

Centro Universitário Carlos Drummond de Andrade

Alex Dias Julião RA: 192000682

Fabiano dos Santos Lins RA: 192000766

Guilherme Marques Sande RA: 201000163

Luã Augusto Santana RA : 999001352

Rodrigo de Barros Rebelo Almendros RA: 201000106

Carteira de Ações com Inteligência Artificial

São Paulo - SP 2021

Análise e Desenvolvimento de Sistemas – **Centro Universitário Carlos Drummond de Andrade**. Eduardo Palhares Júnior . JD Ponte Rasa, 2021.

RESUMO

Quando falamos em Mercado de Ações, entramos em uma questão que pode ser um pouco complexa no entendimento das pessoas, muitas pelo fato de não terem nenhum tipo de base sobre educação financeira ou mesmo nunca ter ouvido falar. Com base nisso necessitamos descobrir um pouco sobre o que seria uma compra de ação, que nada mais é que o investidor leva uma pequena parte de uma empresa de terceiros e passa a ser chamado de acionista minoritário. Para facilitar as pessoas a tomarem a decisão certa na hora de realizar investimentos, um sistema de Inteligência Artificial utilizando técnicas como Machine Learning e Processamento de Linguagem Natural, ensinaremos a IA sobre o comportamento e movimento das cotações dos ativos listados nos fundamentos das empresas, com apoio do Data Mining prevendo resultados, para que no final nos dê a tomada de decisão na compra de ações.

When we talk about Stock Markets, we enter into an issue that can be a little complex for people to understand, many of them because they have no basis in financial education or even have never heard of it. Based on this, we need to find out a little about what a share purchase would be, which is nothing more than the investor taking a small part of a third-party company and is now called a minority shareholder. To make it easier for people to make the right decision when making investments, an Artificial Intelligence system using techniques such as Machine Learning and Natural Language Processing, we will teach AI about the behavior and movement of quotations for assets listed in the fundamentals of companies, with support from Data Mining predicting results, so that in the end it gives us the decision to purchase shares.

Palavras-Chave: Ciências de Dados, Machine Learning, Árvore de decisão, Inteligência Artificial.

SUMÁRIO

| | |
|--|----|
| 1. INTRODUÇÃO | 4 |
| 2. DESENVOLVIMENTO | 6 |
| 2.1.OBJETIVOS | 6 |
| 2.2. JUSTIFICATIVA E DELIMITAÇÃO DO PROBLEMA | 6 |
| 2.3.LIMPEZA E FORMATAÇÃO DOS DADOS | 7 |
| 2.4.DADOS FUNDAMENTAIS DE RELATÓRIOS TRIMESTRAIS | 8 |
| 2.5. EXPLORANDO OS DADOS | 10 |
| 2.6.ÍMPORTAÇÕES E CARREGAMENTO DA BIBLIOTECA NO DF QUE CRIAMOS ANTERIORMENTE | 10 |
| 2.7.CRIAR DICIONÁRIOS DE DATA FRAMES COM CADA EMPRESA | 12 |
| 2.8.ITERANDO ATRAVÉS DE CADA BALANÇOS PARA TRANSFORMAR OS DADOS | 12 |
| 2.9.CORRELAÇÕES EM NOSSOS DADOS | 13 |
| 2.10.SELEÇÃO DE RECURSOS | 14 |
| 2.11.CLASSIFICADORES DE APRENDIZADO DE MÁQUINA | 18 |
| 2.12. METODOLOGIA | 25 |
| 2.13.NOSSOS DADOS | 26 |
| 2.14.CLASSIFICADOR FICTÍCIO | 27 |
| 2.15.CODIFICANDO O MODELO DE CLASSIFICAÇÃO | 27 |
| 2.16.CRIANDO O CLASSIFICADOR FICTÍCIO | 28 |
| 2.17.MATRIZ DE CONFUSÃO | 30 |
| 2.18.CONCLUSÃO | 30 |
| 3. RESULTADOS | 31 |
| 3.1. SOLUÇÃO INICIAL | 31 |
| 3.2. SOLUÇÃO FINAL | 31 |
| 4. CONSIDERAÇÕES FINAIS | 33 |
| 5. REFERÊNCIAS | 34 |

1. INTRODUÇÃO

Quando se trata de negociar no Mercado de Ações, existem muitas abordagens diferentes para encontrar a ação certa. Muitas formas de análise surgiram para detectar qual ação vale o dinheiro. Técnicas de Inteligência Artificial (IA), especialmente uma de suas áreas, o aprendizado de máquina (AM) do inglês, Machine Learning (ML), o processamento de linguagem natural (PLN) é um ramo da inteligência artificial que ajuda os computadores a entender, interpretar e manipular a linguagem humana. A PLN se baseia em muitas disciplinas, incluindo ciência da computação e linguística computacional, em sua busca por preencher a lacuna entre a comunicação humana e a compreensão do computador, a máquina aprende dados para tomar decisões e prever valores conforme aprendeu. Existem diferentes problemas que são utilizados, como: detecção de e-mail spam, detecção de fraudes, recomendação de produtos para um usuário, reconhecimento de padrões e imagem, entre outras.

“O aprendizado de máquina é uma área que provê aos computadores a habilidade de aprender sem que algo seja programado explicitamente.”

(SAMUEL, 2000, minha tradução).

Deste modo o aprendizado de máquina tem o objetivo de desenvolver técnicas computacionais sobre o aprendizado, sistemas capazes de adquirir conhecimento automático.

Esse sistema é capaz de tomar decisões baseado no que aprendeu anteriormente, nas suas experiências com sucesso anteriores.

Esse projeto tem como objetivo criar uma inteligência artificial, fins de tomada de decisão para comprar uma carteira de ações, é de uso acadêmico sem fins lucrativos e comerciais, não recomendamos seu uso para investimentos, o objetivo é aplicar ciência de dados.

A inteligência artificial (IA) é usada de várias formas na bolsa de valores, especialmente para avaliar a variação das ações e escolha de quando você poderá comprar e quando deverá vender.

O projeto tem como objetivo fazer uma análise fundamentalista, onde pegamos uma lista de várias empresas extraímos seus fundamentos que é o seu faturamento, seu lucro, quanto tem de ativos, quanto tem passivo, e nossa inteligência artificial com base nas análises irá indicar quando deveremos realizar a venda ou compra de uma carteira de ação, em outras palavras irá fazer uma análise e baseado nessa análise deverá decidir qual empresa deverá ou não comprar

Essa análise consiste em aplicar IA em todas as empresas no período de tempo por trimestre, irá escolher quais são essas empresas que serão compradas e no final do trimestre iremos apresentar o resultado, e saber se a análise deu certo ou não.

2. DESENVOLVIMENTO

2.1 Objetivos

Desenvolver uma inteligência artificial para analisar o comportamento e movimento das cotações dos ativos listados nos fundamentos das empresas, com apoio da mineração de dados, para auxiliar na tomada de decisão na compra de ações.

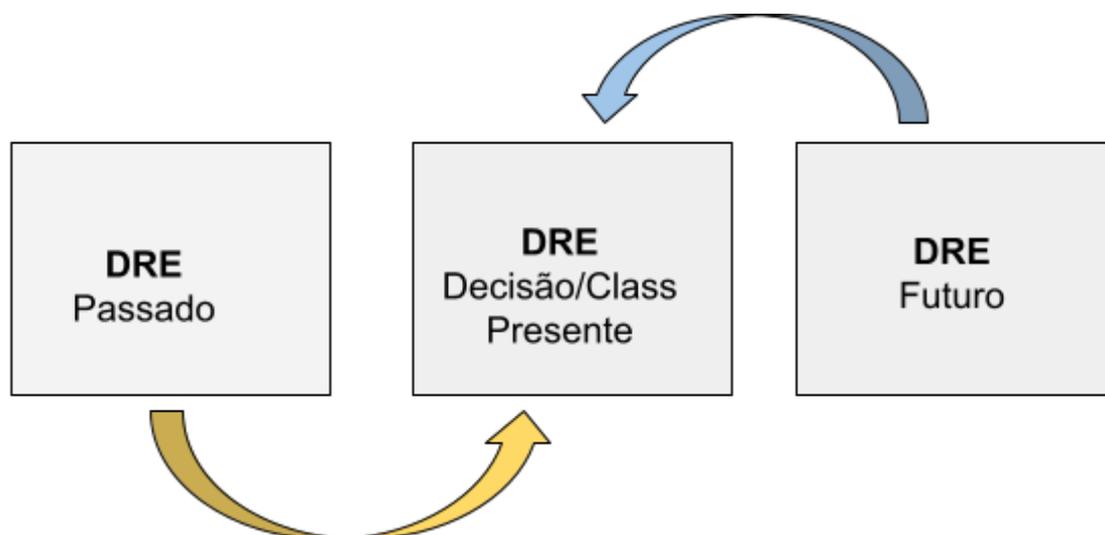
2.2. Justificativa e delimitação do problema

Para uma pessoa que não possui conhecimento em economia ou no mercado financeiro, é muito difícil decidir qual a melhor carteira de ações para se ter no momento, pois a oscilação do mercado é muito grande. São muitas variáveis a serem analisadas sobre as empresas para que uma pessoa possa decidir se vale a pena comprar essas ações ou não. Com isso, a inteligência artificial fará o trabalho de analisar esses dados das empresas a cada trimestre e ajudar a pessoa a decidir se determinada ação de uma empresa é viável de se adquirir ou não.

Vamos implementar um classificador de aprendizado de máquina para determinar se uma ação é uma Compra, Venda ou Espera. Para determinar se uma ação ou empresa se enquadra em uma dessas três categorias ou classes, examinaremos o relatório trimestral de cada empresa. Esses relatórios trimestrais contêm as informações financeiras necessárias de que precisamos para treinar nosso classificador de aprendizado de máquina nas formas de Análise Fundamental.

Como a Análise Fundamental será implementada na inteligência Artificial: com base em um relatório trimestral de qualquer trimestre escolhido, será observado o quanto os valores mudaram do relatório anterior para o atualmente selecionado. Em seguida, observará os valores de preço do próximo relatório trimestral para ver se houve algum salto significativo no preço. Finalmente, com o presente relatório contendo as mudanças do relatório anterior e o comportamento do preço do relatório futuro, será determinado se é uma Compra, Espera ou Venda.

Basicamente, estamos detectando se mudanças fundamentais do trimestre anterior para o atual afetam os preços futuros. Avaliaremos o desempenho de cada relatório trimestral em relação ao último relatório e, em seguida, observando o comportamento dos preços no futuro.



Aplicamos esse método de análise a cada relatório trimestral para criar nossos dados fundamentais. Este método classifica se vale a pena investir em uma ação no trimestre. Evidentemente, não poderemos usar este método no primeiro DRE ou no DRE mais recente porque a natureza da análise requer um relatório trimestral passado e futuro.

2.3.Limpeza e formatação dos dados

Para categorizar ou classificar cada relatório trimestral, usamos a seguinte lógica: se o preço subir um valor significativo no próximo trimestre, então é uma Compra; se for baixo, então é uma venda; se não, é um Não Comprar.

Aqui estão nossos requisitos específicos de classe para cada relatório trimestral observado:

Criando os rótulos: Comprar, Não Comprar ou Vender

Não queremos saber quando vender, mas incluí essa categoria para conseguir identificar quando que o nosso modelo vai sugerir uma compra quando na verdade o melhor momento era vender. Isso significa que o modelo errou "mais" do que quando sugeriu comprar e simplesmente o certo era não comprar

Regra:

- **Comprar** (Valor = 2) Subiu mais do que o Ibovespa (ou caiu menos)
- **Não Comprar** (Valor = 1) Subiu menos do que o Ibovespa até Ibovespa - 2% (ou caiu mais do que Ibovespa até Ibovespa -2%)
- **Vender** (Valor = 0) Subiu menos do que o Ibovespa - 2% (ou caiu mais do que Ibovespa -2%)

Existem outros resultados possíveis que poderíamos considerar, mas para simplificar, vamos manter essas condições como estão. Não saberemos a alta e a baixa do preço do próximo trimestre para o relatório trimestral mais recente, porque isso seria conhecer o futuro, o que é impossível.

2.4.Dados fundamentais de relatórios trimestrais

Para nossos dados fundamentais (Ações, Ativos etc.) como mencionado antes, observamos dois DREs para criar valores. As mudanças do DRE anterior para o atual ou atual serão medidas como uma mudança percentual em vez de seu valor real. Por exemplo: Digamos que no trimestre anterior, as ações mostraram um valor de R\$1.000. Então, no trimestre atual, as ações agora mostram um valor de R\$1.100; um aumento de 10%. Agora, substituímos o valor da ação de R\$1.100 por 10% para nosso DRE atual.

Fazemos esse processo para cada DRE (excluindo o primeiro). Agora, cada DRE tem alteração percentual para cada valor fundamental.

2.5.Codificando nossa Análise Fundamental

Com os DREs formatados com alterações percentuais e rotulados como Comprar, Não Comprar ou Vender, passamos para o processo de programação:

Importando Bibliotecas e Conjunto de Dados

```
1. import pandas as pd
2. import os
3.
4. empresas = ["ABEV3", "AZUL4", "BTOW3", "B3SA3", "BBSE3", "BRML3", "BBDC4", "BRAP4", "BBAS3",
              "BRKM5", "BRFS3", "BPAC11", "CRFB3", "CCRO3", "CMIG4", "HGTX3", "CIEL3", "COGN3", "CPL6", "CSAN3",
              "CPFE3", "CVCB3", "CYRE3", "ECOR3", "ELET6", "EMBR3", "ENBR3", "ENGI11", "ENEV3", "EGIE3", "EQT3",
              "EZTC3", "FLRY3", "GGBR4", "GOAU4", "GOLL4", "NTCO3", "HAPV3", "HYFE3", "IGTA3", "GNDI3", "ITSA4", "ITUB4",
              "JBSS3", "JHSF3", "KLBN11", "RENT3", "LCAM3", "LAME4", "LREN3", "MGLU3", "MRFG3", "BEEF3", "MRVE3",
              "MULT3", "PCAR3", "PETR4", "BRDT3", "PRIO3", "QUAL3", "RADL3", "RAIL3", "SBSP3", "SANB11", "CSNA3",
              "SULA11", "SUZB3", "TAE11", "VIVT3", "TIMS3", "TOTS3", "UGPA3", "USIM5", "VALE3", "VVAR3", "WEGE3",
              "YDUQ3"]
5. fundamentos = {}
6. arquivos = os.listdir('/content/drive/MyDrive/TCC 2021/Balanco')
7. for arquivo in arquivos:
8.     nome = arquivo[-9:-4]
9.     if "11" in nome:
10.         nome = arquivo[-10:-4]
11.         if nome in empresas:
12.             print(nome)
```

2.5. Explorando os dados

Para nos familiarizarmos mais com nossos dados, teremos que realizar algumas análises exploratórias simples de dados. Fazemos isso para entender melhor nossos dados e garantir que não haja problemas persistentes que possamos ter perdido quando transformamos os dados. Portanto, vamos começar a codificá-lo com algumas visualizações.

```
HAPV3
WARNING *** file size (24947) not 512 + multiple of sector size (512)
WARNING *** OLE2 inconsistency: SCS size is 0 but SSAT size is non-zero
WARNING *** file size (24947) not 512 + multiple of sector size (512)
WARNING *** OLE2 inconsistency: SCS size is 0 but SSAT size is non-zero
BDBC4
WARNING *** file size (68284) not 512 + multiple of sector size (512)
WARNING *** OLE2 inconsistency: SCS size is 0 but SSAT size is non-zero
WARNING *** file size (68284) not 512 + multiple of sector size (512)
WARNING *** OLE2 inconsistency: SCS size is 0 but SSAT size is non-zero
```

2.6. Importações e carregamento da biblioteca no DF que criamos anteriormente

Precisamos buscar o preços em nossa planilha de cotações, dessa forma baixamos um arquivo em excel e importamos com o panda, realizamos algumas modificações para que tenham as mesmas estrutura de acordo com as tabelas do fundamentos nomes das colunas com as informações nome da empresa data valores e tabela de cotação, criamos um for para percorrer o arquivo e realizar as modificações necessárias .

```
1. cotacoes_df = pd.read_excel('/content/drive/MyDrive/TCC
2021/Cotacoes.xlsx')
2. cotacoes = {}
3. for empresa in cotacoes_df["Empresa"].unique():
4.     cotacoes[empresa] =
cotacoes_df.loc[cotacoes_df['Empresa']==empresa, :]
'ABEV3':
   Date      High      Low  ...  Volume Adj Close Empresa
0  2012-12-20  17.376226  16.920275  ...  1243126.0  12.422499  ABEV3
1  2012-12-21  17.202246  16.808285  ...  1390141.0  12.472404  ABEV3
```

| | | | | | | | |
|------|------------|-----------|-----------|-----|------------|-----------|-------|
| 2 | 2012-12-26 | 17.090256 | 16.754292 | ... | 1076609.0 | 12.306555 | ABEV3 |
| 3 | 2012-12-27 | 16.870279 | 16.670300 | ... | 1307633.0 | 12.448023 | ABEV3 |
| 4 | 2012-12-28 | 16.916273 | 16.738293 | ... | 2807786.0 | 12.418349 | ABEV3 |
| ... | ... | ... | ... | ... | ... | ... | ... |
| 2073 | 2021-04-14 | 15.900000 | 15.620000 | ... | 21466700.0 | 15.810000 | ABEV3 |
| 2074 | 2021-04-15 | 16.030001 | 15.720000 | ... | 17705700.0 | 15.870000 | ABEV3 |
| 2075 | 2021-04-16 | 15.960000 | 15.720000 | ... | 13563400.0 | 15.800000 | ABEV3 |
| 2076 | 2021-04-19 | 16.070000 | 15.510000 | ... | 20032500.0 | 15.730000 | ABEV3 |
| 2077 | 2021-04-20 | 15.800000 | 15.530000 | ... | 12697600.0 | 15.620000 | ABEV3 |

Como podemos ver, há um problema de desequilíbrio de classe em nossos dados. Isso pode ser um problema, mas não queremos descartar pontos de dados para que cada classe seja igual à classe com a menor quantidade. Mesmo que essa seja uma solução viável para o desequilíbrio de classes, existem outras opções que podemos explorar. Outra opção seria usar uma métrica de avaliação diferente ao validar nosso modelo de classificação. Expandiremos essa opção quando chegarmos à fase de modelagem.

O arquivo Fundamentos é um dicionário de Data Frames que contém os DREs para cada balanço das empresas encontrado em <https://fundamentus.com.br>. Para continuar, é necessário substituir o arquivo Fundamentos que usamos por um DataFrame de DREs para cada balanços formatado como um dicionário de DFs.

Criamos um dicionário com todos os Data Frames das empresas, o mesmo é feito com as cotações, pegamos todas as listas que criamos e juntamos os balanço e DRE e armazenamos em nossa lista de fundamentos que recebe nome (das empresas), com a função `append`, assim criando uma chave para cada lista de fundamento

```
empresas = ["ABEV3", "AZUL4", "BTOW3", "B3SA3", "BBSE3", "BRML3", "BBDC4", "BRAP4",
"BBAS3", "BRKM5", "BRFS3", "BPAC11", "CRFB3", "CCRO3", "CMIG4", "HGTX3", "CIEL3",
"COGN3", "CPL6", "CSAN3", "CPFEB3", "CVCB3", "CYRE3", "ECOR3", "ELET6", "EMBR3",
"ENBR3", "ENGI11", "ENEV3", "EGIE3", "EQTL3", "EZTC3", "FLRY3", "GGBR4", "GOAU4",
"GOLL4", "NTCO3", "HAPV3", "HYPE3", "IGTA3", "GNDI3", "ITSA4", "ITUB4", "JBSS3",
"JHSF3", "KLB11", "RENT3", "LCAM3", "LAME4", "LREN3", "MGLU3", "MRFG3", "BEEF3",
"MRVE3", "MULT3", "PCAR3", "PETR4", "BRDT3", "PRIO3", "QUAL3", "RADL3", "RAIL3",
"SBSP3", "SANB11", "CSNA3", "SULA11", "SUZB3", "TAEE11", "VIVT3", "TIMS3", "TOTS3",
"UGPA3", "USIM5", "VALE3", "VVAR3", "WEGE3", "YDUQ3"]

fundamentos = {}
```

2.7.Criar Dicionários de Data Frames com cada empresa

Na primeira coluna colocamos o título com o nome da empresa , pegamos a 1ª linha e criamos um cabeçalho, tornar a 1ª coluna que agora tem o nome da empresa colocar o título com o nome da empresa,

```
[2078 rows x 8 columns],
'VIVT3':      Date      High      Low ...      Volume Adj Close Empresa
131322 2012-12-20 42.599998 41.400002 ...      34300.0 22.331593 VIVT3
131323 2012-12-21 42.939999 42.080002 ...      3400.0 22.509842 VIVT3
131324 2012-12-26 43.139999 42.680000 ...      28700.0 22.593714 VIVT3
131325 2012-12-27 43.500000 42.450001 ...      10900.0 22.787678 VIVT3
131326 2012-12-28 43.650002 42.549999 ...      31100.0 22.882025 VIVT3
...      ...      ...      ...      ...      ...      ...      ...
133395 2021-04-14 45.080002 44.060001 ...      2750500.0 43.798393 VIVT3
133396 2021-04-15 45.040001 44.369999 ...      2120800.0 43.808182 VIVT3
133397 2021-04-16 44.230000 43.330002 ...      2601600.0 44.139999 VIVT3
133398 2021-04-19 44.070000 43.549999 ...      1808800.0 43.669998 VIVT3
133399 2021-04-20 44.020000 43.439999 ...      2056200.0 43.849998 VIVT3

[2078 rows x 8 columns],
'VVAR3':      Date      High      Low ...      Volume Adj Close Empresa
141746 2012-12-20 4.5000 4.5000 ...      26800.0 3.835596 VVAR3
141747 2012-12-21 4.5025 4.5025 ...      14000.0 3.837726 VVAR3
141748 2012-12-26 4.5025 4.5025 ...      0.0 3.837726 VVAR3
141749 2012-12-27 4.5000 4.5000 ...      1600.0 3.835596 VVAR3
141750 2012-12-28 4.5000 4.5000 ...      6800.0 3.835596 VVAR3
...      ...      ...      ...      ...      ...      ...      ...
143819 2021-04-14 12.9300 12.5400 ...      40757900.0 12.780000 VVAR3
143820 2021-04-15 13.0300 12.7100 ...      37052900.0 12.830000 VVAR3
143821 2021-04-16 12.8900 12.5800 ...      35421000.0 12.750000 VVAR3
143822 2021-04-19 12.9500 12.6500 ...      42285500.0 12.800000 VVAR3
143823 2021-04-20 12.8300 12.5000 ...      31459600.0 12.670000 VVAR3
```

2.8.Iterando através de cada balanços para transformar os dados

Todos os Data Frames do dicionário de DFs são combinados como um para treinar nosso modelo de classificação mais tarde.

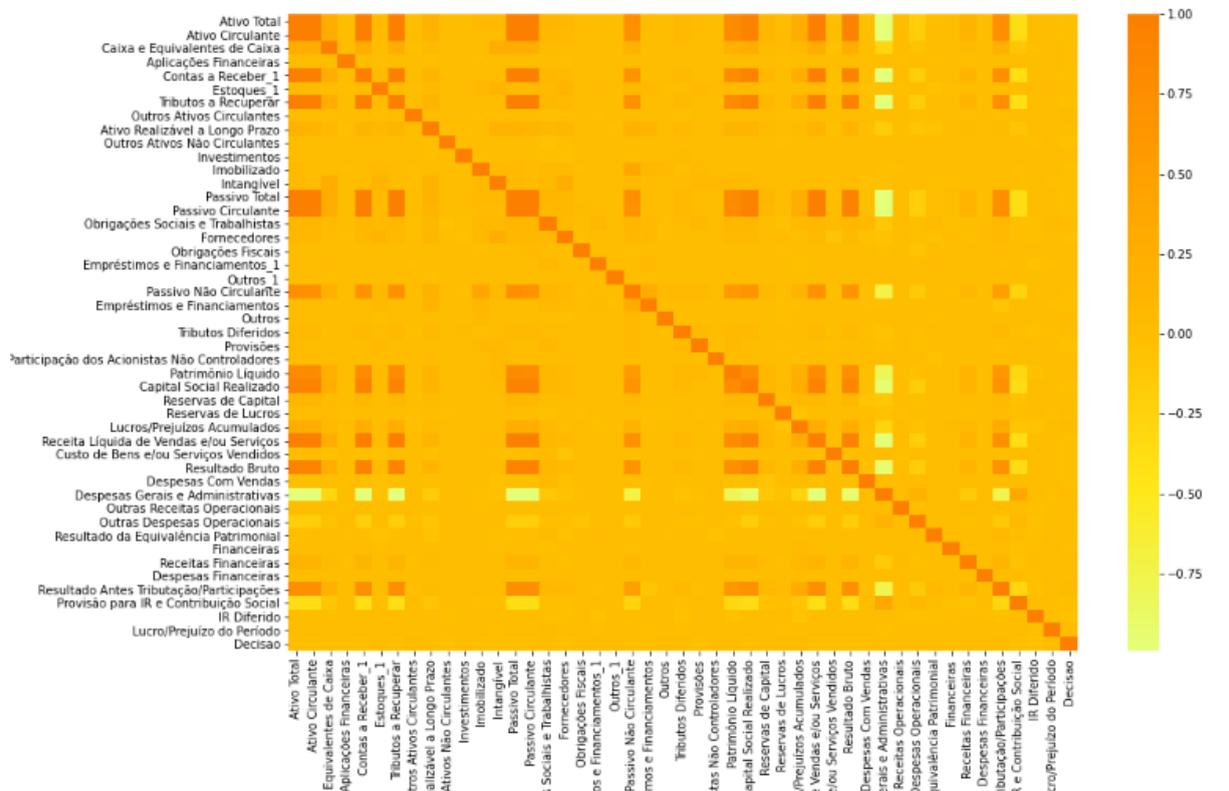
O índice é redefinido porque as datas não são mais necessárias. Cada linha ou DRE agora contém informações dos DREs anteriores e futuros, portanto as datas não são mais importantes para o modelo.

Recursos ou colunas relacionados a preços são eliminados para evitar vazamento de dados. Em DREs normais, esses recursos não estão incluídos.

Finalmente, temos nossos dados fundamentais corretamente formatados e transformados da maneira que desejamos. Exportamos o DF final para exploração de dados e para treinar o modelo de classificação.

2.9. Correlações em nossos dados

1. `correlacoes = base_dados.corr()`
- 2.
3. `fig, ax = plt.subplots(figsize=(15, 10))`
4. `sns.heatmap(correlacoes, cmap="Wistia", ax=ax)`
5. `plt.show()`
6. `display(correlacoes)`



Pelo que podemos ver, alguns recursos são influentes na determinação do rótulo da classe. Alguns têm pouca ou nenhuma correlação com a decisão de se vale ou não a pena comprar, vender ou manter o estoque / relatório trimestral. Agora que sabemos que alguns recursos do DRE não são importantes para determinar o rótulo da classe, podemos remover esses recursos do conjunto de dados.

Engenharia de recursos e seleção

Agora que exploramos um pouco nossos dados, poderíamos continuar fazendo mais explorações ou passar para o processo de Engenharia de Recursos ou, mais especificamente, Seleção de Recursos . Engenharia de recursos é o processo de alteração de nosso conjunto de dados para aprimorar nossos modelos de aprendizado de máquina. Existem várias opções para a engenharia dos recursos de nosso conjunto de dados. Eles incluem, mas não estão limitados a:

Criação de recursos interativos. Faça com que dois recursos distintos interajam entre si para criar um novo recurso.

Reduzindo o número de recursos. Como temos mais de 30 recursos fundamentais, poderíamos eliminar recursos sem importância para melhorar o desempenho de nossos modelos.

Usando o conhecimento do domínio. Se soubéssemos quais características são mais essenciais para o preço de uma ação com base em pesquisas econômicas e financeiras, poderíamos atribuir pesos a essas características.

2.10. Seleção de recursos

Neste projeto, vamos eliminar alguns recursos menos importantes. Fazendo isso, podemos melhorar potencialmente a precisão do nosso modelo e também reduzir o tempo de treinamento. Essa ação é conhecida como Seleção de recursos; o processo de seleção de recursos que consideramos necessários para o nosso modelo. Mostraremos 2 maneiras diferentes de seleção de recursos. Desta forma, podemos comparar e contrastar os

desempenhos do nosso modelo com conjuntos de dados ligeiramente diferentes devido aos nossos métodos de seleção.

Método #1: Selecionando os top 10 principais recursos com base em valores de correlação

| | Fornecedores | Outros Ativos Circulantes | Lucros/Prejuízos Acumulados | Resultado Antes Tributação/Participações | Ativo Total | Obrigações Sociais e Trabalhistas | Obrigações Fiscais | Custo de Bens e/ou Serviços Vendidos | Equivalência Patrimonial | Resultado da Equivalência Patrimonial | Tributos Diferidos | Decisao |
|------|--------------|---------------------------|-----------------------------|--|-------------|-----------------------------------|--------------------|--------------------------------------|--------------------------|---------------------------------------|--------------------|---------|
| 0 | -0.128485 | -0.034116 | 0.191517 | -0.038799 | -0.057810 | -4.396845 | -0.054656 | 0.033336 | -0.004714 | -0.090125 | 2 | |
| 1 | 2.127262 | -0.027903 | 11.816382 | 33.145593 | 43.006573 | 4.066346 | 0.063274 | -0.123545 | 0.116959 | 2.210854 | 0 | |
| 2 | 0.156604 | -0.032969 | 0.090433 | 0.004217 | -0.038324 | 0.630917 | -0.056533 | 0.020923 | 4.755571 | -0.028989 | 2 | |
| 3 | 0.847219 | -0.032883 | -0.460858 | 0.180279 | 0.099339 | 1.239917 | -0.008098 | -0.138281 | 0.105892 | 0.164927 | 2 | |
| 4 | -0.440745 | -0.032706 | 0.191517 | -0.123777 | -0.132114 | -1.283224 | -0.059637 | 0.110711 | 0.002491 | -0.209187 | 0 | |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | |
| 1877 | 0.216744 | -0.034214 | -0.020459 | -0.072927 | -0.021564 | 0.169052 | -0.047162 | 0.044559 | -0.004714 | -0.217314 | 2 | |
| 1878 | -0.353377 | -0.032037 | -0.460858 | -0.121127 | -0.093283 | -1.717283 | -0.047889 | 0.023173 | -0.004714 | -0.085117 | 0 | |
| 1879 | 0.644715 | -0.032350 | 0.191517 | 0.219528 | 0.333928 | 1.088392 | 0.022044 | 0.051061 | -0.004714 | -0.228295 | 2 | |
| 1880 | 0.625836 | -0.032183 | -0.289212 | -0.176608 | 0.165020 | 4.644577 | 0.016299 | -0.009419 | -0.004714 | 0.550222 | 0 | |
| 1881 | -0.194607 | -0.033410 | 0.280608 | 0.099011 | -0.040758 | -0.843650 | -0.058216 | 0.036808 | -0.004714 | -0.095431 | 0 | |

1882 rows x 11 columns

Nossos Top 10 principais recursos correlacionados (ordenados do mais correlacionado para o menos)

Usando a matriz de correlação que criamos enquanto explorávamos o conjunto de dados, selecionamos os top 10 principais recursos correlacionados com os rótulos de classe de decisão (veja a imagem acima). Isso é feito:

- Obtendo o valor absoluto de cada recurso (para lidar com correlações negativas).
- Classificando pelos valores absolutos.
- Cortando os recursos numéricos para incluir apenas os Top 10 principais mais correlacionados.
- Unindo os rótulos da classe Decisão para que sejam incluídos no novo DF.
- Exportando os Top 10 principais DF de correlação como um arquivo Fundamentos.

Com isso, temos um novo conjunto de dados reduzido, mas concentrado para nossos modelos de classificação. Este novo conjunto de dados aprimora nossos modelos de classificação e acelera o tempo de treinamento.

Método 2: Seleção dos Top 10 principais recursos com base em um classificador de árvore

O método de seleção de recursos é um pouco mais complicado. Usando a biblioteca do Sci-Kit Learn, estaremos implementando um classificador baseado em Árvore de Decisão para determinar quais recursos são mais importantes. Usando este classificador, seremos capazes de determinar quais recursos são mais importantes ao classificar o DRE ou estoque.

```
1. from sklearn.ensemble import ExtraTreesClassifier
2. modelo = ExtraTreesClassifier(random_state=1)
3. x = base_dados.drop("Decisao", axis=1)
4. y = base_dados["Decisao"]
5. modelo.fit(x, y)
6. caracteristicas_importantes =
   pd.DataFrame(modelo.feature_importances_,
   x.columns).sort_values(by=0, ascending=False)
7. display(caracteristicas_importantes)
8. top10 = list(caracteristicas_importantes.index)[:10]
9. print(top10)
```

A seguir, com nossos recursos importantes todos classificados, vamos visualizá-los para ver exatamente quais recursos foram determinados como os mais importantes. Para conferir quais recursos foram selecionados.

De acordo com nosso `forest_feature_importances_method`, estes são os top 10 recursos mais importantes:

- Fornecedores 0.032418
- Outros Ativos Circulantes 0.031265
- Lucros/Prejuízos Acumulados 0.031182
- Resultado Antes Tributação/Participações 0.030940
- Ativo Total 0.030832
- Obrigações Sociais e Trabalhistas 0.030454
- Obrigações Fiscais 0.030394
- Custo de Bens e/ou Serviços Vendidos 0.030036
- Resultado da Equivalência Patrimonial 0.029478
- Tributos Diferidos 0.028723
- Outros Ativos Não Circulantes 0.028596
- Caixa e Equivalentes de Caixa 0.028161
- Receitas Financeiras 0.028154
- Ativo Realizável a Longo Prazo 0.028082
- Outros 0.027782
- Despesas Financeiras 0.027619
- Provisões 0.027550
- Empréstimos e Financiamentos_1 0.027481
- Imobilizado 0.027472
- Estoques_1 0.027316
- Lucro/Prejuízo do Período 0.027289
- Outros_1 0.027205
- Investimentos 0.026858
- Participação dos Acionistas Não Controladores 0.026848
- Despesas Com Vendas 0.026749
- Outras Despesas Operacionais 0.026696
- Aplicações Financeiras 0.026536
- Empréstimos e Financiamentos 0.026502
- Provisão para IR e Contribuição Social 0.025756
- Passivo Não Circulante 0.025710
- IR Diferido 0.025659
- Intangível 0.025375
- Financeiras 0.025102
- Outras Receitas Operacionais 0.024958
- Reservas de Lucros 0.024749
- Reservas de Capital 0.024076

A última linha em nosso código exportou o DF dos top10 recursos mais importantes. Agora temos dois Top 10 DFs com alguns recursos diferentes e alguns mesmos representados em ambos. Podemos usar esses dois DFs para descobrir qual tem melhor desempenho com nossos classificadores de aprendizado de máquina

2.11. Classificadores de aprendizado de máquina

Com o objetivo de encontrar o modelo de classificação de Aprendizado de Máquina de melhor desempenho, executaremos nossos dados por meio de vários modelos. Cada modelo tem suas próprias vantagens e desvantagens. Para garantir que estamos usando o modelo ideal para nossos dados, temos que experimentar cada classificador. Não se preocupe, uma vez que escrevermos o código para um modelo, podemos simplesmente copiar o mesmo código para cada novo modelo com apenas algumas modificações de uma ou duas linhas.

Os modelos de classificação que usaremos:

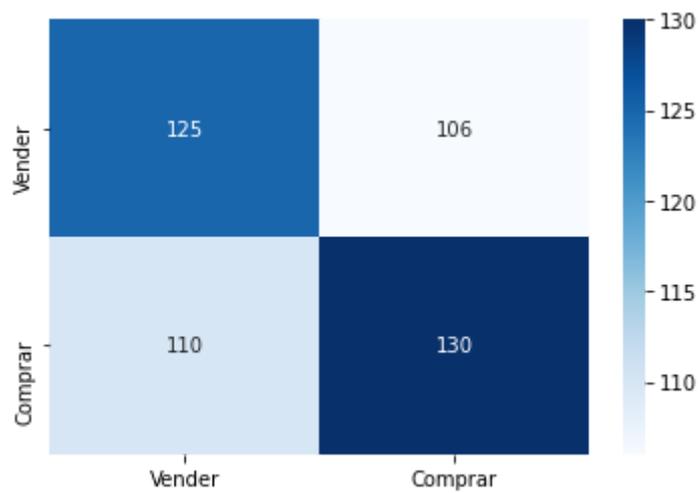
- AdaBoost
- Árvore de Decisão
- Dummy (modelo de linha de base)
- Gradient Boost
- K Vizinhos mais próximos (KNN)
- Regressão Logística
- Baías ingênuas
- Floresta Aleatória
- Máquina de vetor de suporte (SVM)
- Aumento de gradiente extremo (XGBoost)

Com nossos dados limpos, formatados e selecionados, prosseguiremos para a classificação de estoques com base em seus relatórios trimestrais.

Executando o código nos dará:

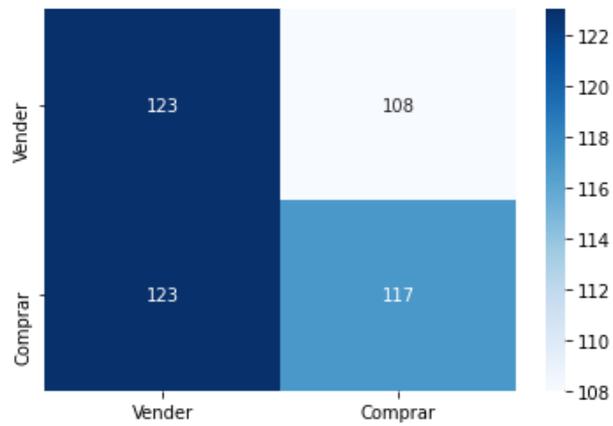
```
AdaBoost
  precision  recall  f1-score  support
    0        0.53    0.54     0.54     231
    2        0.55    0.54     0.55     240

accuracy                    0.54     471
macro avg                   0.54     471
weighted avg                0.54     471
```



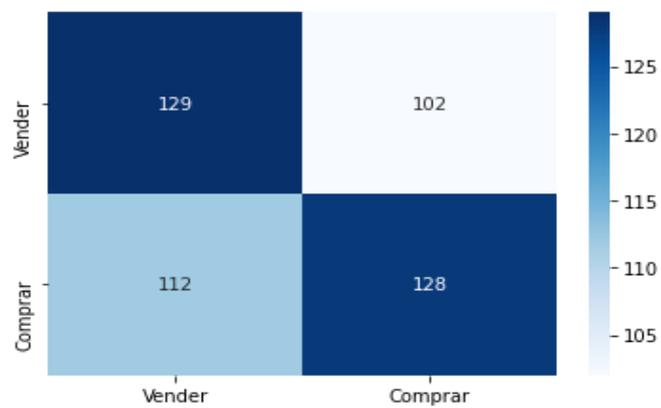
DecisionTree

| | <i>precision</i> | <i>recall</i> | <i>f1-score</i> | <i>support</i> |
|---------------------|------------------|---------------|-----------------|----------------|
| 0 | 0.50 | 0.53 | 0.52 | 231 |
| 2 | 0.52 | 0.49 | 0.50 | 240 |
| <i>accuracy</i> | | | 0.51 | 471 |
| <i>macro avg</i> | 0.51 | 0.51 | 0.51 | 471 |
| <i>weighted avg</i> | 0.51 | 0.51 | 0.51 | 471 |

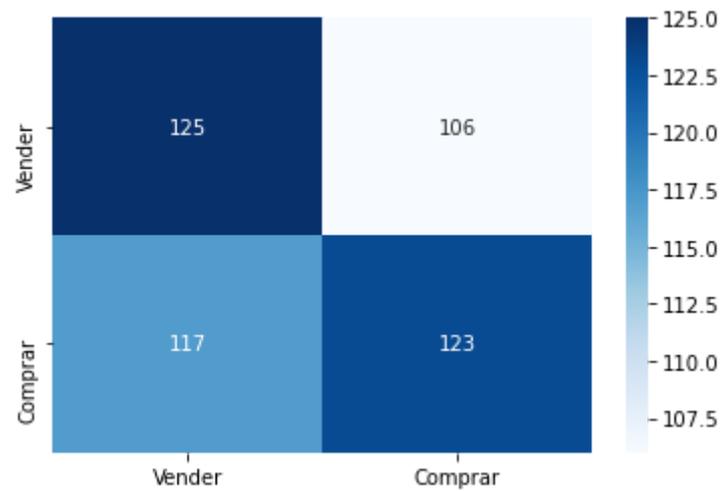


RandomForest

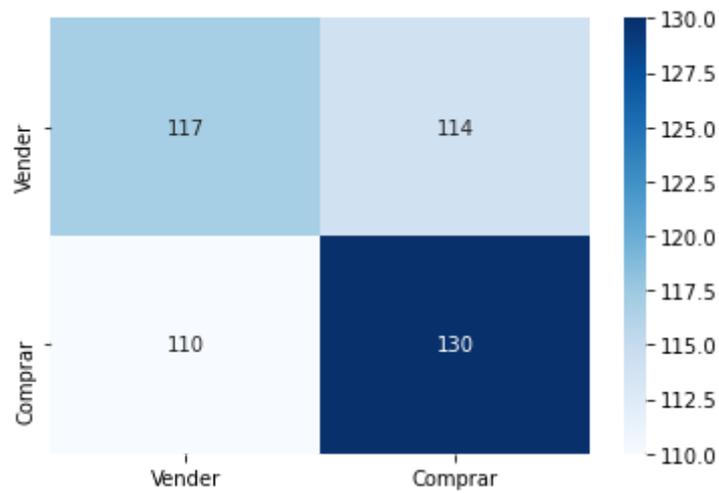
| | <i>precision</i> | <i>recall</i> | <i>f1-score</i> | <i>support</i> |
|---------------------|------------------|---------------|-----------------|----------------|
| 0 | 0.54 | 0.56 | 0.55 | 231 |
| 2 | 0.56 | 0.53 | 0.54 | 240 |
| <i>accuracy</i> | | | 0.55 | 471 |
| <i>macro avg</i> | 0.55 | 0.55 | 0.55 | 471 |
| <i>weighted avg</i> | 0.55 | 0.55 | 0.55 | 471 |



| <i>ExtraTree</i> | | | | |
|---------------------|------------------|---------------|-----------------|----------------|
| | <i>precision</i> | <i>recall</i> | <i>f1-score</i> | <i>support</i> |
| 0 | 0.52 | 0.54 | 0.53 | 231 |
| 2 | 0.54 | 0.51 | 0.52 | 240 |
| <i>accuracy</i> | | | 0.53 | 471 |
| <i>macro avg</i> | 0.53 | 0.53 | 0.53 | 471 |
| <i>weighted avg</i> | 0.53 | 0.53 | 0.53 | 471 |

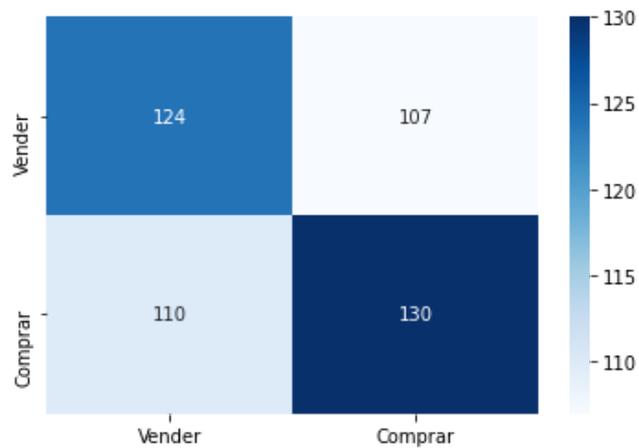


| <i>GradientBoost</i> | | | | |
|----------------------|------------------|---------------|-----------------|----------------|
| | <i>precision</i> | <i>recall</i> | <i>f1-score</i> | <i>support</i> |
| 0 | 0.52 | 0.51 | 0.51 | 231 |
| 2 | 0.53 | 0.54 | 0.54 | 240 |
| <i>accuracy</i> | | | 0.52 | 471 |
| <i>macro avg</i> | 0.52 | 0.52 | 0.52 | 471 |
| <i>weighted avg</i> | 0.52 | 0.52 | 0.52 | 471 |



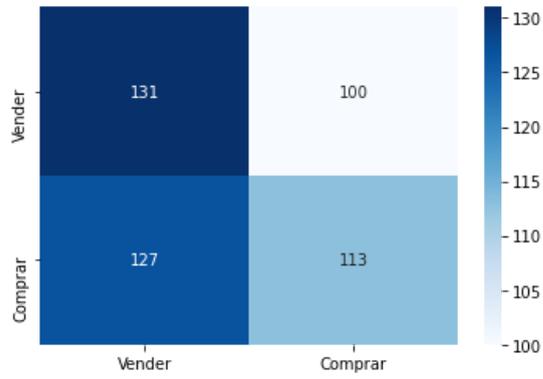
KNN

| | <i>precision</i> | <i>recall</i> | <i>f1-score</i> | <i>support</i> |
|---------------------|------------------|---------------|-----------------|----------------|
| 0 | 0.53 | 0.54 | 0.53 | 231 |
| 2 | 0.55 | 0.54 | 0.55 | 240 |
| <i>accuracy</i> | | | 0.54 | 471 |
| <i>macro avg</i> | 0.54 | 0.54 | 0.54 | 471 |
| <i>weighted avg</i> | 0.54 | 0.54 | 0.54 | 471 |



LogisticRegression

| | <i>precision</i> | <i>recall</i> | <i>f1-score</i> | <i>support</i> |
|---------------------|------------------|---------------|-----------------|----------------|
| 0 | 0.51 | 0.57 | 0.54 | 231 |
| 2 | 0.53 | 0.47 | 0.50 | 240 |
| <i>accuracy</i> | | | 0.52 | 471 |
| <i>macro avg</i> | 0.52 | 0.52 | 0.52 | 471 |
| <i>weighted avg</i> | 0.52 | 0.52 | 0.52 | 471 |



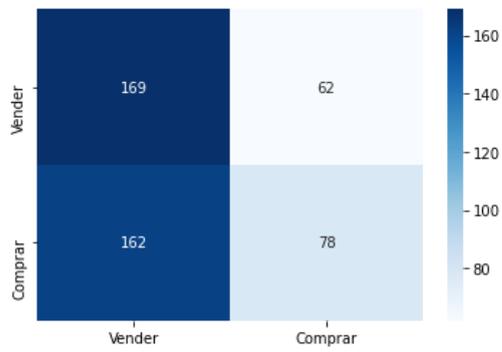
NaiveBayes

| | <i>precision</i> | <i>recall</i> | <i>f1-score</i> | <i>support</i> |
|---------------------|------------------|---------------|-----------------|----------------|
| 0 | 0.48 | 0.89 | 0.63 | 231 |
| 2 | 0.46 | 0.09 | 0.15 | 240 |
| <i>accuracy</i> | | | 0.48 | 471 |
| <i>macro avg</i> | 0.47 | 0.49 | 0.39 | 471 |
| <i>weighted avg</i> | 0.47 | 0.48 | 0.39 | 471 |



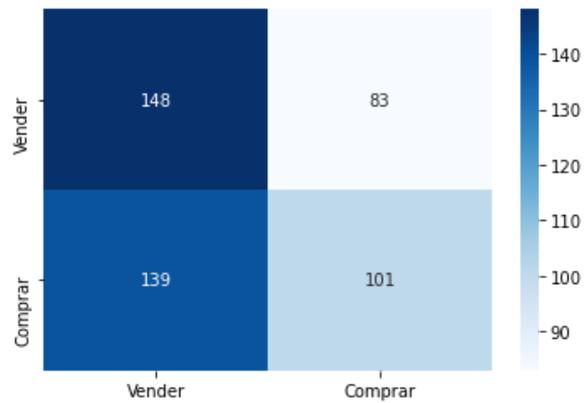
SVM

| | <i>precision</i> | <i>recall</i> | <i>f1-score</i> | <i>support</i> |
|---------------------|-------------------------|---------------|-----------------|----------------|
| 0 | 0.51 | 0.73 | 0.60 | 231 |
| 2 | 0.56 | 0.33 | 0.41 | 240 |
| <i>accuracy</i> | | | 0.52 | 471 |
| <i>macro avg</i> | 0.53 | 0.53 | 0.51 | 471 |
| <i>weighted avg</i> | 0.53 | 0.52 | 0.50 | 471 |



RedeNeural

| | <i>precision</i> | <i>recall</i> | <i>f1-score</i> | <i>support</i> |
|---------------------|------------------|---------------|-----------------|----------------|
| <i>0</i> | <i>0.52</i> | <i>0.64</i> | <i>0.57</i> | <i>231</i> |
| <i>2</i> | <i>0.55</i> | <i>0.42</i> | <i>0.48</i> | <i>240</i> |
| <i>accuracy</i> | | | <i>0.53</i> | <i>471</i> |
| <i>macro avg</i> | <i>0.53</i> | <i>0.53</i> | <i>0.52</i> | <i>471</i> |
| <i>weighted avg</i> | <i>0.53</i> | <i>0.53</i> | <i>0.52</i> | <i>471</i> |



2.12. Metodologia

Uma grande quantidade de relatórios trimestrais precisará ser coletada para treinar nosso modelo de classificador. Depois de baixar os dados fundamentais necessários (relatórios trimestrais), podemos ver como os dados são formatados abaixo, usando um **DataFrame Pandas**:

| | Quarter end | Shares | Shares split adjusted | Split factor | Assets | Current Assets | Liabilities | Current Liabilities | Shareholders equity | Non-controlling interest | Preferred equity | Goodwill & intangibles | Long-term debt |
|---|-------------|-----------|-----------------------|--------------|------------|----------------|-------------|---------------------|---------------------|--------------------------|------------------|------------------------|----------------|
| 0 | 2019-04-30 | 315993352 | 315993352 | 1 | 9022000000 | 3812000000 | 3897000000 | 1118000000 | 5125000000 | 0 | 0 | 3650000000 | 1798000000 |
| 1 | 2019-01-31 | 317515869 | 317515869 | 1 | 8952000000 | 3712000000 | 3916000000 | 1095000000 | 5036000000 | 0 | 0 | 3699000000 | 1798000000 |
| 2 | 2018-10-31 | 318533054 | 318533054 | 1 | 8541000000 | 3848000000 | 3970000000 | 1171000000 | 4567000000 | 4000000 | 0 | 3464000000 | 1799000000 |
| 3 | 2018-07-31 | 318769547 | 318769547 | 1 | 8349000000 | 3667000000 | 3781000000 | 1014000000 | 4564000000 | 4000000 | 0 | 3448000000 | 1799000000 |

Aqui podemos observar todas as diferentes colunas e datas correspondentes a cada coluna. Agora, vamos dar uma olhada nas colunas Preço, Preço alto e Preço baixo. Essas informações geralmente não são fornecidas em relatórios trimestrais, mas essa base tem, o que é importante para determinar se o estoque é uma Compra, Não Comprar ou Venda.

Existem várias maneiras exclusivas de determinar se vale a pena investir em uma ação ou não. Poderíamos potencialmente classificá-lo como Compra se os Ativos aumentassem e os Passivos diminuíssem ao longo de três trimestres? Ou talvez se as ações aumentassem e a dívida de longo prazo diminuísse? Podemos constatar que existem inúmeras opções para determinar a classe das ações.

Ao criar um modelo de classificação de aprendizado de máquina que utilize a análise fundamentalista de ações por meio da análise de dados do Relatório Trimestral, determinando assim se vale a pena investir em uma ação naquele momento. Para isso, as informações necessárias passaram por um processo de engenharia de dados envolvendo coleta de dados, limpeza, formatação, visualização e seleção de recursos. Todo esse procedimento preparou nossos dados para a próxima etapa, a modelagem de dados.

A modelagem de aprendizado de máquina envolve a obtenção de dados para treinar modelos a fim de fazer previsões com base nesses dados. No entanto, esses modelos não

podem aceitar quaisquer dados que fornecemos - eles precisam atender a um padrão específico. Esse padrão foi definido subjetivamente assim que decidimos que tipo de problema estávamos resolvendo. Cada problema requer seu próprio conjunto de dados exclusivos e, na maioria das vezes, esses dados passam por todo o processo de engenharia de dados.

2.13. Nossos Dados

Como estávamos examinando explicitamente os relatórios trimestrais, decidimos avaliar o valor do investimento de uma ação observando o desempenho do DRE em relação ao DRE anterior e armazená-lo como uma variação percentual.

Os rótulos de classe de Comprar, Não Comprar, Vender foram determinados pela observação do próximo DRE Preço alto e Preço baixo e se eles aumentaram ou diminuíram por uma quantidade significativa.

Os dois conjuntos de dados que temos: valores de mudança percentual e rótulos de classe, constituirão os valores de X e Y respectivamente. Os valores de alteração percentual (valor X) são conhecidos como independentes. Os rótulos de classe (valor Y) são conhecidos como variável dependente. Essas variáveis serão escaladas para melhorar o desempenho de nossos modelos. Em seguida, os dados escalados serão divididos em conjuntos de treinamento e teste para avaliar nossos modelos. Assim que os dados estiverem prontos, poderemos finalmente ajustar o modelo aos nossos dados e fazer algumas previsões / classificações.

Criamos dois conjuntos diferentes de dados com dez conjuntos de recursos principais ligeiramente diferentes em cada um (os recursos relacionados ou as importâncias dos recursos do classificador de árvore). Isso nos deu a oportunidade de comparar e contrastar os desempenhos do modelo com base nos dados fornecidos para o modelo. Um conjunto de dados pode ter um desempenho significativamente melhor do que o outro. Para fazer essa comparação, temos que alterar os dados carregados no modelo, o que requer a alteração de apenas uma única linha de código.

2.14. Classificador fictício

O classificador fictício funciona de maneira semelhante a adivinhar aleatoriamente um rótulo de classe para cada DRE. Geralmente é baseado na probabilidade - os rótulos das classes seriam determinados com base no tamanho da amostra.

Com o classificador fictício estabelecido como nosso modelo de linha de base, um padrão de desempenho foi definido para cada modelo. O modelo deve ser capaz de superar o classificador dummy para ser considerado um modelo viável. Se ele não for capaz de superar o modelo de linha de base, o modelo não é melhor do que adivinhação aleatória.

2.15. Codificando o Modelo de Classificação

Agora que temos nossos dados prontos e estabelecemos o modelo que usaremos, podemos codificar nosso classificador. Este será nosso modelo de linha de base (classificador fictício).

Aqui, carregamos nas bibliotecas Python, precisamos desenvolver um modelo de classificação fictício. A maioria dessas bibliotecas permanecerá a mesma - basta substituir a linha de importação Dummy Classifier. Quando se trata de importar nossos dados, temos a opção de escolher qual conjunto de dados queremos.

Escalonar e treinar, testar, dividir os dados

```
1. from sklearn.model_selection import train_test_split
2.
3. x = nova_base_dados.drop("Decisao", axis=1)
4. y = nova_base_dados["Decisao"]
5.
6. x_treino, x_teste, y_treino, y_teste = train_test_split(x, y,
    random_state=1)
```

Escalamos os dados aqui para melhorar o desempenho geral da maioria dos nossos modelos. O método de dimensionamento a ser usado é com você. Cada scaler tem suas próprias vantagens .

Scalers são transformadores lineares (ou mais precisamente afins) e diferem uns dos outros na maneira como estimam os parâmetros usados para mudar e dimensionar cada recurso. ([sckit learn](#))

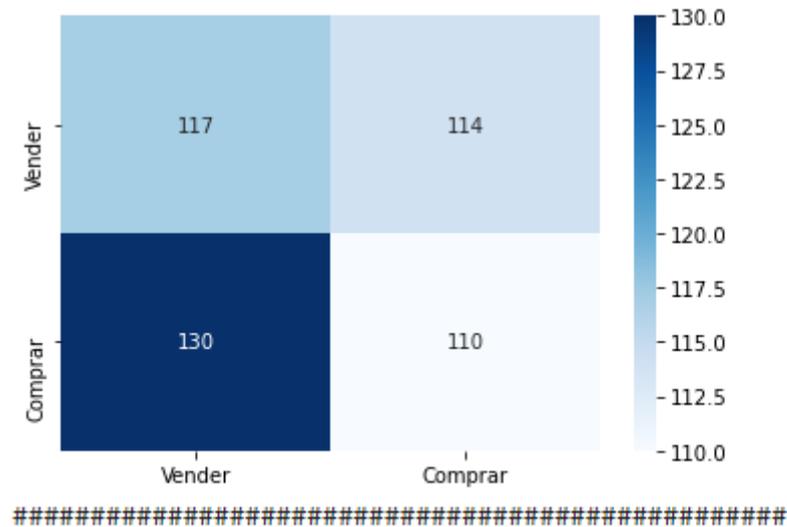
Os dados são então divididos em conjuntos de treinamento e teste para as variáveis X e Y , que precisaremos ajustar e avaliar nossos modelos.

2.16.Criando o classificador fictício

Criamos o classificador Dummy instanciando o classificador e, em seguida, ajustando-o aos nossos conjuntos de treinamento. Em seguida, usamos o conjunto de testes para fazer rótulos previstos para comparar com os rótulos reais. Finalmente, imprimimos um relatório de classificação que contém as métricas de avaliação de que precisamos:

```
1. def avaliar(y_teste, previsoes, nome_modelo):
2.     print(nome_modelo)
3.     report = classification_report(y_teste, previsoes)
4.     print(report)
5.     cf_matrix = pd.DataFrame(confusion_matrix(y_teste, previsoes),
6.                             index=["Vender", "Comprar"], columns=["Vender", "Comprar"])
7.     sns.heatmap(cf_matrix, annot=True, cmap="Blues", fmt=',')
8.     plt.show()
9.     print("#" * 50)
10.
11. avaliar(y_teste, previsao_dummy, "Dummy")
```

| Dummy | precision | recall | f1-score | support |
|--------------|-----------|--------|----------|---------|
| 0 | 0.47 | 0.51 | 0.49 | 231 |
| 2 | 0.49 | 0.46 | 0.47 | 240 |
| accuracy | | | 0.48 | 471 |
| macro avg | 0.48 | 0.48 | 0.48 | 471 |
| weighted avg | 0.48 | 0.48 | 0.48 | 471 |



Métricas de Avaliação

Como estamos lidando com um conjunto de dados desequilibrado, não podemos confiar na precisão como uma métrica de avaliação com nossos modelos. É por isso que não está incluído no relatório de classificação. A precisão pode ser uma medida importante, mas devido à natureza do nosso problema e ao desequilíbrio de classes, ela será ignorada em favor da precisão e da recuperação.

Estamos usando Precision porque lida com o número de falsos positivos. Quando se trata de decisões de investimento, não queremos que o classificador classifique incorretamente a Vender como a Comprar. Em nossa estratégia de investimento conservadora, preferimos perder uma oportunidade de investir (Recall) do que investir na ação errada (Precision). Obviamente, há um equilíbrio em ter cada um, e é aí que o F1-Scorejogo entra em ação. No entanto, o julgamento geral para cada modelo de

classificação será baseado no aumento de sua Precisão Pontuação em relação ao modelo de linha de base.

2.17. Matriz de confusão

Pode ser melhor visualizar os resultados do nosso modelo. Para cada classificador, estaremos visualizando sua matriz de confusão para melhor interpretabilidade. A matriz de confusão nos mostrará a quantidade de DREs que foram classificados incorretamente.

```
1. from sklearn.dummy import DummyClassifier
2. from sklearn.metrics import classification_report,
   confusion_matrix
3.
4. dummy = DummyClassifier(strategy="stratified", random_state=2)
5. dummy.fit(x_treino, y_treino)
6. previsao_dummy = dummy.predict(x_teste)
```

2.18. Conclusão

Decidir o valor de uma ação com base em relatórios trimestrais não é uma conquista nova. Para a maioria dos Analistas Fundamentais, a estratégia que utilizamos pode ser considerada muito simples ou desatualizada. Mas, para fins de aprendizagem de classificação de aprendizado de máquina, foi o suficiente. Poderíamos potencialmente adicionar mais recursos para melhorar o modelo ou alterar os recursos que temos com base em novas estratégias.

Existem também técnicas para agilizar o processo, como a criação de um Pipeline. Ou podemos até mesmo experimentar redes neurais para ver se elas têm um desempenho melhor do que nossos modelos atuais.

3. RESULTADOS

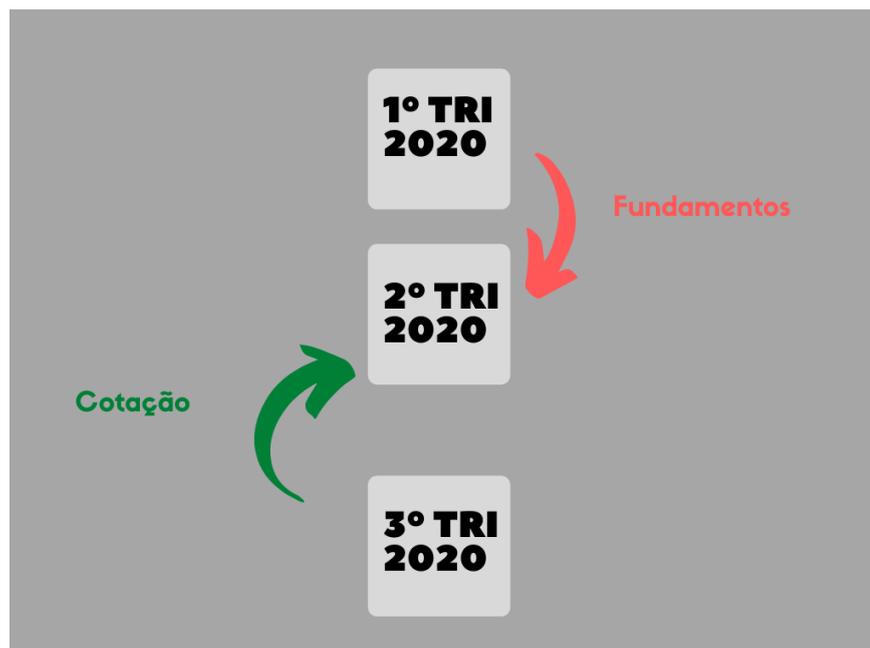
3.1. Solução inicial

Nesta parte do projeto decidimos apenas escolher a base de dados e como faríamos para implementar. Então pra isso escolhemos algumas empresas, depois pegamos os fundamentos das ações dessas empresas (balanço patrimonial e demonstrativo de resultados). Tivemos a ideia de criar um “gabarito” também, para a I.A para ela saber onde deveria investir em ações de determinadas empresas. Por último resolvemos pegar as cotações e definir uma forma de análise.

Basicamente neste projeto estamos informando para a I.A com o histórico de informações de cada trimestre onde ela deveria ou não investir. Vamos analisar os fundamentos com base no trimestre anterior e a cotação pelo trimestre seguinte fazendo assim um balanceamento de informações para melhorar a nossa Inteligência Artificial na tomada de decisões.

3.2. Solução Final

Na solução final resolvemos implementar tudo que havíamos discutido na ideia inicial.



Fizemos um tratamento de dados criando de fato o “gabarito” e limpando a base de dados, removendo valores vazios, indicadores que não são comuns, indicadores com o mesmo nome e juntando várias tabelas em uma tabela só.

Depois partimos para a parte de análise exploratória, que é uma parte bastante importante em qualquer projeto de Análise de Dados. Basicamente pegamos todas as informações disponíveis e analisamos se precisaria fazer mais um tratamento, para entender qual a melhor forma para construir.

A próxima etapa foi a parte de Feature Selection, onde fizemos uma outra análise na base de dados e decidimos com isso pegar menos informações para que ela fique melhor no seu desempenho e precisão, pois em alguns casos muitas informações em uma I.A ela pode acabar ficando confusa e não acabar chegando no resultado já que ela está analisando muita coisa ao mesmo tempo, então de 47 fundamentos resolvemos cortar para 10 fundamentos para utilizar na Inteligência Artificial.

```
[ ] # vamos treinar uma arvore de decisao e pegar as características mais importantes dela
    from sklearn.ensemble import ExtraTreesClassifier

    modelo = ExtraTreesClassifier(random_state=1)
    x = base_dados.drop("Decisao", axis=1)
    y = base_dados["Decisao"]
    modelo.fit(x, y)

    caracteristicas_importantes = pd.DataFrame(modelo.feature_importances_, x.columns).sort_values(by=0, ascending=False)
    display(caracteristicas_importantes)
    top10 = list(caracteristicas_importantes.index[:10])
    print(top10)
```

Criamos um comparativo e avaliação, treinamos os modelos e analisamos qual seria o melhor e mais apropriado para esse tipo de situação.

Agora precisamos saber se deu certo, fizemos um cálculo em uma tabela das onde podemos identificar o rendimento, calculando as cotações iniciais sobre as cotações finais

Meu saldo inicial 30000 Final 31374 Rendimento 4.6 % / índice bovespa 0.97%

4. CONSIDERAÇÕES FINAIS

Ao falarmos de inteligência Artificial (IA) podemos citar diferentes áreas, utilizando o aprendizado de máquina, mais conhecido como Machine Learning (MA), que tem como característica fazer com que as máquinas interpretem a partir dos dados sem programação, e o processamento de linguagem natural (PLN), que é uma subárea da IA ajudando computadores a entender, interpretar e manipular a linguagem humana. Neste contexto o Machine Learning tem como objetivo automatizar o conhecimento das técnicas computacionais.

Com a utilização da Inteligência artificial, tentamos demonstrar um meio de tomada de decisão para a compra de carteira de ações, realizando uma análise fundamentalista, extraíndo informações sobre o faturamento, lucro, quanto possui de ativos e passivos. Afim de passar estas informações para a IA e ela nos dar o retorno de quando devemos realizar a venda ou compra de uma carteira de ação, para que no final nos mostre o resultado de quais decisões devem ser tomadas para obter resultados. Lembrando que esta Inteligência Artificial será apenas para aplicar a ciência de dados e não para uso em seus investimentos.

A utilização da Inteligência Artificial foi com o objetivo de realizar uma análise do comportamento e movimento das cotações dos ativos listados nos fundamentos das empresas, para auxiliar nas compras de ações. Esta IA foi criada para ajudar pessoas que ainda não possuem um certo conhecimento sobre o mercado de ações, executando um trabalho de análise de dados a cada trimestre ela irá auxiliar na tomada de decisão.

5. REFERÊNCIAS

- [1] NORVING, S. R. P. Artificial Intelligence A Modern Approach. 3. ed. [S.l.]: Pearson, 1995.
- [2] SIMAS, E. Introdução às Redes Neurais Artificiais. <<http://www.dee.eng.ufba.br/home/simas/introdRNA.pdf>>.
- [3] CORREA, T. D. P. B. D. A evolução das técnicas de inteligência artificial. SBGAMES, 2012.
- [4] MYERS, G. J.; SANDLER, C.; BADGETT, T. The art of software testing. [S.l.]: John Wiley & Sons, 2011.
- [5] RICH, E. Artificial Intelligence. 2. ed. [S.l.]: McGraw-Hill Publishing Co., 1991.
- [6] NIGRIN, A. Neural networks for pattern recognition. American Psychological Association (APA), 1993.