



UNIVERSIDADE DO MINHO

DEPARTAMENTO DE INFORMÁTICA

Dados e Aprendizagem Automática

Trabalho Prático

Grupo N<sup>o</sup> 29

Ariana Lousada (PG47034)

Márcia Teixeira (A80943)

Carlos Gomes (PG47083)

Tiago Sousa (PG47684)

2 de janeiro de 2022

# Conteúdo

<b>1</b>	<b>Introdução</b>	<b>3</b>
<b>2</b>	<b>Análise dos dados</b>	<b>4</b>
2.1	Dataset 1 . . . . .	4
2.2	Dataset 2 . . . . .	7
<b>3</b>	<b>Tratamento dos dados</b>	<b>9</b>
3.1	Dataset 1 . . . . .	9
3.2	Dataset 2 . . . . .	10
<b>4</b>	<b>Previsão</b>	<b>13</b>
4.1	Modelos Utilizados . . . . .	13
4.2	Treino de modelos . . . . .	13
4.2.1	Gradient Boosting . . . . .	13
4.2.2	XGBoost . . . . .	16
<b>5</b>	<b>Resultados</b>	<b>18</b>
5.1	Dataset 1 - Fluxo de tráfego rodoviário na cidade do Porto . . . . .	18
5.2	Dataset 2 - Housing Price . . . . .	18
5.3	Análise de resultados . . . . .	19
<b>6</b>	<b>Conclusão</b>	<b>20</b>

# Capítulo 1

## Introdução

No âmbito da Unidade Curricular de Dados e Aprendizagem Automática, foi proposta a realização de um trabalho prático cujo objetivo consistiu no desenvolvimento de um modelo de Machine Learning, através da utilização de modelos de aprendizagem que foram abordados ao longo do semestre.

Para a construção de uma solução para o presente trabalho prático, foram utilizados dois datasets.

O primeiro dataset foi fornecido pela equipa docente da unidade curricular e contém dados referentes ao tráfego de veículos na cidade de Porto durante um período superior a 1 ano. Este dataset vai cobrir o período entre 24 de Julho de 2018 até 02 de Outubro de 2019. O objetivo neste dataset consiste em elaborar modelos de previsão do atributo "Average\_Speed\_Diff".

Relativamente ao segundo dataset, foi necessário uma consulta e análise de vários datasets por parte da equipa de trabalho. Após várias pesquisas, foi escolhido um dataset que contém informação sobre várias habitações e como as suas qualidades influenciam o seu preço de compra. O objetivo neste dataset consiste desenvolver modelos de previsão do preço(de venda) das habitações.

Ao longo deste documento vão ser abordadas todas as decisões tomadas por parte da equipa de trabalho na análise e tratamento de dados e escolha de modelos de Machine Learning.

## Capítulo 2

# Análise dos dados

Antes de iniciar a construção dos modelos de Machine Learning, é importante fazer uma análise concisa ao dataset.

Esta fase é essencial para a globalidade do projeto, mesmo que seja muitas vezes desprezada, uma vez que permite uma melhor compreensão a cerca do dataset, como as relações entre os atributos e a identificação de dados desnecessários para o problema.

Também é importante verificar se existem *missing values*, visto que a sua presença se trata de um dos problemas mais comuns que se pode detetar, principalmente em datasets de grande volume. Este tipo de problemas pode ser consequência de erros humanos, questões de privacidade, falhas de leitura por parte de equipamento, entre outros. Este tipo de valores afetam negativamente o desempenho dos modelos e devem ser tratados devidamente e de acordo com o objetivo do modelo a construir.

### 2.1 Dataset 1

Ao analisar o primeiro dataset, foi possível observar que no atributo "average\_precipitation", o valor é sempre o mesmo para todo o dataset. Para uma análise mais precisa, foi construído o gráfico da figura 2.1.

Portanto, uma vez que a variável mantém sempre o mesmo valor, esta irá ser removida na fase de **Tratamento de Dados**.

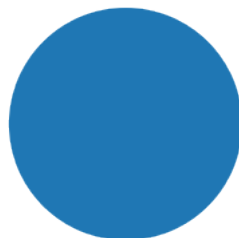


Figura 2.1: Gráfico representante dos valores tomados pela variável `average_precipitation`.

Uma vez que o objetivo consiste numa previsão do atributo "Average.Speed.Diff", foi em primeiro lugar feita uma análise à distribuição deste mesmo.

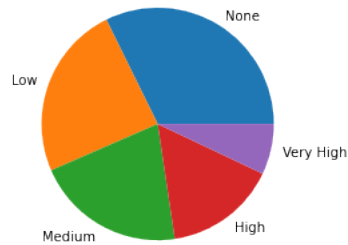


Figura 2.2: Gráfico representante dos valores tomados pela variável `average_speed_diff`

Como é de possível análise na figura, este atributo trata-se de uma variável categórica que vai tomar os valores "None", "Low", "Medium", "High" e "Very High". Visto isto, com o gráfico da figura 2.2 é possível ter uma ideia do trânsito que existe ao longo do dia na cidade de Porto. Existe uma maior concentração no parâmetro "None", portanto é possível concluir que ao longo do dia existem bastantes períodos onde não existe praticamente nenhum trânsito, principalmente de madrugada. Por outro lado, verifica-se que "Very High" corresponde a períodos como início do dia ou final de tarde, que correspondem a horários de maior movimento.

De seguida, fez-se um estudo para tentar analisar relações entre o atributo "average\_speed\_diff" e as restantes variáveis.

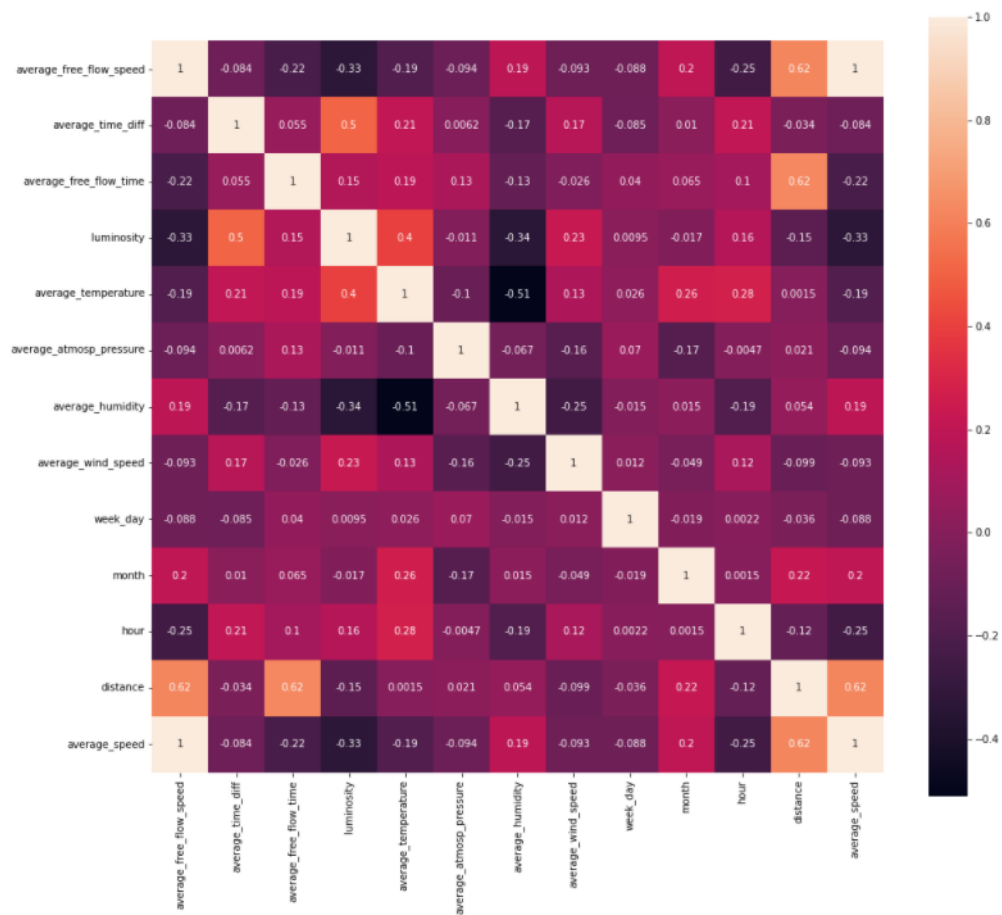


Figura 2.3: Heatmap das correlações entre as diferentes variáveis do dataset.

Com o gráfico da figura 2.3 podem-se observar várias variáveis com correlações negativas e positivas com a *target variable* (average\_speed\_diff). As variáveis mais importantes para a sua futura previsão irão portanto ser as que possuem um valor de correlação mais elevado.

## 2.2 Dataset 2

No segundo dataset, optou-se por um procedimento semelhante: uma vez que a variável da qual pretendemos fazer uma previsão é o atributo "SalePrice", começamos então por observar as correlações entre o "SalePrice" e os restantes atributos.

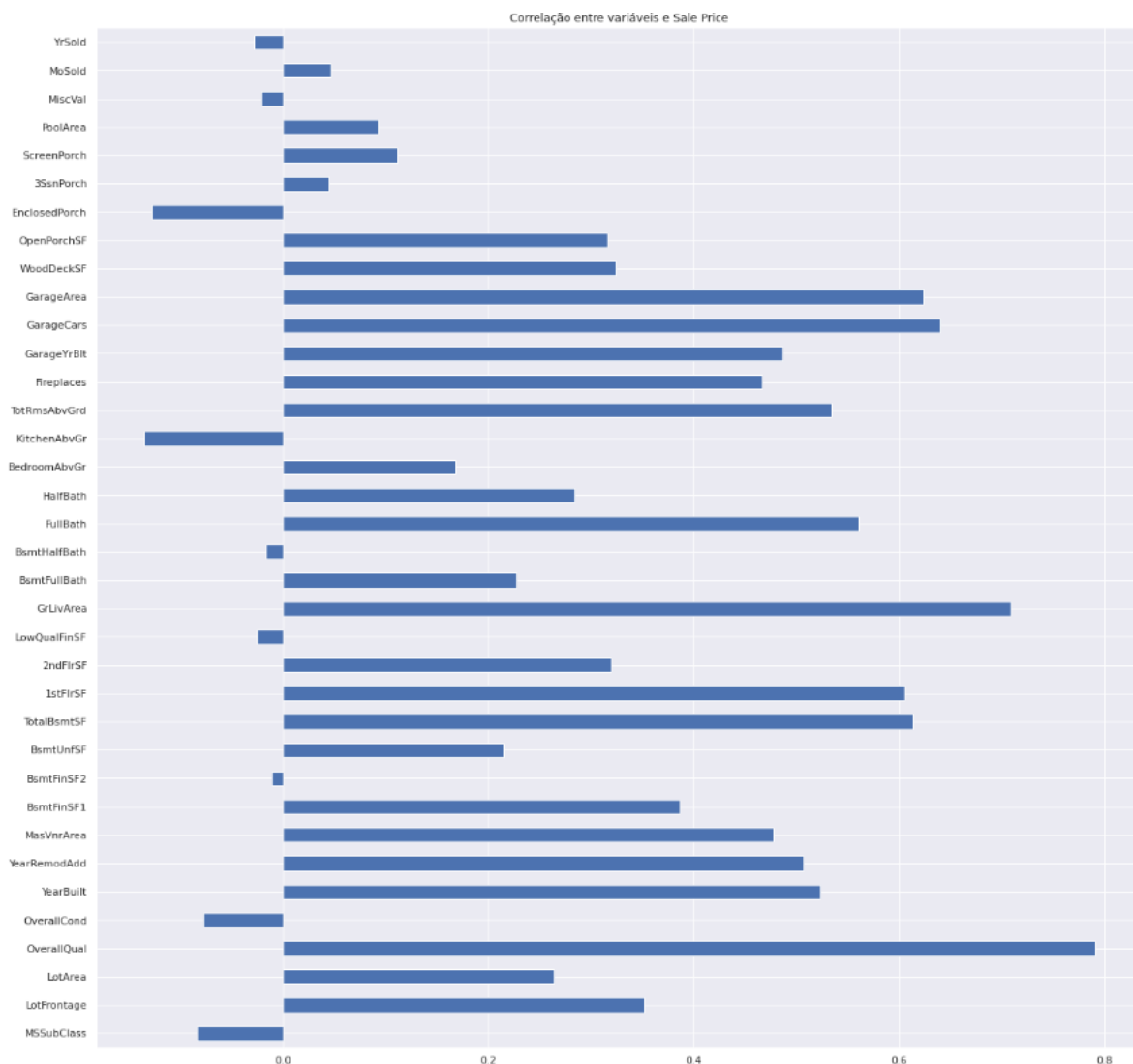


Figura 2.4: Análise das correlações entre as variáveis do dataset 2 com a *target variable* 'Sale-Price'.

De seguida, apenas foram selecionadas as variáveis que possuíam correlação positiva com a variável target, através de uma matriz de correlação. O seguinte gráfico foi construído de modo a ser possível fazer uma seleção das variáveis que apresentam maior correlação com o SalePrice.

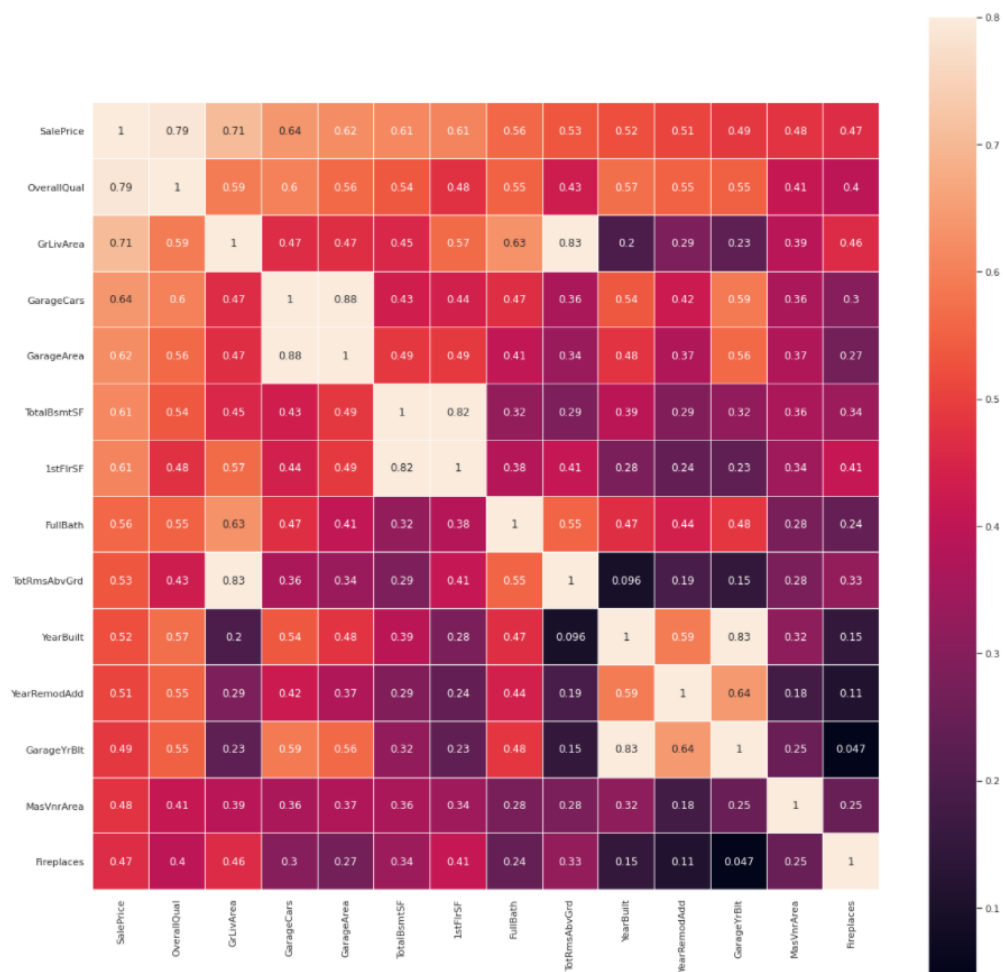


Figura 2.5: Heatmap das correlações entre as diferentes variáveis do dataset.

Através da análise do *heatmap* anterior, é possível concluir que as variáveis **GrLivArea** e **TotalBsmntSF** estão relacionadas linearmente com a *target variable*. Quando estas variáveis sofrem um aumento no seu valor, 'SalePrice' também aumenta, assim como as variáveis **OverallQual** e **YearBuilt**.

Esta análise irá auxiliar a seleção das variáveis cujo tratamento será de maior prioridade, uma vez que terão uma maior importância para gerar mais tarde previsões da *target variable*.



## Capítulo 3

# Tratamento dos dados

Após uma análise concisa aos dados, segue-se o seu tratamento. Nesta fase, serão removidos valores de menor importância. Também se irá fazer um tratamento de dados duplicados e *outliers*.

### 3.1 Dataset 1

O primeiro passo nesta fase de tratamento de dados consistiu na separação da variável "record\_date", para fornecer dados mais úteis. Dada a hipótese colocada durante a análise de average\_speed\_diff, podemos expandir a record\_date para os seguintes atributos: 'month', que nos providencia com as diferenças de average\_speed\_diff ao longo do ano, 'week\_day' com a diferença de average\_speed\_diff ao longo dos dias da semana e 'hour' para permitir uma melhor análise e previsão do tráfego ao longo do dia.

De seguida, foi alterado o atributo "luminosity". Este atributo, visto que se trata de uma variável categórica não contém informação numérica (toma valores como 'Dark' apenas, por exemplo). Contudo, os modelos de Machine Learning não utilizam informação neste formato, pelo que é necessário fazer uma conversão. Uma hipótese seria descartar este atributo, no entanto, a luminosidade é um fator que afeta a segurança de condução e, por consequência, a velocidade máxima aconselhada. Desta forma a equipa de trabalho decidiu incluir este atributo no modelo a construir. Com isto, foi feita uma conversão dos valores deste atributo para valores numéricos do tipo "0" ou "1" ou "2". Uma vez que existiam mais atributos categóricos no dataset com relevância para o problema, aplicou-se a mesma técnica de conversão numérica aos restantes.

```
# Process the columns that have qualitative values in strings.
data.luminosity = data.luminosity.map({'DARK': 0, 'LOW_LIGHT': 1, 'LIGHT': 2})
if 'average_speed_diff' in data.columns:
    data.average_speed_diff = data.average_speed_diff.map({'None':0, 'Low':1, 'Medium':2, 'High':3, 'Very_High':4})

data.average_cloudiness = data.average_cloudiness.map({'NULL':0, 'ceu limpo':1, 'nuvens quebradas':2,
    'tempo nublado':3, 'nublado':3})

data['average_cloudiness'].fillna(0, inplace=True)
data['average_rain'].fillna(0, inplace=True)

data.average_rain = data.average_rain.map({'NULL':0, 'chuvisco fraco':1, 'chuvisco e chuva fraca':2, 'chuva fraca':2,
    'chuva leve':2, 'aguaceiros fracos':2, 'chuva':3, 'aguaceiros':3, 'chuva moderada': 4, 'chuva forte': 5,
    'chuva de intensidade pesado': 5, 'chuva de intensidade pesada': 5, 'trovoada com chuva leve': 6, 'trovoada com chuva': 7})
```

Figura 3.1: Tratamento das variáveis categóricas.

De modo a ter à disposição os atributos que influenciam a velocidade (velocidade = distância/tempo) utilizaram-se os atributos `average_speed_diff`, `average_free_flow_speed`, `average_time_diff` e `average_free_flow_time` para calcular os valores de distância em falta.

```
# Produce a list of the total distance
data['distance'] = data["average_free_flow_speed"] * data["average_free_flow_time"] / 3.6 # 60 * 60 / 1000
```

Figura 3.2: Código Python: Adição da coluna 'distance' ao dataset.

Como foi referido anteriormente, também foi removido o atributo "average\_precipitation", assim como "city\_name", "average\_cloudiness" e "average\_rain", cuja informação era irrelevante para o problema.

```
# Remove columns that don't provide valuable data.
data.drop('city_name', inplace=True, axis=1)
# 100% of the data provided is '0.0'.
data.drop('average_precipitation', inplace=True, axis=1)
# average_cloudiness shows 57% of it's data as empty or null.
data.drop('average_cloudiness', inplace=True, axis=1)
# average_rain shows 91.7% of it's data as empty or null.
data.drop('average_rain', inplace=True, axis=1)
```

Figura 3.3: Código Python: Remoção de colunas de dados com informação não pertinente.

## 3.2 Dataset 2

Para o segundo dataset, foi analisada primeiramente a presença de valores nulos. Para uma mais fácil e rápida observação foi construído o gráfico da figura 3.4 que permite observar a percentagem de valores nulos em cada atributo.

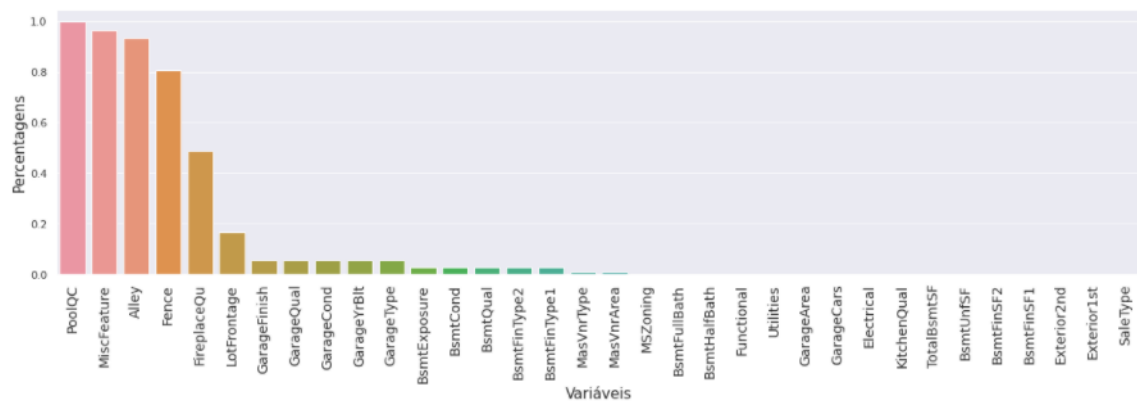


Figura 3.4: Gráfico das percentagens de valores nulos em cada variável do dataset.

Apesar de ser justificável remover colunas cuja percentagem de valores nulos ultrapasse um certo valor, foi decidido substituir todos os valores por 'None' ou '0'(zero), consoante a variável fosse categórica ou quantitativa, respetivamente. Observando a variável 'PoolQC' como exemplo: os valores nulos associados a esta variável correspondem a casas que não possuem qualquer tipo de piscina, o que neste dataset corresponde a cerca de 98% das habitações. Como se trata

de informação potencialmente relevante a possíveis investidores das habitações, decidiu-se não remover esta coluna de dados.

Após a substituição de todos os valores nulos, estabeleceu-se como objetivo tratar também de *outliers*, visto que podem mais tarde influenciar negativamente o modelo a construir para a previsão de dados.

Visto isto, observou-se em primeiro lugar o comportamento da *target value* 'SalePrice' ao longo do dataset (com outliers).

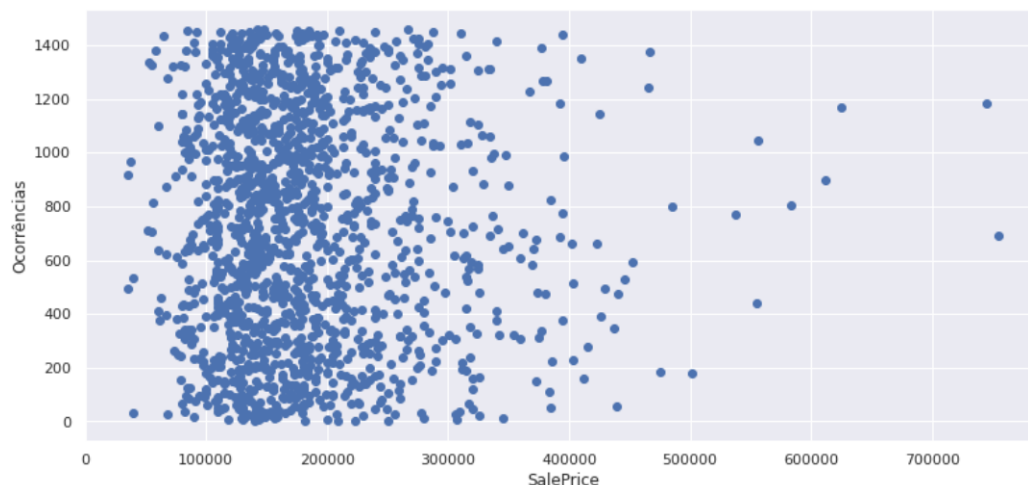


Figura 3.5: Comportamento da variável 'SalePrice' ao longo do dataset.

A partir da análise do gráfico anterior, podemos observar vários outliers, nomeadamente a partir do valor 5000000 de SalePrice. Para a sua remoção alterou-se o upperlimit da target value de modo a regularizar estes valores:

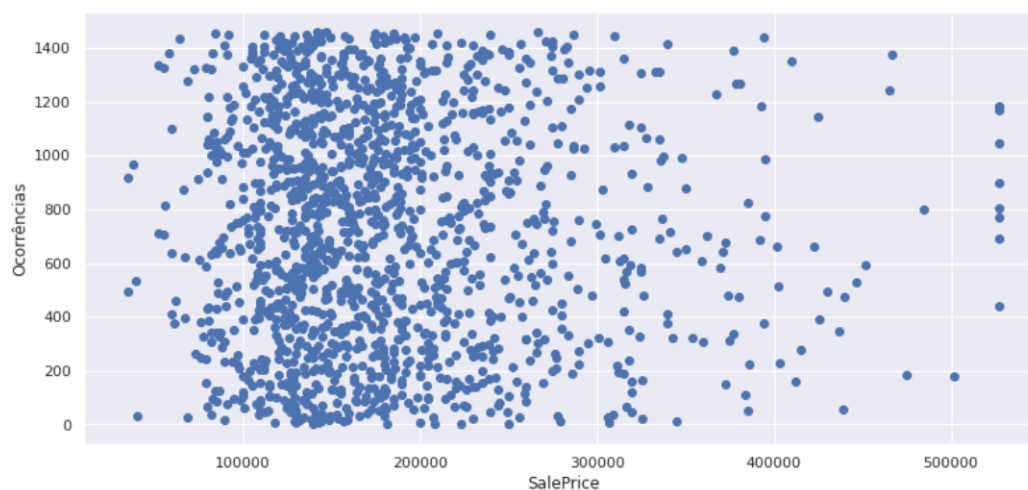


Figura 3.6: Comportamento da variável 'SalePrice' após o tratamento de *outliers*.

Por fim, foi feita uma conversão dos valores de todas as variáveis categóricas em valores numéricos. Uma vez que todos os modelos requerem que todos os inputs/outputs sejam do tipo numérico, para prevenir potenciais erros na comparação de variáveis e obter os melhores resultados possíveis, foi utilizada a biblioteca **pandas** para converter essas variáveis categóricas em *dummy-variables*.

## Capítulo 4

# Previsão

### 4.1 Modelos Utilizados

Após uma pesquisa extensiva acerca de possíveis modelos a utilizar, foi possível encontrar vários exemplos de modelos aplicáveis ao problema. No entanto, um dos modelos de maior eficiência encontrados foi o **Gradient Boosting**, uma vez que este algoritmo é considerado uma das técnicas mais poderosas para fazer previsões. Este modelo é capaz de ser utilizado para ambos os problemas de regressão e classificação, utilizando árvores de decisão.

Uma outra alternativa também bastante utilizada foi o modelo **XGBoost**, que tem por base também o modelo Gradient Boosting, que tem em conta também o desempenho e a velocidade, que por sua vez também utiliza árvores de decisão.

### 4.2 Treino de modelos

Como foi referido anteriormente, a equipa de trabalho teve à sua disposição dois dataset **training\_data** (um de cada dataset) que irá ser utilizado para "alimentar" o modelo para que este ganhe a capacidade de fazer previsões. Sem estes dados, até os modelos mais eficientes não seriam capazes de encontrar relacionamentos nem fazer uma previsão corretamente. Isto é uma das razões pela qual é importante um tratamento rigoroso dos dados.

Este treino possibilita a própria aprendizagem do modelo, o que por sua vez irá melhorar a sua precisão(*accuracy*) à medida que vários dados lhe são fornecidos.

#### 4.2.1 Gradient Boosting

##### Dataset 1

Ao treinar deste modelo, foi necessário descobrir os melhores parâmetros para a sua utilização. Estes parâmetros podem afetar o desempenho do algoritmo e consequentemente a previsão final. Este passo é feito com a utilização de **GridSearchCV** no qual é indicado o modelo, assim como os parâmetros. Esta ferramenta calcula os melhores argumentos para utilizar, apesar de ser um pouco demorado.

```

parameters = {'learning_rate': [0.01,0.02,0.03,0.04],
              'subsample'      : [0.9, 0.5, 0.2, 0.1],
              'n_estimators'   : [100,500,1000, 1500],
              'max_depth'      : [4,6,8,10]}

```

Figura 4.1: Parâmetros utilizados na **GridSearchCV**.

```

Results from Grid Search

The best estimator across ALL searched params:
GradientBoostingRegressor(learning_rate=0.01, max_depth=4, n_estimators=1000,
                           subsample=0.5)

The best score across ALL searched params:
0.9020962609007598

The best parameters across ALL searched params:
{'learning_rate': 0.01, 'max_depth': 4, 'n_estimators': 1000, 'subsample': 0.5}

```

Figura 4.2: Resultados da Grid Search.

Uma vez descobertos os parâmetros ideais para o treino do modelo, fez-se então uma previsão para o dataset **train** e outra previsão para o dataset **test**.



Figura 4.3: Resultados da previsão de dados com o Gradient Boosting.

## Dataset 2

Utilizando um raciocínio semelhante ao aplicado no primeiro dataset, obtiveram-se os seguintes resultados no segundo:

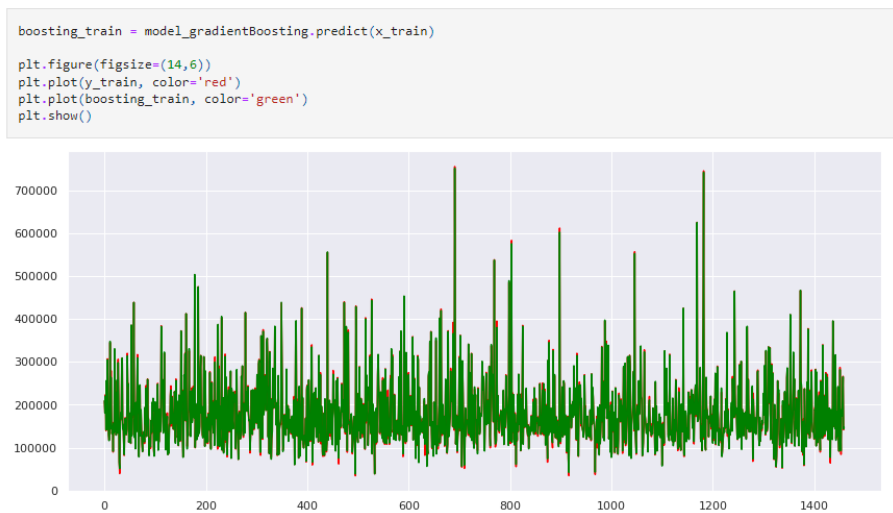


Figura 4.4: Comparação dos dados contidos no dataset `train.csv`(vermelho) com os dados previstos pelo modelo construído(verde).

## 4.2.2 XGBoost

### Dataset 1

De modo a ser possível comparar vários dados finais resultantes de diferentes modelos, utilizou-se também para o primeiro dataset o **XGBoost**. Após várias pesquisas e testes com diferentes valores possíveis de parâmetros, a equipa de trabalho encontrou os melhores valores para o dataset em questão:

```
XGB = xgb.XGBRegressor(n_estimators=2500, max_depth=7, learning_rate=0.01, subsample=0.7, colsample_bytree=1)
```

Figura 4.5: Parâmetros utilizados no **XGBoost**.

Uma vez descobertos os parâmetros ideais para o treino do modelo, fez-se então uma previsão para o dataset:

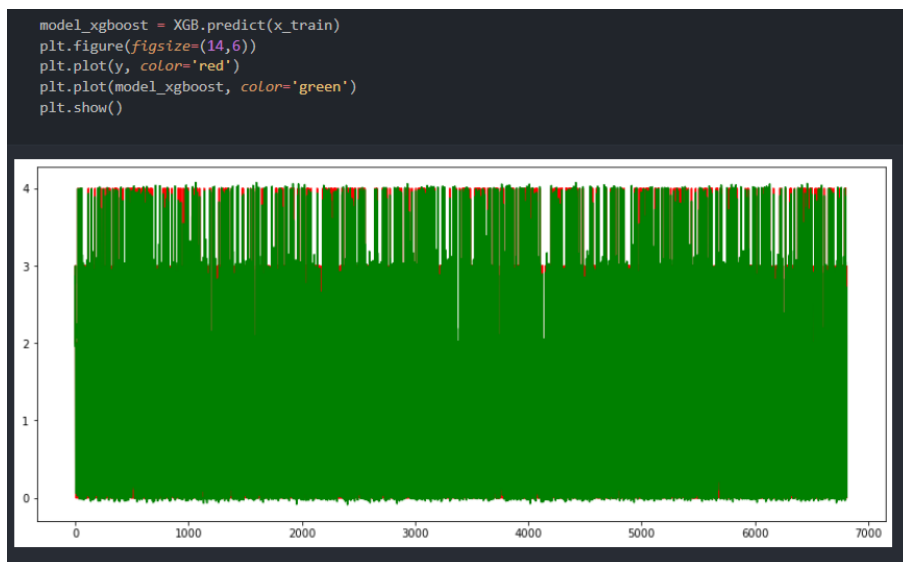


Figura 4.6: Resultados do segundo dataset com utilização do XGBoost.

### Dataset 2

Utilizando um raciocínio semelhante ao aplicado no primeiro dataset, foram utilizados os parâmetros expostos na figura 4.7, resultando na solução da figura 4.8.

```
model_XGB = xgb.XGBRegressor(n_estimators=2200, max_depth=7, learning_rate=0.05, subsample=0.7, colsample_bytree=1)
```

Figura 4.7: Parâmetros utilizados no **XGBoost**.



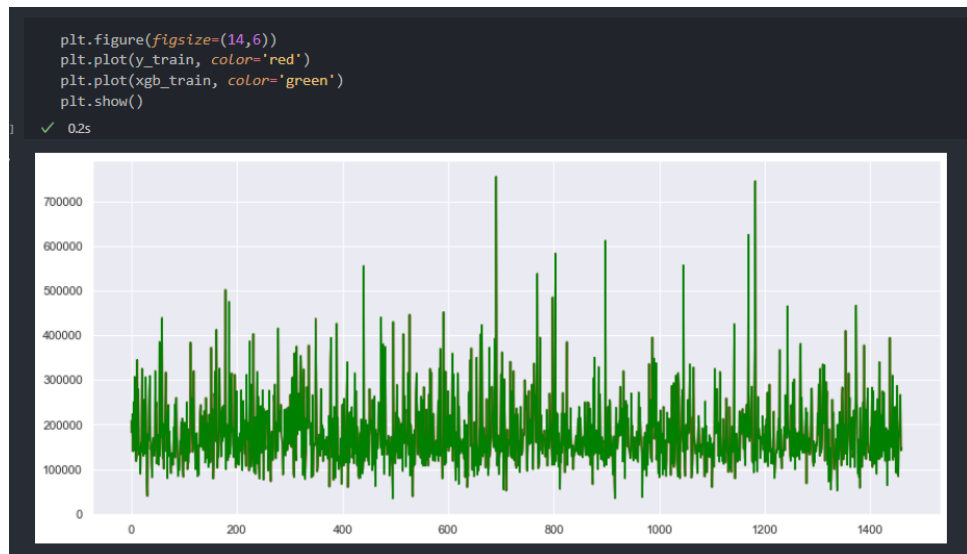


Figura 4.8: Resultados do segundo dataset com utilização do XGBoost.

## Capítulo 5

# Resultados

### 5.1 Dataset 1 - Fluxo de tráfego rodoviário na cidade do Porto

Com a utilização dos modelos *Gradient Boosting* e *XGboost* obtiveram-se os seguintes resultados <sup>1</sup>:

<b>XGBoost</b>	<b>Gradient Boosting</b>
Id,Speed_Diff	Id,Speed_Diff
1,None	1,None
2,Medium	2,Low
3,None	3,None
4,High	4,High
5,Low	5,Low
6,Medium	6,Medium
7,Medium	7,Medium
8,Medium	8,Medium
9,Low	9,Low
10,Medium	10,Medium
11,None	11,None
12,None	12,None
13,Medium	13,Medium

Figura 5.1: Amostra dos resultados do primeiro dataset.

### 5.2 Dataset 2 - Housing Price

Com a utilização dos modelos *Gradient Boosting* e *XGboost* obtiveram-se os seguintes resultados<sup>2</sup>:

---

<sup>1</sup>Para uma análise mais aprofundada, consultar os ficheiros dataset1XGBoost.csv e dataset1Gradient.csv, respetivamente.

<sup>2</sup>Para uma análise mais aprofundada, consultar os ficheiros dataset2XGBoost.csv e dataset2Gradient.csv, respetivamente.

XGBoost	Gradient Boosting
Id,SalePrice	Id,SalePrice
1461.0,124961.19265741957	1461.0,124961.19265741957
1462.0,155634.43176358478	1462.0,155634.43176358478
1463.0,188742.75098360935	1463.0,188742.75098360935
1464.0,200193.27446366608	1464.0,200193.27446366608
1465.0,190192.4612168257	1465.0,190192.4612168257
1466.0,168811.50119056305	1466.0,168811.50119056305
1467.0,166766.42627492957	1467.0,166766.42627492957
1468.0,168156.47873728766	1468.0,168156.47873728766
1469.0,178152.63910810827	1469.0,178152.63910810827
1470.0,134898.93012598227	1470.0,134898.93012598227
1471.0,186743.10475360855	1471.0,186743.10475360855
1472.0,92425.67809522928	1472.0,92425.67809522928
1473.0,95435.74041576113	1473.0,95435.74041576113

Figura 5.2: Amostra dos resultados do segundo dataset.

### 5.3 Análise de resultados

Apesar de só se terem apresentado resultados de dois modelos distintos, foram também testados mais alguns modelos como o Lasso e LightGBM. No entanto, obtiveram-se melhores resultados com os modelos Gradient Boosting e XGBoost.

Com o modelo Gradient Boosting obteve-se uma pontuação(*accuracy*) de 0.82666, enquanto que com o modelo XGBoost obtêve-se uma pontuação de 0.83111. Sendo assim, para obter o melhor destes dois modelos, foram combinadas as duas previsões numa só. Esta previsão foi por sua vez testada com a média do atributo no `teste.csv`, obtendo a melhor pontuação(*accuracy*) de 0.83555.

Com este procedimento, o grupo de trabalho acabou numa posição relativamente boa na competição do *Kaggle*.





#	Δpub	Team Name	Notebook	Team Members	Score ?	Entries	Last
13	▲ 13	GRUPO_MEI_29		   	0.79714	10	16d

Figura 5.3: Resultados da equipa de trabalho na competição do *Kaggle*.

## Capítulo 6

# Conclusão

Com a elaboração deste trabalho prático foi possível ter uma perspetiva perto de um contexto real de como os **datasets** têm de ser tratados de modo a produzirem bons resultados. Também observou-se como esse tratamento se trata de um dos passos mais importantes na área de Machine Learning.

Foi também possível consolidar várias formas de construção de modelos, assim como a importância da escolha do modelo a utilizar para um determinado dataset. Esta escolha juntamente com o tratamento correto do dataset pode melhorar consideravelmente a exatidão (*accuracy*) dos dados previstos.

Em suma, este trabalho prático foi crucial para consolidar e aplicar os vários tópicos e conceitos abordados e adquiridos na unidade curricular Dados e Aprendizagem Automática. Através de uma análise geral da solução implementada, podemos dizer que a equipa foi capaz de atingir todos os objetivos inicialmente estabelecidos.