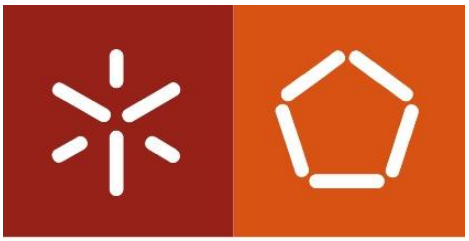

UNIVERSIDADE DO MINHO

DEPARTAMENTO DE INFORMÁTICA

Dados e Aprendizagem Automática

Duarte Oliveira `pg47157` Tiago Barata `pg47695`

Melânia Pereira `pg47520` António Guerra `pg47032`

3 de janeiro de 2022

Índice

1 Introdução

Este documento é o relatório de um trabalho de conceção e otimização de modelos de *Machine Learning*, realizado na UC de Dados e Aprendizagem Automática, do Mestrado em Engenharia Informática da Universidade do Minho.

Este trabalho pode ser dividido em duas partes:

- `a primeira` - tratamento, exploração e análise de dados e extração de conhecimento relevante no contexto do problema;
- `a segunda` - conceção e otimização de modelos de *Machine Learning* e obtenção, análise e interpretação de resultados.

Para a realização do trabalho foi fornecido ao grupo um *dataset* que contém dados referentes ao tráfego de veículos. Foi ainda indicado que deveria ser escolhido um outro *dataset*, para além do anterior, para ser trabalhado, que será indicado e detalhado no presente relatório.

O *dataset* fornecido pelos docentes foi trabalhado no contexto de uma competição na plataforma *kaggle* entre os vários grupos de trabalho inscritos na UC. Esta competição permitia a submissão de resultados obtidos e apresentava um *score* para os mesmos que correspondia à *accuracy* do modelo desenvolvido que os originou.

Nas próximas páginas do presente documento pode encontrar-se uma descrição da metodologia usada para a extração de conhecimento, assim como de ambos os *datasets* trabalhados e dos respetivos modelos desenvolvidos; pode ainda encontrar-se um capítulo referente à demonstração e análise dos resultados obtidos.

2 Objetivos

Os objetivos deste trabalho baseiam-se na prática exploração e extração de conhecimento de um conjunto de dados e de conceção de modelos de *machine learning*.

Para tal serão tratados **dois *datasets***, um relativo a tráfego automóvel e outro, escolhido pelo grupo, relativo a *preferências em tipos de pontos de interesse de um conjunto de utilizadores*.

Pretende-se, para o primeiro *dataset*, que seja desenvolvido um modelo de *machine learning* capaz de prever um atributo alvo, que no caso é o nível de trânsito para um determinado momento. Já para o segundo *dataset*, é pretendido que seja desenvolvido um modelo que possa ajudar num sistema de recomendação, para isso é necessária a criação de um conjunto de *clusters* que irão corresponder a diferentes grupos de utilizadores e, dependendo de em que *cluster* um utilizador se enquadrar, recomendações de acordo com as características desse *cluster* serão feitas.

Para conseguir alcançar estes objetivos, o grupo propõe-se a seguir uma metodologia de **extração de conhecimento**, fazendo a exploração e consequente tratamento dos dados de cada um dos *datasets* para passar ao desenvolvimento do modelo que mais se adeqúe a cada um dos casos, seguindo uma técnica iterativa que inclui **testes de *performance***, procura pelos melhores parâmetros para obter os melhores resultados e ajustes a decisões tomadas.

3 Metodologia

A metodologia seguida para a extração de conhecimento em ambos os *datasets* foi a `CRISP-DM` (CRoss Industry Standard Process for Data Mining).

Primeiro, a equipa começou por perceber os objetivos do projeto, que entendeu ser o treino de modelos de *machine learning* com determinados *datasets* de treino para preverem o comportamento de um certo atributo alvo.

Seguiu-se, então, a análise e exploração dos dados de cada *dataset* para perceber a qualidade dos mesmos e entender quais os ajustes e pré processamento que serão necessários realizar; nesta fase são feitas pesquisas por possíveis valores nulos, duplicados, *outliers* ou até valores possivelmente errados, com recurso a gráficos que representam a distribuição [1] dos valores e relação [2] entre eles, e ainda à função `unique` do *python* que devolve o conjunto de valores existentes numa determinada coluna de um *dataset*.

Esta análise serve de base para a seguinte fase, que é a preparação, ou pré processamento, do conjunto de dados, aqui é feita a seleção de atributos e a limpeza de dados; para o caso dos *datasets* tratados, foram eliminados um conjunto de atributos que se mostraram dispensáveis tanto na exploração dos dados como depois de realizados alguns testes com modelos de ML. Além desta eliminação, foi ainda usada *feature engineering* para extrair dados concretos da coluna `record_date` e criar novas colunas com informação da hora, dia, mês e dia de semana; e ainda usada discretização de valores nominais com o *Label Encoder* em alguns atributos cujos valores eram nominais para passar para valores discretos numéricos.

Depois de ter os dados preparados, o grupo passou à fase de modelação, onde foram experimentados vários tipos de modelos, e testada a *performance* de cada um para o *dataset* em causa.

No caso do *dataset* fornecido pelos docentes, tratando-se de um problema de classificação, passou-se pelos modelos *DecisionTreeClassifier*, *KNeighborsClassifier*, *RandomForestClassifier* e *XGBClassifier*, tendo ainda usado o auxílio de uma *grid search* para encontrar os melhores parâmetros para o modelo em causa.

No caso do *dataset* escolhido e tratando-se de um problema de *clustering*, o modelo testado foi o *KMeans*.

O grupo recorreu ainda às redes neuronais, não obtendo, no entanto, resultados razoáveis, e deparando-se com um problema que não conseguiu resolver: primeiramente seguiu-se o exemplo de construção de uma rede neuronal à imagem do que foi feito nas aulas práticas da disciplina.

No entanto, esse método não estaria correto, pois o `KerasRegressor` que estava a ser utilizado, prevê dados contínuos e não discretos, como o caso do problema apresentado. Após alguma pesquisa, decidiu-se experimentar o `KerasClassifier`, que também não obteve um resultado satisfazível pois ao fazer a previsão, o resultado era constante para os 1500 valores de saída. Tentou-se ainda voltar ao método de regressão utilizado inicialmente e fazer uma discretização dos valores obtidos. No entanto, este método também não foi satisfatório, pois o resultado ia ao encontro da classificação (1500 valores de saída constantes).

A fase que se segue é a de avaliação, aqui são feitas comparações dos resultados obtidos com os esperados, para tal foram usadas métricas de qualidade como a *accuracy* e a *confusion_matrix* e gráficos relativos aos *clusters* no caso do *dataset* escolhido pelo grupo.

Seguindo esta metodologia, o grupo foi sempre adaptando e melhorando as decisões, voltando a fases anteriores, como a *data preparation* ou *business understanding* para melhorar os resultados obtidos ao máximo possível.

4 Datasets

4.1 *Dataset* fornecido

O *dataset* fornecido pelos docentes desta unidade curricular contém dados referentes ao tráfego de veículos na cidade do Porto, ao longo de um período ligeiramente superior a um ano. Este *dataset* é composto por diferentes colunas com diversos atributos a si associados, os quais estão correlacionados com diversos fatores, entre os quais, características meteorológicas, variações associadas à velocidade e tempo do fluxo de trânsito e a data da recolha dos dados.

```
In [243]: training.columns
Out[243]:
Index(['city_name', 'record_date', 'AVERAGE_SPEED_DIFF',
       'AVERAGE_FREE_FLOW_SPEED', 'AVERAGE_TIME_DIFF',
       'AVERAGE_FREE_FLOW_TIME', 'LUMINOSITY', 'AVERAGE_TEMPERATURE',
       'AVERAGE_ATMOSP_PRESSURE', 'AVERAGE_HUMIDITY', 'AVERAGE_WIND_SPEED',
       'AVERAGE_CLOUDINESS', 'AVERAGE_PRECIPITATION', 'AVERAGE_RAIN'],
      dtype='object')
```

Figura 4.1: Lista de atributos do *dataset*

city_name	record_date	RAGE_SPEED	E_FREE_FLOW	RAGE_TIME_D	E_FREE_FLOW	LUMINOSITY	AGE_TEMPERA	E_ATMOSP_PR	RAGE_HUMID	AGE_WIND_S	/ERAGE_CLOUDINES	AGE_PRECIPIT.	AVERAGE_RAIN
Porto	2019-08-29 07:00:00	Medium	41.5	11.5	71.4	LIGHT	15	1019	100	3	nan	0	nan
Porto	2018-08-10 14:00:00	High	41.7	48.3	87.4	LIGHT	21	1021	53	5	céu claro	0	nan
Porto	2019-09-01 16:00:00	High	38.6	38.4	85.2	LIGHT	26	1014	61	4	nan	0	nan
Porto	2019-02-26 11:00:00	High	37.4	61	94.1	LIGHT	18	1025	48	4	céu claro	0	nan
Porto	2019-06-06 12:00:00	Medium	41.6	50.4	77	LIGHT	15	1008	82	10	nan	0	nan
Porto	2018-11-15 07:00:00	Medium	52.4	5.6	68.5	LOW_LIGHT	13	1014	72	4	nuvens dispersas	0	nan
Porto	2018-10-03 21:00:00	None	45.7	4	79.8	DARK	16	1020	58	0	céu claro	0	nan
Porto	2018-08-25 19:00:00	Low	40.9	11.8	87.8	LIGHT	19	1014	64	5	céu claro	0	nan
Porto	2019-06-30 12:00:00	Low	36.4	10.6	72.6	LIGHT	21	1019	82	5	céu pouco nublado	0	nan
Porto	2019-04-20 09:00:00	Low	34.8	10.1	84.4	LIGHT	19	1018	55	3	céu limpo	0	nan
Porto	2019-08-25 22:00:00	Medium	42.5	14.1	75.1	DARK	18	1015	100	1	nan	0	chuva fraca

Figura 4.2: Atributos e alguns objetos do *dataset*

Tendo em conta o objetivo deste *dataset*, foi necessário preparar o mesmo para desenvolver um modelo. Para isso teve que se fazer um prolífero tratamento de dados, de forma a reduzir ao máximo qualquer tipo de inconsistências que o *dataset* pudesse ter a si associado. De notar que as alterações a apresentar são as finais, fruto de um persistente e seletivo estudo dos atributos do *dataset*. Para tal foram usados diferentes tipo de funções auxiliares que, conforme apresentadas nas aulas práticas da disciplina, auxiliam bastante a perceber de que forma estão distribuídos os dados relativos a cada um dos atributos que compõem o *dataset* a ser estudado.

Assim, procedeu-se a efetuar diferentes ajustes, que se listam de seguida, para que houvesse a menor propensão a erros possível.

- **Remoção da coluna** `AVERAGE_PRECIPITATION`

```
In [231]: training['AVERAGE_PRECIPITATION'].unique()
Out[231]: array([0.])
```

Figura 4.3: Valores únicos do atributo *AVERAGE_PRECIPITATION*

Após se verificar os valores únicos desta coluna percebeu-se facilmente que seria extremamente difícil alterar ou prever os seus valores, que se encontram todos a **zero**. Por esse motivo e por se tratar de um atributo que não fornece informação relevante foi mais conveniente eliminar a coluna do *dataset* pois caso contrário estaríamos apenas a por em causa a *accuracy* do *dataset* em si.

- **Remoção da coluna** `city_name`

```
In [219]: training['city_name'].unique()
Out[219]: array(['Porto'], dtype=object)
```

Figura 4.4: Valores únicos do atributo *city_name*

Não havia real utilidade na existência desta coluna. Apenas indicava qual a cidade onde o estudo foi feito. Não havia forma de empregar este dado no desenvolvimento do modelo de *Machine Learning*.

- **Conversão e extração de conhecimento a partir do** `record_date`

```
In [220]: training['record_date'].unique()
Out[220]:
array(['2019-08-29 07:00:00', '2018-08-10 14:00:00',
       '2019-09-01 16:00:00', ..., '2018-10-02 04:00:00',
       '2019-01-30 01:00:00', '2019-06-15 21:00:00'], dtype=object)
```

Figura 4.5: Valores únicos do atributo *record_date*

A coluna *record_date* dispunha um formato de data *ano-mês-dia hora:minuto:segundo*. Nesta disposição, tal não nos é de grande utilidade. Assim, foi necessário converter este formato e a partir dele criar quatro novas colunas - *hour*, *day*, *month* , *day_name* - que trazem mais informação ao *dataset* e permitem aperfeiçoar o nosso modelo.

```
training.record_date = pd.to_datetime(training.record_date)
training['Hour'] = training.record_date.dt.hour
training['Day'] = training.record_date.dt.day
training['Month'] = training.record_date.dt.month
training['Day_Name'] = training.record_date.dt.day_name(locale='pt')
```

Figura 4.6: Extração de dados relevantes do atributo *record_date*

```
In [239]: training.Hour.unique()
Out[239]:
array([ 7, 14, 16, 11, 12, 21, 19,  9, 22,  6,  4,  5, 23, 18,  2, 10,  3,
        1, 13,  0,  8, 20, 17, 15])

In [240]: training.Day.unique()
Out[240]:
array([29, 10,  1, 26,  6, 15,  3, 25, 30, 20, 18, 27,  8,  9,  2, 17, 13,
       16,  7, 19, 12,  5, 24, 14, 11, 31, 21, 22, 28,  4, 23])

In [241]: training.Month.unique()
Out[241]: array([ 8,  9,  2,  6, 11, 10,  4,  7,  1, 12,  5,  3])
```

Figura 4.7: Valores únicos das novas colunas

- **Remoção das colunas** `AVERAGE_CLOUDINESS` **e** `AVERAGE_RAIN`

Apesar de serem colunas diferentes, ambos atributos induzem características e valores de tipos iguais, e para ambos definimos a mesma estratégia, sendo por isso que se aglomera num só tópico o tratamento de ambos.

Para melhor perceber o nosso raciocínio, é melhor ser definido todo o desenrolar e mutações se foram sofrendo.

Inicialmente, e devido a alguma inexperiência, após avaliarmos as colunas sentimos que seria acima de tudo necessário fazer relações de equivalência entre os diferentes dados, para cada coluna. Para isso começamos logo por verificar os valores únicos de cada uma das colunas.

```
In [30]: training['AVERAGE_CLOUDINESS'].unique()
Out[30]:
array([nan, 'céu claro', 'nuvens dispersas', 'céu pouco nublado',
       'céu limpo', 'algumas nuvens', 'nuvens quebrados', 'tempo nublado',
       'nuvens quebradas', 'nublado'], dtype=object)
```

```
In [31]: training['AVERAGE_RAIN'].unique()
Out[31]:
array([nan, 'chuva fraca', 'chuva', 'chuva leve', 'chuvisco fraco',
       'chuva moderada', 'trovoada com chuva leve', 'aguaceiros',
       'aguaceiros fracos', 'chuva de intensidade pesada',
       'trovoada com chuva', 'chuva de intensidade pesado', 'chuva forte',
       'chuvisco e chuva fraca'], dtype=object)
```

Figura 4.8: Valores únicos dos atributos *AVERAGE_CLOUDINESS* e *AVERAGE_RAIN*

```
AVERAGE_SPEED_DIFF          0
AVERAGE_FREE_FLOW_SPEED     0
AVERAGE_TIME_DIFF           0
AVERAGE_FREE_FLOW_TIME      0
LUMINOSITY                  0
AVERAGE_TEMPERATURE         0
AVERAGE_ATMOSP_PRESSURE     0
AVERAGE_HUMIDITY            0
AVERAGE_WIND_SPEED          0
AVERAGE_CLOUDINESS       2682
AVERAGE_RAIN             6249
```

Figura 4.9: Quantidade de valores nulos presentes nos vários atributos do *dataset*

Numa tentativa de colmatar esta lacuna, e após algum estudo e bastante afinco na aplicação de métodos de preenchimento/previsão de valores, essencialmente a partir de diferentes tipos

de *interpolação* (entre outros métodos) percebemos que a quantidade de valores em falta era extremamente significativa e que, portanto, a decisão mais coerente (e mais correta, dado que verificou-se que a diferença de valores de *accuracy* para o mesmo modelo de *ML* usado revelou-se significativamente mais acentuado), tendo em conta os interesses associados ao *dataset* seria a **eliminação de ambas as colunas**.

```
training.drop('AVERAGE_CLOUDINESS',axis=1, inplace=True)
test.drop('AVERAGE_CLOUDINESS',axis=1, inplace=True)

training.drop('AVERAGE_RAIN',axis=1,inplace=True)
test.drop('AVERAGE_RAIN',axis=1,inplace=True)
```

Figura 4.10: Eliminação dos atributos *AVERAGE_CLOUDINESS* e *AVERAGE_RAIN*

- **Conversão dos valores de `LUMINOSITY` e `Day_Name`**

```
training["Day_Name"] = LabelEncoder().fit_transform(training[["Day_Name"]])
test["Day_Name"] = LabelEncoder().fit_transform(test[["Day_Name"]])

training["LUMINOSITY"]  = LabelEncoder().fit_transform(training[["LUMINOSITY"]]
test["LUMINOSITY"] = LabelEncoder().fit_transform(test[["LUMINOSITY"]])
```

Figura 4.11: Aplicação da função *LabelEncoder*

De modo a valorizar o conhecimento associado aos valores das colunas *LUMINOSITY* e *Day_Name* (de notar que a última foi por nós gerada, correspondendo ao dia da semana) decidimos utilizar a função `LabelEncoder` permitindo converter estes valores qualitativos sem potencial matemático em valores de cariz numérico, com relevância para o desenvolvimento do modelo.

LUMINOSITY
LIGHT
LIGHT
LIGHT
LIGHT
LIGHT
LOW_LIGHT
DARK
LIGHT
LIGHT
LIGHT
DARK

Figura 4.12: Valores da coluna *LUMINOSITY*

Day_Name
Quinta-feira
Sexta-feira
Domingo
Terça-feira
Quinta-feira
Quinta-feira
Quarta-feira
Sábado
Domingo
Sábado
Domingo

Figura 4.13: Valores da coluna *DAY_NAME*

Finalmente, conforme as seguintes figuras ilustram, verifica-se e aprecia-se o potencial do *LabelEncoder* permitindo esta útil transformação dos valores, a serem aplicados então ao *dataset*

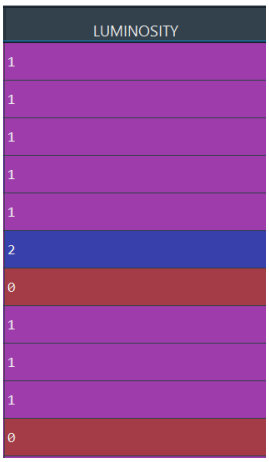

Figura 4.14: Valores da coluna *LUMINOSITY*

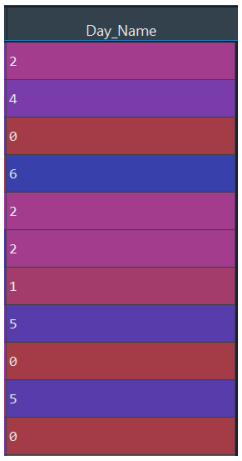

Figura 4.15: Valores da coluna *DAY_NAME*

Concluindo, é de notar que foi aplicado exatamente o mesmo tratamento ao *dataset* de treino e de teste. Será importante também referir que este terá sido o processo mais exaustivo de todo o projeto, visto que a qualidade dos modelos está diretamente correlacionada com a qualidade dos dados do *dataset* a ser usado. Consequentemente houve muitas abordagens que foram tomadas, especialmente em tentativa-erro, muitas das quais não merecem a menção, seja por nossa inexperiência ou por total desvio com o que se realmente pretendia para este projeto. Será relevante retirar que estes erros, etapa a etapa, nos trouxeram uma maior ambientação a como abordar o tratamento de dados.

4.2 *Dataset* escolhido

O segundo *dataset*, escolhido pelo grupo, foi retirado do *kaggle*, de um conjunto de *datasets* fornecidos na plataforma para prática de *machine learning* que podem ser encontrados em `https://www.kaggle.com/kkhandekar/all-datasets-for-practicing-ml?select=Clus`.

Do conjunto disponibilizado, o *dataset* escolhido foi o `Clus_BuddyMove.csv`, da categoria de *clustering*.

Decidiu-se por um *dataset* nesta categoria por se achar que obrigaria o grupo a seguir caminhos diferentes daqueles seguidos com o *dataset* fornecido pelos docentes e ainda por envolver algoritmos e um tratamento de dados e resultados algo diferente daquilo que foi estudado maioritariamente nas aulas desta UC.

Passa-se, então, à sua descrição.

Os dados presentes neste conjunto foram recolhidos de *reviews* de utilizadores publicadas em `holidayiq.com` sobre vários tipos de pontos de interesse no sul da Índia, que revelam os interesses de cada um dos utilizadores.

Os atributos do *dataset* são o *id* único do utilizador, seguido dos tipos de pontos de interesse

em causa, que são os seguintes:

- *Sports* - estádios, complexos desportivos, etc;
- *Religious* - instituições religiosas;
- *Nature* - praias, lagos, rios, etc;
- *Theatre* - teatros, exibições, etc;
- *Shopping* - centros comerciais, locais de compras, etc;
- *Picnic* - parques, locais de piquenique, etc.

```
In [247]: df.columns
Out[247]:
Index(['User Id', 'Sports', 'Religious', 'Nature', 'Theatre', 'Shopping',
       'Picnic'],
      dtype='object')
```

Figura 4.16: Lista de atributos do *dataset*

User Id	Sports	Religious	Nature	Theatre	Shopping	Picnic
User 1	2	77	79	69	68	95
User 2	2	62	76	76	69	68
User 3	2	50	97	87	50	75

Figura 4.17: Atributos e alguns objetos do *dataset*

O objetivo deste conjunto de dados é permitir o desenvolvimento de um modelo capaz de fazer recomendações de pontos de interesse a utilizadores. Para isso, é necessário, numa primeira fase, chegar a um conjunto de *clusters* ou segmentos que represente grupos de preferências dos utilizadores. Por exemplo, um utilizador que prefira pontos relacionados com desporto a pontos relacionados com *shopping* pertencerá a um grupo (*cluster*) diferente de um utilizador que prefira pontos de *shopping* a pontos de desporto.

Para desenvolver tais modelos, foi necessário, primeiramente, proceder à exploração e tratamento de dados.

Começou por se eliminar a coluna de identificação de utilizador visto que, sendo o identificado único, este não traz nenhuma informação relevante para a realização de recomendações ou para a divisão em segmentos de preferências.

```
X=df.drop('User Id', axis=1)
```

Figura 4.18: Eliminação do atributo 'User Id'

De seguida, foi feita uma procura por valores nulos no *dataset*.

```
In [252]: X.isnull().sum()
Out[252]:
Sports       0
Religious    0
Nature       0
Theatre      0
Shopping     0
Picnic       0
dtype: int64
```

Figura 4.19: Verificação da existência de valores nulos no *dataset*

Como se pode ver pela imagem, não existem valores nulos, por isso - após uma cuidada análise qualitativa aos valores de todas as colunas do *dataset* - decidiu-se manter todos os atributos.

Sendo que os algoritmos de *clustering* se servem de medidas de distância, nomeadamente a distância Euclediana, foi decidido que era necessário realizar um redimensionamento de todos os valores do *dataset* para não haver grandes discrepâncias no caso de valores muito mais distantes e de *outliers*. Para isso foi usado o `MinMaxScaler`

```
scaler = MinMaxScaler()
scaler.fit(X)
X=scaler.transform(X)
```

Figura 4.20: Redimensionamento dos valores do *dataset*

0	1	2	3	4	5
0	0.176471	0.101504	0.0649351	0.0983607	0.216561
0	0.0784314	0.0902256	0.11039	0.103825	0.044586
0	0	0.169173	0.181818	0	0.089172
0	0.117647	0.093985	0.233766	0.142077	0
0	0.313725	0.0075188	0	0.245902	0.159236

Figura 4.21: Objetos do *dataset* redimensionados

Depois destes ajustes, considera-se que os dados estão preparados para passar à fase de modelação.

5 Modelos

5.1 *Dataset* fornecido

Depois de efetuar toda a parte da extração do conhecimento e tratamento dos dados, chega a parte de escolher o modelo.

Desde logo se percebeu que o modelo teria que ser da categoria de classificação visto se tratar de uma problemática nesse domínio.

Começou-se, então, por testar modelos como o *DecisionTreeClassifier* passando-se depois a explorar outros modelos como o *RandomForest*. Esta exploração por diferentes modelos foi necessária devido à obtenção de resultados insatisfatórios de precisão nas previsões efetuadas. Algo que levou o grupo a explorar e experimentar oiutros modelos foi o facto de querer obter uma melhor classificação na competição do *kaggle*.

Depois de vários testes com vários modelos diferentes, chegou-se à conclusão que o melhor modelo seria o *XGBClassifier*, um algoritmo baseado no *GBClassifier* que, tal como o *RandomForest* e o *DecisionTree*, utiliza árvores de decisão, mas com uma diferença: usa uma melhor técnica de regularização para reduzir *overfitting*.

Procedeu-se à utilização de uma *Grid Search* para descobrir os melhores parâmetros para o modelo. Inicialmente usou-se uma *Grid Search* completa, mas, após uma pequena investigação, chegou-se à conclusão que uma *Random Search* teria uma melhor relação *accuracy* obtida/tempo gasto.

Depois deste *fine-tuning* dos hiperparâmetros chegou-se à conclusão de que os melhores valores para o modelo *XGBClassifier* são os seguintes:

- *n_estimators* = 120
- *learning_rate* = 0.1
- *criterion* = 'friedman_mse'
- *max_depth* = 5

5.2 *Dataset* escolhido

O *dataset* em questão foi tratado como *clustering*, isto é, agrupamento de dados. Usou-se o algoritmo *KMeans*, que consiste em agrupar os dados em torno de centros (chamados *centroids*),

criando segmentos de dados. A decisão do centro ao qual pertence cada elemento é calculado através da distância mínima, tendo em conta a média dos dados desse grupo.

Após se decidir a técnica a usar, fez-se o *tuning* do modelo, de modo a otimizar a sua utilização. Para tal, foi usada uma *Grid Search* que permite através da procura dos parâmetros previamente definidos, mostrar o resultado mais satisfatório (no caso em questão, chegou-se à conclusão que eram 3 *clusters*).

Foi, então, possível definir 3 grupos e, com a utilização de um algoritmo de classificação, desenvolver um modelo de previsão que permite saber a que grupo pertence determinado utilizador. Estes grupos servirão para a apresentação de sugestões que possam vir de encontro ao gosto de cada utilizador.

6 Resultados e Discussão

Ao longo deste projeto foram tomadas diferentes decisões, ideias e procedimentos. No final de todo o trabalho é sempre importante retirar ilações do produto final para que futuramente não se cometam os mesmos erros noutro tipo de projetos e para também avaliar qualitativamente a prestação do grupo.

Assim sendo, relativamente ao *dataset* dos docentes, a percentagem de *accuracy* revelou-se bastante boa, tendo a classificação final ficado sensivelmente no **top dez** de todas as submissões da competição do *kaggle*. O grupo teve um arranque lento, não só para perceber bem o que era pedido, mas na análise inicial dos dados, utilizando técnicas que no final foram desnecessárias. No final, o grupo já tinha um pré-processamento 100% definido e foi apenas a experimentação de modelos e o *tuning* dos mesmos que fez o bom resultado obtido.

O *dataset* de *clustering* consistiu em aplicar uma técnica utilizada na aula e testar os resultados. Após o *clustering*, foi feito um modelo de previsão, utilizando o `RandomForestClassifier`, o que possibilitou chegar à conclusão que o modelo definido através da aprendizagem não supervisionada era bastante bom. Dado isto, o grupo decidiu que a solução obtida tinha um resultado satisfatório.

Realça-se ainda a tentativa que houve de usar Redes Neuronais Artificiais, mas sem sucesso. Não se percebe o porquê de não terem funcionado nem obtido resultados agradáveis, no entanto perderam-se alguns dias de trabalho em pesquisa, *tuning* e experimentação de vários modelos.

7 Conclusão

Concluindo, é consensual que este projeto se revelou extremamente interessante para todos os elementos deste grupo.

A forma como foi necessário verificar a validade dos modelos que fomos desenvolvendo e a associação à competição criada no *kaggle* trouxe uma nova dinâmica à forma como abordamos o nosso projeto. Foi importante sentir um evoluir figurativo que paralelamente evoluía consoante o trabalho que íamos dedicando a este projeto. Isso trouxe muita motivação ao grupo de trabalho que findou com o **décimo primeiro** lugar na classificação geral da competição do *kaggle*.

Noutra variante, a liberdade dada para podermos escolher outro *dataset*, de nosso interesse, e de o trabalhar a nosso gosto foi também muito bem recebida por nós. A partir deste, pudemos simplesmente dedicar os nossos esforços a outro tipo de abordagem que contrasta com aquela a que nos tínhamos dedicado para o *dataset* dos docentes - escolhemos outro tipo de problemática, que requeria outro tipo de ferramentas, modelos e métodos de trabalho, de forma a distinguir o trabalho feito num *dataset* em relação ao outro. Foi, portanto, excelente, visto que pudemos aprofundar os nossos conhecimentos de uma forma mais ampla, ganhando bastante mais experiência nesta área.

Findando, sentimos que tivemos um desempenho exemplar neste trabalho prático, seguindo e servindo todas as diretivas e sugestões do enunciado prático, e esperamos que tal seja reconhecido pela equipa docente.

8 Anexos

1 Gráficos de distribuição dos dados

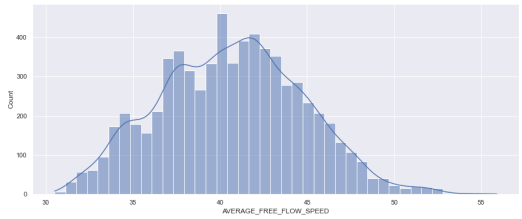

Figura 8.1: Histograma que define a densidade de valores do atributo *AVG_FREE_FLOW_SPEED*

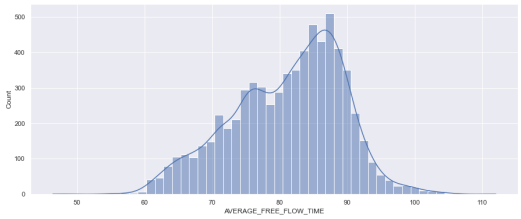

Figura 8.2: Histograma que contabiliza o número de valores do atributo *AVG_FREE_FLOW_TIME*

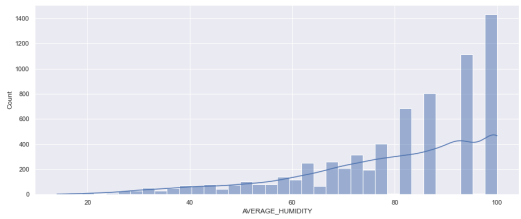

Figura 8.3: Histograma que contabiliza o número de valores do atributo *AVG_HUMIDITY*

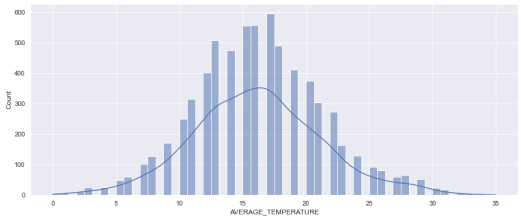

Figura 8.4: Histograma que contabiliza o número de valores do atributo *AVG_TEMPERATURE*

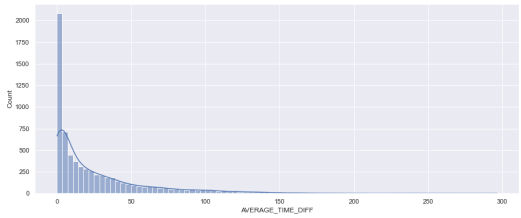

Figura 8.5: Histograma que contabiliza o número de valores do atributo *AVG_TIME_DIF*

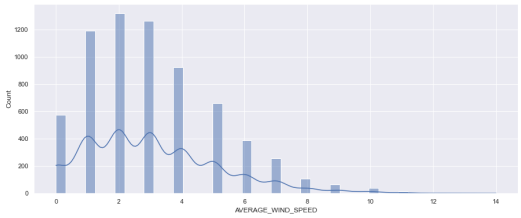

Figura 8.6: Histograma que contabiliza o número de valores do atributo *AVG_WIND_SPEED*

2 Gráficos de relação entre atributos do *dataset*

Desenvolvidos para melhor se compreender de que forma os diferentes atributos do *dataset* se correlacionam e de que forma essas relações podem influenciar o objetivo empregue a este projeto.

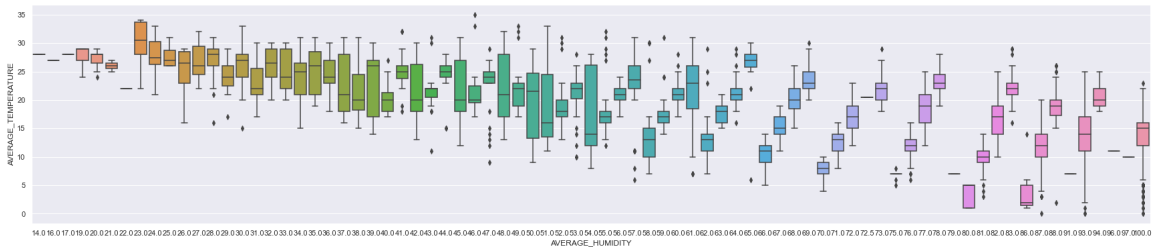

Figura 8.7: Relação entre os valores médios de temperatura e humidade.
Pode observar-se uma diminuição de temperatura com o aumento da humidade.

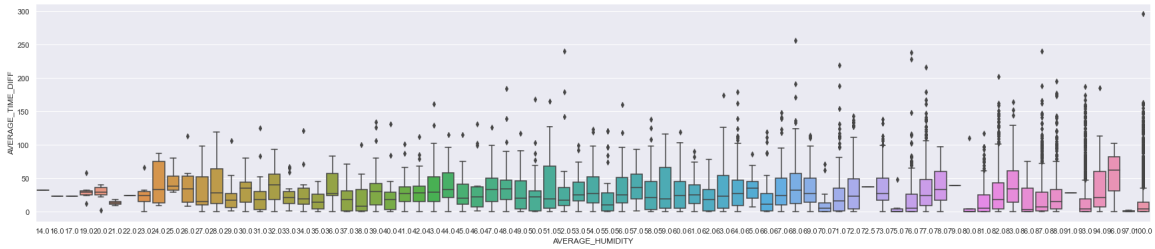

Figura 8.8: Relação entre os valores médios de humidade e a diferença entre o tempo que se demora a percorrer determinado conjunto de ruas e o tempo que se deveria demorar sem trânsito.
Pode aqui reparar-se que, na generalidade, não parece haver uma afetação do tempo pelas condições de humidade, no entanto, é de notar que há um aumento de casos *outliers*, com valores de diferença de tempo maiores, com o aumento da humidade.

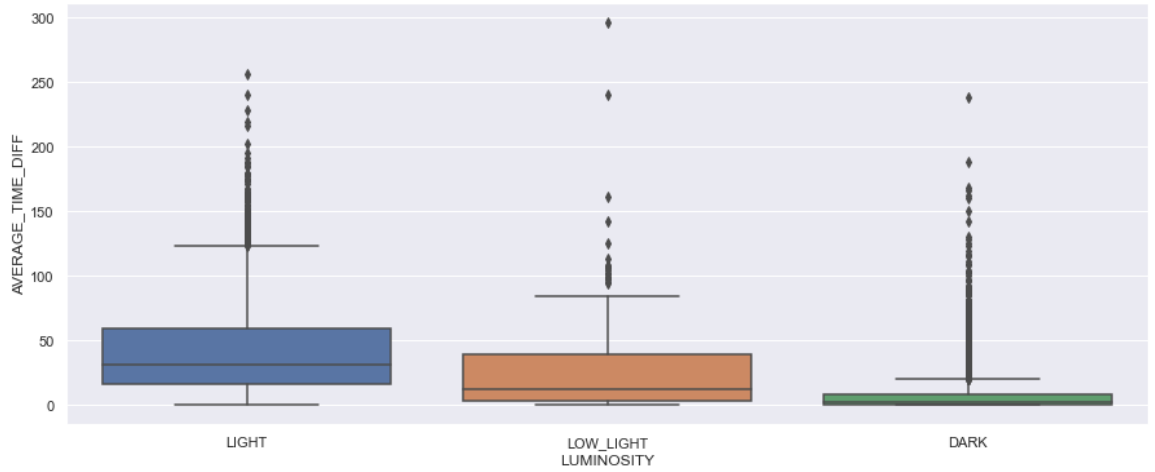

Figura 8.9: Denota-se uma diminuição dos valores do atributo *AVERAGE_TIME_DIFF* consoante a diminuição da luminosidade.

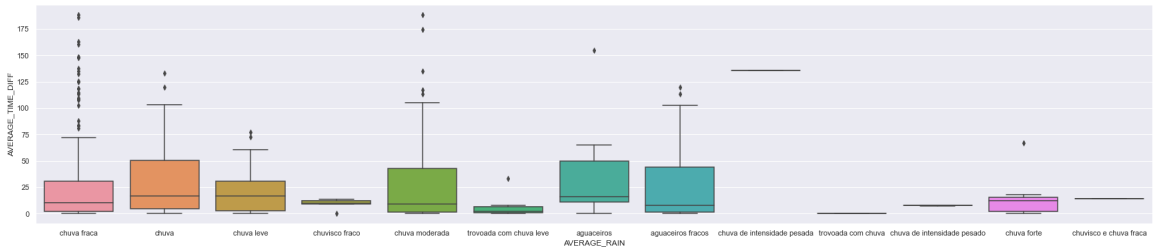

Figura 8.10: Relação dos valores da *AVERAGE_TIME_DIFF* com os diferentes tipos de precipitação.
Pode notar-se que em alguns tipos de precipitação, há uma menor e quase nula quantidade de casos, o que pode indicar que alguns dos tipos de precipitação presentes são dispensáveis do *dataset*.

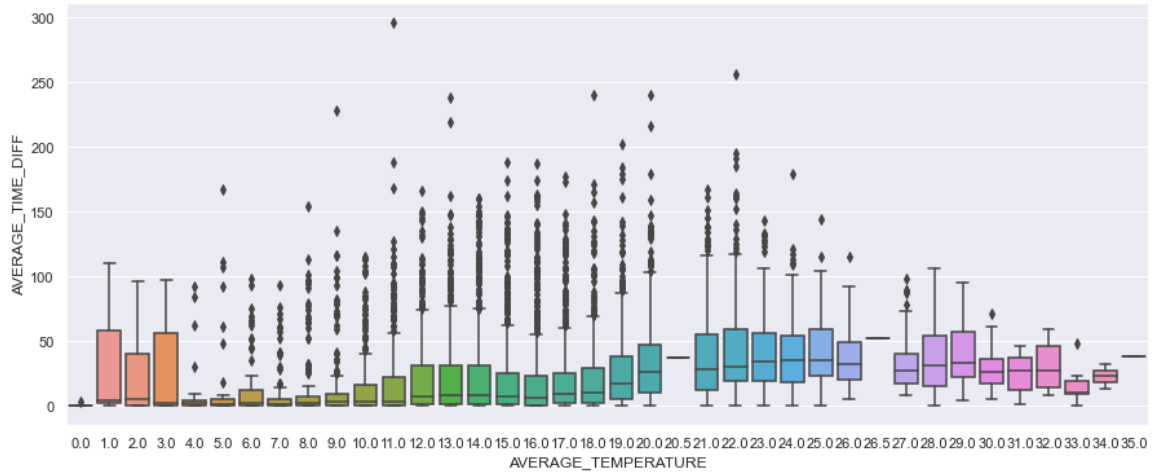

Figura 8.11: Relação entre os valores da *AVERAGE_TIME_DIFF* e a *AVERAGE_TEMPERATURE*.
É possível notar que com temperaturas amenas existem mais casos *outliers* com valores de diferença de tempo maiores, o que pode ser indicador de que com temperaturas amenas há maior trânsito.

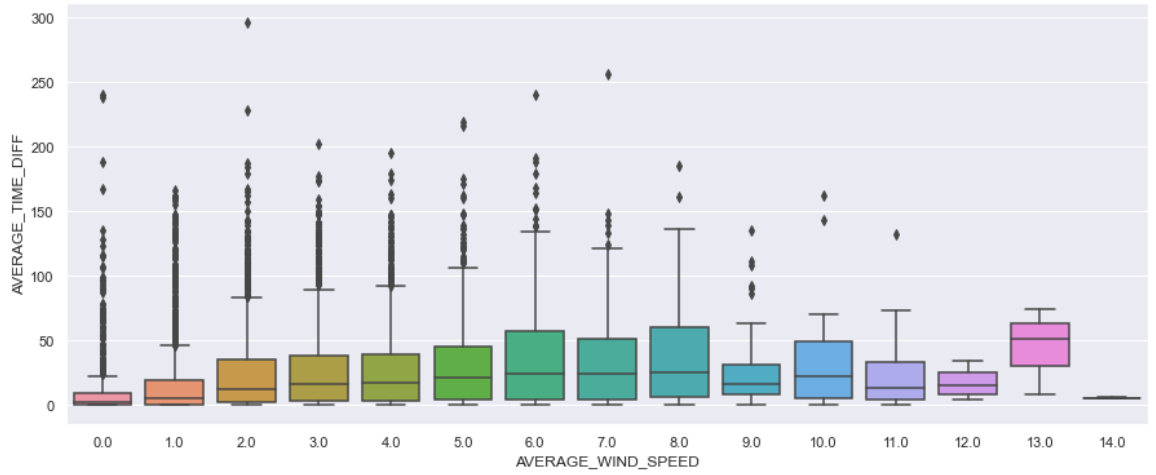

Figura 8.12: Relação dos valores da *AVERAGE_TIME_DIFF* com os valore da *AVERAGE_WIND_SPEED* .
É possível reparar que a velocidade do vento é um fator que, de alguma maneira, afeta a diferença de velocidades, o que, a princípio, estará relacionado com o trânsito.

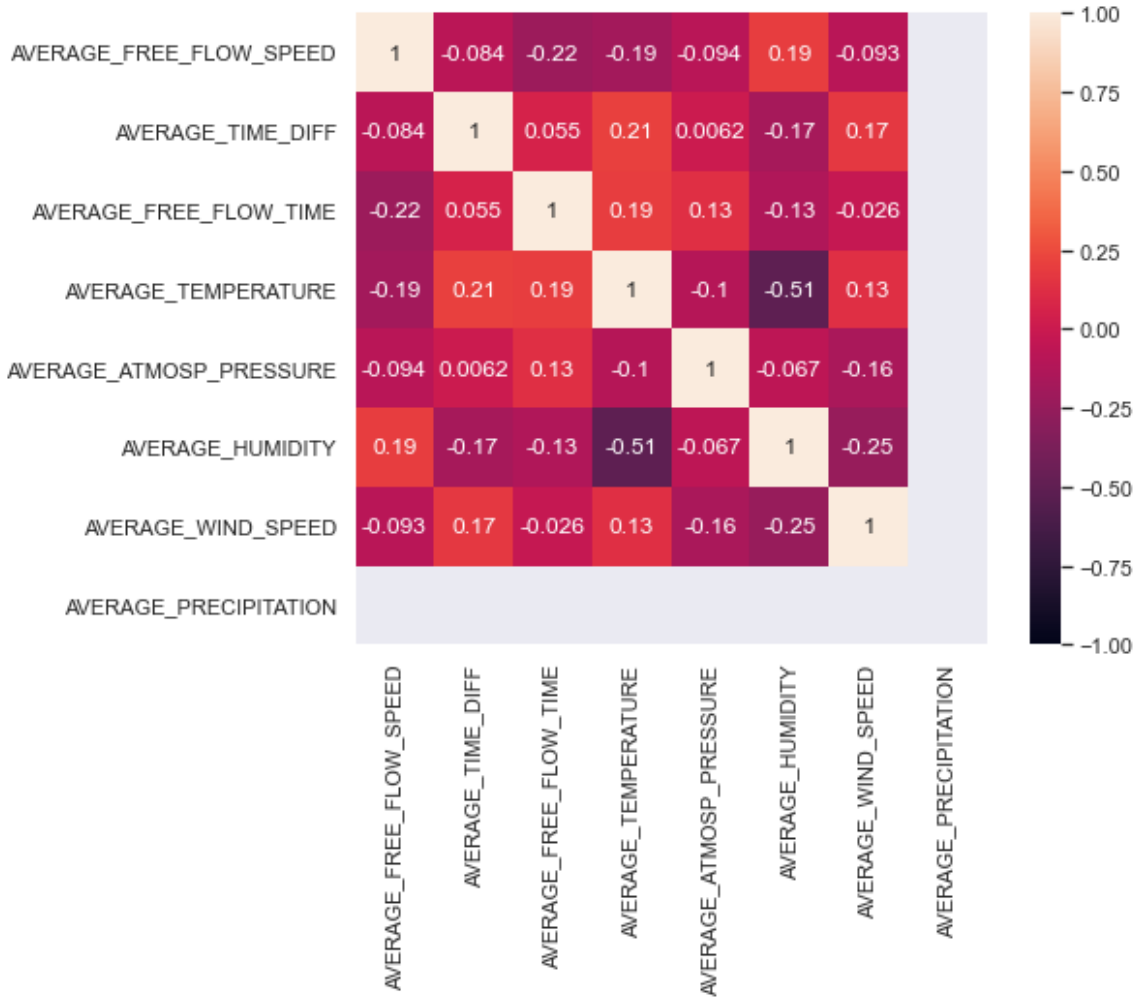

Figura 8.13: Matriz de correlação