Trabalho prático

Diogo Pires, a93239 Gonçalo Soares, a93286
Marco Costa, a93283 Rita Teixeira, a89494

Grupo 16

22 de fevereiro de 2023

Aprendizagem e Decisão Inteligentes

Licenciatura em Engenharia Informática

Conteúdo

1 Introdução e metodologia utilizada **2**

2 Wine Quality **3**
2.1 Domínio a tratar e objetivo . . . 3
2.2 Descrição e exploração do dataset e tratamento de dados . . . 3
2.3 Modelos desenvolvidos . . . 5
2.4 Resultados obtidos . . . 6
2.4.1 Informações relativas à partição do dataset sobre *wine quality*: . . . 6
2.4.2 Modelos e respetivos resultados . . . 7
2.5 Reflexão final . . . 8

3 Global Super Store Dataset **9**
3.1 Domínio a tratar e Objetivo . . . 9
3.2 Descrição do dataset e tratamento de dados . . . 10
3.2.1 Significado dos dados . . . 10
3.2.2 Exploração/Visualização dos dados . . . 10
3.2.3 Tratamento de dados . . . 12
3.3 Modelos desenvolvidos . . . 13
3.4 Resultados obtidos . . . 13
3.4.1 Informações relativas à partição do dataset *Global Superstore* . . . 14
3.4.2 Critérios de performance . . . 14
3.4.3 Modelos e respetivos resultados . . . 14
3.5 Reflexão final . . . 15

4 Ferramentas comuns aos datasets **16**
4.1 Tratamento de dados . . . 16
4.2 Modelos de aprendizagem: . . . 17

5 Conclusão **18**

6 Anexos **19**
6.1 Wine Quality . . . 19

Capítulo 1

Introdução e metodologia utilizada

A importância do *Machine Learning* tem aumentando ao longo dos anos, e a sua utilização tem atravessado vários setores de atividade. Por esta razão, consideramos muito importante aprender mais sobre o funcionamento, e implementação de técnicas de aprendizagem e decisão inteligente baseadas em dados, sendo este trabalho o resultado disso.

Sendo assim, neste trabalho exploramos dois datasets com diferentes caraterísticas, e de áreas diferentes. Apesar de usarem algumas ferramentas comuns, decidimos analisá-los separadamente, porque tomamos várias decisões diferentes. Desta forma, decidimos seguir a metodologia *CRISP-DM* pela sua flexibilidade e facilidade na construção de projetos de análise de dados. Assim, para este projeto seguimos os seguintes passos:

1. **Estudo do Negócio -** onde explicamos qual o domínio a ser tratado e qual o objetivo.
2. **Estudos dos Dados -** onde percebemos o que significam os dados, que certos padrões ou caracterizações estes possem.
3. **Preparação dos Dados -** onde explicamos o cuidado com a preparação dos dados e as decisões tomadas para tal. De notar que este preparamento pode variar consoante os modelos aplicados.
4. **Modelação -** onde são demonstrados os diferentes modelos de aprendizagem aplicados, porquê estes e certos cuidados que tivemos.
5. **Avaliação -** onde é feita uma análise dos resultados obtidos pelos modelos e é explicado o seu significado.
6. **Desenvolvimento -** onde é efetuada uma revisão geral de todo o projeto, e esta fase encontra-se descrita nas secções de Reflexão final em cada um dos capítulos relativos aos datasets.

Capítulo 2

Wine Quality

2.1 Domínio a tratar e objetivo

A qualidade de um vinho depende de diversos parâmetros definidos nas caraterísticas que possui. Assim sendo, é possível analisar a qualidade do vinho através das mesmas. Acreditamos então que é possível prever esta qualidade através da análise das caraterísticas que diferenciam os vários vinhos no mercado. Esta análise deve ser feita a partir de vários fatores objetivos, como a densidade, o pH, o açúcar residual, etc.

Deste modo, nesta parte do trabalho temos como objetivo prever a qualidade dos vinhos a partir de vários parâmetros, como densidade ou pH. Este tipo de análise é útil para que compradores de vinho possam prever a qualidade antes de os provar. Também os produtores podem tentar prever a qualidade do seu vinho.

Por isto, nós pretendemos produzir um modelo que consiga prever com eficácia a qualidade de um vinho a partir das suas várias caraterísticas. Utilizaremos modelos de classificação supervisionada, em que uma parte do dataset será utilizada para treino do modelo e outra para teste. Desta forma, conseguiremos criar um modelo que possa aprender com o dataset dado, e de seguida verificar se está correto.

2.2 Descrição e exploração do dataset e tratamento de dados

O dataset possui várias informações sobre os vinhos, e a qualificação da sua qualidade. De seguida descrevemos as várias informações presentes no dataset, que contém 1599 entradas, bem como o tipo dos parâmetros:

- **1. fixed acidity 6.1 -** (relacionado com a acidez do vinho): contínua.
- **2. volatile acidity6.2 -** (quantidade de ácido acético no vinho, que em níveis muito altos pode levar a um sabor desagradável de vinagre): contínua.
- **3. citric acid 6.3 -** (ácido cítrico, que em pequenas quantidades pode adicionar 'frescura' e sabor aos vinhos): contínua.
- **4. residual sugar 6.4 -** (quantidade de açúcar que resta após a fermentação parar, e é raro encontrar vinhos com menos de 1 grama/litro e vinhos com mais de 45 gramas/litro são considerados doces): contínua.

- **5. chlorides 6.5 -** (quantidade de sal no vinho): contínua.
- **6. free sulfur dioxide 6.6 -** (quantidade livre de SO2, e existe em equilíbrio entre SO2 molecular (como gás dissolvido) e íon bissulfito; previne o crescimento microbiano e a oxidação do vinho): contínua.
- **7. total sulfur dioxide 6.7 -** (pode afetar o aroma e sabor do vinho): contínua.
- **8. density 6.8 -** (densidade do vinho): contínua.
- **9. pH 6.9 -** (descreve a acidez do vinho numa escala de 0 (muito ácido) a 14 (muito básico); a maioria dos vinhos está entre 3-4 na escala de pH): contínua.
- **10. sulphates 6.10 -** (um aditivo de vinho que pode contribuir para os níveis de dióxido de enxofre (S02), que atua como antimicrobiano e antioxidante): contínua.
- **11. alcohol 6.11 -** (percentagem de álcool presente no vinho): contínua.
- **12. quality -** (qualidade do vinho): binária.

Antes de ser possível iniciar o tratamento de dados, é necessário analisar os mesmos de maneira a saber quais a caraterísticas a manipular e de que modo. Assim sendo, o grupo de trabalho achou por bem utilizar os nodos de exploração de dados como o nodo *Statistics* e *Data Explorer*. Estudando os dados fornecidos, apresenta-se abaixo em forma de tabela a informação que o grupo achou relevante de toda a informação dada:

	Minímo	**Média**	**Máximo**
Fixed Acidity	4.6	8.32	15.9
Volatile Acidity	0.12	0.5278	1.58
Citric Acid	0	0.271	1
Residual Sugar	0.9	2.539	15.5
Chlorides	0.012	0.087	0.611
Free Sulfur Dioxide	1	15.875	72
Total Sulfur Dioxide	6	46.468	289
Density	0.99	0.997	1.004
pH	2.74	3.311	4.01
Sulphates	0.330	0.658	2
Alcohol	8.4	10.423	14.9

Apenas a partir da tabela acima, já é possível verificar algumas discrepâncias entre o valor da média e o máximo, o que pode indicar valores chamados de *outliers*. Apesar da sua existência, decidimos não os retirar, porque o tamanho do dataset é reduzido, e não haverem razões fortes para os remover. Para além disso, também utilizamos o nodo *Linear Correlation*, de maneira a obter uma matriz que demonstrasse um indicador de correlação linear contínua ou inversa que as várias caraterísticas têm umas com as outras. Este passo é importante pois permite filtrar certas colunas que definimos como "não necessárias"à previsão da qualidade. Isto é, se duas caraterísticas aumentam/diminuem proporcionalmente, então é apenas preciso uma para prever o objetivo.

Uma nota importante a fazer acerca desta secção do relatório, é que não estamos a distinguir as várias tentativas que realizamos de modo a obter a maior percentagem de *Accuracy*. Assim sendo, estamos a generalizar tratamentos de dados que foram realizados.

Por isso, nem a todos os tratamentos foi aplicado, por exemplo, o nodo de normalização de dados (*Normalizer*).

Abaixo encontra-se a matriz da correlação linear que foi obtida:

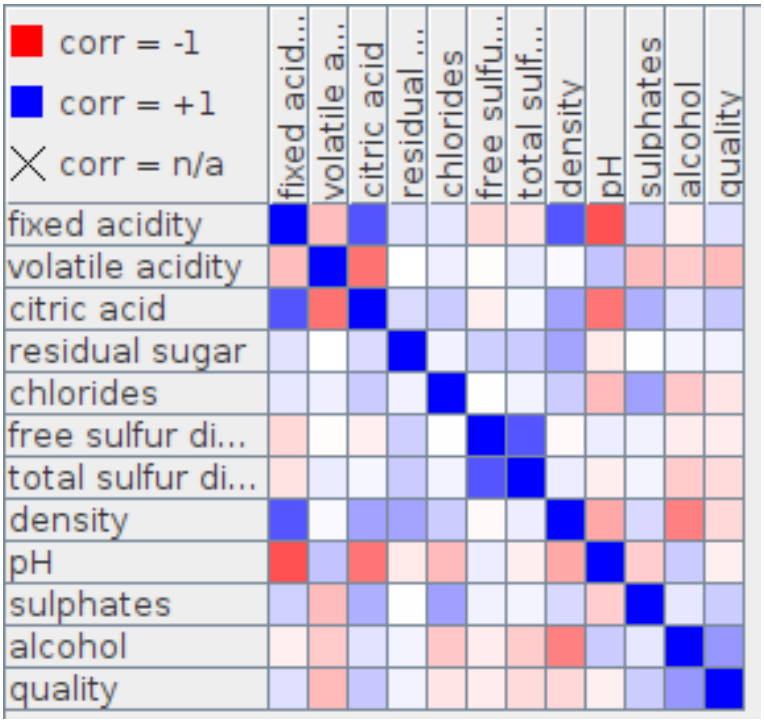

Figura 2.1: Matriz de Correlação Linear

Através da imagem acima imagem, é possível analisar que era prudente filtrar as caraterísticas que apresentavam elevado nível de correlação como:

- **Citric Acid** e **Fixed Acidity**;
- **Total sulfur dioxide** e **Free sulfur dioxide**;
- **pH** e **Fixed acidity**;

Nos anexos encontram-se os vários histogramas de cada um dos parâmetros que definem a qualidade dos vinhos, e é possível ver a sua distribuição. Algumas conclusões podem ser tiradas, como alguns parâmetros terem uma distribuição não simétrica, e com *skew* elevado, como por exemplo o *Residual Sugar* e os *Chlorides*. Por outro lado, a distribuição do pH ou da *density* é bastante positiva para a aprendizagem dos modelos.

Como o tratamento de dados deve ser adequado ao tipo de modelo, nós decidimos descrever no capítulo seguinte como utilizamos várias ferramentas de forma a ter melhores modelos de previsão. No entanto, para todos os modelos utilizados na análise deste dataset, a coluna *target* deve estar em formato nominal, e por isso convertemos o parâmetro qualidade, utilizando o nodo *Number to String*.

2.3 Modelos desenvolvidos

De forma a produzirmos modelos com elevada qualidade de previsão, tivemos de preparar os datasets em função dos modelos. Para além disso, também utilizámos:

- **Parameter Optimization Loop Start/End** para os modelos baseados em redes neuronais, de forma a afinar alguns parâmetros que consideramos importantes, como o número de níveis/camadas e número de neurónios. Assim, conseguimos saber qual a *accuracy* em função destes parâmetros.

- **X-Partitioner e X-Aggregator** que utilizámos para os modelos baseados em árvores de decisão, para que os modelos criados sejam mais fiáveis, e não sejam treinados apenas uma vez. Assim, os indicadores de qualidade são mais fiáveis, porque não tem tanto viés (tradução do inglês *bias*).

Relativamente aos modelos produzidos, vamos descrever agora os 4 melhores modelos que utilizámos:

- **RProp MLP Learner -** com 100 iterações, mas com número variável de níveis/camadas e neurónios por nível, em função do *Parameter Optimization Loop*. A *seed* é -766 177 212.

- **Decision Tree Learner -** Em que não utilizámos métodos de *prunning*, porque este modelo é leve em termos de recursos consumidos, por comparação aos outros. Desta forma, a árvore de decisão gerada testa mais casos.

- **Random Forest Learner -** Com 100 modelos (árvores), sendo que cada uma tem profundidade máxima de 10 níveis. A *seed* para a *Forest Option* é 1651528385605.

- **Gradient Boosted Trees Learner -** que configurámos com profundidade máxima de 4 níveis, 100 modelos, e um *rating* de aprendizagem de 0.1.

2.4 Resultados obtidos

De forma a podermos comparar melhor a qualidade dos diferentes modelos, decidimos utilizar o mesmo conjunto de dados para os treinar individualmente. Assim, todos os modelos foram treinados com os mesmos dados. Os restantes dados foram utilizados para avaliar a qualidade de previsão dos modelos. Assim, os resultados apresentados a seguir são baseados no mesmo conjunto de dados, e tentamos reduzir ao máximo a aleatoriedade dos resultados.

2.4.1 Informações relativas à partição do dataset sobre *wine quality*:

- **Partição entre dados de aprendizagem e de teste 80% -** Consideramos este valor equilibrado entre a quantidade de dados que serve para treinar o modelo e para o seu treino;

- **Stratified sampling -** Desta forma, o modelo treina com dados representativos do dataset, e o critério de divisão foi a qualidade.

- **Seed 2022 -** Desta forma, é possível replicar os resultados que obtemos, para análises futuras.

2.4.2 Modelos e respetivos resultados

Para averiguar a qualidade dos modelos produzidos, usamos como principal critério a *accuracy*. Decidimos usar este critério porque é bastante adequado em problemas de classificação, e dado o contexto do problema achamos que é o mais coerente. Isto porque as consequências de um falso positivo ou falso negativo são as mesmas, e não é preciso um critérios que distinga estas duas situações.

É de destacar que algumas conclusões, como a remoção de certos parâmetros, é congruente com o tratamento de dados. Nalgumas situações remove-se o pH, explicável pela elevada correlação com a *fixed acidity*, ou por exemplo a escolha do parâmetro *density*, que tem uma distribuição bastante positiva.

No fim, descrevemos qual o modelo que consideramos melhor.

Como dito atrás, utilizamos sempre a mesma partição, e obtemos os seguintes resultados para cada um dos modelos.

Redes neuronais (nodo RProp MLP Learner)

Sem qualquer tipo de tratamento de dados, e apenas para servir de ponto de partida, obtemos 86% de *accuracy*.

De forma a obter um correto funcionamento deste tipo de modelos, é necessário normalizar os dados antes de os analisar. Então, após normalizarmos, e utilizando o nodo de Parameter Optimization Loop Start/End, a fazer variar o número de neurónios e níveis/camadas, alcançamos 88%. Este valor é alcançado com 18 neurónios por camada, e duas camadas. Por fim, e utilizando a informação disponibilizada pelo nodo de Backward Feature Elimination, decidimos remover o parâmetro do pH, o que faz sentido tendo em conta a informação contida na matriz de correlação. Com estas caraterísticas, obtemos 90%, sendo o melhor resultado utilizando um modelo baseado em redes neuronais.

Decision Tree Learner:

Para este, e os modelos seguintes que são baseados em árvores de decisão, decidimos apenas filtrar as colunas indicadas na Backward Feature Elimination (BFE).

Para começar, e sem otimizações e tratamento do dataset, para além da conversão do parâmetro qualidade, começamos com um valor de 90.6% de accuracy. Removendo os parâmetros de *citric acid*, pH e *chlorides*, alcançamos 90.3%, o que é bastante positivo, porque diminuimos a complexidade do modelo, e a accuracy é pouco afetada.

Utilizando Cross-Validation, que confere mais fiabilidade ao modelo, conseguimos uma accuracy de 89%.

Random tree learner:

Este e o próximo modelo foram preparados de forma similar ao anterior.

Começando sem qualquer otimização nem tratamento de dados, começamos com 90.9% de accuracy.

Utilizando apenas os parâmetros de *citric acid*, *total sulfur dioxide* e *density*, conforme indica o nodo BFE, conseguimos 90.6%. Mais uma vez, e apesar de baixar a qualidade de previsão, a diminuição da complexidade do modelo compensa bastante essa pequena descida.

Por fim, utilizando a remoção de colunas baseada no BFE, e usando os nodos de Cross-Validation, conseguimos 90.2%.

Gradient tree learner:

Por fim, decidimos também utilizar este modelo, que é mais eficiente que o anterior, e também bastante popular.

Começamos mais uma vez sem qualquer tipo de otimização, e conseguimos 90% de accuracy.

Após a análise do nodo de BFE, excluimos os parâmetros *fixed acidity*, *volatile acidity*, e *free sulfur dioxide*, e utilizando também Cross-Validation, conseguimos 90.4%.

Comparação de modelos e escolha final

Portanto, e após vários testes, decidimos escolher o modelo *Gradient tree learner*, com a seguinte configuração: profundidade máxima de 4 em cada árvore; 100 modelos associados, e um rating de aprendizagem de 0.1.

Essa escolha é baseada na segurança do modelo, por comparação aos outros, pois o seu valor é resultado de Cross-Validation, que confere mais fiabilidade ao modelo gerado, e pela qualidade das previsões, nomeadamente **90.4%** de accuracy.

2.5 Reflexão final

Para concluir a análise e exploração deste dataset, consideramos que o nosso modelo consegue prever com bastante sucesso a qualidade de um vinho, baseado nos vários parâmetros fornecidos. Num contexto real não acreditamos que o nível de precisão do nosso modelo fosse tão elevado, porque a variedade de vinhos é muito elevada.

No entanto, existem algumas coisas que podiam ser feitas de forma a melhorarmos o nosso trabalho, nomeadamente:

- Mais dados, para termos um modelo mais treinado;
- A qualidade ser avaliada com um intervalo de valores maiores, porque considerarmos ser mais útil para potenciais clientes;
- Estudar mais a área da enologia, de forma a saber o que realmente afeta a qualidade de um vinho.

Capítulo 3

Global Super Store Dataset

Para o segundo *dataset* escolhemos o *"Global Super Store Dataset"*.

3.1 Domínio a tratar e Objetivo

Este *dataset* é composto por dados relativos às vendas online de vários produtos. Este caso enquadra-se bem na situação pós-pandêmica atual em que houve um aumento significativo das compras online. Assim, este *dataset* torna-se útil para empresas que pretendam expandir o seu negócio para este mercado. Deste modo, a partir destes dados, o nosso objetivo é criar um modelo capaz de prever o lucro gerado por uma venda.

3.2 Descrição do dataset e tratamento de dados

3.2.1 Significado dos dados

O dataset que escolhemos é composto por 24 colunas, e 51 mil entradas. As colunas são: De notar que temos que este dataset apenas contém missing values no coluna *Postal Code*.

- **Row id, Order id e customer id -** Onde estão armazenados os identificadores da entrada, da ordem do cliente, e do cliente.
- **Order date e ship date -** Onde está registada a data do pedido e de quando foi enviado.
- **Ship mode -** Onde está indicado o modo de transporte, que podem ser, por exemplo: segunda classe, no mesmo dia, etc.
- **Customer name -** Onde se indica o nome do cliente.
- **Segment -** indica o segmento de mercado no qual a compra pertence, como por exemplo: consumo; empresarial, etc.
- **City, state e country -** Cidade, estado e país de destino da encomenda.
- **Postal code e region -** Còdigo postal e região de destino da encomenda. A região pode ser: sul da ásia, África, etc.
- **Market -** O mercado de destino da encomenda, podendo ser: APAC (*Asian-Pacific*); LATAM (*Latin-American*), etc.
- **Product id e product name-** Identificador e nome do produto.
- **Category e subcategory -** Categoria e subcategoria do produto.
- **Sales -** Custo da encomanda.
- **Quantity -** Quantidade de material encomendada.
- **Discount -** Desconto da encomanda.
- **Shipping cost -** Preço de envio da encomenda.
- **Order priotiry -** Prioridade da encomenda, podendo ser: alta; média, crítica, etc.

3.2.2 Exploração/Visualização dos dados

Depois de ficar a perceber o que cada *feature* significa, procuramos perceber como os dados estão distribuídos e que características têm. Para isto, utilizamos o nodo *Data Explorer* e obtemos as seguintes informações:

	No. Missings	**Minimum**	**Maximum**	**Mean**	**Skewness**
Row ID	0	1	51290	25645.5	0
Postal Code	41296	1040	99301	55190.379	-0.129
Sales	0	0.444	22638.48	246.491	8.138
Quantity	0	1	14	3.477	1.36
Discount	0	0	0.85	0.143	1.388
Profit	0	-6599.978	8399.976	28.611	4.157
Shipping Cost	0	0	933.57	26.376	5.863

Tabela 3.1: Exploração dos dados numéricos.

	No. Missings	**Unique values**
Order id	0	>1000
Order Date	0	1000
Ship Date	0	1000
Ship Mode	0	4
Customer id	0	>1000
Customer name	0	795
Segment	0	3
City	0	1000
State	0	>1000
Country	0	147
Market	0	7
Region	0	13
Product id	0	>1000
Category	0	3
Sub-Category	0	17
Product name	0	>1000
Order Priority	0	4

Tabela 3.2: Exploração dos dados nominais.

3.2.3 Tratamento de dados

Inicialmente retiramos as *features* que não continham informação pertinente para a previsão do *target* **Profit**. Estas são: ***Row ID, Order ID, Customer ID (String), Customer Name, Product ID, Product Name***. De notar que antes da remoção destas colunas, decidimos também remover os registos que tivessem com **Row ID** duplicados. Também excluímos a *feature* **Postal Code** pelo elevado número de *missing values*, cerca de 80.5%.

Depois, analisamos a correlação presente entre as features, através do nodo *Linear Correlation* e concluímos que as seguintes colunas estavam bastante relacionadas:

- **Region** e **Market**
- **Sales** e **Shipping cost**
- **Category** e **SubCategory**

Desta forma, optamos por remover as seguintes *features*: **Region**, **Sub-Category** e **Sales**.

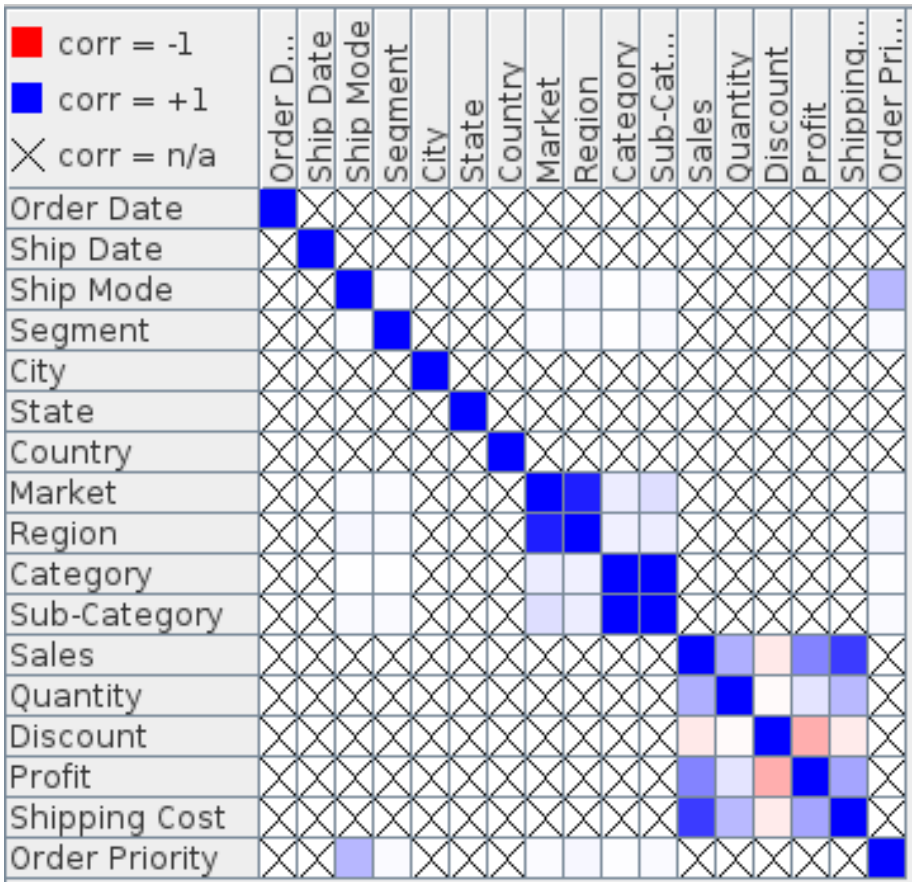

Figura 3.1: Matriz de Correlação Linear.

De seguida, passamos ao tratamento dos *outliers* que separamos em duas partes:
- *All outliers* aplicada à *feature* **Shipping Cost**, de modo a tratar os outliers inferiores e superiores de forma equivalente.
- *Outliers below lower bound* aplicada à *feature* **Profit**, visto que a concentração *outliers* inferiores é relativamente distinta.
- *Outliers above upper bound* aplicada à *feature* **Profit**, por causa da mesma razão referida em cima.

Ainda no tratamento dos *outliers* é de notar que todos foram substituidos pelo valor mais próximo de modo a não perder informação.

Além disto, também tratamos das *features* **Order Date** e **Ship Date**, extraindo apenas o mês destas datas.

Por fim, convertemos todos os dados categóricos em números, de forma a podermos usar modelos como o de regressão. Com este objetivo, utilizamos o nodo *Category to number*.

3.3 Modelos desenvolvidos

Por forma a perceber qual o melhor modelo para o nosso problema, decidimos testar vários modelos de aprendizagem automática.

Assim, testamos o nosso dataset com os seguintes modelos:

1. **Linear Regression;**
2. **Simple Regression;**
3. **Random Forest;**
4. **Gradient Boosted Trees;**
5. **Tree Enseble;**
6. **Redes Neuronais;**

Para cada um destes modelos, executamos o modelo sem qualquer tipo de alteração relativa às configurações do modelo e observamos resultados obtidos. De seguida, caso o modelo se mostrasse promissor, testamo-lo com várias configurações e modos de particionamento diferentes para observar o seu impacto, não só na exatidão dos resultados, mas também no custo computacional de treinar o modelo.

3.4 Resultados obtidos

Ao analisarmos os resultados da primeira iteração de testes, concluímos que os modelos de regressão linear e simples, com valores de R^2 de 0,471 e 0.27 respetivamente, não iriam produzir resultados muito interessantes. Por causa disto, decidimos não continuar a explorar estes modelos. No entanto, os restantes modelos, mostraram todos resultados semelhantes com valores a rondar os 0.6-0.65.

Com os modelos que nos interessaram, procedemos a testa-los com *optimization loops* para observar o resultado de alterar as variáveis dos modelos, X-Partitioner/X-Aggregator para testar usar o dataset inteiro para treino e teste e, por fim, usar o *feature selection* para testar quais as colunas que se podem remover sem afetar a performance significativamente.

3.4.1 Informações relativas à partição do dataset *Global Superstore*

- **Partição entre dados de aprendizagem e de teste 80% -** De igual forma ao dataset anterior, achamos este um valor equilibrado e que produz resultados ligeiramente melhores que um particionamento 70/30;
- **Draw Randomly -** Visto que este se trata de um problema de regressão, não é possível usar o *stratified sampling*. Por isto, decidimos usar o *draw randomly* com a seed **12345**.

3.4.2 Critérios de performance

Sendo o problema deste dataset um de regressão, usamos como critério de performance as seguintes métricas:

- **R^2 -** Permite-nos saber o quão bem as previsões do modelo de regressão se encaixam nos dados. Também serve como um indicador geral do quão bom o modelo é.
- ***Mean Absolute Error* (MAE) -** Tal como o nome indica, este valor dá-nos a média dos erros resultantes da previsão do modelo. Esta métrica é um bom indicador quando pretendemos reduzir os erros mais baixos.
- ***Mean Squared Error* (MSE) -** De forma semelhante ao MAE, esta métrica calcula o quadrado dos erros de previsão do modelo. Visto que o erro de cada previsão é elevado ao quadrado no cálculo do MSE, este é um bom critério quando pretendemos reduzir os erros maiores.

3.4.3 Modelos e respetivos resultados

Redes neuronais (nodo RProp MLP Learner)

Começando por executar o metanodo *Backward Feature Selection*, observamos que podemos apenas manter as colunas *Category*, *Quantity*, *Discount*, *Shipping Cost* e *Profit* sem ter um grande impacto na qualidade do modelo. Com esta alteração o modelo obteve um R^2 de 0.6, um MAE de 0.121 e um MSE de 0.029.

De seguida, executamos a partir deste o *Parameter Optimization Loop* em que concluímos que a partir de 10 neurónios não há um impacto significativo na qualidade dos resultados.

Por fim, com a ajuda do *X-Partitioner/X-Aggregator* obtivemos um R^2 de 0.602, um MAE de 0.122 e um MSE de 0.029.

Árvores de decisão (Random Tree Learner)

Através do *Backward Feature Selection* concluímos que ao mantermos as colunas *Category*, *Quantity*, *Discount*, *Profit*, *Shipping Cost* e *Order Priority*, conseguimos reduzir o número de features, preservando uma boa qualidade de previsão.

Ao correrrmos o *Parameter Optimization Loop*, tendo como variável o número de modelos, percebemos que ao aumentarmos esta variável, não obtemos nenhuma melhoria nas previsões do modelo. Assim, optamos por manter o valor predefinido de 100.

Com estas alterações o nosso modelo obteve um R^2 de 0.646, um MAE de 0.109 e um MSE de 0.026.

Por fim, e juntando aos nodos anteriores, testamos com o nodo *X-Partioning* e obtivemos os seguintes resultados: R^2 de 0.636, um MAE de 0.109 e um MSE de 0.027.

Árvores de decisão (Gradient Boosted Tree Learner)

Mais uma vez, ao usarmos o *Backward Feature Selection*, determinamos que manter apenas as colunas *Country*, *Category*, *Quantity*, *Discount*, *Profit*, *Shipping Cost* e *Order Priority* nos dará o melhor balanço entre qualidade de previsão e custo de computacional para o treino do modelo. Com esta seleção de colunas obtivemos um R^2 de 0.64, um MAE de 0.103 e um MSE de 0.026.

Seguidamente, ao observarmos os resultados de variarmos o *learning rate* do modelo com o *Parameter Optimization Loop*, entendemos que o valor que produz melhores resultados é 0.1.

Finalmente, com o *X-Partitioner*, obtivemos um R^2 de 0.635, um MAE de 0.105 e um MSE de 0.027.

Árvores de decisão (Ensemble Tree Learner)

Como nos outros casos, começamos por passar os dados pela *Backward Feature Selection*. A partir daqui, determinamos que se mantivermos apenas das colunas *Market*, *Category*, *Quantity*, *Discount*, *Profit*, *Shipping Cost* e *Order Priority*, não há uma grande perda na qualidade do modelo. Após a filtragem das colunas, obtivemos um modelo com um R^2 de 0.644, um MAE de 0.107 e um MSE de 0.026.

Posteriormente, executamos o *Parameter Optimization Loop* para verificar o efeito resultante de alterar o número mínimo de divisões de cada nodo. Após observarmos os resultados, apercebemos que alterar esta variável não tem nenhum efeito nas previsões do modelo. Por causa disto, decidimos não alterar este valor.

Por fim, tal como nos outros casos, corremos o modelo com o *X-Partitioner* e observamos os resultados. Assim, obtivemos um R^2 de 0.637, um MAE de 0.109 e um MSE de 0.027.

3.5 Reflexão final

Para concluir a análise e exploração deste dataset, consideramos que não seja possível prever o lucro gerado por uma venda a partir deste conjunto de dados. Sendo assim, num contexto real, não acreditamos que o nosso modelo possa ser utilizado de forma consistente, visto existirem vários fatores que influenciam o lucro de uma venda que não estamos aqui a considerar, nomeadamente o custo de produção e o preço de venda.

Para além disso, acreditamos que esta foi a razão pela qual não obtivemos nenhum ganho significativo com as alterações feitas aos modelos, e também pela qual nenhum modelo se destacou relativamente aos outros.

Capítulo 4

Ferramentas comuns aos datasets

4.1 Tratamento de dados

- **Remoção de colunas (Column Filter)-** A remoção de colunas tem como objetivo retirar parâmetros de todas as entradas. Esta remoção pode ser baseada em vários fatores, nomeadamente: simplificar o dataset para que os modelos usem menos recursos na análise; quando certas colunas não são úteis para a qualidade de previsão; ou quando alguns parâmetros prejudicam a qualidade da previsão.

- **Remoção de colunas baseada na matriz de correlação -**Se após a análise desta matriz com todos os parâmetros, notamos uma elevada correlação entre duas colunas, então um dos valores pode ser retirado. Assim, e utilizando o nodo descrito anteriormente, decidimos remover um dos parâmetros. Esta decisão serve para remover informação duplicada do dataset, porque se duas colunas tem muita correlação, então não precisam de ser analisadas. Para além disso, a remoção dessas colunas melhora a eficácia dos modelos.

- **Remoção de outliers -** Decidimos remover os outliers do nosso dataset para alguns modelos porque os podem afetar negativamente. Para esta transformação, utilizamos o nodo *Numeric outliers*.

- **Normalização de dados -** Decidimos normalizar todos os dados exceto a qualidade que era nominal, para o modelo baseado em redes neuronais. Essa normalização converteu os valores para um intervalo entre 0 e 1. Tomamos esta decisão porque este tipo de modelos é mais eficiente quando analisa dados normalizados, e isso é demonstrado na secção de resultados obtidos. Nos modelos baseados em árvores de decisão, não decidimos normalizar porque não deve afetar a qualidade final do modelo.

- **Backward feature elimination (BFE) -** Através desta ferramenta, podemos analisar a influência que um conjunto de parâmetros tem na qualidade de previsão do modelo. Assim, e removendo as *features* que pioram a eficácia do modelo, podemos tratar o dataset da melhor forma.

4.2 Modelos de aprendizagem:

- **Parameter Optimization Loop Start -** Com este nodo, conseguimos testar automaticamente um dado modelo com vários parâmetros definidos por nós. Desta forma, encontramos mais facilmente os valores que maximizam a qualidade de previsão do modelo.
- **X-Partitioner e X-Aggregator -** Para podermos aumentar a qualidade do modelo, decidimos utilizar esta ferramenta de *Cross-Validation*, para utilizarmos todo o dataset na aprendizagem do modelo.

Capítulo 5

Conclusão

Por fim, consideramos que fizemos um trabalho interessante, e que aprendemos mais sobre *Machine Learning*, e tratamento de dados. Também tivemos de explorar vários algoritmos de aprendizagem e podemos compará-los com exemplos reais, averiguando a sua eficiência e qualidade de previsão.

Capítulo 6

Anexos

6.1 Wine Quality

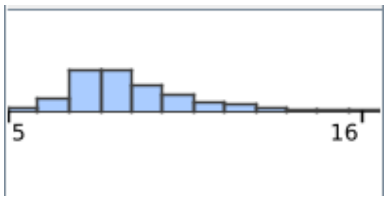

Figura 6.1: Histograma do parâmetro *Fixed Acidity*

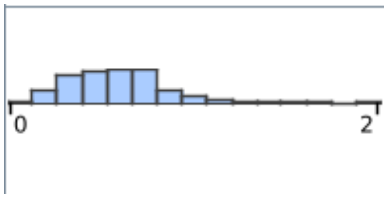

Figura 6.2: Histograma do parâmetro *Volatile Acidity*

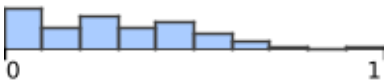

Figura 6.3: Histograma do parâmetro *Citric Acid*

Figura 6.4: Histograma do parâmetro *Residual Sugar*

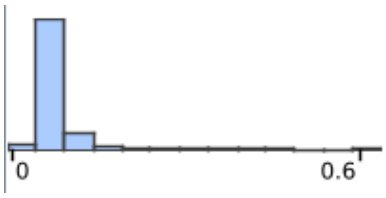

Figura 6.5: Histograma do parâmetro *Chlorides*

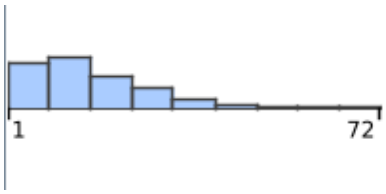

Figura 6.6: Histograma do parâmetro *Free Sulfur Dioxide*

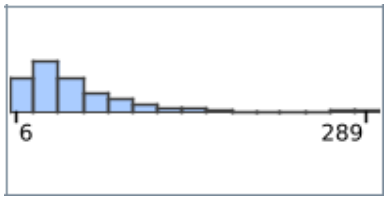

Figura 6.7: Histograma do parâmetro *Total Sulfur Dioxide*

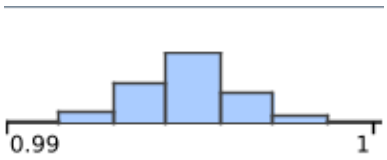

Figura 6.8: Histograma do parâmetro *Density*

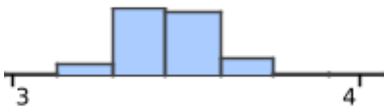

Figura 6.9: Histograma do parâmetro *pH*

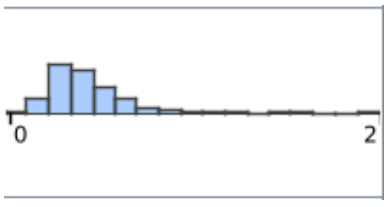

Figura 6.10: Histograma do parâmetro *Sulphates*

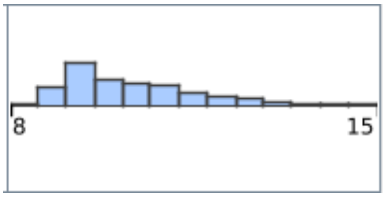

Figura 6.11: Histograma do parâmetro *Alcohol*