



Universidade de Aveiro
2023

Leonel Olímpio Silima

**Exploração de Técnicas de Machine Learning em
Análise de Dados Financeiros na Vision Factory**



Universidade de Aveiro

2023

**Leonel Olímpio
Silima**

**Exploração de Técnicas de Machine Learning
em Análise de Dados Financeiros na Vision
Factory**

Relatório de estágio apresentado à Estga - Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Informática Aplicada, realizada sob a orientação científica do Doutor Ciro Alexandre Domingues Martins, Professor Adjunto da Universidade de Aveiro e Doutor Mário Jorge Ferreira Rodrigues, Professor Adjunto da Universidade de Aveiro.

O júri

Presidente

Professor Doutor Pedro Alexandre de Sousa Gonçalves, Professor Adjunto,
Universidade de Aveiro

Vogais

Professor Doutor Fernando Miguel Rocha de Oliveira, Professor Adjunto,
Universidade de Aveiro

Professor Doutor Mário Jorge Ferreira Rodrigues, Professor Adjunto,
Universidade de Aveiro

Agradecimentos

Ao longo desta jornada de estudo no âmbito de mestrado em Informática Aplicada, expresso o meu profundo agradecimento a todos os que contribuíram para o sucesso deste trabalho.

Em primeiro lugar, agradeço a Deus, cuja sabedoria e orientação divina me acompanharam em todos os momentos deste percurso académico.

À minha equipa de supervisão, o Doutor Ciro Alexandre Domingues Martins e o Co-orientador Doutor Mário Jorge Ferreira Rodrigues, agradeço pela vossa orientação crítica e insights valiosos na área de Machine Learning, que foram fundamentais para o desenvolvimento deste estudo.

À minha família, meu pai Olímpio Silima e minha mãe Elisa João Silima, bem como todos os meus irmãos, agradeço pelo amor incondicional, apoio constante e compreensão ao longo dos anos.

Aos meus amigos, em especial a Glaidys Do Céu Leão Taka, Ilídio Paulino, Janeth Santos, Marielle Creão, Marcílio Correia, Bruno Lopes, Miguel Oliveira, Beatriz Henriques, Beatriz Triane, Ozlem Eryurur, Ece Miray, Natsuki Ono, Mevlut Akçakuyu, Edimar Ramalho, Higiney Paquete, Cláudio Nhancale, Mário Uacane, Marko Smilevski, Aleksei Dmitriev e a todos os outros amigos que estiveram ao meu lado, agradeço pelo encorajamento e momentos especiais partilhados.

Por último expresso a minha gratidão a todos os colegas da EESTEC (Electrical Engineering Students' European assoCiation), que enriqueceram a minha experiência académica, proporcionando oportunidades de aprendizado e intercâmbio cultural durante minhas viagens pela Europa como estudante.

O vosso apoio e contribuições foram fundamentais para o sucesso desta dissertação e para a minha jornada académica. O meu sincero obrigado a todos.

Palavras-chave

Análise de Dados Financeiros, Exploração de Técnicas Machine Learning, Negociação de Ações, Estratégias de Investimento, Python, Backtesting, Chatbot, Inteligência Artificial.

Resumo

Este documento enfatizou a análise de dados financeiros e o tratamento de desequilíbrios de dados, com ênfase na aplicação de técnicas de Machine Learning. Foi conduzida uma exploração aprofundada das complexidades financeiras, identificando padrões cruciais para embasar decisões assertivas. Foram desenvolvidas competências sólidas na manipulação de dados e assegurou-se a confiabilidade das análises dos dados para garantir os resultados almejados. Foram realizadas operações de negociação de ativos, implementaram-se estratégias com Python e efetuados testes retrospectivos para fortalecer a base de investimento. A automação de tarefas, através da implementação de um chatbot com a API da OpenAI, otimizou significativamente o atendimento ao cliente. Contribuiu-se ativamente para a pesquisa, com publicações de artigos e tutoriais educativos na área de análise de dados e inteligência artificial aplicada ao setor financeiro. Este estágio proporcionou uma preparação sólida para enfrentar desafios financeiros complexos, destacando a necessidade de implementar estratégias algorítmicas personalizadas minuciosamente testadas em simulações antes da aplicação nos mercados reais, a fim de mitigar o viés emocional nas decisões financeiras.

Keywords

Financial Data Analysis, Exploration of Machine Learning Techniques, Stock Trading, Investment Strategies, Python, Backtesting, Chatbot, Artificial Intelligence.

Abstract

This document focused on the analysis of financial data and the treatment of data imbalances, with an emphasis on the application of Machine Learning techniques. A thorough exploration of financial complexities was conducted, identifying crucial patterns to support informed decisions. Solid skills were developed in data manipulation, ensuring data analysis reliability for desired outcomes. Asset trading operations were carried out, Python-based strategies were implemented, and retrospective tests were conducted to strengthen the investment foundation. Task automation, through the implementation of a chatbot using the OpenAI API, significantly optimized customer service. Active contributions were made to research, with the publication of articles and educational tutorials in the field of data analysis and artificial intelligence applied to the financial sector. This internship provided a robust preparation for facing complex financial challenges, highlighting the necessity of implementing meticulously tested customized algorithmic strategies through simulations before applying them to real markets to mitigate emotional bias in financial decisions.

Índice de Conteúdo	
Índice de Conteúdo.....	9
Lista de figuras	11
Lista de tabelas.....	12
SIGLAS E ABREVIATURAS	13
1.INTRODUÇÃO	14
1.2Motivação e Justificativa	15
1.3Problema e questões de investigação.....	15
1.4Questões de investigação.....	16
1.5Objetivo geral do trabalho.....	16
1.6Objetivos específicos	17
1.7Metodologia.....	17
1.8 Estrutura do documento.....	18
2.ENQUADRAMENTO TEORICO.....	20
2.1Técnicas de tratamento de dados	20
2.2Algoritmos de Machine Learning	24
2.2.1Tipos de Aprendizagem de Máquina.....	24
2.3 Análise de dados financeiros e estratégias de negociação.....	29
2.3.1 Estratégias de negociação.....	30
2.3.2 Principais técnicas utilizadas para análise das estratégias de negociação	31
3.TRABALHO INICIAL.....	32
3.1. Análise e visualização do conjunto de dados	33
3.2Preprocessamento de dados.....	41
3.2Treinamento do modelo.....	42
3.3Hiperparametrização dos algoritmos.....	47
3.4Validação Cruzada.....	50
3.5Geração do modelo final.....	51
3.6Testes dos modelos.....	52
3.7Teste de significância.....	54
4DESCRIBÇÃO DA EMPRESA E AS ATIVIDADES DESENVOLVIDAS DURANTE O ESTÁGIO.....	56
4.1.Descrição da empresa.....	56
4.2.Atividades desenvolvidas durante o estágio.....	56
5.RESULTADOS E DISCUSSÃO	64
5.1.Critérios para investimentos com alto potencial de crescimento.....	64
5.2.Análise técnica do mercado financeiro para adoção das estratégias de negociação.....	66
5.3.Conhecimentos adquiridos durante o estágio	67
6. CONCLUSÕES E SUGESTÕES	69

Bibliografia..... 71

Lista de figuras

Figure 1: Função de decisão de reamostragem	21
Figure 2: Formulação matemática de NearMiss 3	22
Figure 3: Função de decisão e de reamostragem do NearMiss – 3	22
Figure 4: Principais diferenças entre as técnicas de reamostragem	23
Figure 5: Diferentes técnicas de reamostragem baseadas em SMOTE	24
Figure 6: Árvores de decisão no modelo de regressão	25
Figure 7: Árvores de decisão no modelo de classificação	26
Figure 8: Exemplo de geração de um conjunto de árvores	26
Figure 9: Um MLP com camada oculta e saída escalar	27
Figure 10: Diferentes parâmetros das SVMs	28
Figure 11: Exemplo de classificação com K-vizinhos mais próximos com pesos uniformes e distantes	28
Figure 12: Uma comparação dos algoritmos de clustering	29
Figure 13: Fluxograma do modelo	32
Figure 14: Algumas das bibliotecas utilizadas para leitura e visualização de dados	33
Figure 15: Conjunto de dados	33
Figure 16: Verificação dos valores nulos	34
Figure 17: Valores categóricos	34
Figure 18: Relação entre clientes fiéis e os que abandonam cartões de crédito	35
Figure 19: Frequência de uso do cartão de crédito	35
Figure 20: Idades dos clientes	36
Figure 21: Limite de crédito	36
Figure 22: Meses de trabalho	36
Figure 23: Total de pagamentos de créditos	36
Figure 24: Totais de transações	37
Figure 25: Frequências de compras	37
Figure 26: Totais de transações	37
Figure 27: Totais de variação de montantes	37
Figure 28: Totais de variação de contas de crédito	38
Figure 29: Matriz de correlação	38
Figure 30: Relação entre atributos e o target de destino	39
Figure 31: Características dos clientes contatados nos últimos 12 meses	40
Figure 32: Tempo de inatividade e target final	40
Figure 33: Género	41
Figure 34: Matriz de confusão de avaliação do modelo com random forest	44
Figure 35: Matriz de confusão de avaliação do modelo com árvore de decisão	45
Figure 36: Matriz de confusão de avaliação do modelo com SVM	45
Figure 37: Matriz de confusão de avaliação do modelo com KNN	46
Figure 38: Matriz de confusão de avaliação da rede neural	47
Figure 39: HDFCBANK long position	61
Figure 40: SPX short position	62
Figure 41: Teste de parâmetros de ADX com chatGPT	63
Figure 42: Resultados do ADX com chatGPT	63

Lista de tabelas

[Tabela 1: Critérios para investimentos de alto potencial de crescimento](#) 65

SIGLAS E ABREVIATURAS

ADASYN	Adaptive Synthetic Sampling
ADX	Average Directional Index
API	Application Programming Interface
CMVM	Comissão do Mercado de Valores Mobiliários
DBSCAN	Density-Based Spatial Clustering of Applications with Noise
DTs	Decision Trees
ENN	Edited Nearest Neighbors
EUA	Estados Unidos de América
H0	Hipótese primária
H1	Hipótese alternativa
HTTPS	HyperText Transfer Protocol Secure
IA	Inteligência Artificial
IBM	International Business Machines Corporation
ID	identificação
ISBN	International Standard Book Number
KNN	K-Nearest Neighbors
MLP	Multi-layer Perceptron
N	NearMiss
NN	Neural Network
NP	Numpy
PD	Pandas
PDF	Portable Document Format
ROS	Random OverSampling
RSI	Relative Strength Index
SMOTE	Synthetic Minority Over-sampling Technique
SVM	Support Vectorial Machine
VaR	Value at Risk
WWW	World Wide Web

1. INTRODUÇÃO

O presente documento delinea exhaustivamente as atividades empreendidas durante o estágio curricular na Vision Factory, uma empresa situada em Barcelona que se dedica as soluções de negócios e estudo do mercado financeiro. O escopo primordial deste relatório consiste em apresentar as atividades meticulosamente desenvolvidas ao longo do período de seis meses, abrangendo o intervalo temporal de 03 de janeiro a 03 de julho. Dentro deste período, um conjunto diversificado de tarefas foi meticulosamente abordado, destacando-se:

- Identificação de oportunidades de investimento com potencial substancial de crescimento.
- Análise de dados financeiros mediante a aplicação de metodologias de análise de mercados.
- Elaboração de um relatório fundamentado na análise técnica e fundamental de mercados específicos.
- Criação de representações gráficas através da utilização de ferramentas profissionais destinadas à análise de dados.
- Formulação de sinais de negociação direcionados à clientes da empresa.
- Monitoramento contínuo dos mercados de ações e engajamento em atividades relacionadas ao mercado.
- Exploração de conjuntos de dados financeiros em múltiplas plataformas, com vista à previsão do comportamento de mercado, bem como à concepção de estratégias de negociação embasadas em técnicas de aprendizado de máquina.
- Contribuição para o cenário de pesquisa, através da publicação de conceitos teóricos e aplicações práticas nas plataformas internas da empresa.

Previamente ao início do estágio, um esforço de preparação pré-estágio foi empreendido. Este estágio preliminar concentrou-se na etapa de preprocessamento de dados, incorporando técnicas para abordar desequilíbrios de dados, treinamento de modelos, aplicação de diversas abordagens de avaliação, e a geração de modelos preditivos com base em algoritmos de aprendizado de máquina. Importante destacar que todas as fases concernentes à concepção de

modelos de Aprendizado de Máquina foram observadas, e o projeto recebeu validação por parte da equipe de supervisão.

Este trabalho preparatório contribuiu substancialmente para o fortalecimento das aptidões necessárias para a execução das tarefas previamente mencionadas durante o estágio formal, consolidando assim um alicerce sólido para o cumprimento das atividades propostas.

1.2 Motivação e Justificativa

A justificativa para o estágio está relacionada à importância de adquirir experiência prática e aplicar os conhecimentos teóricos adquiridos durante a formação acadêmica.

A oportunidade de trabalhar em uma empresa real, lidando com dados e problemas reais do mercado financeiro, proporcionou um ambiente propício para o desenvolvimento de habilidades e competências fundamentais na área de análise financeira e tomada de decisões aplicando diferentes técnicas de inteligência artificial.

Além disso, a motivação científica para o estágio está relacionada ao caráter inovador e desafiador das atividades realizadas.

O estágio ofereceu a oportunidade de explorar técnicas avançadas de análise de dados financeiros, inclusive aplicação de algoritmos de aprendizado de máquina. A aplicação dessas técnicas permitiu o domínio de utilização de técnicas de IA para análise de comportamento do mercado financeiro, o que contribuiu para a compreensão e previsão de tendências e potenciais oportunidades de investimento.

O trabalho realizado antes do estágio, com foco no pré-processamento de dados, treinamento de modelos e avaliação de técnicas de aprendizado de máquina, trouxe uma base teórica sólida e metodológica para as atividades desenvolvidas ao longo do estágio. Essa etapa preparatória fortaleceu as habilidades necessárias e contribuiu para a aplicação prática de conhecimentos científicos na área inteligência artificial e sua aplicação real em dados financeiros.

Portanto, a justificativa e motivação científica para o estágio na Vision Factory estão relacionadas à aquisição de experiência prática, aplicação de conhecimentos teóricos, utilização de técnicas avançadas de análise de dados e geração de modelos preditivos, visando a tomada de decisões financeiras fundamentadas e o desenvolvimento de habilidades e competências na área de IA especificamente para dados financeiros.

1.3 Problema e questões de investigação

Durante os estudos, o autor identificou um dos principais problemas que motivou a realização de estudos aprofundados na área financeira: a compreensão de que, num futuro próximo, senão mesmo no presente, a gestão será amplamente controlada pelas técnicas

de aprendizado de máquina.

No entanto, é importante notar que, no mundo capitalista em que vivemos, a área financeira tem sido o principal foco de atenção em nossas vidas. Nesse contexto, é crucial garantir que os algoritmos que lidam com dados financeiros sejam treinados, avaliados e gerem modelos de forma robusta, a fim de evitar riscos quando aplicados na vida real. Essas motivações levaram ao pesquisador a levantar a seguinte questão de partida:

Como aprimorar a análise financeira e a tomada de decisões no mercado financeiro utilizando os algoritmos de machine learning?

Essa questão reflete a necessidade de explorar e desenvolver estratégias de machine learning para melhorar a tomada de decisões no mercado financeiro. A aplicação de técnicas avançadas de análise de dados, juntamente com inteligência artificial, é uma abordagem promissora para melhorar a compreensão do mercado e a tomada de decisões concisas. Ao investigar essa questão, busca-se encontrar formas de utilizar a inteligência artificial de maneira mais eficaz, garantindo uma gestão financeira mais sólida e segura.

1.4 Questões de investigação

Para responder à questão de investigação anteriormente formulada, foram elaboradas as seguintes questões de investigação:

- Que técnicas de análise comportamental do mercado financeiro devem ser aplicados de maneira a torná-lo mais compreensível?
- Quais são as ferramentas mais adequadas para a elaboração de gráficos de negociação?
- Como os sinais de negociação podem ser aprimorados para melhor compreensão e identificação dos pontos de ação?
- Quais são as melhores estratégias de rastreamento de mercado de ações para identificação de oportunidades de investimento?
- Quais procedimentos de IA devem ser aplicados para permitir a seleção de melhores estratégias de negociação?
- Que estratégias aplicar para incentivar o público e garantir adesão às plataformas da empresa?

As respostas a essas questões contribuíram para o desenvolvimento de estratégias mais eficientes, maior precisão na identificação de oportunidades de investimento e uma abordagem mais fundamentada para a tomada de decisões financeiras com base nas técnicas de machine learning.

1.5 Objetivo geral do trabalho

O objetivo geral deste trabalho é aprimorar a análise financeira e a tomada de decisões no

mercado financeiro, utilizando algoritmos de machine learning orientadas à análise de dados financeiros.

1.6 Objetivos específicos

Como premissas para dar resposta ao objetivo geral, foram formulados os seguintes objetivos específicos:

- Identificar critérios eficazes para investimentos com alto potencial de crescimento.
- Aprimorar as técnicas de análise de mercado visando uma compreensão mais precisa e abrangente do comportamento do mercado financeiro.
- Desenvolver sinais de negociação mais eficazes, evitando riscos de perda de valores financeiros.
- Definir estratégias eficientes de rastreamento de mercado de ações e trabalho de mercado para identificação de oportunidades de investimento.
- Utilizar conjuntos de dados financeiros de maneira eficaz para prever o comportamento do mercado e melhorar as técnicas de negociação com base em técnicas de machine learning.
- Publicar ideias teóricas e práticas nas plataformas da empresa, promovendo sua reutilização interna de forma eficiente.

Os objetivos específicos visam abordar questões específicas relacionadas ao aprimoramento da análise financeira e da tomada de decisões no mercado financeiro. Cada objetivo direciona – se para áreas-chave, fornecendo uma estrutura clara para a investigação e desenvolvimento de estratégias que possam melhorar a eficiência e a precisão da análise financeira.

1.7 Metodologia

Para garantir que os estudos para soluções dos problemas enfrentados durante o estágio, fossem reconhecidos como sólidos e potencialmente relevantes, tanto pela empresa quanto pela sociedade científica em geral, foram aplicados diferentes métodos com base na natureza do problema. De entre eles se destacam:

- a) **Observação participante:** Observação participante possibilita obter uma perspectiva holística e natural das matérias a serem estudadas. (Mónico, Alferes, & Castro, 2017)

Porém, a observação participante foi aplicada para encontrar padrões comuns ou diferenças significativas nas abordagens das estratégias de negociação, identificando quais parecem mais eficazes em diferentes situações. Além disso, o método possibilitou a seleção das literaturas com informações mais concisas sobre dados financeiros e estratégias de negociação.

O método de observação foi auxiliado pelas técnicas de experimentação e análise de dados

através da projeção de experimentos rápidos e simples para identificar as melhores opções ou abordagens em um estágio inicial. De enaltecer que as etapas de experimentação foram efetuadas com dados não comprometedoras com vista a evitar riscos.

- b) **Pesquisa bibliográfica:** Uma pesquisa bibliográfica, é um método sistemático, explícito e reproduzível para identificar, avaliar e sintetizar o corpo existente de trabalhos concluídos e registados produzidos por investigadores, académicos e profissionais. (Fink, 2020)

Este método foi relevante tanto para pré-estágio quanto para o estágio em si. As pesquisas desenvolvidas por outros autores, foram fundamentais para obter uma visão geral dos procedimentos técnicos no processo de tratamento de dados, treinamento de modelos com algoritmos de machine learning e identificação de técnicas de avaliação e validação de modelos.

Durante o estágio, esse método foi aplicado para obter um conhecimento mais aprofundado sobre as estratégias de negociação e os procedimentos a serem observados nos gráficos, a fim de identificar os melhores pontos de entrada para a compra e venda de ações. Porém, este método foi conciliado por técnica de modelagem e simulação o que permitiu a avaliação rápida de diferentes cenários e soluções além da tomada de decisões rápidas.

- c) **Colaboração e brainstorming:** brainstorming é uma técnica realizada em grupo, com o intuito de obter o máximo de ideias possíveis para uni-las em uma solução ideal para o problema a ser solucionado. Esta etapa exige criatividade dos participantes, para que ao decorrer da aplicação desta etapa, possam surgir diversos pensamentos e até mesmo experiências que possam ser aplicadas de forma eficiente. (Pereira, Vasconcelos, Rodrigues, & Marques, 2023)

A troca de ideias com profissionais da área, especialmente aqueles com conhecimentos e perspectivas diferentes, proporcionou insights e soluções inovadoras. O brainstorming em grupo revelou-se a forma mais eficaz de gerar ideias rapidamente e obter conhecimentos sólidos sobre as atividades do dia a dia. Esta foi a técnica mais utilizada durante estágio principalmente nas tarefas de menor domínio pelas equipas de estagiários.

1.8 Estrutura do documento

Este trabalho apresenta a seguinte estrutura:

Capítulo I: Introdução – com breve contextualização das atividades desenvolvidas durante o estágio e na fase preparatória, justificativa e motivação, problema e questões de investigação, objetivos, Metodologias aplicadas e estrutura do documento;

Capítulo II: Enquadramento do trabalho – onde foram apresentados trabalhos relacionados ao tratamento de dados desbalanceados, treinamento com algoritmos de machine learning

e geração de modelos;

Capítulo III: Estudos iniciais – apresentação dos resultados obtidos no trabalho desenvolvido antes do estágio;

Capítulo IV: Apresentação das atividades do estágio;

Capítulo V: Discussão de resultados e recomendações para futuros trabalhos – último capítulo do trabalho que apresenta os resultados obtidos e por último foram apresentadas as referências bibliográficas.

2. ENQUADRAMENTO TEORICO

Neste capítulo, são apresentados os principais conceitos relacionados ao processo de tratamento de dados para a criação de modelos de aprendizagem de máquina, desde a análise de dados brutos, pré-processamento até o treinamento com algoritmos de machine learning. Além disso, foram abordados conceitos sobre dados financeiros e estratégias de negociação de ações, que foram o foco principal das atividades desenvolvidas durante o estágio.

2.1 Técnicas de tratamento de dados

A qualidade dos modelos de aprendizagem de máquina depende do cuidado no processo de análise e tratamento de dados. Tratar dados não numéricos, dados nulos e normalizar os dados são pontos importantes ao utilizar técnicas de machine learning. (Luchese, 2021) Dados do mundo real podem conter irregularidades e corrupções, comprometendo a qualidade do conjunto de dados. Problemas típicos de qualidade de dados incluem dados incompletos e inconsistentes.

Ter dados de qualidade é essencial para modelos preditivos precisos. É importante realizar uma triagem dos dados para identificar problemas precocemente e tomar medidas correspondentes de processamento e limpeza dos dados. (Microsoft, 2022)

Na preparação dos dados, a limpeza dos dados (incluindo a remoção de valores em falta, outliers e inconsistências), análise multivariada, combinação ou criação de novas variáveis a partir das existentes e redução do número de variáveis são aspectos a serem considerados. (ManagementSolutions, 2018)

- **Tratamento de dados desbalanceados**

O balanceamento de dados visa equilibrar as instâncias entre classes majoritárias e minoritárias, quando há uma diferença no número de instâncias entre essas classes, resultando em dados desequilibrados. O desequilíbrio pode afetar a avaliação e a qualidade do treinamento se a classe minoritária não estiver adequadamente representada. Para abordar esse desequilíbrio, são utilizadas técnicas de subamostragem (removendo instâncias) e/ou super-amostragem (adicionando instâncias) em relação às classes, buscando corrigir o desequilíbrio. (Saias, Maia, Rato, & Gonçalves, 2018)

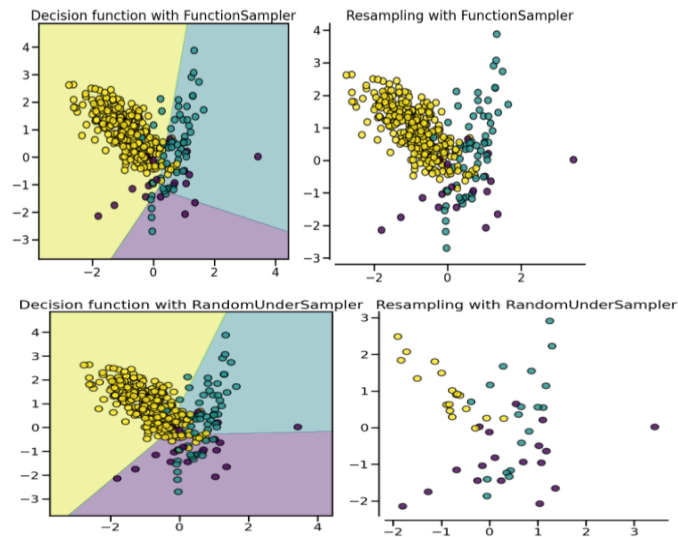
No entanto, esses mecanismos possuem vantagens e desvantagens dependendo da natureza dos dados e do método aplicado. Diferentes técnicas são apresentadas para reduzir ou aumentar as instâncias desequilibradas.

- a) UnderSampling ou redução da classe majoritária**

O método de subamostragem aleatória (RandomUnderSampling) é uma técnica para equilibrar dados desiguais, reduzindo o número de observações da classe majoritária.

Embora seja rápido e fácil de implementar, pode levar à perda de informações importantes. A seguir apresentamos uma figura ilustrativa da função de decisão e de reamostragem.

Figure 1: Função de decisão de reamostragem



Fonte: (imbalanced-learn, 2022)

- **NearMiss**

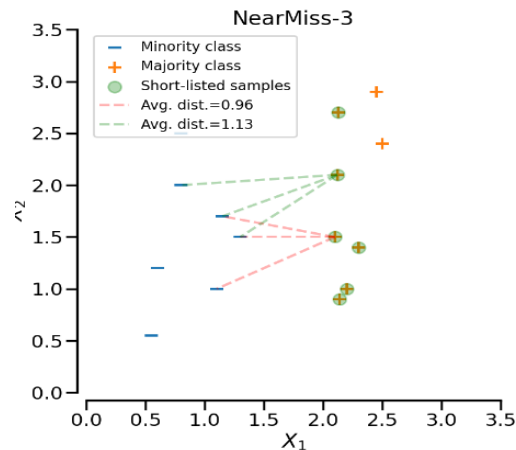
NearMiss é uma técnica de subamostragem que utiliza regras heurísticas baseadas no algoritmo dos vizinhos mais próximos (knn) e no classificador de aceitação derivados (KNeighborsMixin) do scikit-learn. Existem três tipos de heurísticas implementadas: NearMiss-1, NearMiss-2 e NearMiss-3.

NearMiss-1 seleciona as amostras positivas com a menor distância média aos N vizinhos mais próximos da classe negativa. NearMiss-2 seleciona as amostras positivas com a menor distância média aos N vizinhos mais distantes da classe negativa.

NearMiss-3 é um algoritmo de dois passos. Primeiro, mantém os vizinhos mais próximos de cada amostra negativa. Em seguida, seleciona as amostras positivas com a maior distância média aos N vizinhos mais próximos.

Essas heurísticas visam diminuir a diferença entre as classes, selecionando amostras positivas de forma estratégica em relação às amostras negativas, melhorando o balanceamento dos dados. (imbalanced-learn, 2022) A figura a seguir ilustra a formulação matemática de NearMiss 3.

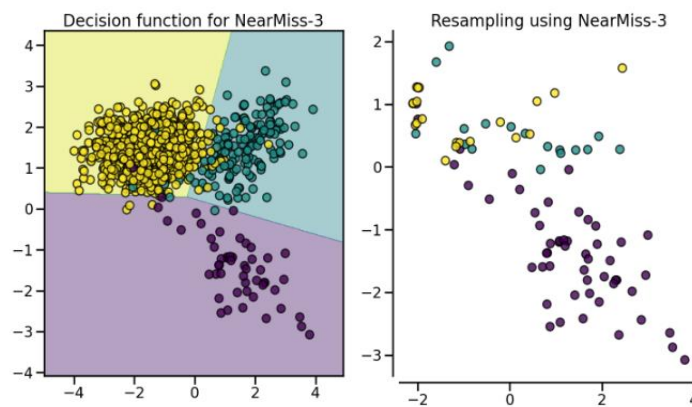
Figure 2: Formulação matemática de NearMiss 3



Fonte: (imbalanced-learn, 2022)

De entre as várias técnicas de UnderSampling, o método de NearMiss tem sido o mais utilizado pelo fato de que não há muita perda de informação importante contrariamente aos outros métodos. A seguir é apresentada a figura ilustrativa da técnica de reamostragem de NearMiss – 3:

Figure 3: Função de decisão e de reamostragem do NearMiss – 3



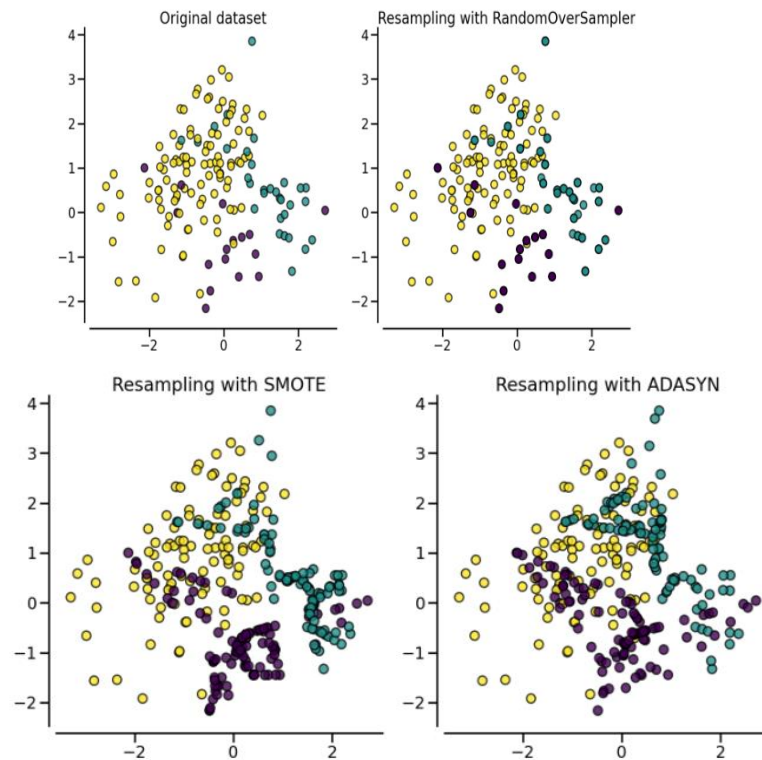
Fonte: (imbalanced-learn, 2022)

b) OverSampling ou aumento da classe majoritária

A técnica de OverSampling consiste na inserção de instâncias na classe minoritária, este procedimento era anteriormente resolvido com aplicação muito simples e de menos consumo computacional designada por Random Over-Sampling (ROS) e foi descontinuado justificadamente por aumento da probabilidade de overfitting. (Soares, 2017)

Como solução, foram implementados dois novos métodos para reamostrar classes minoritárias: (i) a técnica de reamostragem de minorias sintéticas (SMOTE) e (ii) o método de amostragem sintético adaptativo (ADASYN). Estes dois algoritmos podem ser utilizados da mesma maneira. As figuras abaixo apresentam as principais diferenças entre os três métodos mencionados.

Figure 4: Principais diferenças entre as técnicas de reamostragem



Fonte: (imbalanced-learn, 2022)

Entretanto, investigadores perceberam que a técnica SMOTE que responde os problemas da ROS também apresenta problemas de sobre-generalização relacionado com o modo de geração de padrões sintéticos, vistos que durante o processo de computação de instâncias sintéticas não tem em consideração os exemplos próximos da classe majoritária o que aumenta deste modo a ocorrência de sobreposição de classes. Victoria López et al (2013) citado por (Soares, 2017). Contudo, esta técnica foi evoluindo com vista a solucionar as inconsistências anteriormente mencionadas tendo surgido SMOTE + ENN e SMOTE + Tomek Links.

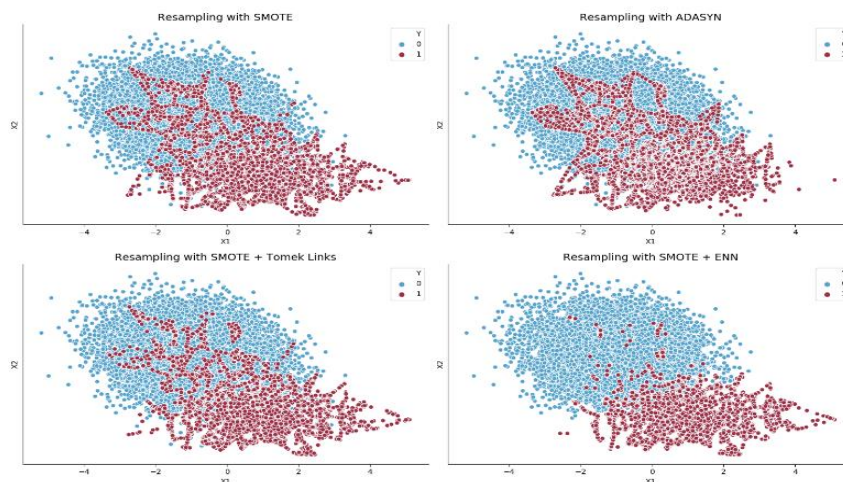
SMOTE + ENN e SMOTE + Tomek Links são técnicas utilizadas para lidar com o desequilíbrio de classes em conjuntos de dados. O SMOTE + ENN combina a técnica de OverSampling do SMOTE com a técnica de UnderSampling do ENN. O ENN estima os vizinhos mais próximos de cada classe majoritária e remove instâncias dessa classe se forem classificadas incorretamente pelos vizinhos mais próximos. Essa combinação ajuda a realizar uma limpeza mais extensa dos dados, resultando em uma separação mais clara entre as classes. (Soares, 2017)

Por outro lado, o SMOTE + Tomek Links utiliza o método de limpeza de dados baseado em Tomek Links. Os links Tomek são pares de instâncias de classes opostas que são os vizinhos mais próximos um do outro. A maioria das observações de classe desses links é

removida, o que aumenta a separação das classes próximas aos limites de decisão. Essa técnica é menos intrusiva em comparação com o ENN.

Enquanto o SMOTE + ENN realiza a limpeza dos dados, o SMOTE + Tomek Links equilibra as instâncias e remove inconsistências do SMOTE. Ambos os métodos visam lidar com o desequilíbrio de classes em conjuntos de dados. (Analytics Vidhya, 2022) As figuras a seguir ilustram as características dos dados após a aplicação dessas técnicas.

Figure 5: Diferentes técnicas de reamostragem baseadas em SMOTE



Fonte: (Analytics Vidhya, 2022)

2.2 Algoritmos de Machine Learning

Neste ponto são abordados alguns algoritmos de machine learning, mas, para melhor esclarecimento em relação aos algoritmos antes foram apresentados os conceitos fundamentais de machine learning e sua respectiva classificação.

2.2.1 Tipos de Aprendizagem de Máquina

Machine learning ou aprendizagem de máquina é um subcampo da inteligência artificial que se concentra em técnicas para automatizar tarefas inteligentes por meio do uso de algoritmos baseados em matemática e estatística. O objetivo é desenvolver sistemas capazes de adquirir conhecimento automaticamente. Existem diferentes abordagens e subdivisões no campo do Machine Learning. (Escovedo & Koshiyama, 2020) e (Machado, Souza, & Santo, 2017)

Uma das maneiras de categorizar os algoritmos de machine learning é com base no estilo de aprendizado. Segundo (AMAZON, 2022), existem quatro tipos principais: aprendizado supervisionado, aprendizado não supervisionado, aprendizado semi-supervisionado e aprendizado por reforço. No entanto, (Machado, Souza, & Santo, 2017) e (Silva & Borges, 2019), agrupam aprendizagem de máquina em três categorias principais: aprendizado supervisionado, aprendizado não supervisionado e aprendizado por reforço.

O aprendizado supervisionado é aplicado quando os dados de treinamento possuem

rótulos pré-determinados e confiáveis. Os modelos são treinados para prever atributos desconhecidos ou classificar dados em categorias predefinidas. Isso é feito por meio da aplicação de um conjunto de variáveis de entrada em uma variável de saída, com base em um conjunto de dados de treinamento. Dependendo do tipo de variável de saída (contínua ou discreta), temos problemas de regressão ou classificação.

Existem diversos algoritmos utilizados no aprendizado supervisionado, cada um com suas características e aplicações específicas. No entanto, a escolha do algoritmo depende do contexto e dos requisitos do problema em questão. (ManagementSolutions, 2018)

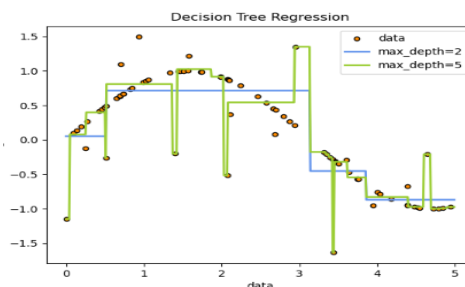
a) Aprendizagem Supervisionada

A aprendizagem supervisionada pressupõe que as classificações passadas estejam disponíveis e os dados estejam rotulados. Os modelos são treinados para prever atributos desconhecidos e determinar se o resultado pertence a categorias predefinidas, como fraudulento ou legítimo. As informações utilizadas para construir o algoritmo estão presentes nos dados de treinamento. O treinamento envolve a aplicação de um conjunto de variáveis de entrada (x) em uma variável de saída (y) usando uma amostra de treinamento (Δ) com pares de dados (x_i, y_i). Dependendo do tipo de variável de saída, temos problemas de regressão ou classificação. (ManagementSolutions, 2018) e (Silva & Borges, 2019), De entre vários algoritmos de aprendizagem supervisionada destacamos os seguintes:

- **Decision Trees (DTs)**

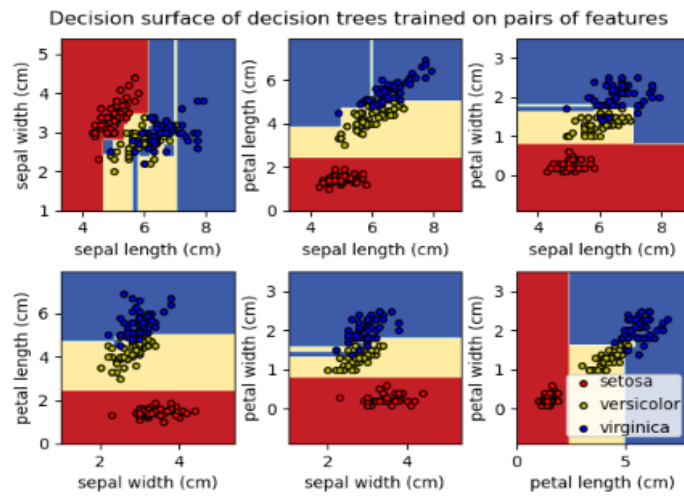
As árvores de decisão são um método de aprendizado supervisionado usado para classificação e regressão. Elas criam um modelo que prevê o valor de uma variável de destino com base em regras de decisão simples extraídas dos dados. Quanto mais profunda a árvore, mais complexas as regras e melhor o modelo. As árvores de decisão são uma forma de aproximação por partes, aprendendo a representar curvas com conjuntos de regras if-then-else. (scikit-learn, 2022) Como representamos na figura a seguir.

Figure 6: Árvores de decisão no modelo de regressão



Fonte: (scikit-learn, 2022)

Figure 7: Árvores de decisão no modelo de classificação

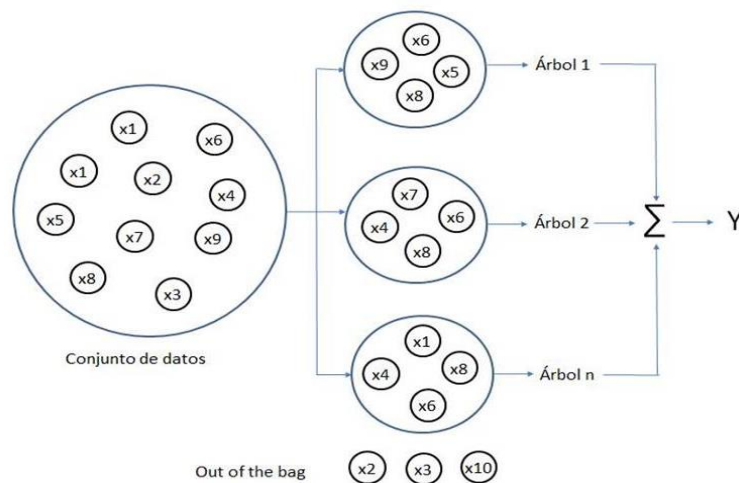


Fonte: (scikit-learn, 2022)

- **Random Forest**

O algoritmo Random Forest é uma técnica de aprendizado supervisionado que combina múltiplas árvores de decisão para criar um modelo robusto. Cada árvore é gerada em duas etapas: um subconjunto aleatório de variáveis é selecionado e cada árvore é expandida ao máximo. As árvores são construídas usando amostras aleatórias (bootstrap) e as observações não utilizadas são aplicadas para validar o modelo. As saídas de todas as árvores são combinadas para obter uma saída final por meio de regras como média ou contagem de votos. Esse processo resulta em um modelo mais preciso e generalizado. (Zúñiga, 2020), Isso é mostrado graficamente na figura abaixo.

Figure 8: Exemplo de geração de um conjunto de arvores

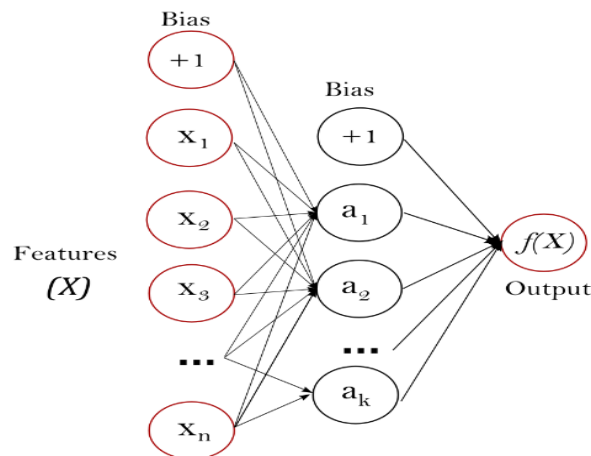


Fonte: (Zúñiga, 2020)

- **Neural network (NN)**

As redes neurais, também conhecidas como Multi-layer Perceptron (MLP), são algoritmos de aprendizado supervisionado que aprendem a função de um conjunto de dados. Elas são capazes de aproximar funções não lineares para classificação ou regressão. Diferentemente da regressão logística, as redes neurais podem ter camadas ocultas entre a camada de entrada e a camada de saída. Essas camadas adicionais permitem que o modelo aprenda representações mais complexas dos dados. (scikit-learn, 2022). A figura abaixo ilustra um MLP com uma camada oculta e saída escalar.

Figure 9: Um MLP com camada oculta e saída escalar



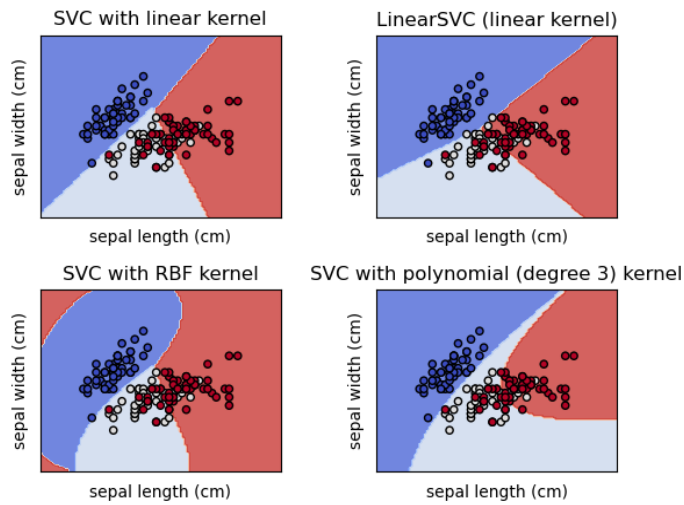
Fonte: (scikit-learn, 2022)

A camada de entrada contém neurônios representando as características de entrada. O neurônios da camada oculta transformam os valores da camada anterior aplicando uma soma ponderada seguida de uma função de ativação não linear. A camada de saída recebe os valores da última camada oculta e os transforma em valores de saída.

- **Support Vector Machines (SVM)**

As Máquinas de Vetores de Suporte (SVM) são métodos de aprendizado supervisionado utilizados para classificação, regressão e detecção de outliers. Elas são eficazes em espaços de alta dimensionalidade e em casos onde o número de dimensões é maior que o número de amostras. SVMs utilizam um subconjunto de pontos de treinamento na função de decisão, o que as torna eficientes em termos de memória. Elas também são versáteis, permitindo a especificação de diferentes funções de kernel. No entanto, as SVMs não fornecem estimativas de probabilidade diretamente e há riscos de overfitting quando o número de recursos é maior que o número de amostras. (scikit-learn, 2022)

Figure 10: Diferentes parametros das SVMs

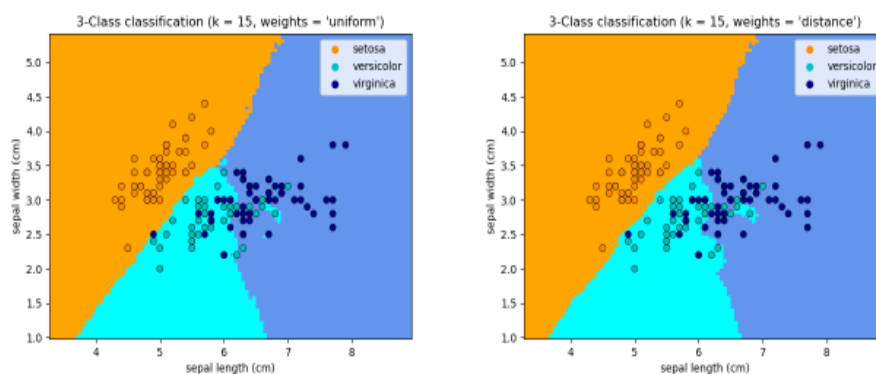


Fonte: (scikit-learn, 2022)

- **Nearest Neighbors (KNN)**

Os métodos de vizinho mais próximo (KNN) buscam encontrar um número pré-definido de amostras de treinamento mais próximas em termos de distância a um novo ponto e utilizam essas amostras para prever o rótulo desse ponto. O número de amostras pode ser fixo (vizinho mais próximo k) ou variar com base na densidade local de pontos (vizinho baseado no raio). A distância entre pontos pode ser medida de diversas formas, sendo a distância euclidiana a mais comum. Os métodos de vizinho mais próximo são considerados não generalizantes, pois memorizam todo os dados de treinamento. (scikit-learn, 2022)

Figure 11: Exemplo de classificação com K-vizinhos mais próximos com pesos uniformes e distantes



Fonte: (scikit-learn, 2022)

b) Aprendizagem não supervisionada

Aprendizagem não supervisionada é uma técnica de aprendizado de máquina em que não há informações sobre a classificação das transações. Os modelos aprendem iterativamente com os dados, buscando identificar padrões e relações para adaptar seu

comportamento na análise de informações desconhecidas.

Não há uma variável de saída definida, e o objetivo é encontrar padrões e relações nos dados. Os algoritmos de agrupamento são amplamente utilizados nessa abordagem, onde o objetivo é encontrar grupos de dados com características semelhantes.

Alguns algoritmos comumente usados incluem K-Means, Mean-shift, DBSCAN, entre outros. A escolha do algoritmo e da métrica de distância depende do objetivo e dos critérios de semelhança desejados entre os clusters. (Soares, 2017) e (scikit-learn, 2022), A seguir apresentamos a comparação dos algoritmos de agrupamento anteriormente mencionados.

Figure 12: Uma comparação dos algoritmos de clustering

Method name	Parameters	Scalability	Usecase	Geometry (metric used)
K-Means	number of clusters	Very large <code>n_samples</code> , medium <code>n_clusters</code> with <code>MiniBatch</code> code	General-purpose, even cluster size, flat geometry, not too many clusters, inductive	Distances between points
Affinity propagation	damping, sample preference	Not scalable with <code>n_samples</code>	Many clusters, uneven cluster size, non-flat geometry, inductive	Graph distance (e.g. nearest-neighbor graph)
Mean-shift	bandwidth	Not scalable with <code>n_samples</code>	Many clusters, uneven cluster size, non-flat geometry, inductive	Distances between points
Spectral clustering	number of clusters	Medium <code>n_samples</code> , small <code>n_clusters</code>	Few clusters, even cluster size, non-flat geometry, transductive	Graph distance (e.g. nearest-neighbor graph)
Ward hierarchical clustering	number of clusters or distance threshold	Large <code>n_samples</code> and <code>n_clusters</code>	Many clusters, possibly connectivity constraints, transductive	Distances between points
Agglomerative clustering	number of clusters or distance threshold, linkage type, distance	Large <code>n_samples</code> and <code>n_clusters</code>	Many clusters, possibly connectivity constraints, non Euclidean distances, transductive	Any pairwise distance
DBSCAN	neighborhood size	Very large <code>n_samples</code> , medium <code>n_clusters</code>	Non-flat geometry, uneven cluster sizes, outlier removal, transductive	Distances between nearest points
OPTICS	minimum cluster membership	Very large <code>n_samples</code> , large <code>n_clusters</code>	Non-flat geometry, uneven cluster sizes, variable cluster density, outlier removal, transductive	Distances between points
Gaussian mixtures	many	Not scalable	Flat geometry, good for density estimation, inductive	Mahalanobis distances to centers
BIRCH	branching factor, threshold, optional global clusterer.	Large <code>n_clusters</code> and <code>n_samples</code>	Large dataset, outlier removal, data reduction, inductive	Euclidean distance between points
Bisecting K-Means	number of clusters	Very large <code>n_samples</code> , medium <code>n_clusters</code>	General-purpose, even cluster size, flat geometry, no empty clusters, inductive, hierarchical	Distances between points

Fonte: (scikit-learn, 2022)

Existem muitas outras técnicas de aprendizagem não supervisionada apenas salientar que as mencionadas acima são as utilizadas com maior frequência nesta subcategoria de machine learning.

2.3 Análise de dados financeiros e estratégias de negociação

De acordo com a (IVESTOPEDIA, 2023), análise de dados financeiros refere-se ao processo de examinar e interpretar informações financeiras para obter insights sobre o desempenho financeiro de uma empresa, setor ou mercado. Essa análise é crucial para tomar decisões informadas relacionadas a investimentos, estratégias de negócios e gestão financeira.

A análise de dados financeiros envolve várias etapas, incluindo a coleta de dados financeiros relevantes, organização e preparação dos dados, aplicação de técnicas

analíticas e interpretação dos resultados.

A análise de dados financeiros desempenha um papel fundamental na identificação de tendências, padrões e oportunidades nos mercados financeiros. Essa análise ajuda os investidores, traders e gestores de portfólio a tomar decisões informadas sobre estratégias de negociação.

2.3.1 Estratégias de negociação

Estratégias de negociação são planos ou métodos estabelecidos para orientar a tomada de decisões de compra e venda de ativos financeiros, como ações, títulos e moedas. Essas estratégias são projetadas para maximizar os lucros, minimizar as perdas e gerir o risco ao realizar transações nos mercados financeiros. (etoro, 2023)

Para (The balance, 2023) e (Ivestopedia, 2023), afirmam que as estratégias de negociação podem variar amplamente, dependendo dos objetivos, estilo de negociação e horizonte temporal do investidor ou trader. Algumas das estratégias mais comuns incluem:

- **Negociação de tendência:** Os traders identificam e seguem a direção predominante de uma tendência de mercado e buscam entrar em posições alinhadas com essa tendência. Isso envolve a análise de gráficos de preços, indicadores técnicos e padrões de mercado.
- **Negociação de reversão:** Ao contrário da negociação de tendência, essa estratégia busca identificar pontos de reversão ou mudanças de direção nos mercados. Os traders buscam entrar em posições contrárias à tendência atual, apostando em uma reversão de preço.
- **Negociação de intervalo:** Os traders identificam níveis de suporte e resistência nos quais o preço tende a oscilar entre eles. Eles compram quando o preço atinge o suporte e vendem quando atinge a resistência, aproveitando os movimentos de preço dentro desse intervalo.
- **Negociação de breakout:** Nessa estratégia, os traders procuram identificar pontos em que o preço rompe um nível de suporte ou resistência significativo. Eles entram em posições após o rompimento, esperando que o preço continue a se mover na direção do breakout.
- **Negociação de eventos:** Os traders se baseiam em eventos econômicos, políticos ou corporativos para realizar transações. Eles procuram aproveitar a volatilidade e as oportunidades que surgem em torno desses eventos, como anúncios de resultados financeiros, decisões de bancos centrais ou lançamentos de dados econômicos.
- **Negociação algorítmica:** Essa estratégia envolve o uso de algoritmos e sistemas

automatizados para realizar transações nos mercados financeiros.

2.3.2 Principais técnicas utilizadas para análise das estratégias de negociação

A análise das estratégias de negociação é uma parte fundamental do processo de tomada de decisões dos investidores e traders. (etoro, 2023), para (Quantified, 2023) algumas das principais técnicas são:

- **Backtesting:** Essa técnica envolve o teste retrospectivos da estratégia em dados históricos para avaliar seu desempenho passado. Os traders usam software especializado para aplicar a estratégia a dados históricos e verificar como ela teria se saído em diferentes períodos e condições de mercado. O backtesting ajuda a identificar os pontos fortes e fracos da estratégia e permite que os traders ajustem e refinem a abordagem.
- **Análise estatística:** A análise estatística é utilizada para avaliar o desempenho da estratégia com base em métricas e indicadores quantitativos. Isso pode incluir a análise de métricas de desempenho, como taxa de acerto, razão de recompensa/risco, retorno sobre o investimento, entre outros.
- **Simulações e modelagem:** As simulações e modelagem são utilizadas para criar cenários hipotéticos e testar a eficácia da estratégia em diferentes condições de mercado. Isso ajuda a identificar os pontos fortes e fracos da estratégia em diferentes cenários e a tomar decisões com base nessas informações.
- **Análise de sensibilidade:** Essa técnica envolve a análise do impacto das mudanças nos parâmetros da estratégia. Os traders podem realizar análises de sensibilidade para avaliar como o desempenho da estratégia é afetado por alterações nos parâmetros, como o tamanho da posição, o período, as regras de entrada/saída, entre outros. Isso ajuda a otimizar a estratégia e identificar os parâmetros mais adequados.
- **Análise de risco:** A análise de risco é crucial para avaliar a viabilidade e a robustez da estratégia. Isso envolve a identificação e a quantificação dos riscos associados à estratégia, como o risco de mercado, o risco de liquidez e o risco operacional. Os traders podem utilizar técnicas de análise de risco, como a simulação de Monte Carlo, para avaliar a distribuição de retornos e estimar métricas de risco, como o VaR (Value at Risk).

Essas são algumas das principais técnicas utilizadas para analisar as estratégias de negociação.

3. TRABALHO INICIAL

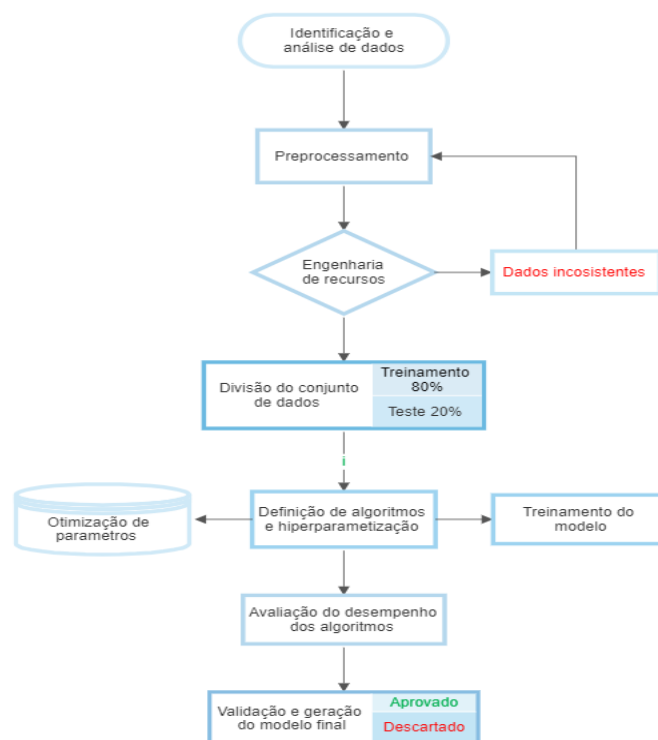
O projeto consistiu na criação de um modelo de Aprendizagem de Máquina projetado para prever clientes propensos ao abandono dos serviços de crédito. Essa iniciativa teve como objetivo preparação para o estágio em análise de dados financeiros.

Os dados históricos dos clientes foram coletados e analisados, incluindo informações como histórico de pagamentos, saldo do devedor, utilização de crédito e outras variáveis relevantes. Com base nesses dados, um modelo de Aprendizagem de Máquina foi treinado para identificar padrões e tendências nos comportamentos dos clientes que indicassem um risco de abandono.

A avaliação do modelo quanto à sua precisão e eficácia na previsão de clientes propensos ao abandono foi um passo importante do projeto. Essa previsão pode ser valiosa para instituições financeiras, pois permite a implementação de medidas proativas para reter esses clientes e evitar perdas financeiras.

Além disso, a preparação desse projeto proporcionou a oportunidade de adquirir e aprimorar habilidades essenciais em análise de dados, pré-processamento de dados, seleção de recursos e desenvolvimento de modelos de aprendizagem de máquina. Tais competências foram fundamentais no campo da análise de dados financeiros e prepararam significadamente para o estágio realizado. A seguir apresentamos os procedimentos observados em forma dum fluxograma:

Figure 13: Fluxograma do modelo



Adaptado por autor

3.1. Análise e visualização do conjunto de dados

Foi utilizado um conjunto de dados contendo informações de 10.000 clientes com o problema de perda de clientes nos serviços de cartão de crédito. O objetivo principal foi realizar análise de dados para descobrir os motivos por trás da perda e prever quais clientes têm maior probabilidade de desistir. O primeiro passo foi a instalação e importação das bibliotecas de Python necessárias para a leitura, análise e visualização dos dados. Essas etapas são essenciais para manipular e explorar o conjunto de dados, preparando-o para a análise antes de trabalhá-los com os algoritmos de aprendizado de máquina. A seguir observamos as bibliotecas utilizadas para o tratamento de dados.

Figure 14: Algumas das bibliotecas utilizadas para leitura e visualização de dados

```
import pandas as pd
import matplotlib.pyplot as plt
import missingno as mn
import plotly.express as px
import seaborn as sns
from sklearn.preprocessing import PowerTransformer
from sklearn.model_selection import train_test_split
from imblearn.under_sampling import RandomUnderSampler, NearMiss, OneSidedSelection
from sklearn.preprocessing import LabelEncoder, StandardScaler
import pickle
```

Após esta etapa, seguiu o processo de visualização e análise de dados como podemos observar os resultados a seguir.

Figure 15: Conjunto de dados

```
[2]: dados = pd.read_csv('BankChurners.csv') #CARREGAMENTO DE DADOS
dados
```

Out[2]:

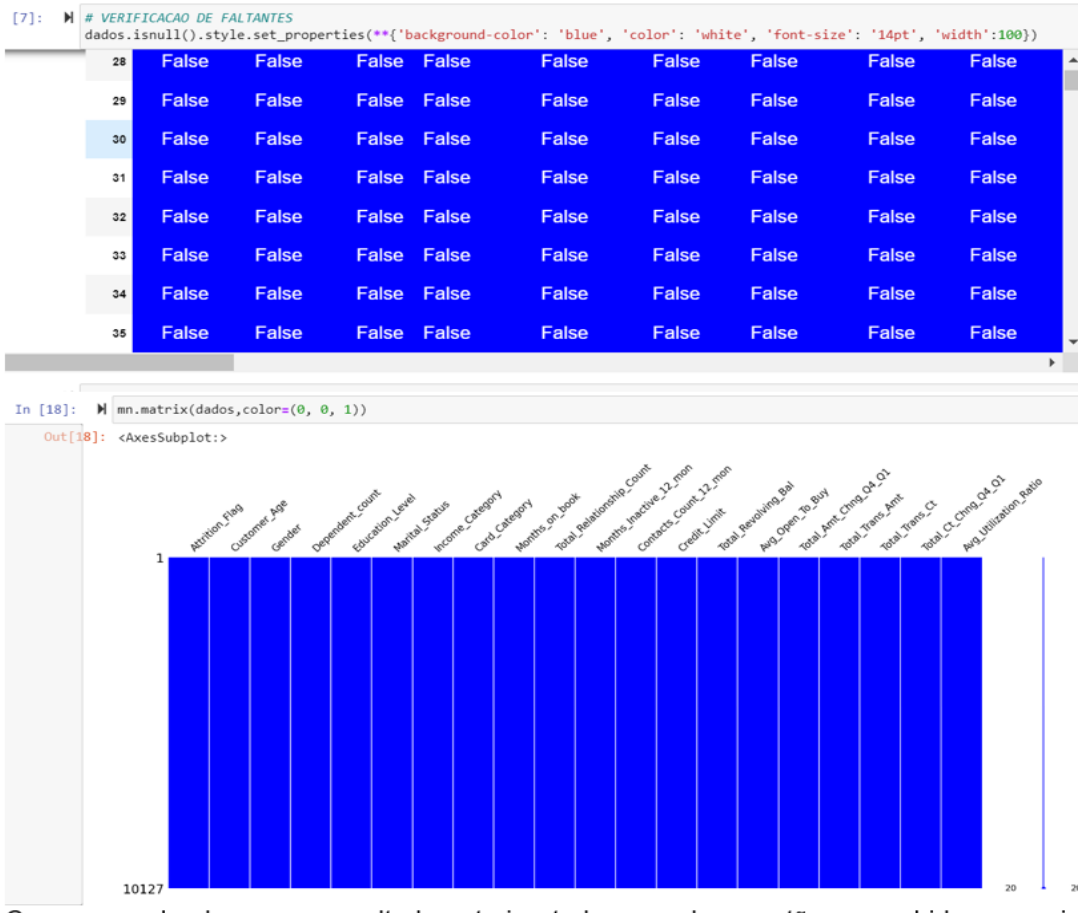
	CLIENTNUM	Attrition_Flag	Customer_Age	Gender	Dependent_count	Education_Level	Marital_Status	Income_Category	Card_Category	Months_or
0	768805383	Existing Customer	45	M	3	High School	Married	60K – 80K	Blue	
1	818770008	Existing Customer	49	F	5	Graduate	Single	Less than \$40K	Blue	
2	713982108	Existing Customer	51	M	3	Graduate	Married	80K – 120K	Blue	
3	769911858	Existing Customer	40	F	4	High School	Unknown	Less than \$40K	Blue	
4	709106358	Existing Customer	40	M	3	Uneducated	Married	60K – 80K	Blue	
...
10122	772366833	Existing Customer	50	M	2	Graduate	Single	40K – 60K	Blue	
10123	710638233	Attrited Customer	41	M	2	Unknown	Divorced	40K – 60K	Blue	
10124	716506083	Attrited Customer	44	F	1	High School	Married	Less than \$40K	Blue	
10125	717406983	Attrited Customer	30	M	2	Graduate	Unknown	40K – 60K	Blue	
10126	714337233	Attrited Customer	43	F	2	Graduate	Married	Less than \$40K	Silver	

10127 rows x 23 columns

- **Identificação de valores faltantes**

Tendo sido visualizado o conjunto de dados seguiu o processo de análise dos dados brutos sendo a primeira etapa a identificação de valores nulos. A seguir apresentamos o resultado da função (isnull) do pandas que apresenta os detalhes em relação aos valores nulos.

Figure 16: Verificação dos valores nulos

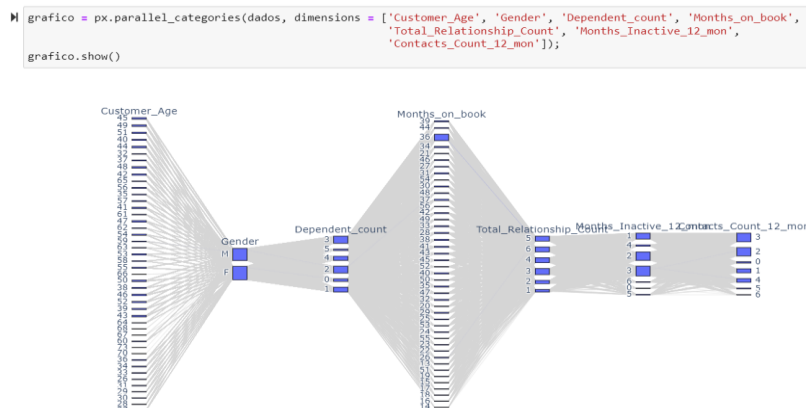


Como se pode observar o resultado anterior, todos os valores estão preenchidos, ou seja, o resultado false presente em cada atributo indica que nenhum valor nulo foi encontrado.

- **Visualização de valores categóricos**

O conjunto de dados foi subdividido entre categóricos e numéricos com vista a facilitar a análise e tratamento. A seguir apresentamos os relacionamentos entre os valores categóricos.

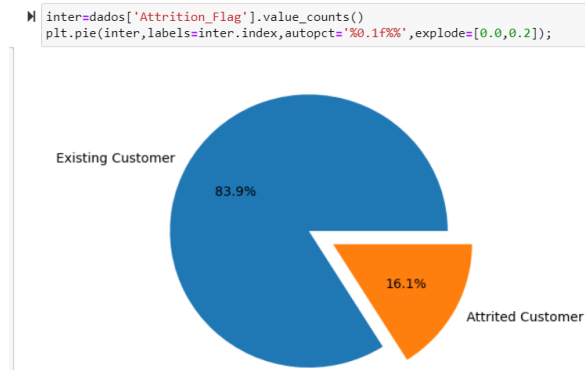
Figure 17: Valores categóricos



- **Análise das instâncias negativas e positivas**

Em seguida, podemos visualizar a percentagem de clientes leais e aqueles que abandonam os cartões de crédito com base nas características dos dados brutos, conforme mostrado no gráfico.

Figure 18: Relação entre clientes fiéis e os que abandonam cartões de crédito



Como podemos observar os dados apresentam maior desequilíbrio entre as instâncias positivas (clientes leais) e instâncias negativas (clientes que abandonam cartões de crédito), posteriormente serão apresentados os métodos utilizados para o tratamento deste tipo do problema.

- **Análise dos valores numéricos**

Para melhor compreensão dos atributos que mais influenciam no desequilíbrio do modelo esta subcategoria de dados foi analisada atributo por atributo. A seguir apresentamos o código e os resultados obtidos.

```

def analise_num(var):
    plt.subplot(1, 2, 1)
    sns.histplot(data=dados, x=var, hue='Attrition_Flag', kde=True);

    plt.subplot(1,2,2)
    sns.boxplot(x='Attrition_Flag', y=var, data=dados);
    plt.subplots_adjust(left=0.2,
                        bottom=0.1,
                        right=1.3,
                        top=0.9,
                        wspace=0.4,
                        hspace=0.8)

```

Figure 19: Frequência de uso do cartão de crédito

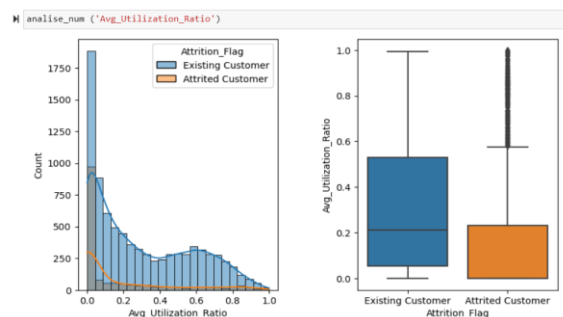


Figure 20: Idades dos clientes

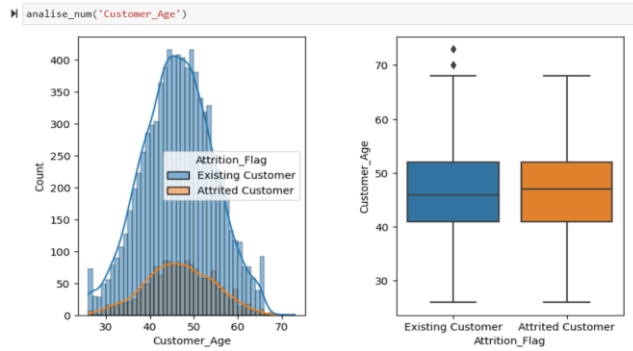


Figure 21: Limite de crédito

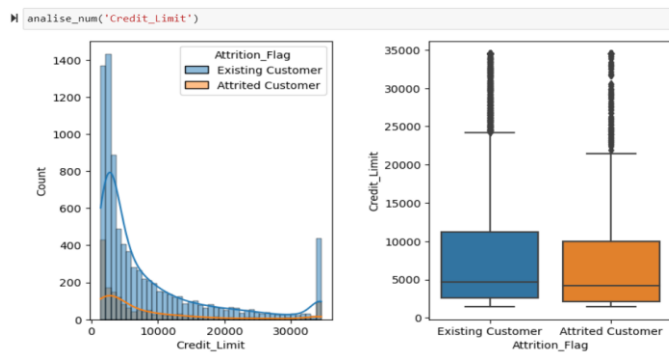


Figure 22: Meses de trabalho

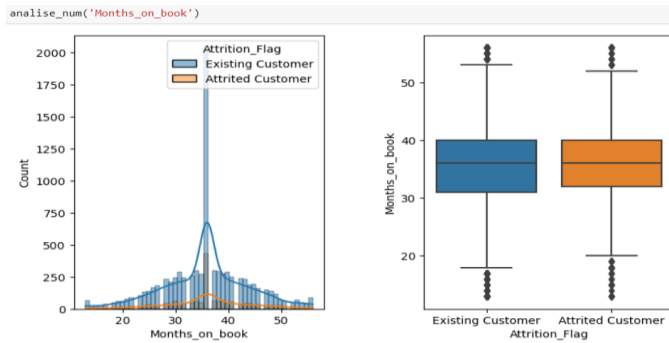


Figure 23: Total de pagamentos de créditos

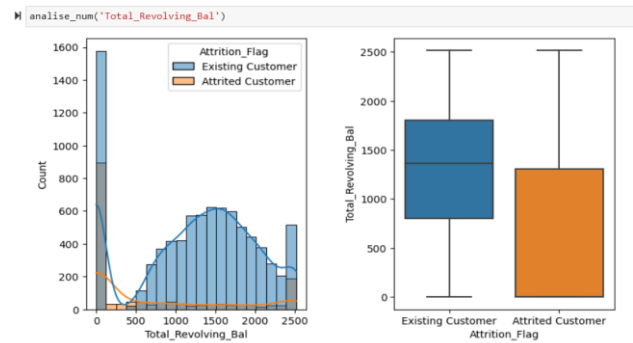


Figure 24: Totais de transações

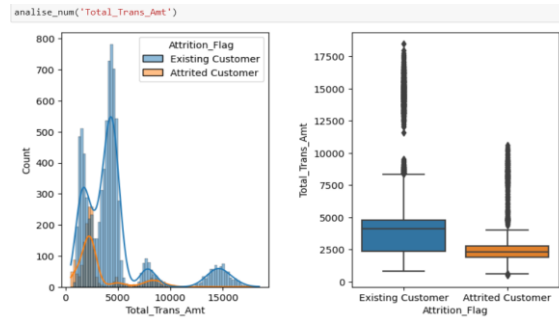


Figure 25: Frequências de compras

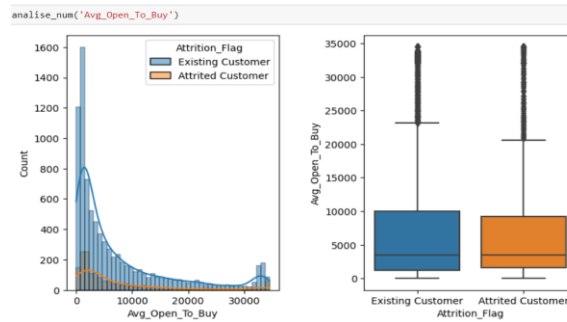


Figure 26: Totais de transações

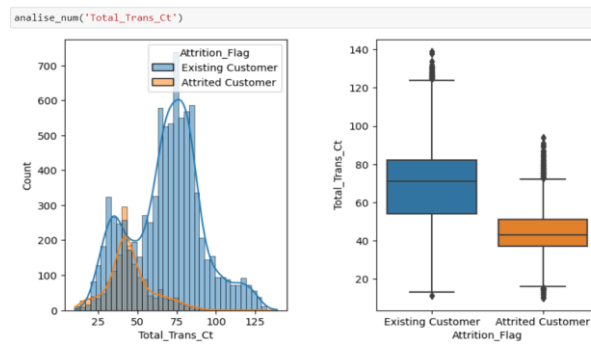


Figure 27: Totais de variação de montantes

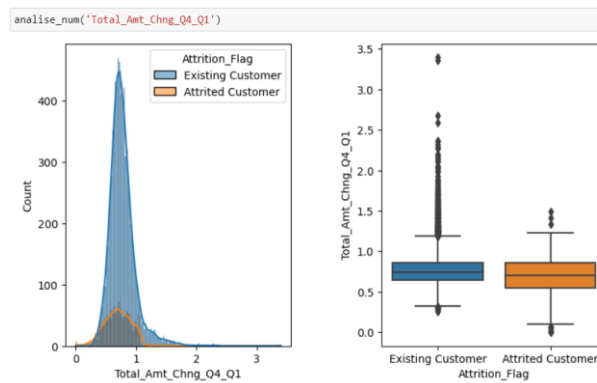
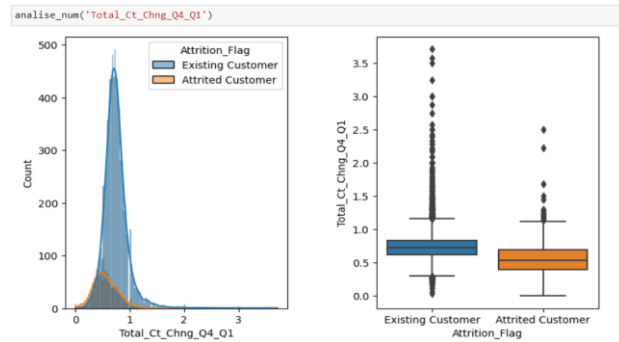


Figure 28: Totais de variação de contas de crédito



Os atributos como: Total_Trans_Ct, Total_Trans_Amt, Total_Revolving_Bal, Total_Ct_Chng_Q4_Q1 e Avg_Utilization_Ratio apresentam maior influencia em determinar se um cliente deixa o serviço ou não. Enquanto as variáveis, Age e month_on_book são irrelevantes para decidir se um cliente sai ou não do serviço de credito. Para representação da correlação entre pares de variáveis nos lados, foi construída a matriz de correlação calculada através de coeficiente de correlações que quantifica a associação entre as características do conjunto de dados, como se pode verificar na figura abaixo.

Figure 29: Matriz de correlação



Nota – se que as diagonais desta matriz apresentam o valor 1 na interseção de x e y. Nas restantes interseções, um valor próximo de 1 significa que existe uma forte correlação positiva entre essas 2 variáveis. A figura a seguir apresenta o comportamento dos atributos em relação ao target de destino.

Figure 30: Relação entre atributos e o target de destino



As características dos dados em relação ao target de destino, ilustra que há uma relação não linear entre a variável de destino e a variável de entrada e muitas variáveis de entrada não possuem distribuição gaussiana.

- **Análise de valores categóricos em relação as instâncias de destino**

Entretanto, as variáveis categóricas também apresentam suas inconsistências em relação ao target de destino como se pode observar nas características a seguir.

```

> analise_cat=['Attrition_Flag', 'Gender', 'Dependent_count', 'Education_Level', 'Marital_Status',
              'Income_Category', 'Card_Category', 'Total_Relationship_Count', 'Months_Inactive_12_mon',
              'Contacts_Count_12_mon' ]
    
```

```

> def call():
    for i in range(len(analise_cat)):
        print(dados[analise_cat[i]].unique())
        print(dados[analise_cat[i]].value_counts())
        print("*****20*\n")
    call()

['Existing Customer' 'Attrited Customer']
Existing Customer    8500
Attrited Customer    1627
Name: Attrition_Flag, dtype: int64
*****
    
```

Como se pode verificar no resultado do código acima, nota-se desbalanceamento entre os clientes fiéis aos cartões de crédito e os que abandonam estes serviços. Contudo, a análise foi refinada para identificar os atributos categóricos que mais influenciam o desequilíbrio entre as instâncias finais, graficamente apresentamos a seguir os resultados das

características de cada atributo em relação a saída.

```
# CODIFICACAO DE TARGET SENDO 0 = CLIENTES EXISTENTES E 1 CLIENTES COM CARACTER DE ABANDONO
dados['Attrition_Flag']=dados['Attrition_Flag'].replace({'Existing Customer':0,'Attrited Customer':1})

def Aval_cat(var):

    inter=dados.groupby(var)['Attrition_Flag'].mean()
    order=inter.index

    plt.subplot(1,2,1)
    sns.countplot(x=var,data=dados,order=order);

    plt.subplot(1,2,2)
    plt.pie(inter,labels=inter.index,autopct="%0.1f%%",radius=1.2);

    plt.tight_layout()
    plt.subplots_adjust(left=0.1,
                        bottom=0.1,
                        right=1.3,
                        top=0.9,
                        wspace=0.4,
                        hspace=0.8)

# À medida que aumenta o número de contactos com os clientes, aumenta a percentagem de clientes que abandonam os serviços.
Aval_cat('Contacts_Count_12_mon')
```

Figure 31: Características dos clientes contactados nos últimos 12 meses

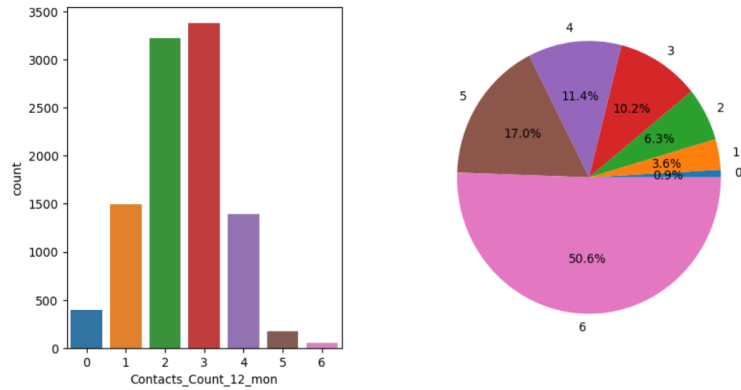


Figure 32: Tempo de inatividade e target final

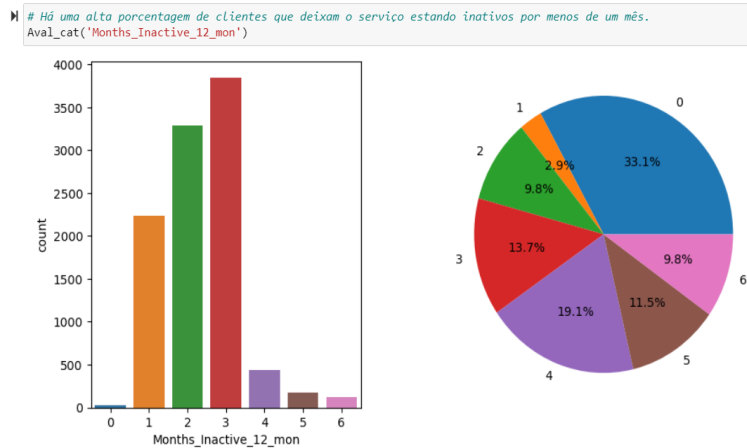
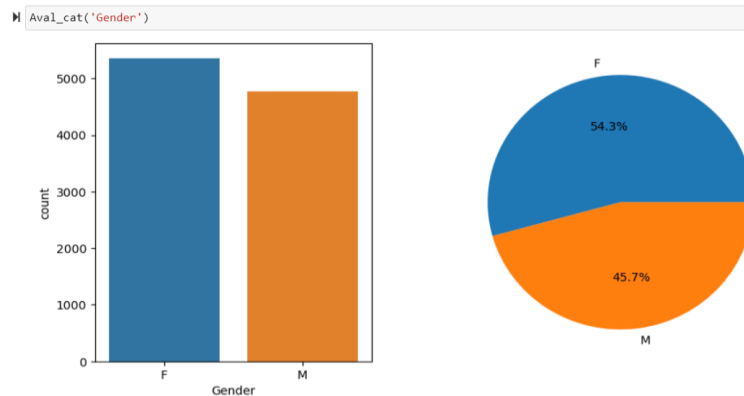


Figure 33: Género



Os resultados ilustram que entre os atributos que mais influenciam no abandono dos serviços de cartão de crédito se destacam contatos e tempo de serviço sendo género um atributo menos importante, mas interessante para o modelo, pois apesar da diferença não ser igual em relação aos outros atributo verifica-se que as mulheres são mais fiéis aos serviços em relação aos clientes de sexo masculino.

3.2 Preprocessamento de dados

Após a visualização e análise dos dados brutos, foram aplicados métodos de transformação, escalonamento e balanceamento para preparar os dados para o processo de treinamento e obter melhores resultados com os algoritmos de machine learning. A primeira etapa desse processo foi a divisão das instâncias entre previsores e a classes. Em seguida, os atributos categóricos foram concatenados com os atributos numéricos para criar um único conjunto de dados submetido às técnicas de preprocessamento.

Observou-se a necessidade de estabilizar a variância e reduzir a assimetria dos dados. Para isso, foi aplicada a técnica conhecida como PowerTransformer, que transforma os valores em uma distribuição mais normal. Essa técnica é indicada quando se deseja uma distribuição normal para os dados e suporta métodos de transformação, como observado neste conjunto de dados.

A padronização dos dados é um requisito comum para muitos estimadores de aprendizado de máquina. (scikit-learn, 2022), Portanto, após a aplicação do PowerTransformer, foi aplicado o StandardScaler para padronizar os dados, mantendo os atributos na mesma escala e evitando que os algoritmos sejam influenciados por atributos com valores menores.

Essas etapas de preprocessamento são importantes para garantir que os dados estejam adequados para a aplicação dos algoritmos de machine learning, melhorando a qualidade dos resultados obtidos.

- **Tratamento de dados desbalanceados**

Após a aplicação das técnicas para distribuição normal e padronização dos atributos, foi necessário verificar o equilíbrio entre as instâncias negativas e positivas com vista a evitar que o o comportamento dos dados brutos não influencie o desempenho dos algoritmos. O código a seguir ilustra os resultados obtidos.

```
▶ y.value_counts()
In [ ]: 0    8500
        1    1627
        Name: Attrition_Flag, dtype: int64
```

Observou-se um desequilíbrio entre as instâncias positivas (0) e negativas (1) no conjunto de dados. Para lidar com esse desequilíbrio, duas técnicas foram consideradas: UnderSampling (redução da classe majoritária) e OverSampling (aumento da classe minoritária). Embora o OverSampling seja robusto, pode resultar em replicação excessiva das características e não funcionar bem na prática.

Para evitar esse problema, optou-se por aplicar a técnica de UnderSampling. Entre as técnicas de UnderSampling disponíveis, selecionou-se a NearMiss, que utiliza a menor distância média dos K-vizinhos mais próximos. Essa técnica reduz a perda de informação em comparação com outras técnicas de UnderSampling. A seguir apresentamos os resultados obtidos.

```
▶ y_nm.value_counts()
0    1627
1    1627
Name: Attrition_Flag, dtype: int64
```

De salientar que contrariamente das técnicas de OverSampling que devem ser aplicadas apenas aos dados de treinamento e não aos de testes, as técnicas de UnderSampling podem ser aplicadas aos dados antes da divisão entre os dados de teste e treinamento vistos que nenhuma das características existentes foi replicada senão a redução das já existentes.

A seguir armazenamento os dados pré-processados para testá-los com diferentes algoritmos

```
▶ with open("Preprocessado.pkl", mode = 'wb') as f:
    pickle.dump([X_train, X_test, y_train, y_test], f)
```

3.2 Treinamento do modelo

No estágio de treinamento do modelo, após o processo de preprocessamento dos dados mencionado anteriormente, adotou-se uma abordagem cuidadosa que envolveu o treinamento de modelos utilizando diversos algoritmos, como Random forest, Decision tree, SVM, KNN e Redes neurais. Essa estratégia foi adotada com o objetivo de identificar qual algoritmo produziria os melhores resultados para a tarefa em questão.

Esta abordagem englobou várias etapas críticas, incluindo a otimização de parâmetros

para cada algoritmo, a realização de validação cruzada para avaliar o desempenho dos modelos sob diferentes condições e a execução de testes de significância estatística para determinar quais diferenças nos resultados eram estatisticamente relevantes.

Além disso, a utilização de múltiplos algoritmos também contribuiu para aumentar a confiabilidade dos resultados, reduzindo a dependência de uma única abordagem e garantindo que a escolha final do modelo fosse fundamentada em uma avaliação abrangente e robusta. Para otimizar ainda mais o processo de desenvolvimento e ajuste do modelo, os dados foram armazenados de forma eficiente, evitando a necessidade de reexecutar todo o processo sempre que fossem feitas alterações nos algoritmos ou nos parâmetros, economizando tempo e recursos. A seguir iremos observar os resultados obtidos para cada algoritmo.

a) Resultados de Random Forest Classifier

A floresta aleatória é um estimador que combina várias árvores de decisão para melhorar a precisão das previsões. Ao utilizar o valor predefinido de 100 árvores, alcançou um desempenho de 93,9% e uma taxa de acerto de 94% com os parâmetros-padrão. Como podemos observar os resultados da acurácia abaixo.

```
import pickle
with open ("Preprocessado.pkl", mode = 'rb') as f:
    X_train, X_test, y_train, y_test = pickle.load(f)
```

RANDOMFOREST

```
from sklearn.ensemble import RandomForestClassifier
dt = RandomForestClassifier()
dt.fit(X_train, y_train)
dt.score(X_test, y_test)
```

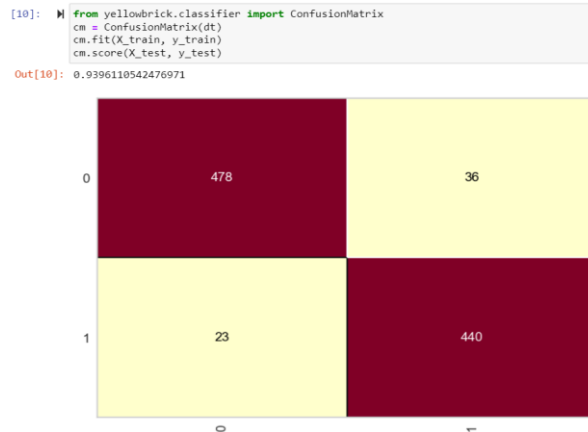
```
7]: 0.9396110542476971
```

```
from sklearn.metrics import classification_report
print(classification_report(y_test, predict))
```

	precision	recall	f1-score	support
0	0.95	0.93	0.94	514
1	0.92	0.95	0.94	463
accuracy			0.94	977
macro avg	0.94	0.94	0.94	977
weighted avg	0.94	0.94	0.94	977

Para medição da taxa de assertividade dos verdadeiros positivos e falsos negativos nas duas instâncias (0 e 1), imprimimos a matriz de confusão. Como se pode observar a seguir:

Figure 34: Matriz de confusão de avaliação do modelo com random forest



Conforme podemos observar nesta matriz, os verdadeiros positivos são pouco confundidos, assim como os falsos negativos. Além disso, a frequência dos resultados das previsões nas categorias positivas e negativas (0 e 1) é aproximada, o que reflete a estabilidade entre as instâncias.

b) Resultados da Árvore de decisão

O objetivo da árvore de decisão é criar um modelo que preveja o valor de uma variável de destino aprendendo regras de decisão simples inferidas dos recursos de dados. (scikit-learn, 2022) Este algoritmo alcançou um percentual de 89.1% com parâmetros predefinidos e uma acurácia de 89%. Conforme os resultados abaixo.

DECISIONTREE

```
[11]: In [11]: from sklearn.tree import DecisionTreeClassifier
Dt = DecisionTreeClassifier()
Dt.fit(X_train, y_train)
Dt.score(X_test, y_test)

Out[11]: 0.8915046059365405

In [13]: In [13]: from sklearn.metrics import accuracy_score, classification_report
accuracy_score(y_test, prevision)

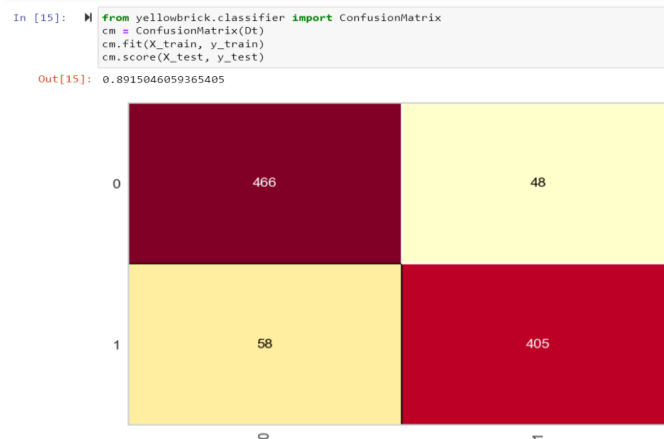
Out[13]: 0.8915046059365405

In [39]: In [39]: print(classification_report(y_test, prevision))
```

	precision	recall	f1-score	support
0	0.89	0.91	0.90	514
1	0.89	0.87	0.88	463
accuracy			0.89	977
macro avg	0.89	0.89	0.89	977
weighted avg	0.89	0.89	0.89	977

Não obstante, foi aplicada a técnica de matriz de confusão para aferir a taxa de acertos nos verdadeiros positivos e falsos negativos. Ela exhibe a distribuição dos registros em termos de suas classes atuais e de suas classes previstas. (IBM, 2023), A seguir observamos os resultados.

Figure 35: Matriz de confusão de avaliação do modelo com arvore de decisão



Como se pode verificar, dos 514 verdadeiros positivos, algoritmos acerta 466 e confunde 48 correspondente a 9.3% de falhas, por outro lado dos 463 das características dos falsos negativos, o modelo acertou 405 e confunde 58 equivalente a 12.5%. Voltaremos a testá-lo após a etapa de hiperparametização.

c) Resultados de Support Vectorial Machine (SVM)

O principal objetivo deste algoritmo é encontrar um hiperplano de decisão que seja capaz de separar de forma distintiva os pontos de dados em classes. No contexto deste conjunto de dados, o algoritmo alcançou um desempenho de 86,4% de precisão, o que também corresponde à taxa de acerto. Os resultados da avaliação, incluindo a matriz de confusão, estão apresentados na figura a seguir.

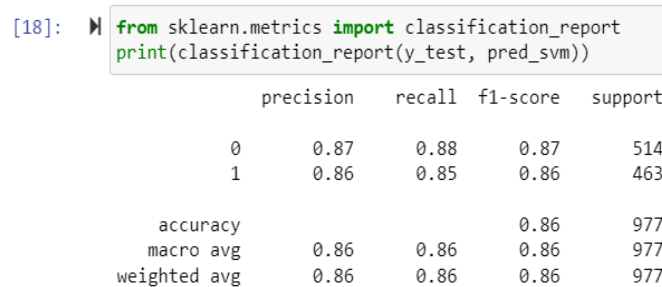
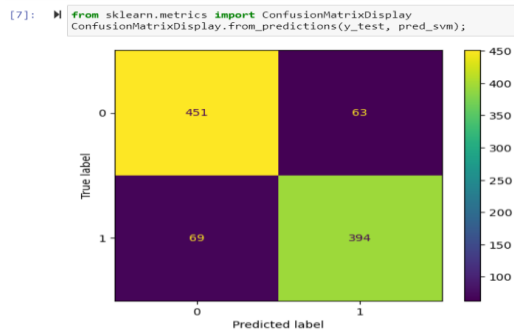


Figure 36: Matriz de confusão de avaliação do modelo com SVM



Como observado, não existe um desequilíbrio significativo entre as instâncias negativas e positivas, o que não compromete seu desempenho quando aplicado em casos reais. Mas enaltecer ainda que os resultados decisivos serão observados após a etapa de hiperparametização.

d) Resultados de KNN

O método de vizinho mais próximo (KNN) busca encontrar amostras de treinamento mais próximas em relação a um novo ponto e prevê seu rótulo com base nelas. O número de amostras pode ser fixo (aprendizado do K vizinho mais próximo) ou variar com base na densidade local (aprendizado do vizinho baseado no raio). A distância pode ser qualquer medida métrica, sendo a distância euclidiana comum. Os métodos baseados em vizinhos são considerados não generalizáveis, pois “lembram” todos os dados de treinamento. No entanto, neste conjunto de dados, o KNN não apresentou um desempenho satisfatório, conforme mostrado nos resultados seguintes.

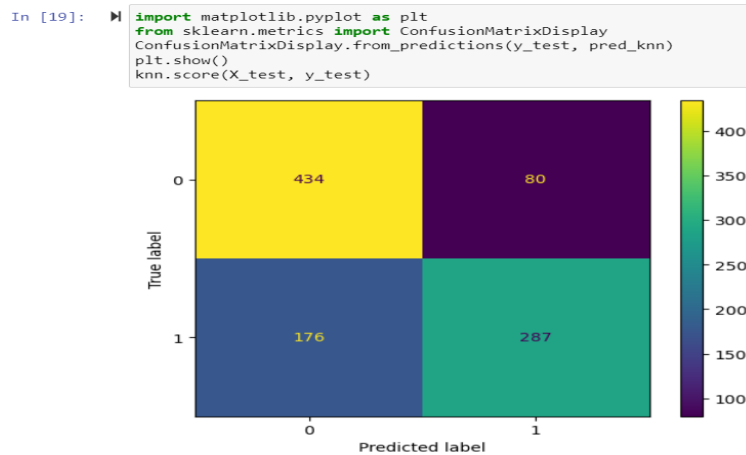
```
In [20]: In [20]: from sklearn.neighbors import KNeighborsClassifier
knn = KNeighborsClassifier().fit(X_train, y_train)
knn.score(X_test, y_test)
```

Out[20]: 0.7379733879222109

```
In [22]: In [22]: from sklearn.metrics import classification_report
print(classification_report(y_test, pred_knn))
```

	precision	recall	f1-score	support
0	0.71	0.84	0.77	514
1	0.78	0.62	0.69	463
accuracy			0.74	977
macro avg	0.75	0.73	0.73	977
weighted avg	0.74	0.74	0.73	977

Figure 37: Matriz de confusão de avaliação do modelo com KNN



e) Resultados da Rede Neural

A Multi-layer perceptron (MLP) também designado por redes neurais são sistemas de computação com nós interconectados que funcionam como os neurônios do cérebro humano. Entretanto, testado este algoritmo com parâmetros predefinidos, obteve resultados que a seguir apresentamos.

Neural network

```
: In ► from sklearn.neural_network import MLPClassifier
nn = MLPClassifier().fit(X_train, y_train)
nn.score(X_test, y_test)
```

C:\Users\leone\Anaconda3\lib\site-packages\sklearn\nn\mlp\mlp.py:100: Warning: Maximum iterations (200) reached at iteration 199. Consider increasing the number of iterations using the parameter max_iter.

[24]: 0.8628454452405322

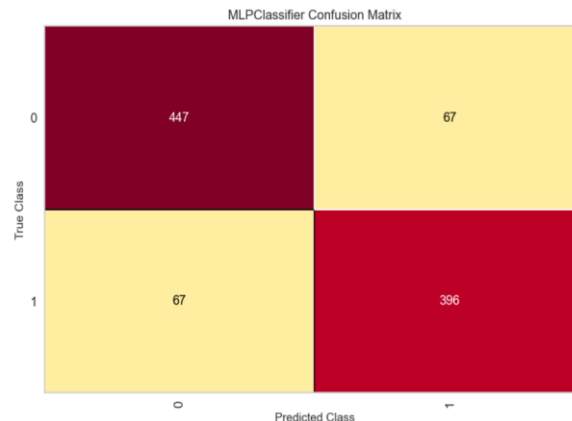
```
from sklearn.metrics import classification_report
print(classification_report(y_test, pred_nn))
```

	precision	recall	f1-score	support
0	0.87	0.87	0.87	514
1	0.86	0.86	0.86	463
accuracy			0.86	977
macro avg	0.86	0.86	0.86	977
weighted avg	0.86	0.86	0.86	977

Como se pode observar, numa precisão de 87% o algoritmo consegue prever clientes leais aos serviços de crédito em 87% correspondente a 514 e numa precisão de 86% o algoritmo deteta 86% clientes que abandonam os serviços de crédito correspondentes a 463 e em geral o algoritmo obteve uma acurácia de 86%.

Não obstante foi necessário identificar onde o modelo se confunde e a taxa de assertividade que o modelo apresenta, para o efeito foi utilizada a técnica de matriz de confusão. A seguir apresentamos os resultados desta técnica.

Figure 38: Matriz de confusão de avaliação da rede neural



3.3 Hiperparametrização dos algoritmos

A hiperparametrização é o processo de encontrar as melhores configurações de parâmetros para um algoritmo de aprendizagem de máquina (MICROSOFT AZURE, 2023). Este processo pode ser realizado de forma discreta, contínua ou aleatória. No contexto deste estudo, optou-se pela abordagem discreta.

A escolha da técnica discreta baseia-se na necessidade de definir valores específicos para os parâmetros, em oposição a intervalos contínuos ou valores aleatórios. Isso proporciona uma maior precisão e controle sobre as configurações dos parâmetros, tornando mais fácil a compreensão e a reprodução dos resultados.

Entretanto, procedeu-se ao teste dos parâmetros de cada um dos cinco algoritmos anteriormente mencionados (RandomForest, DecisionTree, SVM, Knn e Redes Neurais), com o objetivo de identificar as melhores configurações e, posteriormente, refinar o treino dos modelos. A seguir, serão apresentados os parâmetros selecionados para cada algoritmo.

a) Otimização de parâmetros da árvore de decisão

Resultados obtidos quando aplicada a técnica contínua para selecionar os melhores parâmetros da árvore de decisão.

1. Tuning de parametros para DecisionTreeClassifier

```
In [16]: ▶ # Todos parametros da arvore de decisao
parametros = {'criterion': ['gini', 'entropy'],
              'splitter': ['best', 'random'],
              'min_samples_split': [2, 6, 10],
              'min_samples_leaf': [1, 5, 10]}

In [17]: ▶ Grid_search = GridSearchCV(estimator=DecisionTreeClassifier(), param_grid=parametros)
Grid_search.fit(X_concatenado, y_concatenado) # treinamento dos dados concatenados
melhores_parametros = Grid_search.best_params_ # Verificacao dos melhores parametros
melhor_resultado = Grid_search.best_score_ # verificacao do melhor resultado
print(melhores_parametros)
print(melhor_resultado)

{'criterion': 'entropy', 'min_samples_leaf': 5, 'min_samples_split': 10, 'splitter': 'best'}
0.9087316554413329
```

Como observamos acima, aplicados os parâmetros padrões do algoritmo, o modelo alcançou 89.1%, já com a otimização de parâmetros, melhorou para 90.8%.

b) Otimização de parâmetros de Random Forest

Ao contrário das árvores de decisão, as florestas randômicas têm um desempenho inferior quando aplicada a técnicas de otimização de parâmetros. Com os parâmetros predefinidos, o algoritmo alcançou 93,9% de desempenho, enquanto com otimização de parâmetros reduziu para 93,7%. No entanto, é importante ressaltar que em algumas situações, os algoritmos têm um desempenho melhor com os parâmetros predefinidos.

2. Tuning de parametros RandomForestClassifier

```
[18]: ▶ # TODOS PARAMETROS DO ALGORITMO COM VISTA A SELECIONAR OS QUE APRESENTAM MELHORES DESEMPENHOS
      parametros = {'criterion': ['gini', 'entropy'],
                    'n_estimators': [10, 40, 100, 150],
                    'min_samples_split': [2, 6, 10],
                    'min_samples_leaf': [1, 5, 10]}

[19]: ▶ # COMBINACAO DE PARAMETROS
      Random_search = GridSearchCV(estimator = RandomForestClassifier(), param_grid=parametros)
      Random_search.fit(X_concatenado, y_concatenado) # treinamento dos dados concatenados
      Melhores_parametros = Random_search.best_params_ # Verificacao dos melhores parametros
      Melhor_resultado = Random_search.best_score_ # verificacao do melhor resultado
      print(Melhores_parametros)
      print(Melhor_resultado)

      {'criterion': 'entropy', 'min_samples_leaf': 1, 'min_samples_split': 6, 'n_estimators': 100}
      0.93700389932648
```

c) Otimização de parâmetros do SVM

As máquinas de vetores de suporte é um dos algoritmos que melhorou seu desempenho com a técnica de ajuste de parâmetros como podemos observar os resultados abaixo.

3. Tuning de parametros de SVM

```
in [20]: ▶ parametros = {'tol': [0.001, 0.00003, 0.000000001], 'kernel': ['rbf', 'sigmoid', 'linear', 'poly'],
                       'C': [1.0, 2.0, 5.0]}

in [21]: ▶ SVM_grid_search = GridSearchCV(estimator = SVC(), param_grid = parametros)
      SVM_grid_search.fit(X_concatenado, y_concatenado)
      Bom_parametro = SVM_grid_search.best_params_
      Bom_resultado = SVM_grid_search.best_score_
      print(Bom_parametro)
      print(Bom_resultado)

      {'C': 1.0, 'kernel': 'rbf', 'tol': 0.001}
      0.8752305329079523
```

Portanto observa-se uma melhoria vistas que com os parâmetros predefinidos o algoritmo obteve 86.4% do seu desempenho tendo melhorado para 87.5%.

d) Otimização de parâmetros da rede neural

As redes neurais por sua vez apresentaram uma melhoria com a aplicação da técnica de otimização de parâmetros. Portanto, com os parâmetros predefinidos o desempenho foi de 86.2%, já após a seleção de melhores parâmetros melhorou para 88.1% como podemos ver os resultados a seguir:

4. Tuning de parâmetros de neural network

```
[10]: ▶ parametros = {'hidden_layer_sizes': [2,4, 10, 20, 100, 150], 'activation': ['relu', 'logistic', 'tahn'],
'solver': ['adam', 'sgd'], 'batch_size': ['auto', 100, 200, 300]}

[11]: ▶ Rede_grid_search = GridSearchCV(estimator = MLPClassifier(), param_grid = parametros)
Rede_grid_search.fit(X_concatenado, y_concatenado)
melhorRede_parametros = Rede_grid_search.best_params_
Bom_resultado_Rede = Rede_grid_search.best_score_
print(melhorRede_parametros)
print(Bom_resultado_Rede)

0.9325114 0.77258844 0.84265532 0.771352 0.84849675 0.78334019
0.85833534 0.78887345 0.85587101 0.79502162 0.85249581 0.79594423
0.82759778 0.75722226 0.84234858 0.75844925 0.85341416 0.76797259
0.85710315 0.77350538 0.85372232 0.78057332 0.85587569 0.77965402
nan nan nan nan nan nan
nan nan nan nan nan nan
nan nan nan nan nan nan
nan nan nan nan nan nan
nan nan nan nan nan nan
nan nan nan nan nan nan
nan nan nan nan nan nan
warnings.warn(
{'activation': 'relu', 'batch_size': 100, 'hidden_layer_sizes': 150, 'solver': 'adam'}
0.8810710150064989
```

e) Otimização de parâmetros de KNN
A aplicação das técnicas de otimização de parâmetros melhorou de forma significativa o desempenho deste algoritmo, vistos que com os seus parâmetros predefinidos obteve 73.7% de desempenho, e após a seleção das melhores técnicas melhorou para 79.7%, como podemos observar os resultados a seguir.

5. Tuning de parâmetros de Knn

```
[24]: ▶ parametros = {'n_neighbors': [3, 5, 10, 20], 'p': [1, 2]}
KNN_search = GridSearchCV(estimator = KNeighborsClassifier(), param_grid=parametros)
KNN_search.fit(X_concatenado, y_concatenado) # treinamento dos dados concatenados
K_melhores_parametros = KNN_search.best_params_ # Verificacao dos melhores parametros
K_melhor_resultado = KNN_search.best_score_ # verificacao do melhor resultado
print(K_melhores_parametros)
print(K_melhor_resultado)

{'n_neighbors': 20, 'p': 1}
0.7971679073614557
```

3.4 Validação Cruzada

O método de validação cruzada k-fold divide o conjunto de dados em k subconjuntos, sendo um utilizado para teste e os k-1 restantes para treinamento dos modelos. Esse processo é repetido k vezes, alternando o subconjunto de teste. (Silva & Borges, 2019), A k-fold foi a técnica de validação cruzada aplicada neste trabalho, pois permite calcular a acurácia sobre os erros encontrados ao final das k iterações, fornecendo uma medida confiável da capacidade do modelo. O conjunto de dados foi dividido em 10 partes para treinamento e teste, avaliação estatística de precisão e geração de métricas de desempenho para cada algoritmo testado, conforme os resultados a seguir:

Variância e desvio padrao dos algoritmos

```
[20]: ▶ # Criação da lista dos resultados obtidos
import pandas as pd
Resultados = pd.DataFrame({'Arvore': resultados_decision_tree, 'Random': resultados_random_forest, 'SVM': resultados_svm,
                           'Rede_neural': resultados_rede_neural, 'KNN': resultados_knn})

[21]: ▶ Resultados
Out[21]:
```

	Arvore	Random	SVM	Rede_neural	KNN
0	0.907806	0.935465	0.87184	0.87184	0.780576

```
[22]: ▶ Resultados.mean()
Out[22]: Arvore      0.907806
Random      0.935465
SVM         0.871840
Rede_neural 0.871840
KNN         0.780576
dtype: float64
```

A técnica de validação cruzada fornece resultados diferentes dos métodos de avaliação anteriores, pois não apenas mede a precisão de um modelo, mas também indica quão representativo o conjunto de dados é e quão sensível o modelo é a variações nos dados. Em relação aos resultados, observa-se um melhor desempenho do Random Forest, seguido pela árvore de decisão, enquanto as máquinas de suporte de vetor e redes neurais apresentam resultados semelhantes, apenas o KNN não apresenta um bom desempenho. Na próxima etapa, serão gerados modelos preditivos com base nos algoritmos de melhor desempenho, combinando classificadores para determinar qual algoritmo é considerado o melhor, atendendo a um nível mínimo de confiança.

3.5 Geração do modelo final

Nesta etapa, foram selecionados os algoritmos que apresentaram melhor desempenho na fase de validação cruzada e foram submetidos ao treinamento com os dados de treino e parâmetros obtidos da técnica de otimização de parâmetros e por último foram guardados para sua reutilização na fase seguinte. Abaixo apresentamos os procedimentos executados.

- **Carregamento de dados e importação dos algoritmos**

```

In [2]: ▶ import pickle
        with open ("Preprocessado.pkl", mode = 'rb') as f:
            X_train, X_test, y_train, y_test = pickle.load(f)

In [3]: ▶ X_train.shape, y_train.shape
Out[3]: ((2277, 32), (2277,))

In [4]: ▶ X_test.shape, y_test.shape
Out[4]: ((977, 32), (977,))

In [5]: ▶ # UNIR OS DADOS
        import numpy as np
        X_credit = np.concatenate((X_train, X_test), axis=0)
        y_credit = np.concatenate((y_train, y_test), axis=0)

In [6]: ▶ # Verificar apos a uniao dos dados
        X_credit.shape, y_credit.shape
Out[6]: ((3254, 32), (3254,))

In [7]: ▶ # importacao dos algoritmos selecionados como melhores
        from sklearn.ensemble import RandomForestClassifier
        from sklearn.neural_network import MLPClassifier
        from sklearn.tree import DecisionTreeClassifier
        from sklearn.svm import SVC

```

- **Treinamento e geração de modelos**

```

In [1]: ▶ import pickle
        with open ("Preprocessado.pkl", mode = 'rb') as f:
            X_train, X_test, y_train, y_test = pickle.load(f)

In [2]: ▶ X_train.shape, y_train.shape
Out[2]: ((2277, 32), (2277,))

In [3]: ▶ X_test.shape, y_test.shape
Out[3]: ((977, 32), (977,))

In [4]: ▶ # UNIR OS DADOS
        import numpy as np
        X_credit = np.concatenate((X_train, X_test), axis=0)
        y_credit = np.concatenate((y_train, y_test), axis=0)

In [5]: ▶ # Verificar apos a uniao dos dados
        X_credit.shape, y_credit.shape
Out[5]: ((3254, 32), (3254,))

In [11]: ▶ # importacao dos algoritmos selecionados como melhores
        from sklearn.ensemble import RandomForestClassifier
        from sklearn.neural_network import MLPClassifier
        from sklearn.tree import DecisionTreeClassifier
        from sklearn.svm import SVC

```

```

In [12]: ▶ prediction_random_forest = RandomForestClassifier(criterion='entropy', min_samples_leaf=1, min_samples_split=6, n_estimators=
prediction_random_forest.fit(X_credito, y_credito)

Out[12]: ▼ RandomForestClassifier
RandomForestClassifier(criterion='entropy', min_samples_split=6)

In [13]: ▶ # Treinamento da rede neural com as metricas selecionadas
prediction_rede_neural = MLPClassifier(hidden_layer_sizes = 150, activation='relu', batch_size=100, solver='adam')
prediction_rede_neural.fit(X_credito, y_credito)

C:\Users\leone\Anaconda3\lib\site-packages\sklearn\network\_multilayer_perceptron.py:702: ConvergenceWarning: Stochastic
Optimizer: Maximum iterations (200) reached and the optimization hasn't converged yet.
  warnings.warn(

Out[13]: ▼ MLPClassifier
MLPClassifier(batch_size=100, hidden_layer_sizes=150)

In [14]: ▶ # Treinamento com Arvore de decisao
prediction_arvore = DecisionTreeClassifier(criterion='entropy', min_samples_leaf=5, min_samples_split=10, splitter = 'best')
prediction_arvore.fit(X_credito, y_credito)

Out[14]: ▼ DecisionTreeClassifier
DecisionTreeClassifier(criterion='entropy', min_samples_leaf=5,
min_samples_split=10)

In [15]: ▶ #Treinamento com SVM
prediction_SVM = SVC(tol = 0.001, kernel = 'rbf', C = 5.0, probability=True )
prediction_SVM.fit(X_credito, y_credito)

Out[15]: ▼ SVC
SVC(C=5.0, probability=True)

In [16]: ▶ # SALVAR OS CLASSIFICADOR
import pickle
pickle.dump(prediction_random_forest, open('Random_Forest_finalizado.sav', 'wb'))
pickle.dump(prediction_rede_neural, open('NEURAL_NETWORK_finalizado.sav', 'wb'))
pickle.dump(prediction_arvore, open('DECISION_TREE_finalizado.sav', 'wb'))
pickle.dump(prediction_SVM, open('SVM_finalizado.sav', 'wb'))

```

Como se pode observar, os modelos aqui gerados para os quatro algoritmos selecionados, resultam dos melhores parâmetros obtidos na etapa de ajuste de parâmetros.

3.6 Testes dos modelos

Os testes a seguir correspondem a um ID de cliente selecionado de forma aleatória no conjunto de dados submetido aos quatro modelos gerados com intuito de aferir o seu desempenho. Onde 2500 é o ID do cliente, 0 corresponde ao cliente fiel aos serviços de crédito e 1 corresponde ao cliente que abandona estes serviços.

```

In [14]: # Simulacao de funcionamento
cliente_real = X_credit[2500] # A POSICAO DAS CARATERISTICAS DO CLIENTE
cliente_real

Out[14]: array([-0.9434357, -0.21589388, 1.49583822, -0.49808606, -0.23170775,
-0.41485718, -0.42007589, 1.07733799, -0.79850685, -0.28260887,
2.15813389, -0.40085868, -0.42267548, -0.73643691, -0.35121215,
-0.10764411, -0.044484, -0.24079371, -0.04362869, 0.51632686,
0.37112776, -0.53041502, 0.71401146, -0.40608359, -0.75335908,
0.91793895, -1.43858906, 0.05495142, 0.08205876, 0.09421216,
-0.18218596, 1.54874035])

In [15]: cliente_real.shape

Out[15]: (32,)

In [16]: cliente_real = cliente_real.reshape(1, -1)
cliente_real.shape

Out[16]: (1, 32)

In [17]: random_forest.predict(cliente_real) # 1 Corresponde ao cliente que abandona os serviços de credito E 0 clientes fieis

Out[17]: array([0], dtype=int64)

In [18]: rede_neural.predict(cliente_real) |

Out[18]: array([0], dtype=int64)

In [19]: arvore.predict(cliente_real)

Out[19]: array([0], dtype=int64)

```

```
In [20]: SVM.predict(cliente_real)
```

```
Out[20]: array([0], dtype=int64)
```

Como observado, os quatro algoritmos apresentam 0 como saída o que significa que o cliente foi previsto como fiel aos serviços de crédito.

Igualmente foi testado outro cliente com comportamento contrário tendo como ID 902 e os modelos apresentaram os resultados seguintes:

```

In [50]: # Simulacao de funcionamento
cliente_real = X_credit[902] # A POSICAO DAS CARATERISTICAS DO CLIENTE
cliente_real

Out[50]: array([-0.9434357, -0.21589388, -0.66852149, -0.49808606, -0.23170775,
-0.41485718, 2.38052223, -0.92821381, -0.79850685, -0.28260887,
2.15813389, -0.40085868, -0.42267548, -0.73643691, -0.35121215,
-0.10764411, -0.044484, -0.24079371, -0.29300734, 0.51632686,
-0.25989505, -1.16583366, 0.71401146, -0.40608359, -1.40336283,
-1.66614912, -0.5735263, -3.77675749, -1.24560704, -1.20156816,
-2.35297955, -1.20167778])

In [51]: cliente_real.shape

Out[51]: (32,)

In [52]: cliente_real = cliente_real.reshape(1, -1)
cliente_real.shape

Out[52]: (1, 32)

In [53]: random_forest.predict(cliente_real) # 1 Corresponde ao cliente que abandona os serviços de credito E 0 clientes fieis

Out[53]: array([1], dtype=int64)

In [54]: rede_neural.predict(cliente_real)

Out[54]: array([1], dtype=int64)

In [55]: arvore.predict(cliente_real)

Out[55]: array([1], dtype=int64)

In [56]: SVM.predict(cliente_real)

Out[56]: array([1], dtype=int64)

```

A saída para este cliente é 1, portanto, nota-se que quando o cliente apresenta comportamentos de abandono aos serviços de crédito todos os modelos são unânimes em

prevê-lo. Abaixo apresentamos a combinação dos algoritmos com ID 190 diferente dos dois acima testados e foi selecionado de forma aleatória.

COMBINACAO DE CLASSIFICADORES

```
In [20]: # PREVISAO DOS ALGORITMOS
cliente_real = X_credit[190] # POSICAO DOS DADOS A PREVER O RESULTADO - HISTORICOS
cliente_real = cliente_real.reshape(1, -1) # PARA COLOCAR NO FORMATO DE MATRIZ
cliente_real.shape

Out[20]: (1, 32)

In [21]: resposta_rede_neural = rede_neural.predict(cliente_real)
resposta_random_forest = random_forest.predict(cliente_real)
resposta_arvore = arvore.predict(cliente_real)
resposta_SVM = SVM.predict(cliente_real)

In [22]: resposta_rede_neural[0], resposta_random_forest[0], resposta_arvore[0], resposta_SVM[0] # TODOS CLASSIFICADORES INDICAM QUE C

Out[22]: (1, 1, 1, 1)
```

Como observado, todos os modelos apresentaram 1 como saída indicando deste modo que o cliente abandonará os serviços de empréstimo.

3.7 Teste de significância

Por último, para avaliação de resultados dos modelos submetemos aos testes de significância. Portanto, um teste de hipótese ou de significância corresponde a uma regra decisória que nos permite rejeitar ou não rejeitar uma hipótese estatística com base nos resultados de uma amostra. (Peternelli, 2023)

A seguir apresentamos os procedimentos executados para execução do teste de hipótese e a confirmação de H_0 (afirmação ou hipótese primária) e H_1 (hipótese alternativa).

TESTE DE SIGNIFICANCIA

```
In [24]: novo_registro = X_credit[1900] # ID DO CLIENTE OU HISTORICOS
novo_registro = novo_registro.reshape(1, -1)
novo_registro.shape

Out[24]: (1, 32)

In [25]: resposta_random_forest = random_forest.predict(novo_registro)
resposta_rede_neural = rede_neural.predict(novo_registro)
resposta_arvore = arvore.predict(novo_registro)
resposta_SVM = SVM.predict(novo_registro)

In [26]: resposta_rede_neural[0], resposta_random_forest[0], resposta_arvore[0], resposta_SVM[0]

Out[26]: (0, 0, 0, 0)

In [27]: probabilidade_random_forest = random_forest.predict_proba(novo_registro)
probabilidade_random_forest

Out[27]: array([[0.99, 0.01]])

In [28]: confianca_random_forest = probabilidade_random_forest.max()
confianca_random_forest

Out[28]: 0.99
```

```

In [29]: ▶ probabilidade_rede_neural = rede_neural.predict_proba(novo_registro) # Visualizacao das probabilidades das classes
          probabilidade_rede_neural
Out[29]: array([[0.99565945, 0.00434055]])

In [30]: ▶ confianca_rede_neural = probabilidade_rede_neural.max() # para verificar a confianca da probabilidade
          confianca_rede_neural
Out[30]: 0.9956594512983938

In [31]: ▶ probabilidade_arvore = arvore.predict_proba(novo_registro) #100% de probabilidade dar certo
          probabilidade_arvore
Out[31]: array([[1., 0.]])

In [32]: ▶ confianca_arvore = probabilidade_arvore.max() # para verificar a confianca da probabilidade
          confianca_arvore
Out[32]: 1.0

In [33]: ▶ probabilidade_SVM = SVM.predict_proba(novo_registro)
          probabilidade_SVM
Out[33]: array([[0.98701709, 0.01298291]])

In [34]: ▶ confianca_SVM = probabilidade_SVM.max() # para verificar a confianca da probabilidade
          confianca_SVM
Out[34]: 0.9870170855580874

```

Até este ponto, nota-se que a hipótese primária ainda é válida, vistos que os quatro algoritmos estão no intervalo da confiança recomendada (acima de 97.5%). Entretanto, foi definida uma confiança mínima de 97.5% para reafirmar as saídas ou outputs dos modelos acima. A seguir ilustramos os resultados obtidos.

```

In [80]: ▶ fiel = 0
          infiel = 1
          confianca_minima = 0.975000
          algoritmos = 0 # para identificar quantos algoritmos foram utilizados

          if confianca_random_forest >= confianca_minima:
              algoritmos += 1
              if resposta_random_forest[0] == 1:
                  infiel +=1
          else:
              fiel +=1
          if confianca_rede_neural >= confianca_minima:
              algoritmos += 1
              if resposta_rede_neural[0] == 1:
                  infiel +=1
              else:
                  fiel +=1
          if confianca_arvore >= confianca_minima:
              algoritmos += 1
              if resposta_arvore[0] == 1:
                  infiel +=1
              else:
                  fiel +=1
          if confianca_SVM >= confianca_minima:
              algoritmos += 1
              if resposta_SVM[0] == 1:
                  infiel +=1
              else:
                  fiel +=1
          if fiel > infiel:
              texto = ('O cliente será fiel aos serviços de credito com base em {} Algoritmos' .format(algoritmos))
              texto = texto.upper()
              print(texto)
          elif fiel == infiel:
              print('Empate,solicitar mais dados do cliente ou aferir se os dados entregues são reais')
          else:
              print('O cliente é infiel aos servicos de credito')

O CLIENTE SERÁ FIEL AOS SERVIÇOS DE CREDITO COM BASE EM 4 ALGORITMOS

```

Por fim o resultado ilustra que a hipótese primária é validada, uma vez que os quatro algoritmos satisfazem a confiança mínima.

4 DESCRIÇÃO DA EMPRESA E AS ATIVIDADES DESENVOLVIDAS DURANTE O ESTÁGIO

Nesta sessão, apresentamos a visão geral da empresa onde foi realizado o estágio curricular e em seguida os detalhes das atividades desenvolvidas.

4.1. Descrição da empresa

A Vision Factory é uma empresa especializada em negócios e serviços financeiros, estabelecida em 2015, com sede principal em Barcelona e filiais em diversos locais da Espanha e na Europa. Sua missão é fornecer soluções abrangentes em negócios, criação digital e suporte a serviços financeiros.

A empresa possui uma ampla gama de áreas de atuação, incluindo análise financeira, desenvolvimento web, marketing, produção de conteúdos educativos e Mídias digitais, gestão de recursos humanos, turismo e gestão de eventos. Com sua expertise e experiência, a Vision Factory busca atender às necessidades de seus clientes, oferecendo serviços de qualidade e soluções personalizadas para impulsionar o sucesso e o crescimento de seus negócios. (VISION FACTORY, 2023)

4.2. Atividades desenvolvidas durante o estágio

Durante o estágio, realizado de 03 de janeiro a 03 de julho de 2023, fui alocado no departamento de análise de dados financeiros. Minhas atividades envolveram a análise das estratégias de negociação, a criação de sinais de negociação e a publicação de artigos nas plataformas da empresa. Além disso, participei de reflexões conjuntas com a equipe do departamento para discutir soluções futuras.

No período de ambientação, tive a oportunidade de participar de um programa de formação de 15 dias, que me permitiu conhecer a empresa em profundidade, entender seus propósitos e expectativas, além de adquirir conceitos básicos sobre o mercado financeiro e estratégias de negociação. Após esse período, pude desempenhar as atividades abaixo descritas, colocando em prática os conhecimentos adquiridos e desenvolvendo habilidades específicas na área de análise financeira.

a) Estudo e publicação de artigo sobre tratamento de dados desbalanceados

Este estudo teve como objetivo apresentar de forma resumida as principais técnicas a serem aplicadas no tratamento de dados desbalanceados durante o processo de análise de dados.

Esta pesquisa está diretamente relacionada com o trabalho anteriormente desenvolvido, que envolveu o treinamento de modelos para a previsão de clientes propensos a abandonar os serviços de crédito.

No trabalho anterior, uma das etapas cruciais também envolveu o tratamento de dados, tornando-se essencial compreender e aplicar técnicas eficazes para lidar com

desequilíbrios nos dados. Com base na revisão da literatura relevante, este estudo apresenta os seguintes resultados:

- **Subamostragem:** reduz-se o número de instâncias da classe majoritária para equilibrar a proporção entre as classes. Pode-se remover aleatoriamente instâncias da classe majoritária ou utilizar técnicas mais sofisticadas, como algoritmos de clusterização ou seleção de instâncias baseadas em critérios específicos.

Neste caso, concluiu-se que o algoritmo NearMiss-3 seja o mais recomendado pelo fato de selecionar instâncias da classe majoritária que são mais distantes dos três exemplos mais próximos da classe minoritária, calculando a média das distâncias entre as instâncias de diferentes classes. (imbalanced-learn, 2022)

- **Reamostragem:** A técnica de reamostragem envolve a manipulação da distribuição de exemplos em um conjunto de dados, com o objetivo de alcançar um equilíbrio mais adequado entre as classes. Contrariamente da subamostragem, a reamostragem consiste no aumento da classe minoritária.

Os estudos concluíram que a combinação do Smote + ENN é poderosa porque não apenas aumenta a representação da classe minoritária, mas também reduz o impacto da classe majoritária no modelo, resultando em um conjunto de dados mais equilibrado e limpo.

O Smote é aplicado para gerar instâncias sintéticas da classe minoritária, aumentando sua representação no conjunto de dados. Em seguida, o ENN é utilizado para remover instâncias redundantes da classe majoritária. Essa abordagem resulta em dados mais limpos e uma distribuição mais balanceada entre as classes.

Essa estratégia é recomendada para melhorar a capacidade de generalização e o desempenho de modelos de aprendizado de máquina em tarefas com classes desbalanceadas. (Wang, Zheng, Ren, Li, & Han, 2021)

Este estudo aprofundado sobre o tratamento de dados desbalanceados trouxe valiosos insights sobre a importância da análise de dados e a busca por soluções eficazes para torná-los mais coerentes e adequados aos algoritmos de aprendizado de máquina.

Além disso, dentro do contexto da empresa, esse estudo serviu como um impulso crucial para o processo de análise de dados financeiros, considerando sua sensibilidade e os potenciais riscos envolvidos caso não seja dada a devida atenção a essa etapa.

Com uma abordagem cuidadosa e soluções eficientes, é possível melhorar a precisão e a confiabilidade dos modelos de aprendizado de máquina aplicados a dados financeiros,

auxiliando na tomada de decisões informadas e minimizando riscos futuros.

b) Análise de ações

Nesta atividade, aplicamos os conhecimentos adquiridos durante a formação inicial na empresa para analisar 20 ações. O objetivo foi identificar zonas de resistência e suporte, a fim de fornecer aos clientes informações sobre os melhores pontos de entrada e os respectivos níveis de stop loss. Isso ajudou a evitar perdas nas negociações de ações. Esta atividade envolveu estagiários e a equipe mais experiente da empresa para obtenção de melhores resultados. Contudo além da implementação dos conhecimentos adquiridos na formação, serviu para adquerir mais habilidades na leitura de gráficos e configuração de estratégias de negociação.

c) Estudo sobre as tendências da inteligência artificial na análise financeira

Neste artigo, foi conduzida uma análise abrangente das tendências emergentes da inteligência artificial (IA) aplicadas à análise financeira. Teve como objetivo explorar o impacto significativo que a IA exerce no campo financeiro, fornecendo insights valiosos sobre como essa tecnologia está revolucionar a maneira como as instituições financeiras abordam o processo de negociação.

Este artigo ofereceu uma visão abrangente das tendências da IA na análise financeira, destacou suas vantagens e forneceu recomendações práticas para a implementação de ferramentas de IA na negociação financeira. E teve como contributo a compreensão e adoção crescente da IA como um componente fundamental da tomada de decisões financeiras eficazes.

d) Análise das estratégias de negociação com backtesting

Backtesting é uma técnica amplamente utilizada no mercado financeiro para avaliar a eficácia de uma estratégia de negociação. (KERNC, 2023)

Durante o estágio, uma das principais atividades foi encontrar soluções com backtesting para análise prévia de estratégias de negociação. Essa ideia permitiu testar diferentes conjuntos de dados e estratégias, proporcionando aos traders uma base sólida para a seleção das melhores estratégias.

Além disso, essa experiência levou à criação de um artigo intitulado "*Como utilizar backtesting para teste de estratégias de negociação*", que teve como objetivo fornecer uma visão geral do funcionamento do backtesting, suas vantagens e desvantagens.

O backtesting demonstrou ser uma ferramenta valiosa para avaliar o desempenho histórico de uma estratégia de negociação e auxiliar na tomada de decisões futuras. No entanto, é importante ressaltar que resultados passados não garantem resultados futuros.

O backtesting deve ser utilizado como uma ferramenta complementar para avaliar o potencial de uma estratégia antes de sua aplicação em tempo real. A iniciativa foi

implementada pela empresa para análise das estratégias de trading.

e) Um tutorial educativo para construção de uma biblioteca em python

A empresa reconhece a importância de fornecer conteúdos digitais educativos, especialmente sobre tecnologias emergentes, para que as pessoas estejam preparadas para as oportunidades do mundo digital em constante evolução.

Ao oferecer essa atividade, a empresa mostra seu compromisso em fornecer recursos de qualidade que ajudam os clientes e o público a se familiarizarem com a linguagem Python. O objetivo é incentivar os leitores a explorar ativamente essa linguagem, despertando interesse e incentivando-os a adquirir habilidades relevantes.

Essa iniciativa fortalece a posição da empresa como líder no fornecimento de conteúdos educativos sobre tecnologia, consolidando sua reputação como uma fonte confiável de conhecimento. Capacitando os clientes e o público em geral a dominarem a linguagem Python, a empresa contribui para uma comunidade mais preparada e atualizada no cenário tecnológico.

f) Construção dum chatbot baseado na API de OpenAI

A OpenAI é uma organização líder em pesquisa e desenvolvimento de inteligência artificial (IA) com o objetivo de disseminar os benefícios dessa tecnologia para o bem-estar da humanidade. (OPENAI, 2023)

Como parte de suas atividades, a empresa solicitou a criação dum chatbot capaz de responder perguntas por voz ou por texto tendo sido utilizado a API da OpenAI para o efeito. O principal propósito dessa iniciativa foi demonstrar a capacidade técnica da empresa na criação de ferramentas de pesquisa e beneficiar o público em geral.

A criação desse chatbot possui um valor educativo significativo, tanto para os clientes quanto para o público em geral. Os procedimentos técnicos utilizados no desenvolvimento foram compartilhados em um artigo intitulado "Criação de Smart Julie com Python e OpenAI", fornecendo um tutorial detalhado dos passos e técnicas utilizadas. Isso fortalece ainda mais o aspecto educativo da empresa.

Ao publicar esse tutorial, a empresa não apenas compartilha seu conhecimento e experiência, mas também promove a aprendizagem e capacitação dos leitores interessados em explorar as capacidades de IA e da linguagem Python. Essa iniciativa reforça o compromisso da empresa em fornecer conteúdo educativo valioso, estimulando o desenvolvimento de habilidades e conhecimento sobre novas tecnologias.

Além disso, ao utilizar o chatbot como base para o tutorial, a empresa demonstra sua capacidade de inovação e pesquisa, consolidando sua reputação como referência no campo de IA. Isso contribui para a disseminação do conhecimento e o avanço da comunidade científica.

Dessa forma, essa atividade tem um impacto educativo duplo: oferece uma ferramenta interativa para o público explorar e obter respostas relevantes, e compartilha conhecimento técnico e metodologias por meio do tutorial. Ambos os aspectos fortalecem a componente educativa da empresa e contribuem para o progresso da IA como um todo.

g) Um estudo sobre análise crítica das estratégias de negociação

Uma revisão bibliográfica detalhada revela que as estratégias de trading baseadas em análises técnicas e fundamentais têm sido amplamente utilizadas e estudadas. A análise técnica se concentra no estudo de padrões de preços passados, utilizando indicadores e gráficos para identificar oportunidades de compra e venda. Já a análise fundamental considera fatores econômicos, financeiros e políticos para determinar o valor intrínseco dos ativos. (Jiwani, 2021)

No entanto, é importante destacar que o sucesso das estratégias de trading não é garantido, pois, as variações comportamentais do mercado devem ser acompanhadas e as técnicas devem aplicadas conforme a realidade. Além do conhecimento e habilidade necessários para implementar uma estratégia de negociação, é fundamental adotar uma abordagem disciplinada e realizar uma análise crítica contínua. Isso envolve ajustar as estratégias de acordo com as mudanças do mercado e gerir adequadamente os riscos envolvidos. (Graham, 2023)

Estudos acadêmicos mostram que a escolha e combinação das estratégias de trading dependem do perfil do investidor, tolerância ao risco e objetivos financeiros. Além disso, a diversificação de estratégias pode ajudar a reduzir o risco e maximizar os retornos. (Hagstrom, 2014)

Com base a literatura concluiu-se que as estratégias de trading podem ser eficazes para alcançar lucros no mercado financeiro, desde que sejam utilizadas com conhecimento, habilidade e disciplina. A análise crítica e o ajuste contínuo dessas estratégias são fundamentais para se adaptar às mudanças do mercado e maximizar os resultados a longo prazo.

h) Desenvolvimento de ideias de negociação utilizando diferentes técnicas e plataformas da empresa

Esta atividade exigiu um estudo prévio sobre as técnicas utilizadas no trading e o significado das velas. No processo de negociação, o recurso a velas (candles) é uma técnica comum na análise gráfica, também conhecida como análise de candlestick. Cada vela representa um período específico, como um dia ou uma hora, e fornece informações sobre o comportamento dos preços de um ativo financeiro (Nison, 2001). O autor defende que, antes de iniciar o processo de negociação, o trader deve possuir sólidos conhecimentos nos seguintes aspectos:

- **Padrões de velas:** É fundamental compreender os diferentes padrões de velas e a sua interpretação. Isso envolve reconhecer os padrões de reversão, continuação e indecisão, bem como a sua relevância nas tendências de mercado.
- **Análise de tendências:** É importante estar ciente das tendências predominantes no mercado, seja de alta, baixa ou lateral. A identificação correta das tendências permite ao trader tomar decisões mais informadas sobre quando comprar ou vender um ativo.
- **Suportes e resistências:** Os níveis de suporte e resistência são áreas-chave nos gráficos de velas. Esses níveis representam preços nos quais o ativo financeiro tende a encontrar suporte ou resistência. Para identificação dos potenciais pontos de entrada e saída.
- **Indicadores técnicos:** Os indicadores técnicos, como médias móveis, osciladores e bandas de Bollinger, podem ser utilizados em conjunto com as velas para confirmar sinais de compra ou venda. Essas ferramentas ajudam a fornecer uma análise mais completa do mercado.
- **Gestão de risco:** A gestão adequada do risco é essencial na negociação. Isso envolve estabelecer stop-loss (limites de perda) e take-profit (metas de lucro) para cada negociação, além de calcular o tamanho adequado da posição com base no risco máximo aceitável.

O conhecimento aprofundado desses aspectos permitirá ao trader tomar decisões mais fundamentadas e reduzir os riscos envolvidos na negociação. Além disso, a prática e o aperfeiçoamento contínuo são fundamentais para desenvolver habilidades sólidas nessa área (Nison, 2001). A seguir apresentamos algumas ideias das soluções desenvolvidas.

- **Identificação de pontos de ação e o comportamento futuro do stock HDFCBANK**

Figure 39: HDFCBANK long position



Neste gráfico, o preço encontrou suporte na linha de tendência ascendente. Podemos obter

uma vela engolfante de alta semanal, o que deve implicar em mais ação de alta. Esperamos que o preço atinja a resistência ascendente que seria nossa entrada para uma posição curta. A partir daí, direcionaríamos o suporte desse canal ascendente. A invalidação desta hipótese seria se obtivéssemos uma vela semanal fechando acima da linha de resistência ascendente. O RSI encontrou suporte em 48, o que também mostra que a tendência pode ter mais pernas, portanto, esperamos que o preço continue a tendência de alta.

- **Controle de stock com comportamento imprevisível**

Figure 40: SPX short position



Neste caso o índice SPX está aproximando-se de uma resistência inclinada para baixo. É possível que ele simule uma ruptura acima da linha de tendência, a fim de enganar os traders, no entanto, é importante prestar atenção para não cair no erro. Esta alta temporária do mercado de baixa está prestes a finalizar e entrar numa movimentação significativa. As sucessivas formações de topos inferiores sugerem a continuação da tendência de baixa. Neste caso, aconselha-se a encerrar posições longas e considerar a entrada em posições curtas ou manter-se à margem do mercado. Este processo pode levar algum tempo, portanto, recomenda-se ainda a paciência ao aguardar as movimentações, para evitar riscos significativos.

- **Aplicação do chatGPT para obtenção de parâmetros mais confiáveis duma estratégia de negociação**

Outro cenário testado, foi a aplicação da tecnologia da OpenAI, ou seja, chatGPT para imprimir os melhores parâmetros de uma estratégia e garantir indicação dos melhores momentos para entrar em ação. A seguir apresentamos os procedimentos aplicados:

- pedido submetido no chatGPT: Print the best ADX parameters to get in on the

action by showing the best points for buys and sells.

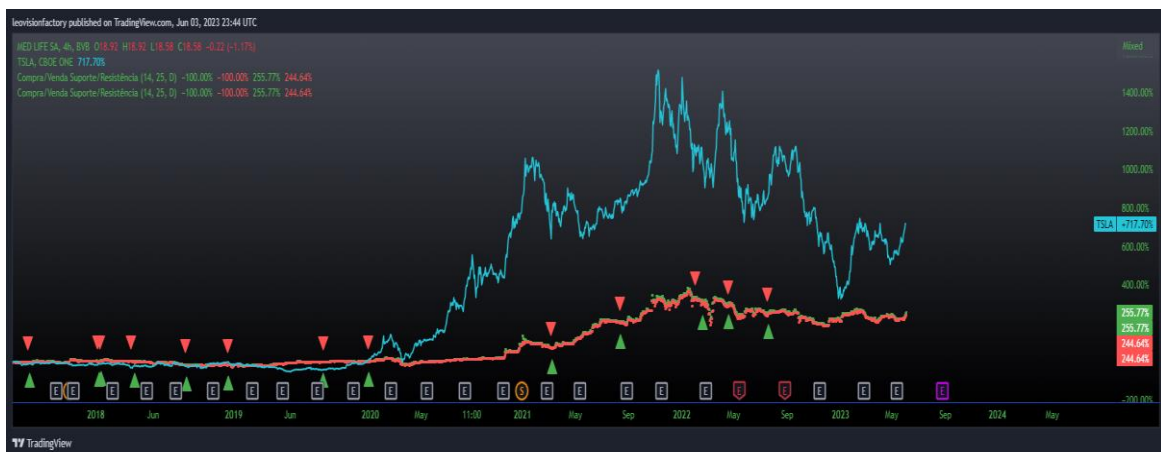
- Resposta do chatGPT:

Figure 41: Teste de parâmetros de ADX com chatGPT

```
Crypto Pairs Screener  Pine Editor  Strategy Tester  Trading Panel
Estratégia de Compra...  4.1
14 tr = max(max(high - low, abs(high - close[1])), abs(low - close[1]))
15 dmPlus = change(high) > change(low) ? max(high - high[1], 0) : 0
16 dmMinus = change(low) > change(high) ? max(low[1] - low, 0) : 0
17 atr = sma(tr, adxPeriod)
18 diPlus = 100 * sma(dmPlus, adxPeriod) / atr
19 diMinus = 100 * sma(dmMinus, adxPeriod) / atr
20 dx = 100 * abs(diPlus - diMinus) / (diPlus + diMinus)
21 adxValue = sma(dx, adxPeriod)
22
23 // Sinal de Compra
24 buySignal = crossover(adxValue, adxThreshold)
25
26 // Sinal de Venda
27 sellSignal = crossunder(adxValue, adxThreshold)
28
29 // Plotando os sinais de Compra e Venda
30 plotshape(buySignal, title="Compra", location=location.belowbar, color=■color.green, style=shape.triangleup, size=size.small)
31 plotshape(sellSignal, title="Venda", location=location.abovebar, color=■color.red, style=shape.triangledown, size=size.small)
32
33 // Cálculo dos Pontos de Pivô
34 pivotHigh = security(syminfo.tickerid, pivotPeriod, high)
35 pivotLow = security(syminfo.tickerid, pivotPeriod, low)
36 pivotClose = security(syminfo.tickerid, pivotPeriod, close)
37
38 // Desenhar Linhas de Suporte e Resistência
39 plot(pivotHigh, color=supportColor, title="Suporte", linewidth=1, style=plot.style_circles)
40 plot(pivotLow, color=resistanceColor, title="Resistência", linewidth=1, style=plot.style_circles)
41
```

Ao adicionar o código fornecido ao Pine Script da plataforma de trading, como o TradingView que foi utilizado para este caso, substituindo o código original da estratégia, obtivemos os seguintes resultados:

Figure 42: Resultados do ADX com chatGPT



Como podemos observar no gráfico, os pontos de entrada para compra e venda são indicados. No entanto, é importante ressaltar que é necessário possuir conhecimento na leitura dos gráficos de negociação para aplicar qualquer técnica de trading. O assistente imprime os melhores parâmetros da estratégia solicitada, mas é crucial avaliar o desempenho dessa estratégia em particular.

5. RESULTADOS E DISCUSSÃO

Neste capítulo, serão apresentados e discutidos os resultados das pesquisas efetuadas durante o estágio sob auxílio do trabalho inicial. É importante ressaltar que esses resultados estarão alinhados com os objetivos do estágio, uma vez que os insights e aprendizados adquiridos no trabalho inicial serão utilizados para fortalecer a discussão dos resultados. Além disso, interpretaremos esses resultados à luz dos objetivos do estágio e do contexto financeiro em que estão inseridos.

Com base nas pesquisas realizadas, destacaremos as descobertas mais relevantes e suas implicações para a análise de dados financeiros. Também faremos conexões com estudos anteriores e teorias científicas relevantes, a fim de posicionar nossa pesquisa dentro do campo acadêmico e contribuir para o avanço do conhecimento.

Por fim, discutiremos as limitações encontradas durante a pesquisa e suas possíveis influências nos resultados, demonstrando uma abordagem crítica e reflexiva. Além disso, exploraremos as implicações práticas dos resultados obtidos e delinearemos possíveis direções futuras de pesquisa com base nos insights adquiridos.

5.1. Critérios para investimentos com alto potencial de crescimento

De acordo (health-inequalities, 2023), os critérios de investimento são os elementos estabelecidos que compõem um conjunto de parâmetros utilizados por investidores financeiros e estratégicos para a análise e avaliação de oportunidades de investimento. Esses critérios têm a finalidade de otimizar o processo de busca e qualificação de novas oportunidades, conferindo maior eficiência ao mesmo.

Tal eficiência é alcançada por meio da definição clara dos critérios de investimento, os quais proporcionam um direcionamento específico e permitem a rápida exclusão de negócios que não atendem às exigências do investidor.

Estudos indicam a segurança, risco, retorno sobre o investimento, sustentabilidade e transferibilidade como critérios de qualidade para investimentos.

A seguir apresentamos resultados obtidos sobre os critérios de investimento de alto potencial com base nos estudos da (CMVM, 2023) & (Poitan, 2014).

Tabela 1: Critérios para investimentos de alto potencial de crescimento

Nº	Critério	Descrição
1	Mercado em crescimento	Pesquisas de análise tendência e projeções do setor para compreender se o mercado para o investimento tem potencial para o crescimento.
2	Produto ou serviço inovador	Avaliação do produto oferecido pela empresa para aferir se é inovador e possui uma proposta única de valor. Pois a diferenciação competitiva constitui um fator importante para o crescimento sustentável a longo prazo.
3	Equipe de gestão qualificada	Uma equipe experiente e com histórico comprovado pode ser um indicador positivo de sucesso futuro.
4	Modelo de negócio escalável	Uma avaliação do modelo de negócio da empresa é necessária para aferir se permite a escalabilidade, ou seja, se é capaz de expandir suas operações e aumentar sua receita de forma significativa sem aumentar proporcionalmente os custos.
5	Finanças sólidas	Uma empresa com uma base financeira sólida tem maior capacidade de investir em crescimento e resistir a períodos de volatilidade econômica.
6	Parcerias estratégicas	Parcerias com empresas estabelecidas ou líderes de mercado podem abrir portas para novas oportunidades.
7	Potencial de expansão internacional	A entrada em novos mercados internacionais pode oferecer oportunidades significativas de crescimento.
8	Tecnologia e tendências de mercado	Empresas que estão na vanguarda de avanços tecnológicos ou que aproveitam tendências emergentes têm maior probabilidade de crescimento acelerado.
9	Barreiras à entrada	Barreiras como patentes, direitos de propriedade

		intelectual, regulamentações governamentais ou vantagens competitivas podem proteger a empresa de uma competição intensa.
10	Análise de risco	É preciso uma avaliação cuidadosa dos riscos associados ao investimento, como volatilidade do mercado, riscos regulatórios, concorrência, mudanças tecnológicas e outros fatores que possam impactar o crescimento da empresa.

Fonte: Adaptado por autor

Com base nos resultados e critérios discutidos, podemos concluir que a avaliação criteriosa de oportunidades de investimento é fundamental para identificar negócios com alto potencial de crescimento. Os investidores devem considerar aspectos como o mercado em crescimento, a inovação dos produtos ou serviços, a qualidade da equipe de gestão, a escalabilidade do modelo de negócio, a solidez financeira, as parcerias estratégicas, o potencial de expansão internacional, a adoção de tecnologia e tendências de mercado, a existência de barreiras à entrada e a análise de riscos.

Ao levar em conta esses critérios, os investidores têm mais chances de tomar decisões assertivas e maximizar suas oportunidades de sucesso nos investimentos.

5.2. Análise técnica do mercado financeiro para adoção das estratégias de negociação

A análise técnica pode ser definida como o estudo da ação do mercado, primariamente por meio de uso de gráficos, com o objetivo de prever comportamento ou tendências futuras. (Machado & Pereira, 2022)

Embora existam evidências a favor e contra sua eficácia, ela se baseia na ideia de que os preços dos ativos refletem todas as informações disponíveis. No entanto, sua validade pode variar dependendo do mercado e das circunstâncias.

O mercado financeiro é constituído de agentes investidores de compras e vendas de ativos, com o objetivo de obter lucros. Estes agentes investidores são compostos de seres humanos, sujeitos às emoções, razões, medos e vontade de ganhar. É comum o desejo dos investidores e operadores do mercado financeiro adotarem critérios objetivos para embasar suas decisões em operações financeiras. (Machado & Pereira, 2022)

A adoção de critérios objetivos no mercado financeiro é respaldada cientificamente, visando reduzir vieses emocionais e tomar decisões mais fundamentadas. Estudos mostram que investidores podem ser influenciados por heurísticas e vieses cognitivos.

Ao utilizar critérios objetivos, como análises fundamentais e indicadores financeiros, busca-se embasar as decisões em análises racionais e maximizar o potencial de lucro, alinhando-

se à eficiência de mercado. Isso contribui para uma tomada de decisão mais eficiente e melhores resultados financeiros.

O estudo de Turcas et al (2020) citado por (Machado & Pereira, 2022), concluiu que uma análise crítica quantitativa e qualitativa dos três principais aspectos da evolução do mercado de ações (previsão, avaliação e retorno da carteira) ilustrou a importância de utilizar ferramentas econométricas para testar a solidez de diferentes teorias do mercado de ações. Além disso, foi identificada a existência de lacunas entre as teorias atuais e sua aplicação prática nos mercados de ações, enfatizando a necessidade de avanços e aprimoramentos nas práticas adotadas pelos investidores.

O estudo mencionado, que envolveu uma análise crítica quantitativa e qualitativa dos aspectos importantes da evolução do mercado de ações, oferece insights valiosos sobre a necessidade de utilizar ferramentas econométricas para testar a robustez das teorias existentes.

Ao identificar lacunas entre as teorias atuais e sua aplicação prática nos mercados de ações, o estudo destaca a importância de avançar e aprimorar as práticas adotadas pelos investidores. Essa abordagem crítica contribui para uma compreensão mais aprofundada do mercado de ações, enfatizando a importância de uma análise rigorosa e fundamentada para tomar melhores decisões de investimento.

Ao reconhecer as limitações das teorias existentes, os investidores podem buscar estratégias mais eficazes e adaptáveis às realidades do mercado, buscando maximizar o retorno e gerir os riscos de forma mais eficiente. Portanto, verifica-se a importância contínua da pesquisa e do desenvolvimento de abordagens mais avançadas e alinhadas com a dinâmica dos mercados de ações, visando melhorar a compreensão e as práticas de investimento.

5.3. Conhecimentos adquiridos durante o estágio

Durante o estágio em análise de dados financeiros, obtive habilidades e conhecimentos fundamentais em diversas áreas como: estratégias de negociação, automação de tarefas e análise de dados financeiro que constituiu o foco principal do estágio. A seguir estão descritas detalhadamente os principais tópicos aprendidos:

- **Exploração de Fontes de Dados**

Adquiri habilidades para extrair e analisar dados financeiros de várias fontes, incluindo Yahoo Finance, eToro, Kaggle e outras plataformas. Essa habilidade permitiu-me acessar informações cruciais para a tomada de decisões de investimento.

- **Backtesting de Estratégias**

Teste de estratégias com dados históricos – Utilizei a técnica de backtesting para avaliar o desempenho de diferentes estratégias de investimento com base em dados históricos. Isso

me permitiu entender como as estratégias teriam funcionado no passado e otimizá-las para o futuro.

- **Plataformas de Negociação**

Domínio de plataformas – Adquiri conhecimentos práticos em várias plataformas de negociação, como eToro e TradingView. Essas plataformas são essenciais para a execução de ordens de compra e venda, bem como para a análise de gráficos de preços e indicadores técnicos. Além de aplicação da funcionalidade do chatGPT para identificação dos melhores parâmetros de uma estratégia.

- **Análise técnica do mercado por meio de gráficos**

Leitura de velas – Aprendi a interpretar padrões de velas, que são cruciais para identificar momentos oportunos para ação nos mercados financeiros. Essa habilidade é vital para a análise técnica e a tomada de decisões acertadas.

- **Tratamento de dados desbalanceados**

Manipulação de dados – Conhecimentos adquiridos durante o período preparatório para o estágio foram fortificados pelas práticas, entretanto, utilizei diversas bibliotecas em Python, como scikit-learn e imbalanced-learn, para tratar dados financeiros desbalanceados. Essas bibliotecas permitem a correção de desequilíbrios nos conjuntos de dados, garantindo análises mais precisas e modelos mais robustos.

- **Automação com API da OpenAI**

Criação de chatbot – Desenvolvi habilidades na criação de chatbot utilizando a API da OpenAI. Este conhecimento reduziu a atividade dos técnicos em interagir com os clientes para questões básicas permitindo maior produtividade com menor esforço possível.

Esses conhecimentos adquiridos durante o estágio em análise de dados financeiros me proporcionaram uma base sólida para análise de investimentos, execução de estratégias de negociação, interpretação de gráficos e automação de processos, preparando-me para enfrentar os desafios do mundo financeiro com confiança e competência.

6. CONCLUSÕES E SUGESTÕES

Após a conclusão do estágio, com base nas atividades desenvolvidas, pode-se concluir que este representou uma experiência enriquecedora centrada principalmente na exploração das técnicas de machine learning na análise de dados financeiros, desempenhando uma função auxiliar na interligação entre a teoria e a prática.

Durante este período de imersão, um repertório abrangente de competências e conhecimentos fundamentais para o campo financeiro foi adquirido, com destaque para a análise de dados e suas implicações. A análise de dados financeiros, em sua essência, constituiu o epicentro desta experiência de estágio, facultando a exploração aprofundada dos intrincados meandros das finanças.

A exploração sistemática desses dados proporcionou uma compreensão minuciosa dos padrões, tendências e determinantes que subjazem às tomadas de decisão financeira, erigindo-se como o alicerce primordial para uma atuação fundamentada e perspicaz.

O exercício concernente ao tratamento de dados desbalanceados desempenhou um papel de apoio crucial ao enriquecimento das aptidões relacionadas à manipulação de dados. Especificamente, estas atividades se prestaram a assegurar que as análises e modelos resultantes mantivessem um grau de robustez e confiabilidade condizente com as expectativas.

Adicionalmente, o estágio compreendeu incursões na negociação de ações e na análise de mercados, viabilizando a aplicação prática das competências adquiridas em um contexto real.

O desenvolvimento de técnicas voltadas à análise das estratégias de negociação, com o emprego de Python como ferramenta preponderante, bem como a execução de testes retroativos (backtesting), conferiram à empreitada uma dimensão avaliativa, permitindo uma análise cuidadosa da efetividade das estratégias, consequentemente contribuindo para o aprimoramento de estratégias de investimento mais sólidas.

A materialização de um chatbot mediante a utilização da API da OpenAI representou uma incursão no domínio da automação de tarefas, melhorando a eficácia e qualidade dos serviços prestados, sobretudo no contexto de atendimento ao cliente.

Por fim, a publicação de artigos científicos no âmbito da análise de dados e aplicação de inteligência artificial no contexto financeiro demonstrou o comprometimento com a pesquisa e a disseminação do conhecimento.

Uma das principais lacunas identificadas nas pesquisas durante o estágio foi a necessidade premente de implementar estratégias algorítmicas personalizadas que sejam minuciosamente testadas por meio de simulações antes de sua aplicação em mercados reais. Tal abordagem destaca-se como um imperativo, pois tem o potencial de mitigar de

forma significativa um dos maiores desafios enfrentados: o viés emocional na tomada de decisões financeiras.

Em síntese, o estágio propiciou uma experiência vasta e abrangente, tendo a análise de dados financeiros como seu foco principal, enquanto o tratamento de dados desbalanceados representou uma faceta secundária complementar. Os conhecimentos e habilidades acumulados auferiram ao estagiário a preparação adequada para enfrentar com confiança e competência os desafios complexos e dinâmicas inerentes ao campo financeiro.

Futuramente, sugere-se a implementação de estratégias algorítmicas personalizadas devidamente testadas para simulação e sua aplicação em mercados reais. Pois, esta iniciativa poderá reduzir viés emocionais, manter a consistência e adaptar-se as mudanças do mercado.

Bibliografia

- AMAZON. (30 de 12 de 2022). *aws*. Obtido de [aws.amazon.com: https://aws.amazon.com/pt/what-is/machine-learning/](https://aws.amazon.com/pt/what-is/machine-learning/)
- Analytics Vidhya. (27 de 12 de 2022). *Analytics vidhya*. Obtido de [analyticsvidhya.com: https://www.analyticsvidhya.com/blog/2020/10/overcoming-class-imbalance-using-smote-techniques/](https://www.analyticsvidhya.com/blog/2020/10/overcoming-class-imbalance-using-smote-techniques/)
- Apolónia, J. D. (2018). *seleção de atributos de dados inconsistentes*. Lisboa. ClearPointStrategies. (19 de 4 de 2023). Obtido de <https://www.clearpointstrategy.com/blog/strategic-analysis>
- Dye, S. (02 de May de 2020). *Towards Data Science. How to Handle SMOTE Data in Imbalanced Classification Problems*.
- Escovedo, T., & Koshiyama, A. (2020). *Introdução a Data Science - Algoritmos de Aprendizagem de Máquina e métodos de análise*. São Paulo: ISBN.
- etoro. (25 de junho de 2023). *etoro.com*. Obtido de <https://www.etoro.com/es/news-and-analysis/copytrader/6-investment-strategies-gaspersopi/>
- Falardo, D. J. (2022). *A utilização de ferramentas de inteligência artificial na gestão comercial das empresas e o seu impacto no processo de vendas*. Lisboa.
- Fink, A. (2020). Conducting reseach literature reviews. *From the internet to paper*, p. 261.
- FOREX. (29 de 06 de 2023). *forex.com*. Obtido de <https://www.forex.com/ie/trading-academy/platform-tutorials/web-trader-introducing-technical-analysis/>
- Google. (10 de Janeiro de 2023). *Google Analytics*. Obtido de [support.google.com: https://support.google.com/analytics/answer/10089681?hl=en](https://support.google.com/analytics/answer/10089681?hl=en)
- Graham, M. (2023). *Day Trading: 10 Best Beginners Strategies to Start Trading Like A Pro and Control Your Emotions in Stock, Penny Stock, Real Estate, Options Trading, Forex, Cryptocurrencies, Futures, Swing Trading*. New York: Tor books.
- Hagstrom, R. G. (2014). *The Warren Buffett Way: +Website, 3rd ed Paperback*. New York: Wiley.
- IBM. (25 de 6 de 2023). *ibm.com*. Obtido de <https://www.ibm.com/docs/pt-br/db2/10.5?topic=visualizer-confusion-matrix-view>
- imbalanced-learn. (27 de 12 de 2022). *imbalanced-learn.org*. Obtido de [imbalanced-learn.org: https://imbalancedlearn.org/stable/references/generated/imblearn.under_sampling.CondensedNearestNeighbour.html#imblearn.under_sampling.CondensedNearestNeighbour](https://imbalancedlearn.org/stable/references/generated/imblearn.under_sampling.CondensedNearestNeighbour.html#imblearn.under_sampling.CondensedNearestNeighbour)
- INVESTOPEDIA. (25 de junho de 2023). *Investopedia.com*. Obtido de <https://academy.investopedia.com/products/advanced-technical-analysis>
- Jiwani, D. (2021). *Buffett's 2-Step Stock Market Strategy: Know When to Buy A Stock, Become a Millionaire, Get The Highest Returns Paperback*. New York.
- Jr, A. d., Amorim, B. P., Tabata, P. I., & Felicissimo, V. R. (10 de Outubro de 2022). the impacts of artificial intelligence on marketing strategies. *Analise e desenvolvimento de sistemas*, p. 24.
- KERN. (26 de 06 de 2023). *kernc.github.io*. Obtido de [Backtesting: https://kernc.github.io/backtesting.py/](https://kernc.github.io/backtesting.py/)
- Kotler, P., Kartajaya, H., & Setiawan, I. (2017). *Marketing 4.0: Do tradicional ao digital. Sextante*, p. 22.
- Luchese, R. L. (2021). *Aplicação de Técnicas de Aprendizagem de Máquina para Classificação ABC Multicritério*. Porto - Alegre.
- Machado, F. d., Souza, P. V., & Santo, P. T. (8 de 12 de 2017). a utilização dos métodos de data mining e aprendizagem de máquina no processo de prevenção à fraudes no mercado segurador. *vii congresso brasileiro de engenharia de produção*, p. 12.
- Management Solutions. (2018). *Aprendizagem de Máquina uma peça-chave na*

- transformação dos modelos de negócios. Espanha. Obtido de <http://www.managementsolutions.com/>
- Mashaushi, K. R. (September de 2017). An Analysis of Technical Trading Strategies. Microsoft. (21 de 12 de 2022). *Preparar dados para aprendizagem de máquinas melhoradas*. Obtido de <https://learn.microsoft.com/pt-pt/azure/architecture/data-science-process/prepare-data>
- MICROSOFT AZURE. (25 de 06 de 2023). *learn.microsoft.com*. Obtido de <https://learn.microsoft.com/es-es/azure/machine-learning/how-to-tunehyperparameters?view=azureml-api-2>
- Mónico, L., Alferes, V., & Castro, P. P. (04 de 07 de 2017). A Observação Participante enquanto metodologia de investigação qualitativa. *CIAIQ 2017*, p. 12.
- Moreira, R. P. (2021). *Inteligência Artificial no Marketing Digital*. Portugal - Lisboa.
- Nison, S. (2001). *Japanese candlestick charting techniques*. New York: NYIF.
- Oliveira, J. C. (2018). *Imputação em datasets médicos – uma comparação entre três métodos*. Aveiro.
- OPENAI. (28 de 06 de 2023). *openai.com*. Obtido de <https://platform.openai.com/docs/api-reference>
- ORACLE. (7 de 3 de 2023). *oracle.com*. Obtido de Oracle Cloud: <https://www.oracle.com/br/artificial-intelligence/what-is-ai/>
- Pereira, V., Vasconcelos, P., Rodrigues, M. E., & Marques, A. B. (1 de 4 de 2023). Utilizando Design Thinking no design de aplicativos educacionais para crianças autistas. *XIV Computer on the Beach*, p. 8.
- Peternelli, L. A. (23 de Dezembro de 2023). Testes de Significância. São Paulo.
- Pinto, E. F. (2023). *Gestão de créditos e cobranças*. Porto: Porto editora.
- Quantified. (16 de 5 de 2023). *Quantified Strategies*. Obtido de <https://www.quantifiedstrategies.com/technical-analysis-strategy/>
- quantifiedstrategies. (25 de junho de 2023). *quantifiedstrategies.com*. Obtido de <https://www.quantifiedstrategies.com/technical-analysis-strategy/>
- Saias, J., Maia, M., Rato, L., & Gonçalves, T. (23 de 12 de 2018). Machine Learning: um estudo sobre conceitos, tarefas e algoritmos relacionados com predição e recomendação. p. 14.
- scikit-learn. (30 de 12 de 2022). *scikit-learn*. Obtido de [scikit-learn.org: https://scikit-learn.org/stable/modules/tree.html](https://scikit-learn.org/stable/modules/tree.html)
- Silva, C. E., & Borges, L. G. (Dezembro de 2019). Uma Abordagem Inteligente para Classificação de Perfis de Aprendizagem de Discentes com Aplicação de Aprendizado de Máquina. p. 280-285.
- Silva, D. F. (2021). *Pré-processamento de Dados e Comparação entre Algoritmos de Aprendizagem de Máquina para a Análise Preditiva de Falhas em Linhas de Produção para o Controlo de Qualidade*. Porto - Portugal.
- Soares, J. P. (2017). *Explorar diferentes estratégias de data mining aplicadas a dois problemas no pré-processamento de dados*. Coimbra.
- The balance. (25 de junho de 2023). *The balance*. Obtido de <https://www.thebalancemoney.com/stocks-4073971>
- VISION FACTORY. (26 de 06 de 2023). *visionfactory.com*. Obtido de <https://www.visionfactory.org/>
- Wang, J. K., Zheng, C. Y., Ren, J., Li, C., & Han, Q. &. (24 de 05 de 2021). Improving Risk Identification of Adverse Outcomes in Chronic Heart Failure Using SMOTE+ENN and Machine Learning. p. 14 Pages 2453—2463.
- Winder, P. (2021). *Reinforcement Learning*. Gravenstein Highway North, Sebastopol - EUA: Beth Kelly.
- Zúñiga, J. J. (2 de 12 de 2020). Aplicação dos algoritmos Random Forest e XGBoost a um banco de dados de aplicativos de cartão de crédito. *Engenharia, pesquisa e tecnologia*, p. 16.

About me

Leonel Silima

Master's degree in applied computing

[Linkedin](#), [Github](#) & [Kaggle](#)

Keywords: *handling unbalanced data with machine learning algorithms*

TECHNIQUES AND METHODS OF HANDLING UNBALANCED DATA WITH MACHINE LEARNING ALGORITHMS

TECHNIQUES AND METHODS OF HANDLING UNBALANCED DATA BEFORE TRAINING WITH MACHINE LEARNING ALGORITHMS (**This step is done before machine learning algorithms**)

Currently financial management software and platforms are built under artificial intelligence algorithms. In fact, these solutions come from raw data that is pre-processed and then subjected to Machine Learning algorithms. However, the question is: What is the problem arising from these applications and what are their motivations? Answering these questions is the main purpose of this article.

What is the need to handle unbalanced data?

One of the biggest problems with machine learning platforms or applications has been the inefficiency of their performance, often the result of not applying the best techniques on imbalanced data. However, the applications end up showing a greater probability of assertiveness in the majority instances and fail in the minority instances. That is, unbalanced data applications present a greater risk of assertiveness only in one direction and not when applied in the opposite direction.

The quality of a machine learning model starts with the quality of the data in the pre-processing stage. However, when this step is not observed, there is a risk of generating misleading models, that is, with difficulties in forecasting. Next, we will talk about the main techniques and methods to be used in the process of handling unbalanced data before the training stage with machine learning algorithms.

Unbalanced Data

Unbalanced data can be defined by the small incidence of a category within a dataset (minority class) compared to the other categories (majority classes). The following example illustrates the characteristics of unbalanced data.

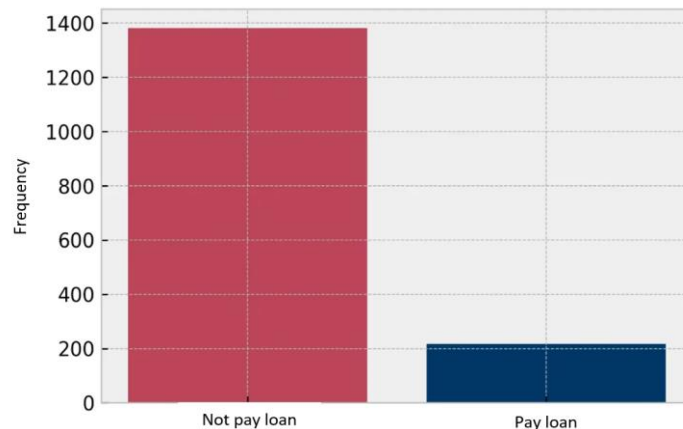
balanceados.Figura 1: unbalanced data characteristics



(Dye, 2020)

However, if we develop a machine learning model without considering the disproportionality between the majority and minority instances, the model will fall victim to the [precision paradoxo](#), in which the algorithm parameters will not differentiate the minority class from the other categories, resulting in better results in the tests while worse when applied in real situations. Below we present a predictive model of bank loan payments based on unbalanced data.

Figure 2: Machine learning model results with unbalanced data



As we see in this example, the incidence of the *not pay loan* category is dominant, being found in 1,400 of the cases, so predicting that each case is in the *not pay loan* category will be more accurate for 1,400 customers than for 200 customers in the *pay loan* category which means that of the 1,400 customers, the model understands that only 14.2% are those who repay loans.

Precision and recall are better measures in such cases. The underlying issue is that there is a class imbalance between the positive class and the negative class. Prior probabilities for these

classes need to be accounted for in error analysis. Precision and recall help, but precision too can be biased by very unbalanced class priors in the test sets.

Techniques and Methods of handling Unbalanced Data

Unbalanced data treatment techniques are subdivided into 2 large groups: OverSampling (increasing the minority instance) and UnderSampling (reducing the majority instance).

1. UnderSampling techniques

UnderSampling is a method that consists of reducing the number of observations of the class with greater frequency to reduce the difference between categories. This method has the following cases:

- **Random UnderSampling**

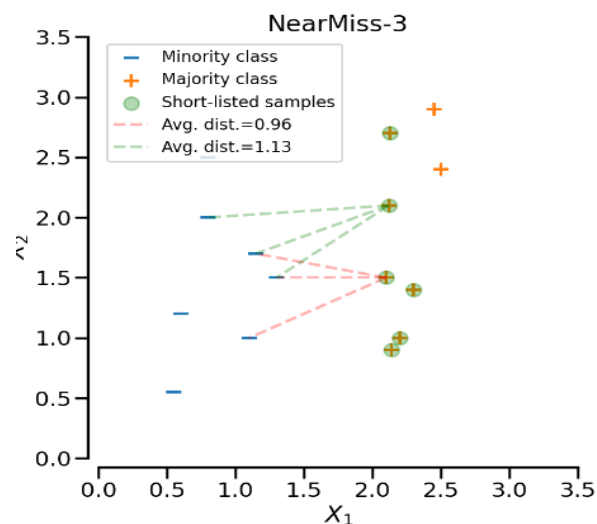
Technique that consists of randomly removing data from the class more frequently which leads to a serious loss of information.

- **NearMiss – 3**

This technique adds some heuristic rules based on the nearest neighbor's algorithm (knn) and the acceptance classifier derived from scikit-learn.

NearMiss-3 is a 2-steps algorithm. First, for each negative sample, their M nearest neighbors will be kept. Then, the positive samples selected are the one for which the average distance to the N nearest-neighbors is the largest, [see more](#). As we can see in the figure below.

Figure 3: Mathematical formulation of NearMiss 3



It should be noted that this technique reduces data loss contrary to previous practice, however this is the most recommended for UnderSampling.

2. OverSampling techniques

Contrary to UnderSampling, OverSampling consists of synthetically creating observations of the minority class, aiming to match the proportion of categories.

However, this technique offers good results when the minority class doesn't have a very large quantitative variation in its parameters. Otherwise, the model may be very good at identifying specific cases of the minority class and not the category as a whole. Among the methods, the following apply:

a) SMOTE (Synthetic Minority Oversampling Technique) - Oversampling

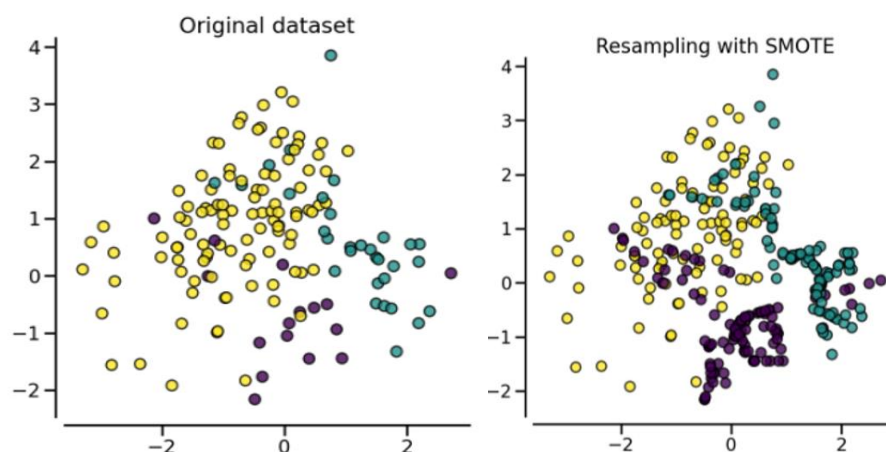
Smote is one of the most used oversampling methods to solve the imbalance problem. It aims to balance the distribution of classes with random increase of examples of minority classes, replicating them. However, inconsistencies were identified in this technique and improvements were applied having created its following subset.

b) SMOTE + ENN

SMOTE + ENN consists of a combination of a synthetic data generation technique and a noisy data cleaning technique to avoid class overlap.

It's important to note that this technique first performs the generation of synthetic data in the minority class followed by the cleaning of misclassified instances in both classes, which makes the data more standardized for machine learning algorithms.

Next, we observe the characteristics of the data before and after applying the Smote technique.



Final remarks

Finally, there are many other techniques for treating unbalanced data, both for UnderSampling and OverSampling, as well as other subsets of the Smote method. However, the practices mentioned in this article are the most applied and have shown better results.

It is also worth emphasizing that during the process of handling unbalanced data, all the above alternatives must be considered and the method that presents the best results in its context must be chosen. In addition, it's interesting to note that more than one type of transformation should not be applied at the same time, under the risk of losing the original information!

In the next article we will talk about processing financial data with machine learning algorithms

Reference List

Dye, S. (02 de May de 2020). Towards Data Science. *How to Handle SMOTE Data in Imbalanced Classification Problems*.

imbalanced-learn. (27 de 12 de 2022). *imbalanced-learn.org*. Obtido de imbalanced-learn.org: https://imbalanced-learn.org/stable/references/generated/imblearn.under_sampling.CondensedNearestNeighbour.html#imblearn.under_sampling.CondensedNearestNeighbour

scikit-learn. (30 de 12 de 2022). *scikit-learn*. Obtido de scikit-learn.org: <https://scikit-learn.org/stable/modules/tree.html>

<https://towardsdatascience.com/how-to-handle-smote-data-in-imbalanced-classification-problems-cf4b86e8c6a1>

https://en.wikipedia.org/wiki/Accuracy_paradox

<https://learn.microsoft.com/pt-br/azure/machine-learning/component-reference/smote>

<https://medium.com/turing-talks/dados-desbalanceados-o-que-s%C3%A3o-e-como-evit%C3%A1-los-43df4f49732b>

AI TRENDS IN FINANCIAL ANALYSIS

The entire world has been experiencing a consistent digital transformation, increasingly powerful and with infinite growth. When it comes to technology, there is no end point. In fact, it is always possible to go one step further, make new discoveries and adopt intelligent digital practices. As a result, it facilitates organizational day-to-day activities and generates greater profitability. However, it would be impossible to talk about **financial analysis**, ignoring **artificial intelligence**. Therefore, we are going to talk about **artificial intelligence (AI)** trends in the financial sector, present the role of AI in the financial market and provide a list of AI tools aimed at financial services.

AI: Current position and future potential

Delays in decision making are costly in electronic trading – whether you're dealing with equities, fixed income, futures, or commodities. So, waiting for end-of-day reports means you're likely to miss profitable opportunities or fail to respond to threats to compliance until it's much too late. Thus, AI gives tools that need to build and deploy their own real-time monitoring and analysis systems without writing any code. For example, such tools are MQL5, etoro, analytics 2go, TWAP and more.

So what is artificial intelligence and its trends?

Simply put, **artificial intelligence** refers to systems or machines that mimic human intelligence to perform tasks based on the information they collect. However, AI is a technology in constant development that allows the advancement of several areas. In particular, in recent years, this area tends to drive algorithms that are smarter than human cognitive capacity.

Brief framework of artificial intelligence

Artificial intelligence is a vast field in which we can find subsets such as: **Machine Learning** and **Deep Learning** as a subset of **Machine Learning**. Below we present the figure with more details.

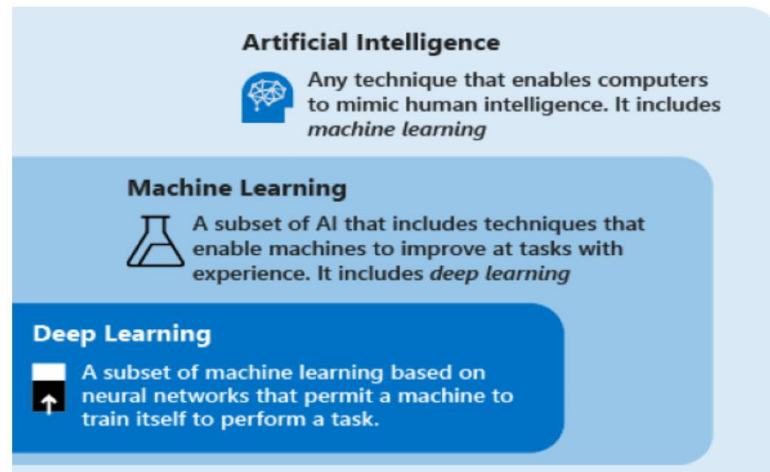


Figura 1: Artificial Intelligence structure

AI effects on our lives and on the financial market

Previously, there was more talk about the dependence on the internet, but now we cannot talk about the internet without mentioning AI. This is because they go hand in hand. In fact, the applications that circulate on the internet are currently developed by **artificial intelligence** tools, either by **machine learning** or **deep learning**.

Moreover, the core principle of AI is to replicate, and then exceed, the way humans perceive and react to the world. And it's quickly becoming a mainstay of innovation fueled by various **machine learning** and **deep learning** techniques. These recognize patterns in data to enable predictions. The following are the main AI actions:

- Provide a more comprehensive understanding of the abundance of data available.
- Rely on predictions to automate tasks that are too complex and difficult for humans and
- Learn from human data and patterns to replicate more intelligently.

Now that you understand the main concepts of **artificial intelligence** and its impact on our lives, let's talk about its trends in **financial analysis**.

Artificial intelligence and financial analysis

Before linking these two major areas, we will present concepts and some **financial analysis** techniques.

Financial analysis can be defined as a study of the company's ability to identify its weaknesses and trends to improve and generate profits. The figure below illustrates the main **financial analysis** techniques.



Figura 2: Financial analysis techniques

What is the role and trend of AI in financial analysis?



Figura 3: The role of AI in financial analysis

Was it clarified? You might think so, but the answer is still no, because the figure above just shows a passage from archaic to smart and the role of AI in this area.

What exactly are the trends of AI in the financial analysis process?

Understanding how **artificial intelligence** works in the [financial market](#) is an important role for finance professionals. Especially this happens for those who seek to enhance their operations or who still don't understand much about investments. Therefore, this technology can be a hand in the wheel for its developing tools that help investors in their choices based on market analysis.

Algorithmic trading

Algorithmic trading consists of executing orders through automated pre-programmed trading instructions, identifying variables such as price, volume and time.

This type of trading attempts to leverage the speed and computational resources of computers relative to human traders. As you can see, AI marked a big step in the financial market. This is what used to be a series of tasks and a trader at the helm is now an AI algorithm capable of running the entire trading process automatically.

Among the algorithms used for automatic trading we can mention: MQL5 & TWA. Below we present an illustrative figure on the trend of AI in market trading.



Figure 4: AI trends in the financial market

Among several AI applications and trends in the finance area, we will present some of the main ones that stand out:

1. Waiting time reduction

The application of AI technologies in financial services offers a more robust and personalized customer experience. For example the predictive analytics help customers manage their finances and get personalized advice. Another example is chatbots that are widely adopted to interpret customer queries and provide relevant answers in real time, thus reducing waiting time.

2. Risk assessment and prevention

Using AI to manage risks is particularly helpful when handling massive amounts of both structured and unstructured data. AI algorithms help in fast data processing, implement cognitive analytics to anticipate and manage risk cases to ensure better planning.

3. Ability to prevent financial crimes.

AI in your subset **Machine learning** models can perform analytics to identify fraud patterns and payment anomalies. In addition, it delivers risk scores, and helps with developing anti-money laundering solutions for real-time payment systems.

4. Support in trading

[AI](#) is used to analyze competitors' prices, market trends and even historical product prices. Apart from that, it monitors the market in real time and readjust product prices as market conditions change. In this way, it is possible to guarantee that the products are always being sold for the best possible price.

Final considerations

Finally, I would like to remind you that those who want to win, need to say goodbye to traditionalism and invest in new technologies to keep up with the desires and needs of the market. Therefore, it's impossible to support the demands and challenges of the current financial market without the support of **artificial intelligence** tools. Below are some of the main **artificial intelligence** tools aimed at **financial analysis** and management:

- [Analytics2go](#)
- [Etoro](#)
- [Excelfomulabot](#)
- [Findly](#)
- [Looka](#)
- [Quickchat](#)
- [Revive](#)
- [Vee](#)
- [Wand](#)

Reference list

- <https://www.btgpactualdigital.com/blog/bolsa-de-valores/algorithmic-trading/amp>
- <https://learn.microsoft.com/en-us/azure/architecture/data-guide/big-data/ai-overview>
- <https://people.com.ai/blog/tipos-de-inteligencia-artificial/>
- <https://smartclick.ai/personal-stories/artificial-intelligence-in-finance/>
- <https://www.mindtek.com.br/2023/01/08-ferramentas-de-inteligencia-artificial/>
- <https://www.oracle.com/br/artificial-intelligence/what-is-ai/>
- <https://www.wallstreetmojo.com/financial-analysis/>

Tags

[Artificial Intelligence](#), [Machine learning](#), [Deep Learning](#), [Algorithmic trading](#), [Financial analysis](#)

How to use backtesting to create a trading test strategy?

You might have read a lot about some data processing techniques in other articles of ours. However, in this article we are getting to work to create our own tool for testing trading strategies.

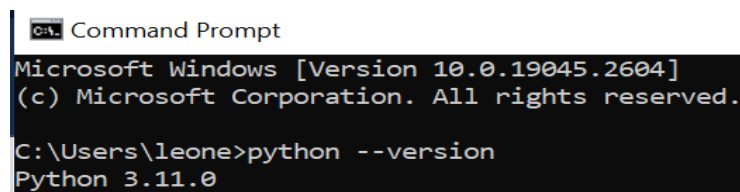
What are the tools needed for this project?

Firstly, you need basic notions of python and programming logic. Next I will present step by step how to create the project. You just need to follow all the points mentioned.

1. Python Installation

You can install python easily and free and to both Windows and Linux systems. Click [here](#) to access the installer according to your operating system and follow the instructions.

Don't worry about the pip package because version 3 of python already has it built in. In fact, you just need to go to its command line and view the version of python installed as shown in the following example.



```
Command Prompt
Microsoft Windows [Version 10.0.19045.2604]
(c) Microsoft Corporation. All rights reserved.

C:\Users\leone>python --version
Python 3.11.0
```

Ready to install python with its incorporated pip? Let's proceed with the installation and import of the necessary libraries for the project.

2. Installation of necessary tools for the project

The first step is to install the text editor (the place where you will write your code). My advice is to install anaconda because it brings all the necessary tools for python including editors like PyCharm, Jupyter and others. In this project we will work with Jupyter. Find [here](#) anaconda installer and enjoy its amazing tools.

3. Installation of libraries

As we mentioned above, python has a package called pip that is used to install, update, and uninstall libraries. So, we are going to use this tool in our text editor to install the libraries that we are going to use in this project. See the following example.

🔗 Pandas installation

Pandas is a library used for data analysis and manipulation, it is open source and easy to use, it was built on the Python language. For its installation you just need to type in your text editor the following command:

```
pip install pandas
```

🔗 TA-Lib installation

TA-Lib is a library used by trading software developers required to perform technical analysis of financial market data. Type following command to install:

```
pip install TA-Lib
```

🔗 Yfinance installation

Yfinance is a open source library developed by Ran Aroussi as a means to access the financial data available on Yahoo Finance. Therefore, install it with the following command:

```
pip install yfinance
```

🔗 Backtesting installation

It's a python framework that uses historical data to infer the viability of trading strategies. Install it with following command:

```
pip install Backtesting
```

4. Library import

With all the libraries installed, we proceed with their import for this purpose, we will apply the following procedures.

```
In [2]: ▶ 1 import talib
          2 import pandas as pd
          3 from backtesting import Backtest, Strategy
          4 from backtesting.lib import crossover
          5 #from backtesting.test import SMA, GOOG
          6 import yfinance as yf
```

Then we define two input variables where we will ask the end user to define his trading range with start and end date. As we can see the example below:

```
date = input(str("Start date: "))
End_date = input(str("End date: "))
```

Now let's instantiate an object with our yfinance library to call our input variables and download the data, based on our input stock also defined by the user.

```
10
11 data = yf.download((input("Stock: ")), start=(date), end=(End_date))
12
```

At this point we create a class that asks us to strategize to test.

```
13 class SMAcross (Strategy):
14     def init(self):
15         self.smaQuick = self.I(talib.SMA, self.data.Close, 50)
16     def next(self):
17         price = self.data.Close[-1]
18
19         if price > self.smaQuick[-1] and not self.position.is_long:
20             self.buy()
21         elif price < self.smaQuick[-1] and self.position.is_long:
22             self.position.close()
23
24 bt = Backtest(data, SMAcross)
```

Next we run backtesting to visualize the results of the strategy.


```
stats = bt.run()
```

Moreover, to improve the visualization, let's call the plot function. In particular, this is incorporated in the backtesting library, to print a graph based on the parameters we defined.

```
bt.plot(plot_equity = True, plot_volume = True, plot_drawdown = False,
        smooth_equity = True, open_browser = True, show_legend = True,
        superimpose = True, resample = True, plot_pl = True)
```

Finally, we will run stats to view the details of the results.

```
stats
```

FULL CODE

```
In [2]: 1 import talib
2 import pandas as pd
3 from backtesting import Backtest, Strategy
4 from backtesting.lib import crossover
5 #from backtesting.test import SMA, GOOG
6 import yfinance as yf
7
8 date = input(str("Start date: "))
9 End_date = input(str("End date: "))
10
11 data = yf.download((input("Stock: ")), start=(date), end=(End_date))
12
13 class SMACross (Strategy):
14     def init(self):
15         self.smaQuick = self.I(talib.SMA, self.data.Close, 50)
16     def next(self):
17         price = self.data.Close[-1]
18
19         if price > self.smaQuick[-1] and not self.position.is_long:
20             self.buy()
21         elif price < self.smaQuick[-1] and self.position.is_long:
22             self.position.close()
23
24 bt = Backtest(data, SMACross)
25
26 stats = bt.run()
27
28 bt.plot(plot_equity = True, plot_volume = True, plot_drawdown = False,
29         smooth_equity = True, open_browser = True, show_legend = True,
30         superimpose = True, resample = True, plot_pl = True)
31 stats
```

RESULTS



```

Out[13]: Start                2022-05-02 00:00:00
        End                  2023-02-24 00:00:00
        Duration              298 days 00:00:00
        Exposure Time [%]     34.951456
        Equity Final [$]      10544.920013
        Equity Peak [$]       12715.48056
        Return [%]            5.4492
        Buy & Hold Return [%] -24.899598
        Return (Ann.) [%]     6.706009
        Volatility (Ann.) [%] 29.381223
        Sharpe Ratio          0.228241
        Sortino Ratio         0.378078
        Calmar Ratio          0.392848
        Max. Drawdown [%]     -17.07022
        Avg. Drawdown [%]     -5.965938
        Max. Drawdown Duration 170 days 00:00:00
        Avg. Drawdown Duration 28 days 00:00:00
        # Trades              3
        Win Rate [%]          33.333333
        Best Trade [%]        9.669568
        Worst Trade [%]       -3.507688
        Avg. Trade [%]        1.808568
        Max. Trade Duration    51 days 00:00:00
        Avg. Trade Duration    34 days 00:00:00
        Profit Factor          2.551637
        Expectancy [%]        1.960004
        SQN                   0.453742
        _strategy              SMAcross
        _equity_curve          ...
        _trades                Size EntryBa...
dtype: object

```

As noted in the result above, we had input data from May 2022 to today's date and AMZN (amazon) is our stock. In your case, you can try any stock available on yfinance.

Last thoughts

This tutorial presented an overview of the capabilities of the backtesting library. Our next step in the future is to develop an installable tool for these purposes. Stay tuned for our updates!

Reference list

<https://www.python.org/downloads/>

https://pandas.pydata.org/getting_started.html

<https://www.python.org/doc/>

<https://www.anaconda.com/products/distribution>

<https://kernc.github.io/backtesting.py/>

Keywords

Smart trading, backtesting, finance data

How to create a PYTHON Library

Welcome to another practical article! Well, in this session we will develop a *python library* capable of the following:

- Download and view *financial data*.
- Perform volume calculations of *geometric solids*.
- And perform *mathematical calculations* such as: square root, logarithms, potency, and other basic operations such as addition, subtraction, divide and multiplication.

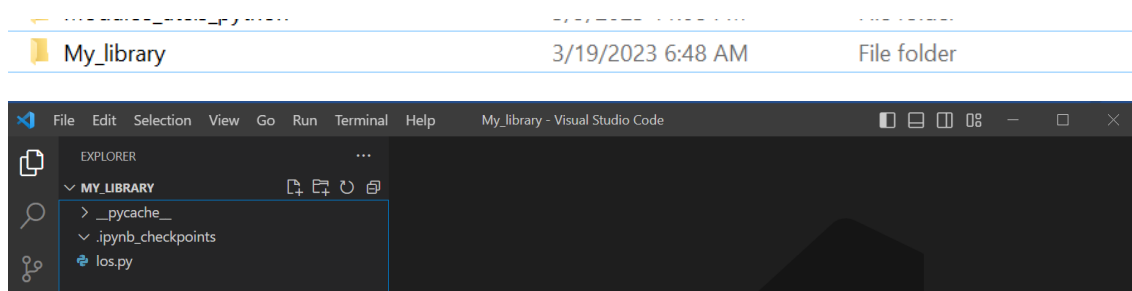
Without further ado let's look at the steps to follow

First step: Creating the library

First, we need to install the python programming language and a text editor. Find in this [article](#) the instructions for installing these tools.

Second step: Project creation

At this point we are going to create a folder where we will store our project and open it with a text editor. In this example I used Vscode but you can use any other editor of your choice. So, inside our text editor we create a file with a py extension where we are going to write our code. In my case the file name is los.py. Below we can see the figure illustrating this point.



Third step: coding

With the project created we can start with the hand in the work. Remember that we only use other libraries when necessary. So, in this project I used math to help me with some mathematical operations and yfinance to download financial data! Therefore, the following is the coding of this library.

```
File Edit Selection View Go Run Terminal Help
* lospy - My Library - Visual Studio Code

EXPLORER
MY_LIBRARY
  _pycache_
  .ipynb_checkpoints
  testing_my_library-checkpoint.ipynb
  lospy
  testing_my_library.ipynb

lospy
1 import math # For some calculation operations
2 import yfinance as yf # To download financial data
3
4 # TO DOWNLOAD FINANCIAL DATAS
5 def fdata(): # Function that will be executed in the front end
6     a = input(str("Enter start date: ")) # To input start date
7     b = input(str("Enter end date: ")) # To input end date
8     c = yf.download(input("Stock: "), start=a, end=b)
9     return c # This object will call the introduces stock to download the data
10
11 # To calculate volumes of geometric solids
12 def pvol(a, h): # prism volume
13     volp = a*b*h # base area multiplied by height
14     return volp
15
16 # pyramid area
17 def pyv(a, h):
18     pyvol = a*b*h/3
19     return pyvol
20
21 # cylinder volume
22 def cvol(h):
23     cvolum = 3.14**2*h
24     return cvolum
25
26 # cone volume
27 def conv(h):
28     cone = 3.14**2*h/3
29     return cone
30
31 # sphere volume
32 def svol(r):
33     svolum = 4*3.14*r**3/3
34     return svolum
35
36 #BASIC OPERATIONS
37 # Square root
38 def sqrt(a):
39     root = a
40     return root
41
42 # logarithm
43 def log(a, b):
44     Log = math.log(a, b)
45     return Log
46
47 # SINE
48 def sin(a):
49     seno = math.sin(a)
50     return seno
51
52 # COSINE
53 def cos(a):
54     cosin = math.cos(a)
55     return cosin
56
57 # SUM
58 def sm(a, b):
59     s = a+b
60     return s
61
62 # subtraction
63 def sub(a, b):
64     su = a-b
65     return su
66
67 # Multiplication
68 def mult(a, b):
69     Mult = a*b
70     return Mult
71
72 # POTENCY
73 def pot(a, n):
74     Pot = a**n
75     return Pot
```

Fourth step: Testing our Library

At this point we test our library. Note that the test file and the library are in the same directory and this is exactly what you should do. In this project the test was carried out in jupyter. However, if you want to use colab for testing, you will need to upload the file to your colab environment.

Though our first step is to import our library into the environment where we want to test it, the procedure is the same as for any **python library**. First we call the name of the library and then we assign an abbreviation that will be used during its execution. You can see the example below:

Importing the library created

```
In [1]: 1 import los as ls
```

Executing the functions

In this first function we want to download data from yfinance as defined in our library. However, we create an object that receives the function to download the data. This happens according to the range and the symbol introduced by the user as seen in the figure below.

```
In [2]: 1 # DAWNLOADING FINANCIAL DATAS
        2 dbase = ls.fdata()

Enter start date: 2020-01-01
Enter end date: 2023-04-01
Stock: AMZN
[*****100%*****] 1 of 1 completed
```

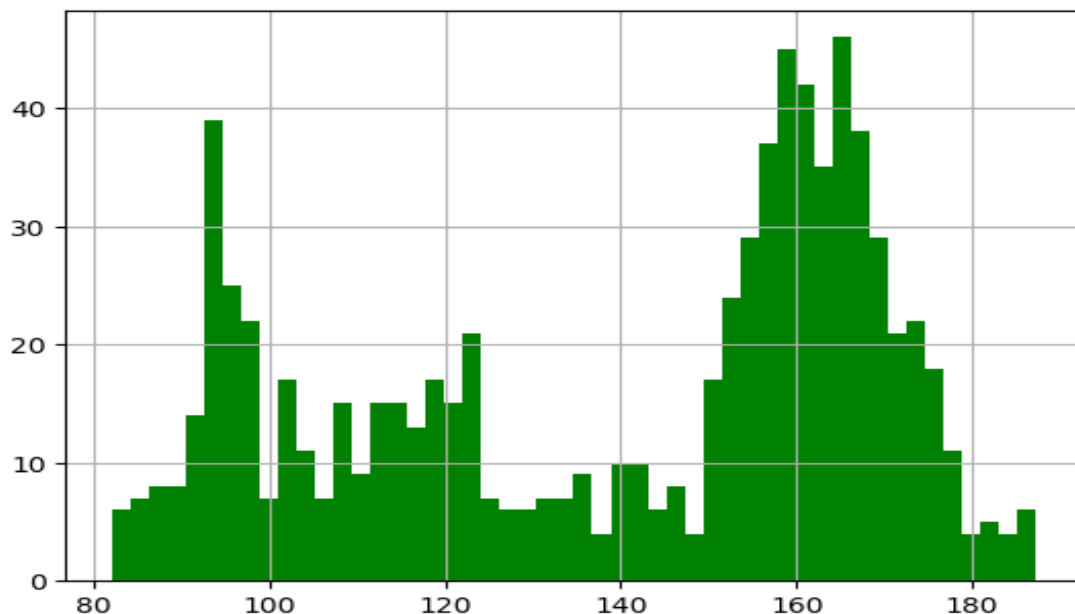
Out[2]:

	Open	High	Low	Close	Adj Close	Volume
count	808.000000	808.000000	808.000000	808.000000	808.000000	8.080000e+02
mean	139.540156	141.387896	137.557357	139.470030	139.470030	8.022596e+07
std	29.403657	29.409765	29.256445	29.283279	29.283279	3.481055e+07
min	82.075500	83.480003	81.301498	81.820000	81.820000	2.903800e+07
25%	113.046000	114.932499	110.750002	113.178751	113.178751	5.677650e+07
50%	152.713249	154.555252	149.894249	152.182755	152.182755	7.029100e+07
75%	164.117249	165.779629	162.180626	164.171745	164.171745	9.334000e+07
max	187.199997	188.654007	184.839493	186.570496	186.570496	3.113460e+08

Now we can also observe the characteristics of the data that we download with the python functions.

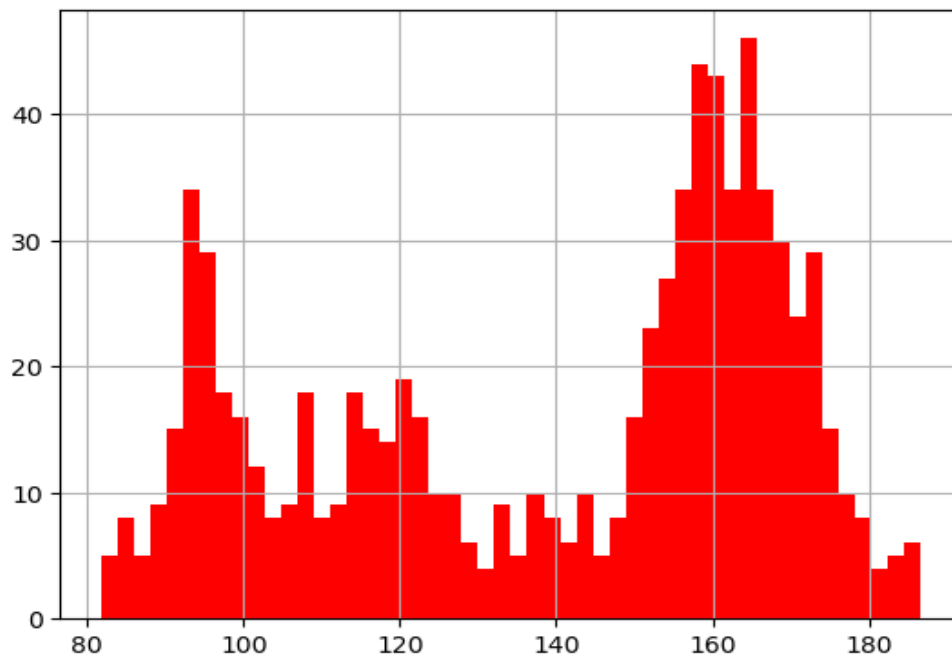
- **When the market is open**

```
1 dbase.Open.hist(bins = 50, color='green');
```



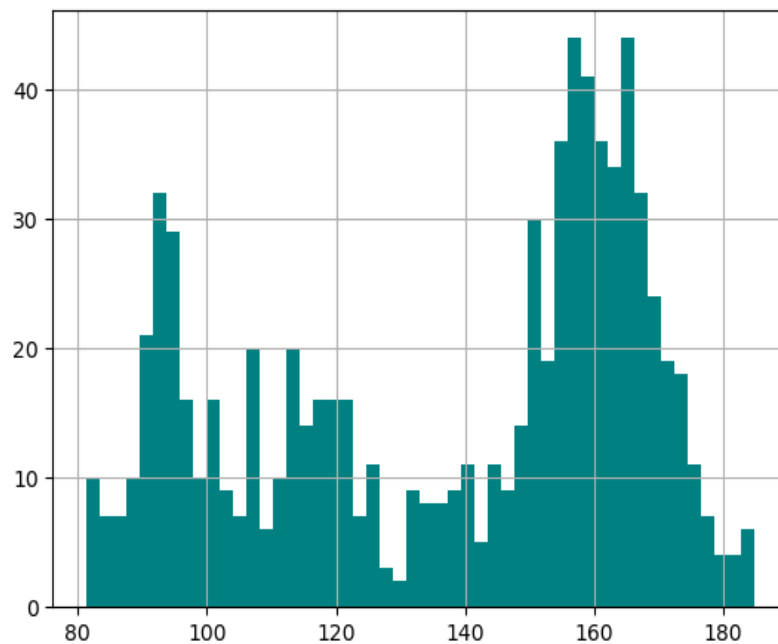
- When market is closed

```
1 dbase.Close.hist(bins = 50, color='red');
```



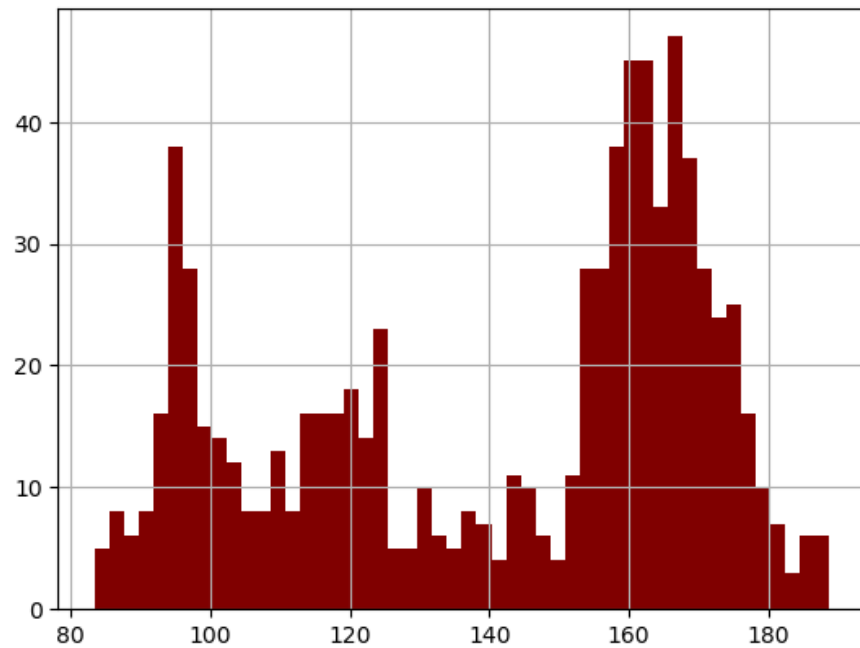
- When the price trend is down

```
[23]: 1 dbase.Low.hist(bins = 50, color='teal');
```



- When the price trend is high

```
[27]: ▶ 1 dbase.High.hist(bins = 50, color = 'maroon');
```



VOLUME CALCULATIONS OF GEOMETRIC SOLIDS

Another operation included in this library is the calculation of volumes of geometric solids. For this reason, we present the figure of the results obtained for each geometric solid.

Volumes of geometric solids

```
In [7]: ▶ 1 #prism volume  
        2 ls.pvolum(12, 3) # input base area and height values
```

Out[7]: 36

```
In [8]: ▶ 1 # pyramid volume  
        2 ls.pyv(12, 3) # input base area and height values
```

Out[8]: 12.0

```
In [9]: ▶ 1 # Cylinder volume  
        2 ls.cvol(2) # input volume value
```

Out[9]: 19.7192

```
In [10]: ▶ 1 # cone volume  
         2 ls.conv(3)# input volume value
```

Out[10]: 9.8596

```
In [11]: ▶ 1 # Sphere volume  
         2 ls.svol(2)# input ray volume
```

Out[11]: 33.49333333333333

Other mathematical operations

```
In [12]: 1 ls.sqrt(9) # Square root
```

```
Out[12]: 9
```

```
In [13]: 1 ls.log(1000, 30000) # Logarithms
```

```
Out[13]: 0.6700734309658197
```

```
In [14]: 1 ls.sin(203) # Sine
```

```
Out[14]: 0.9333097001669604
```

```
In [15]: 1 ls.cos(100) # Cosine
```

```
Out[15]: 0.8623188722876839
```

```
In [16]: 1 ls.pot(3, 2) # Potency
```

```
Out[16]: 9
```

```
In [17]: 1 ls.sm(12, 234) # Sum
```

```
Out[17]: 246
```

```
In [18]: 1 ls.sub(200, 437)# Subtraction
```

```
Out[18]: -237
```

```
In [19]: 1 ls.mult(234, 89)# Multiplication
```

```
Out[19]: 20826
```

```
In [20]: 1 ls.div(100, 2) # Divide
```

```
Out[20]: 50.0
```

Final considerations

We finally realized how amazing the python language is for creating fantastic solutions. Actually, this project was tested on the local machine as it is an educational one. So, in the next session we will learn how to upload it to [pypi](#) platform for online access. Stay tuned for the next article!

KEY WORDS

python library, financial data, geometric solids, mathematical calculations

Reference List

- <https://www.visionfactory.org/post/how-to-use-backtesting-to-create-a-trading-test-strategy>
- <https://pypi.org/project/yfinance/>
- <https://cplusplus.com/reference/cmath/>

BUILDING SMART JULIE WITH PYTHON AND OPENAI API



Contextualization

It seems that the title of this article has piqued your curiosity. Yes, this article was created for this purpose. I think the first question for you is who is smart **Julie**? That's exactly what I wanted you to ask because our main goal is to answer what smart **Julie** is. However, in a nutshell, smart **Julie** is a friend who, in addition to being kind, is also very intelligent, capable of answering all your questions and even giving you advice if you ask for it.

Technically, smart Julie will be a **virtual assistant** that we will build with **python** and **openai API**. So, we will be able to answer any question that is put to her, both academically and socially. However, the question has to be well formulated and the pronunciation of the words sufficient in the language that she intends to use. Therefore, in this project we will use English but any internationally recognized language can be used as well.

What do we need to complete this project?

The first step for this project is to know how to program with **python**, because the knowledge of installing libraries and connecting to an **API** will be essential. But don't worry because we will explain it step by step. So, you just need to follow all the steps and at the end we will provide the link from GitHub with source code.

1. Necessary tools

The tools needed for this project are **Python** programming language and following libraries:

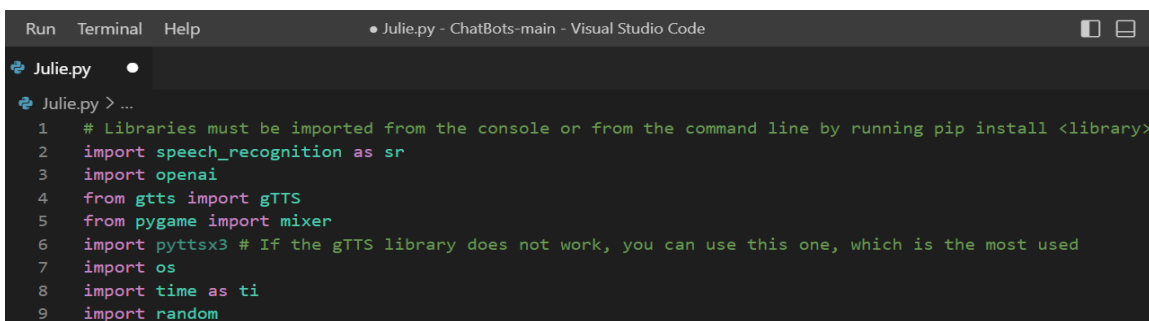
- **SpeechRecognition** is a **Python** library that allows you to recognize human speech and convert it into text. In particular, this can be used in various applications, such as **virtual assistants** and audio transcription. To use the library, you need to have a microphone connected to your computer and install the library using a package manager like pip.
- **OpenAI API** is an application programming interface (**API**) made available by OpenAI. This way developers can use the artificial intelligence models developed by the company in their own applications.
- **GTTS** stands for "Google Text-to-Speech" and is a free **API** made available by Google to convert text to speech. The **API** allows developers to integrate speech synthesis functionality into their own applications, using a variety of voices and languages.
- **Pygame** is a set of **Python** libraries aimed at game and multimedia development. Specifically, libraries include modules for window and event management, 2D graphics, image manipulation, audio, keyboard and mouse input, collision detection, and much more. In our case, we will use it for audio manipulation.
- **Pytsx3** is based on different speech synthesis engines like SAPI5 (Speech **API** 5) on Windows and NSSpeechSynthesizer on macOS. In addition, it supports multiple languages and voices, as well as real-time audio playback.
- **OS** : the **os** module is a standard **Python** library that provides a way to interact with the operating system. Actually, it allows you to work with the file system, get information from the environment, manage processes, change environment variables and much more.
- **TIME** the time module is a **Python** standard library that allows you to work with functions related to time, such as measuring durations. That is waiting for a certain period before executing a certain action, formatting dates and times, among other features.
- **Random**: the random module is a standard **Python** module that provides functions for generating pseudorandom numbers. In fact, these functions are useful for various applications such as games, simulations, and cryptography.

- **Text editor** is where we write our code in this case, we use vscode but you can choose your preferences

2. Installation and import of libraries

Before importing the libraries make sure that you have python installed on your computer and that your microphone is operational because some libraries require it. Regarding the procedures for installing **python** and libraries, find in this [article](#) where we present all the details for the purpose, below we present the imports.

3. Coding



```
Run Terminal Help • Julie.py - ChatBots-main - Visual Studio Code
Julie.py
Julie.py > ...
1 # Libraries must be imported from the console or from the command line by running pip install <library>
2 import speech_recognition as sr
3 import openai
4 from gtts import gTTS
5 from pygame import mixer
6 import pyttsx3 # If the gTTS library does not work, you can use this one, which is the most used
7 import os
8 import time as ti
9 import random
```

Next we call our **API** key in the form of a string

```
openai.api_key = "paste your api key here"
```

We define the function that transforms the voice captured in the mic to text

```

15 # We define the function that transforms the voice captured in the mic to text
16 def transform_audio_in_text():
17
18     r = sr.Recognizer()
19     with sr.Microphone() as origen:
20         r.pause_threshold = 0.9
21         print("Speak something please!")
22         audio = r.listen(origen)
23         try:
24             # The language with which the voice is recognized must be specified
25             request = r.recognize_google(audio, language="en-EN")
26             print("You: " + request)
27             return request
28
29         except sr.UnknownValueError:
30             print("Ups, Sorry I didn't understand!")
31             return "Please can repeat?"
32
33         except sr.RequestError:
34             print("Ups, no service!")
35             return "waiting"
36
37         except:
38             print("Ups, something are wrong!")
39             return "waiting"

```

Afterwards, we define the function that will transform the text (message) into audio, I leave both for gTTS library and for pyttsx3.

```

# We define the function that will transform the text (message) into audio, I leave both for gTTS library and for pyt
def speak (mensaje):
#This section of code is for gTTS library **
    volume = 0.7
    tts = gTTS(mensaje, lang="en", slow=False)
    ran = random.randint(0,9999)
    filename = 'Temp' + format(ran) + '.mp3'
    tts.save(filename)
    mixer.init()
    mixer.music.load(filename)
    mixer.music.set_volume(volume)
    mixer.music.play()

    while mixer.music.get_busy():
        ti.sleep(0.3)

    mixer.quit()
    os.remove(filename)

```

***** This section of code is for pyttsx3 library, just use one removing the numeral

In case the gTTS library doesn't work

Start the pyttsx3 engine

#engine = pyttsx3.init()

Slow down playback speed, default is 200

#engine.setProperty('rate', 150)

speak message

#engine.say(message)

engine.runAndWait()

```
71 def despedida():
72     speak("See you, have a great day!")
73     exit()
74 def main():
75     conversation = ""
76     despedida_phrases = ["tchau", "thank you by", "see you later", "see you soon", "i have to go", "end", "see off"]
77     speak ("Hello I'm Julie, welcome, I'm willing to answer all of exiyour questions, please let's start the conversa
78     while True:
79         question = transform_audio_in_text().lower()
80         conversation += "\nYou: " + question + "\nJulie:"
81         response = openai.Completion.create(
82             model="text-davinci-003",
83             prompt=conversation,
84             temperature=0.5,
85             max_tokens=3500,
86             top_p=0.5,
87             frequency_penalty=0.7,
88             presence_penalty=0.0,
89             stop=["\n", " You:", " Julie:"]
90         )
91         answer = response.choices[0].text.strip()
92         conversation += answer
93         print("Julie: " + answer + "\n")
94         speak(answer)
95         # Check if any of the despedida phrases was said to stop the conversation
96         if any(frase in question for frase in despedida_phrases):
97             despedida()
98     main()
```

FULL CODE

```
Go Run Terminal Help • Julie.py - Smart_Julie - Visual Studio Code
test.py Julie.py Los.py ls.py testes.py 2
Julie.py > main
1 # Libraries must be imported from the console or from the command line by running pip install <library>
2 import speech_recognition as sr
3 import openai
4 from gtts import gTTS
5 from pygame import mixer
6 import pyttsx3 # If the gTTS library does not work, you can use this one, which is the most used
7 import os
8 import time as ti
9 import random
10
11
12 openai.api_key = "sk-8XDtCbMOL5ZEbg14Q1YvT3B1bkFJN5zsZoNuqtBXNQcaSdeA"
13
14
15 # We define the function that transforms the voice captured in the mic to text
16 def transform_audio_in_text():
17
18     r = sr.Recognizer()
19     with sr.Microphone() as origen:
20         r.pause_threshold = 0.9
21         print("Speak something please!")
22         audio = r.listen(origen)
23         try:
24             # The language with which the voice is recognized must be specified
25             request = r.recognize_google(audio, language="en-EN")
26             print("You: " + request)
27             return request
28
29         except sr.UnknownValueError:
30             print("Ups, Sorry I didn't understand!")
31
32             return "Please can repeat?"
33
34         except sr.RequestError:
35             print("Ups, no service!")
36             return "waiting"
37
38         except:
39             print("Ups, something are wrong!")
40             return "waiting"
41
42 # We define the function that will transform the text (message) into audio, I leave both for gTTS library and for pyt
43 def speak (mensaje):
44     #This section of code is for gTTS library **
45     volume = 0.7
46     tts = gTTS(mensaje, lang="en", slow=False)
47     ran = random.randint(0,9999)
48     filename = 'Temp' + format(ran) + '.mp3'
49     tts.save(filename)
50     mixer.init()
51     mixer.music.load(filename)
52     mixer.music.set_volume(volume)
53     mixer.music.play()
54
55     while mixer.music.get_busy():
56         ti.sleep(0.3)
57
58     mixer.quit()
59     os.remove(filename)
```

```

71 def despedida():
72     speak("See you, have a great day!")
73     exit()
74 def main():
75     conversation = ""
76     despedida_phrases = ["tchau", "thank you by", "see you later", "see you soom", "i have to go", "end", "see off"]
77     speak ("Hello I'm Julie, welcome, I'm willing to answer all of exiyour questions, please let's start the conversa
78     while True:
79         question = transform_audio_in_text().lower()
80         conversation += "\nYou: " + question + "\nJulie:"
81         response = openai.Completion.create(
82             model="text-davinci-003",
83             prompt=conversation,
84             temperature=0.5,
85             max_tokens=3500,
86             top_p=0.5,
87             frequency_penalty=0.7,
88             presence_penalty=0.0,
89             stop=["\n", " You:", " Julie:"]
90         )
91         answer = response.choices[0].text.strip()
92         conversation += answer
93         print("Julie: " + answer + "\n")
94         speak(answer)
95         # Check if any of the despedida phrases was said to stop the conversation
96         if any(frase in question for frase in despedida_phrases):
97             despedida()
98     main()

```

Final considerations

Once you have completed all the steps, you can talk to our assistant **Julie** and she is already in a position to answer your questions. So, don't lose time! Find the source code [here](#) and test it on your own computer and have fun. Remember that the next article will be much more interesting because we are going to convert Julie into an installable setup on your computer or smartphone don't miss it out!

Keywords

[Julie](#), [openAI](#), [python](#), [API](#), [Virtual assistant](#)

Reference List

1. <https://pypi.org/project/SpeechRecognition/>
2. <https://openai.com/research/overview>
3. <https://gtts.readthedocs.io/en/latest/>
4. <https://www.pygame.org/docs/>
5. <https://pyttxs3.readthedocs.io/en/latest/>
6. <https://stackoverflow.com/questions/4813238/difference-between-subprocess-popen-and-os-system>
7. <https://docs.python.org/3/library/time.html>
8. <https://docs.python.org/3/library/random.html>

TRANSFORMING SMART JULIE FROM CODE TO APPLICATION



Contextualization

Hello, welcome to our second stage with **Julie**, before I didn't explain the reasons that led me to choose this name, this is due to a tribute to a friend of the same name and as smart as her!

However, in this step you don't need to code practically anything, you just need attention to implement it properly and talk to **Julie** asking her by voice or written messages everything you want to know!

Point out that our friend **Julie** understands English and Spanish well written or well pronounced including phrasal verbs that is everything.

WHAT DO I NEED TO TALK TO HER?

Well, I could even say that you don't need anything, but you do. The question is what if everything is already done, of course everything is done, but for him to hear you he must make sure that your computer's microphone is working correctly otherwise you can connect an external microphone that's all. Also remember that at this stage it can be run on windows or Linux except android.

HOW CAN I SAY GOODBYE TO JULIE AND END OUR CONVERSATION?

Obviously when we want to say goodbye to a friend there are specific expressions so it's no different with **Julie**, for that purpose we need to use one of the following expressions: *bye, I must go, see you later, see off, or end.*

HOW TO IMPLEMENT THE PROJECT?

Most of the necessary tools have already been described in the previous [article](#), so we just added a few technologies described below.

1. Necessary tools

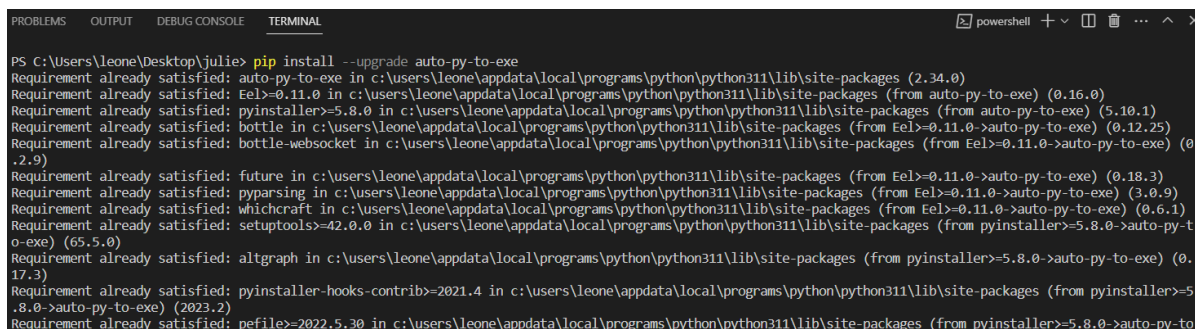
For this phase we need following tools:

- **Tkinter** the Tkinter library is based on the Tk toolkit, which is a cross-platform GUI library. Tk provides the necessary building blocks for creating a graphical interface, and Tkinter is an abstraction layer that allows the use of Tk in Python. Since Tkinter is part of the standard Python library, there is no need to install anything additional to use it. It is available on virtually all platforms where Python runs, including Windows, macOS, and Linux.
- **Threading** The "threading" library provides a simple API for working with threads. Developers can create threads using the "Thread" class and implement the thread's logic within a method. Additionally, the library provides mechanisms for synchronization and communication between threads, such as locks, semaphores, and condition variables, which help avoid concurrency issues like race conditions and simultaneous access to shared resources. It is available on virtually all platforms where Python runs, including Windows, macOS, and Linux.

- **Auto-py-to-exe** Auto-py-to-exe is a Python utility that allows you to convert a Python script into a standalone executable file. It simplifies the process of converting your Python code into an executable format, making it easier to distribute and run on systems that don't have Python installed. Simplifies the usage of PyInstaller by providing a user-friendly interface and pre-configured settings. It abstracts away some of the complexities and technical details involved in the conversion process, making it accessible to users with varying levels of Python and packaging knowledge. To install run: *pip install auto-py-to-exe* on your terminal.

CONVERTING CODE TO EXECUTABLE FILE

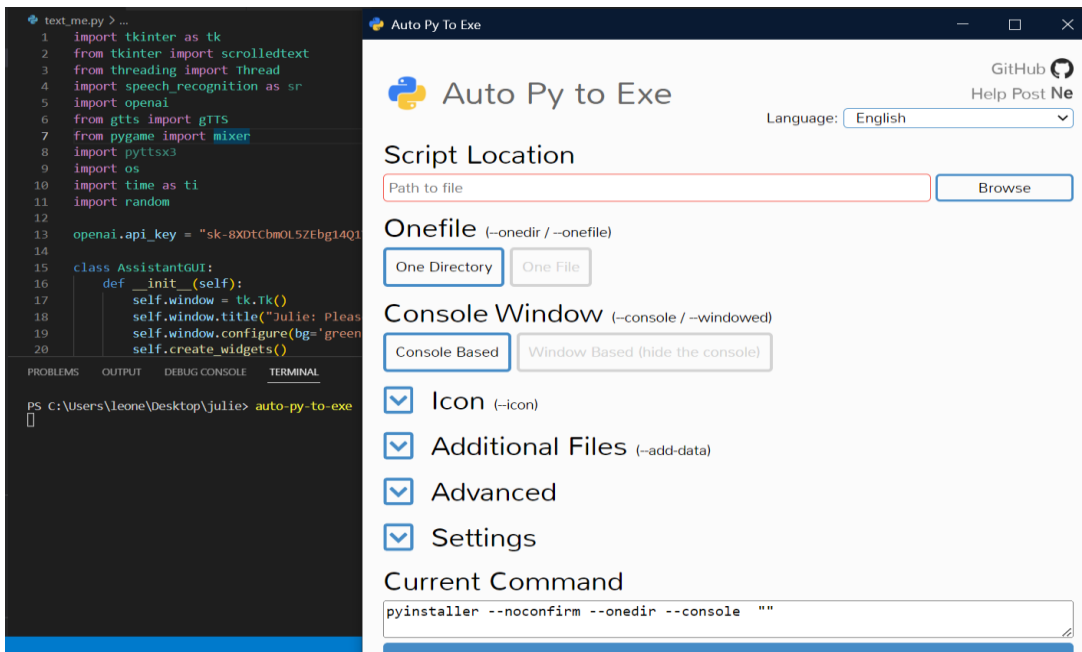
As we explained earlier, at this point we will be using the auto-py-to-exe library. as you can see in the figure below:



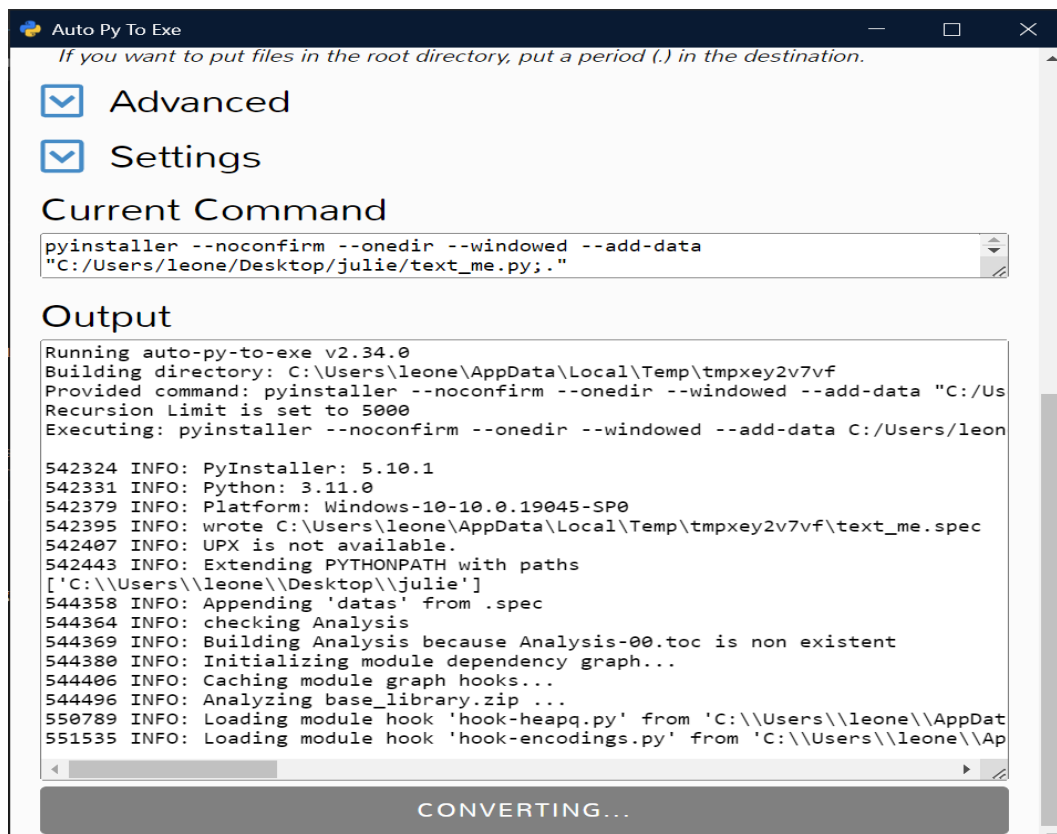
```
PS C:\Users\leone\Desktop\julie> pip install --upgrade auto-py-to-exe
Requirement already satisfied: auto-py-to-exe in c:\users\leone\appdata\local\programs\python\python311\lib\site-packages (2.34.0)
Requirement already satisfied: Eel>=0.11.0 in c:\users\leone\appdata\local\programs\python\python311\lib\site-packages (from auto-py-to-exe) (0.16.0)
Requirement already satisfied: pyinstaller>=5.8.0 in c:\users\leone\appdata\local\programs\python\python311\lib\site-packages (from auto-py-to-exe) (5.10.1)
Requirement already satisfied: bottle in c:\users\leone\appdata\local\programs\python\python311\lib\site-packages (from Eel>=0.11.0->auto-py-to-exe) (0.12.25)
Requirement already satisfied: bottle-websocket in c:\users\leone\appdata\local\programs\python\python311\lib\site-packages (from Eel>=0.11.0->auto-py-to-exe) (0.2.9)
Requirement already satisfied: future in c:\users\leone\appdata\local\programs\python\python311\lib\site-packages (from Eel>=0.11.0->auto-py-to-exe) (0.18.3)
Requirement already satisfied: pyparsing in c:\users\leone\appdata\local\programs\python\python311\lib\site-packages (from Eel>=0.11.0->auto-py-to-exe) (3.0.9)
Requirement already satisfied: whichcraft in c:\users\leone\appdata\local\programs\python\python311\lib\site-packages (from Eel>=0.11.0->auto-py-to-exe) (0.6.1)
Requirement already satisfied: setuptools>=42.0.0 in c:\users\leone\appdata\local\programs\python\python311\lib\site-packages (from pyinstaller>=5.8.0->auto-py-to-exe) (65.5.0)
Requirement already satisfied: altgraph in c:\users\leone\appdata\local\programs\python\python311\lib\site-packages (from pyinstaller>=5.8.0->auto-py-to-exe) (0.17.3)
Requirement already satisfied: pyinstaller-hooks-contrib>=2021.4 in c:\users\leone\appdata\local\programs\python\python311\lib\site-packages (from pyinstaller>=5.8.0->auto-py-to-exe) (2023.2)
Requirement already satisfied: pefile>=2022.5.30 in c:\users\leone\appdata\local\programs\python\python311\lib\site-packages (from pyinstaller>=5.8.0->auto-py-to-exe)
```

In my case I used upgrade because I already have it installed and I'm just updating it and in your case remove upgrade and run it.

Then find full code [here](#) and run auto-to-py-exe to open the GUI window and load the file you want to convert. Like the figure below:



Once done, load your file, select the one file options for just one file and windows based for the graphical interface and in the additional file option insert another file. and click the convert py to exe option. As you can see the figure:



And finally locate your converted project in the previous project directory in an automatically generated folder with the name outpu or click the open output folder button. See it in the figure below:

```
634575 INFO: Building PYZ (ZlibArchive) C:\Users\leone\AppData\Local\Temp\tmpxe
637815 INFO: Building PYZ (ZlibArchive) C:\Users\leone\AppData\Local\Temp\tmpxe
637893 INFO: checking PKG
637914 INFO: Building PKG because PKG-00.toc is non existent
637930 INFO: Building PKG (CArchive) text_me.pkg
638004 INFO: Building PKG (CArchive) text_me.pkg completed successfully.
638025 INFO: Bootloader C:\Users\leone\AppData\Local\Programs\Python\Python311\
638040 INFO: checking EXE
638046 INFO: Building EXE because EXE-00.toc is non existent
638056 INFO: Building EXE from EXE-00.toc
638064 INFO: Copying bootloader EXE to C:\Users\leone\AppData\Local\Temp\tmpxey
638174 INFO: Copying icon to EXE
638189 INFO: Copying icons from ['C:\Users\leone\AppData\Local\Programs\Py
638240 INFO: Writing RT_GROUP_ICON 0 resource with 104 bytes
638261 INFO: Writing RT_ICON 1 resource with 3752 bytes
638277 INFO: Writing RT_ICON 2 resource with 2216 bytes
638283 INFO: Writing RT_ICON 3 resource with 1384 bytes
638292 INFO: Writing RT_ICON 4 resource with 38188 bytes
638307 INFO: Writing RT_ICON 5 resource with 9640 bytes
638322 INFO: Writing RT_ICON 6 resource with 4264 bytes
638338 INFO: Writing RT_ICON 7 resource with 1128 bytes
638364 INFO: Copying 0 resources to EXE
638384 INFO: Embedding manifest in EXE
638406 INFO: Updating manifest in C:\Users\leone\AppData\Local\Temp\tmpxey2v7vf
638550 INFO: Updating resource type 24 name 1 language 0
638584 INFO: Appending PKG archive to EXE
638650 INFO: Fixing EXE headers
639426 INFO: Building EXE from EXE-00.toc completed successfully.
639457 INFO: checking COLLECT
639466 INFO: Building COLLECT because COLLECT-00.toc is non existent
639479 INFO: Building COLLECT COLLECT-00.toc
654305 INFO: Building COLLECT COLLECT-00.toc completed successfully.

Moving project to: C:\Users\leone\Desktop\julie\output
Complete.
```

Something wrong with your exe? Read [this post on how to fix common issues](#) for possible solutions.

CLEAR OUTPUT **OPEN OUTPUT FOLDER**

FINAL CONSIDERATIONS

Finally, we can talk to Julie by simply downloading the files available [here](#) and running them on the operating systems. If you want to contribute to the progress of this project, contact me! Enjoy and visit us often for more python news.

Keywords

[Julie](#), [convert](#), [application](#), [python](#), [API](#), [Virtual assistant](#)

Reference List

- <https://docs.python.org/3/library/tkinter.html>
- <https://docs.python.org/3/library/threading.html>
- <https://pypi.org/project/auto-py-to-exe/>

Critical Analysis of the main Trading Strategies



Welcome to our article about **trading** strategies and their critical **analysis**. These aim to reduce risks and enhance gains through the rational and less emotional use of **trading** strategies.

Concepts and contextualization

According (Mashaushi, 2017), Traders develop strategies to profit from short-term buying and selling decisions of **financial** assets. Consider limitations and challenges when analyzing **trading** strategies. Here are key critical points to ponder:

a) Risk and volatility

Trading strategies entail risks in volatile markets. Asset price unpredictability can cause substantial losses, even for experienced traders. Volatility arises from macroeconomic events, market behavior, or policy changes, making consistent strategy application challenging.

b) Market timing

Trading strategies aim to accurately predict future market movements. However, accurately predicting short-term market behavior is extremely challenging. Market factors, including human psychology, make it challenging to pinpoint precise entry and exit points.

c) Emotions and discipline

Greed and fear can lead to impulsive or irrational decisions, negatively affecting the consistent application of a strategy. Maintaining emotional discipline and adhering to the rules of the strategy are crucial but can be challenging for some traders.

d) Retrospective bias and overfitting

Avoid retrospective bias when developing a trading strategy. Beware of finding profitable patterns in historical data that may not repeat. Overfitting can make a strategy ineffective with new data. Test strategies in various market conditions for reliability.

Some trading strategies and their respective critical analysis

Based (Quantified, 2023), Traders use diverse strategies based on style, risk profile, and **financial** instrument. Here are some main strategies and their critical analysis.

1. Scalping

In this strategy, traders aim to take advantage of small price fluctuations throughout the day by executing multiple short-term trades. They seek to make quick profits by capitalizing on market liquidity.

Analysis

For scalping, fast platform access and price understanding are key. Risk management, profit targets, and stop-loss levels prevent major losses.

1. Day trading

In this strategy, traders open and close their positions within the same day, aiming to profit from daily price fluctuations. They may use technical **analysis**, chart patterns, and indicators to identify trading opportunities.

Analysis

It is important to understand key technical indicators such as moving averages, oscillators, and Bollinger bands. Additionally, fundamental **analysis** can be useful in identifying economic events or news that may impact asset prices.

2. Swing trading

This strategy involves opening positions that can last for several days to several weeks, with the goal of capturing large price moves. Traders may use technical and fundamental analysis to identify medium-term trends and entry and exit points.

Analysis

Swing traders analyze trends, support/resistance, and use indicators for confirmation. Set profit targets and stop-loss levels based on asset volatility.

3. Algorithmic trading

Traders automate trades using algorithms and predefined rules based on technical **analysis**, fundamentals, or other criteria.

Analysis

When using algorithmic **trading**, it is essential to have programming skills or access to specialized platforms and software. Additionally, it is important to test and optimize the algorithms under different market conditions to ensure their effectiveness.

Aspects to consider when analyzing a strategy before its application

According (ClearPointStrategies, 2023), When it comes to analyzing these strategies, it is crucial to consider a few aspects:

- Evaluate the performance history of the strategy under different market conditions.
- Analyze transaction costs, including brokerage fees, spreads, and taxes.
- Take risk management into account by defining appropriate stop-loss and profit target levels.
- Assess the suitability of the strategy to the risk profile and the time available for **trading**.

Final considerations

Trading strategies can be powerful tools to achieve profits in the **financial** market. However, they require knowledge, skill, and a disciplined approach. Therefore, critical **analysis** and continuous adjustment of strategies are essential to adapt to market changes and maximize results.

Tags

- **Trading**
- **Financial Analysis**

Reference List

ClearPointStrategies. (19 de 4 de 2023). Obtido de

<https://www.clearpointstrategy.com/blog/strategic-analysis>

Mashaushi, K. R. (September de 2006). An Analysis of Technical Trading Strategies.

https://etheses.whiterose.ac.uk/696/1/uk_bl_ethos_431997.pdf

Quantified. (16 de 5 de 2023). *Quantified Strategies*. Obtido de

<https://www.quantifiedstrategies.com/technical-analysis-strategy/>

Wikipedia. (14 de 6 de 2023). *Wikipedia*. Obtido de

https://en.wikipedia.org/wiki/Algorithmic_trading