Escola de Engenharia

Universidade do Minho

Trabalho Prático de Grupo Nº2

Realizado por:

Lucas Mello PG40158
Gonçalo Almeida A84610
Nuno Pereira PG42846
Benjamim Oliveira PG42815

No âmbito do Perfil
Machine Learning : Fundamentos e Aplicações
dos cursos:
MIEI/MEI/MMC

Unidade Curricular:
CSC (Classificadores e Sistemas Conexionistas)

23 de Maio, 2021

Conteúdo

1 Introdução

O presente trabalho prático foi desenvolvido no âmbito da unidade curricular "Classificadores e Sistemas Conexistas", do perfil "Machine Learning: Fundamentos e Aplicações", do conjunto de cursos MIEI/MEI/MMC, disponibilizados pela Universidade do Minho. De forma paralela foram explorados dois *datasets*, um fornecido pelos docentes da unidade curricular e o outro escolhido por nós. O nosso objetivo é desenvolver modelos de *Deep Learning* para séries temporais e realizar o respetivo *benchmarking* destes para abordagens como univariável e multivariável, single-step e multi-step, e encontrar o melhor conjunto de híper-parâmetros para cada modelo/abordagem.

2 Metodologia

De forma a garantir a qualidade dos modelos produzidos ao longo deste estudo, a metodologia *Cross Industry Standard Process for Data Mining*, ou CRISP-DM, foi empreendida no intuito de garantir inclusão de abordagens normalmente utilizadas pelos profissionais da área com vantagens imediatas na qualidade do projecto desenvolvido.

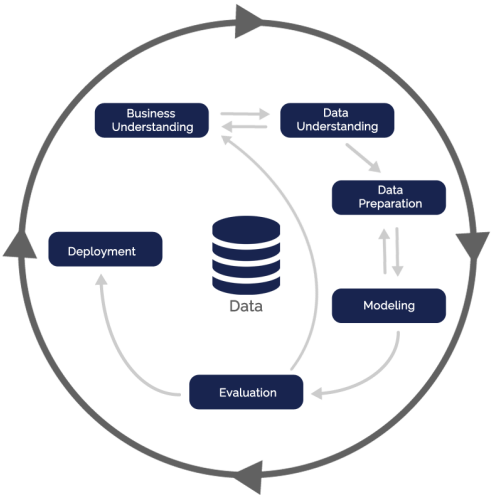


Figura 1: Diagrama ilustrativo da metodologia CRISP-DM.

A figura 1 ilustra os passos considerados por esta metodologia. De imediato, é possível perceber que esta metodologia não possui um início ou fim bem definido, propositadamente este comportamento é definido de forma a conseguir garantir que existe um refinamento do projecto ao longo do tempo, cada iteração contemplando das informações colectadas pela iteração anterior. Esta metodologia pode ser, resumidamente, descrita nos seguintes termos.

1. **Compreensão do Negócio** - É importante perceber qual é o domínio do negócio/tema que estamos a considerar, o que estamos a tentar prever e como utilizar o nosso conhecimento de domínio em nosso prol.

2. **Compreensão dos Dados** - Compreender o conjunto de dados captados, como os utilizar para alcançar o nosso objetivo e explorar estes de forma a descobrir fatores de interesse.

3. **Preparação dos Dados** - Perceber qual a melhor forma para processar os dados de tal forma a que os modelos desenvolvidos possam tirar mais proveito.

4. **Modelação** - Pesar os diferentes modelos utilizados, a diferenças entre si para alcançar o objetivo, bem como ponderar a variação dos hyper-parâmetros.

5. **Apreciação** - Comparar o modelo com os objetivos iniciais propostos e perceber de que forma o modelo permite descrever o desejado.

É importante ter em conta que a fase denotada como *Deployment* da figura 1 foi propositadamente deixada de parte, por não ter sido considerada em grande amplitude ao longo deste projeto.

3 Incidentes de Trânsito em Braga

3.1 Descrição do *Dataset*

O *dataset* fornecido denominado **Traffic_Incidents_Braga.csv** apresenta várias informações sobre incidentes de trânsito ocorridos na cidade de Braga durante o ano de 2019. Este apresenta, para 83347 observações, as 13 variáveis a seguir descritas.

city_name - cidade onde ocorreu o incidente;

description - descrição sobre o tipo de trânsito durante o incidente;

cause_of_incident - causa do incidente;

from_road - via rodoviária onde começou o incidente;

to_road - via rodoviária onde terminou o incidente;

affected_roads - vias rodoviárias afetadas pelo incidente;

incident_category_desc - categoria do incidente;

magnitude_of_delay_desc - magnitude do atraso provocado pelo incidente;

length_in_meters - comprimento do congestionamento de trânsito provocado pelo incidente;

delay_in_seconds - atraso provocado pelo incidente;

incident_date - data do incidente;

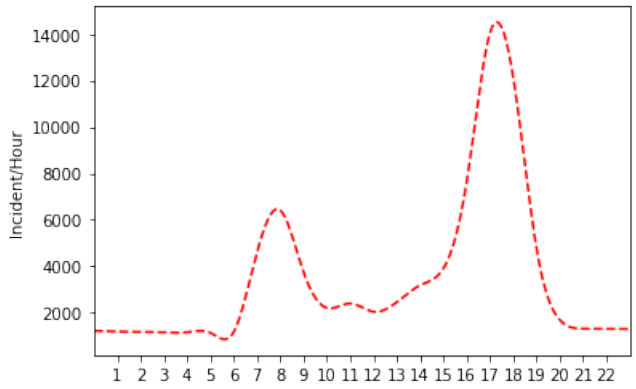
latitude - latitude do incidente;

longitude - longitude do incidente.

3.2 Análise Gráfica dos Dados

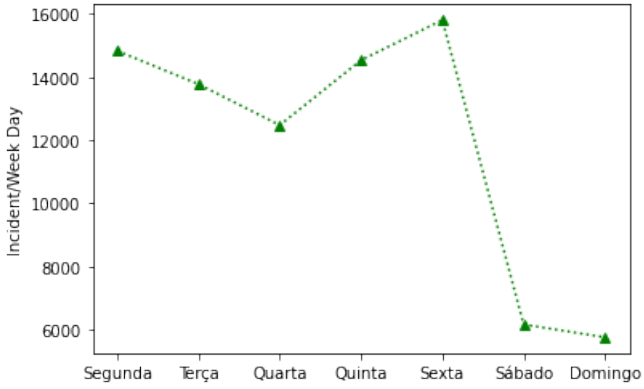
Antes de aplicar os tratamentos necessários para a construção do modelo de previsão é necessário conhecer melhor como os dados estão distribuídos durante o tempo, em particular os incidentes que é o foco deste trabalho.

Distribuição por Hora



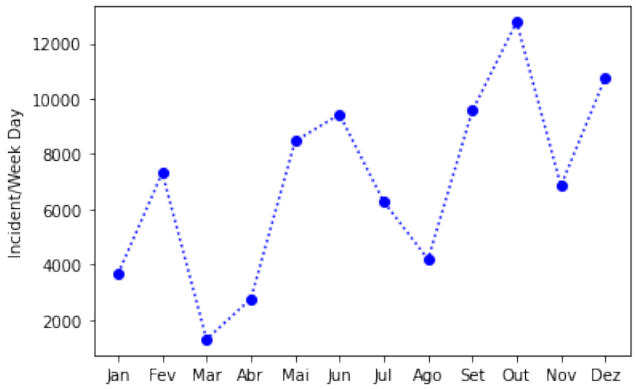
Quando os dados são exibidos em um gráfico de linha em uma abordagem por hora nota-se uma maior elevação em 2 pontos específicos, primeiro entre 6hr até as 9 hr e depois uma elevação ainda maior entre 14hr até as 19hr. E isso faz bastante sentido visto que os horários que ocorrem a primeira elevação na linha é quando as pessoas estão se dirigindo ao trabalho, e/ou levando seus filhos a escola, e na segunda elevação é geralmente o horário em que as pessoas estão voltando do trabalho e/ou buscando os filhos da escola, logo quando existe esse maior volume de trânsito é criado um ambiente mais favorável para ocorrer incidentes.

Distribuição por Semana



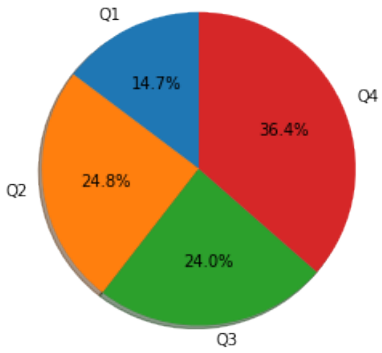
Na abordagem de distribuição de incidentes por semana nota-se que o maior índice de ocorrência de incidentes é durante a semana, de sexta feira para sábado a uma queda enorme no número de incidentes, uma vez que a maior parte das pessoas não trabalham e não precisam levar os filhos a escola durante o final de semana, sendo assim o fluxo de trânsito dentro da cidade diminui drasticamente.

Distribuição por Mês



Na distribuição de incidentes por mês é mais fácil de notar que existe uma sazonalidade no gráfico, geralmente o número de incidentes sobe no período de 3 meses e depois cai no próximo, e então o processo se repete novamente ao longo de todo o ano.

Distribuição por Trimestre



Para visualizar a distribuição de incidentes ao longo do ano foi construído um gráfico que captura a percentagem de incidentes por trimestre, cada fatia representa 3 meses. Podemos notar que em Q4 (Outubro, Novembro e Dezembro) é onde se concentra a maior percentagem de incidentes com mais de 36%, Q2 (Abril, Maio e Junho) e Q3(Julho, Agosto e Setembro) possuem uma distribuição parecida com uma média de 24%, enquanto que Q1(Janeiro, Fevereiro e Março) possui a menor percentagem de incidentes.

3.3 Tratamento dos Dados

Univariável e Multivariável

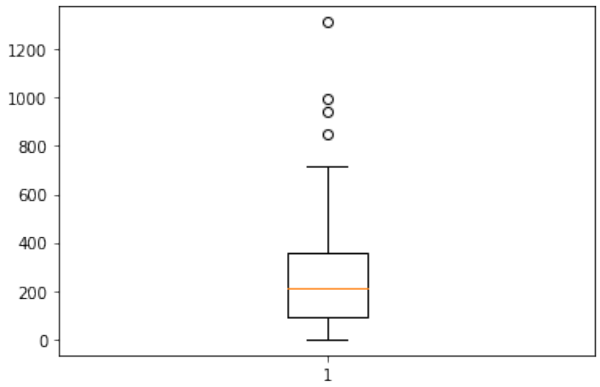
Para abordarmos o problema de forma univariável consideramos agrupar os incidentes de hora em hora ou diariamente contudo, com a primeira abordagem, a quantidade de dados em falta é consideravelmentre superior comparativamente à segunda abordagem. Agrupando então os incidentes diariamente, a variável única que consideramos foi o número de incidentes por dia. Por outro lado, para a abordagem multivariável, agrupamos novamente os dados diariamente mas consideramos, não só o número de incidentes por dia, mas também as variáveis *length_in_meters*, *latitude* e *longitude*, trabalhando assim com 4 variáveis.

Missing Values

De forma a tratar os valores em falta, isto é, dias em que não temos informações sobre nenhum acidente, consideramos duas abordagens, remover estas instâncias (método Dropout), associar o valor *default* -99 (método Masking) ou calcular a média entre os valores antes e depois dos valores em falta (método Interpolation).

Outliers

Na abordagem do trabalho onde se buscou realizar o modelo com base nos incidentes ocorridos diariamente houve poucas ocorrências de outliers como mostrado na figura abaixo. Foi decidido não remover os valores com outilers pois foi obtido melhores resultados em média sem a remoção:



Valor médio das métricas com outliers: loss: 0.1805 mae: 0.1043 rmse: 0.1906 Valor médio das métricas sem outlers: loss: 0.2762 mae: 0.1349 rmse: 0.2729

Normalização

Quando uma rede neuronal é alimentada com dados não escalados e a dimensão dos dados de *input* for suficientemente grande, os processos de aprendizagem e convergência podem ser afetados. Para evitar este problema normalizamos os dados, escalando os seus valores para o intervalo [-1, 1].

Não Supervisionado para Supervisionado

Para tornar o problema supervisionado de modo a que os modelos possam aprender a prever padrões do *output* dos dados de *input*, sendo este um problema de previsões em séries temporais, consideramos para cada sequência de dados temporais o número de incidentes do dia seguinte como *output*.

Conjuntos de Treino e de Teste

Para separar os dados em conjuntos de treino e de teste para estimar a performance dos modelos consideramos os primeiros 90% dos dados para treino e os restantes 10% para teste.

3.4 Modelos de *Deep Learning*

Com a fase de tratamento dos dados concluída e com os mesmos em condições de serem fornecidos aos modelos de Deep Learning, foi possível começar o desenvolvimento dos modelos. Foram desenvolvidos 5 modelos no total

3.4.1 LSTM

Long Short-Term Memory networks, ou LSTMs podem ser aplicadas em aplicações de time series.Existem diversos modelos de LSTM que podem ser usados para variados tipos de problemas relativo a time series. Neste trabalho foram utilizados dois modelos de arquitetura LSTM, o primeiro um modelo comum, e o segundo um modelo de LSTM Autoencoder que é uma implementação de autoencoder para dados de sequência utilizando arquitetura LSTM de codificador-decodificador.

3.4.2 Rede Neuronal Convolucional

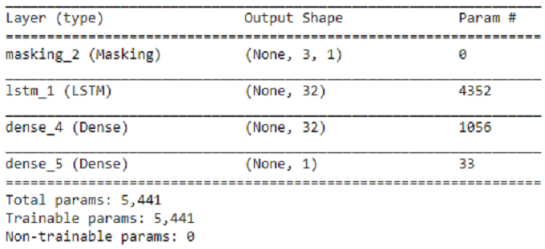
Finalmente, foi aplicada uma *Convolutional Neural Network* (CNN). Embora este tipo de rede seja tipicamente associado a dados como imagens ou modelos 3D (utilizando camadas convolucionais 2D ou 3D), podemos também utilizar este tipo de modelo para trabalhar com dados sequenciais a partir de camadas convolucionais de 1 dimensão (CNN unidimensional). A nossa arquitetura é composta por uma camada convolucional 1D e uma camada AveragePooling para reduzir a dimensão dos dados. Por fim temos uma camada Flatten seguida de duas camadas Dense. Na operação de *pooling* consideramos a abordagem *channels_first* que, em vez de reduzir os *timesteps*, reduz o número de *features*. Consideramos, também, os parâmetros *filters* = 16 (número de filtros de *output* na convolução), *kernel_size* = 5 (comprimento da janela de convolução unidimensional) e *pool_size* = 2 (tamanho da janela de de *max pooling*).

3.5 Modelos Univariantes

Foram treinados 5 modelos diferentes, os melhores parâmetros foram definidos através de um algoritmo de otimização (grid search) executado exaustivamente. O melhor resultado obtido em cada modelo é apresentado a seguir.

3.5.1 LSTM

- Neurónios: 32
- Timesteps: 3
- Batch Size: 10
- Epochs: 20
- Activation: tanh



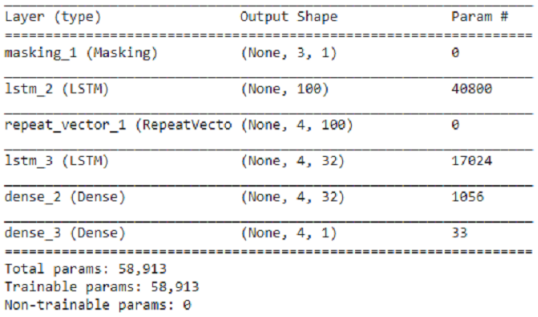
Layer (type)	Output Shape	Param #
masking_2 (Masking)	(None, 3, 1)	0
lstm_1 (LSTM)	(None, 32)	4352
dense_4 (Dense)	(None, 32)	1056
dense_5 (Dense)	(None, 1)	33

Total params: 5,441
Trainable params: 5,441
Non-trainable params: 0

LSTM Univariante optimizado:
LOSS de 0.2918 MAE de 0.2281 e **RMSE de 0.3065**

3.5.2 LSTM Autoencoder

- Neurónios: 32
- Timesteps: 3
- Batch Size: 2
- Epochs: 50
- Activation: tanh



Layer (type)	Output Shape	Param #
masking_1 (Masking)	(None, 3, 1)	0
lstm_2 (LSTM)	(None, 100)	40800
repeat_vector_1 (RepeatVecto	(None, 4, 100)	0
lstm_3 (LSTM)	(None, 4, 32)	17024
dense_2 (Dense)	(None, 4, 32)	1056
dense_3 (Dense)	(None, 4, 1)	33

Total params: 58,913
Trainable params: 58,913
Non-trainable params: 0

LSTM Autoencoder Univariante optimizado:
LOSS de 0.2174 MAE de 0.1360 e **RMSE de 0.2332**

3.5.3 GRU

- Neurónios: 32
- Timesteps: 3
- Batch Size: 10
- Epochs: 40
- Activation: relu

Layer (type)	Output Shape	Param #
masking_2 (Masking)	(None, 3, 1)	0
gru (GRU)	(None, 32)	3360
dense_4 (Dense)	(None, 32)	1056
dense_5 (Dense)	(None, 1)	33

Total params: 4,449
Trainable params: 4,449
Non-trainable params: 0

GRU optimizado:
LOSS de 0.2923 MAE de 0.2253 e **RMSE de 0.3072**

3.5.4 GRU Autoencoder

- Neurónios: 64
- Timesteps: 3
- Batch Size: 10
- Epochs: 50
- Activation: tanh

Layer (type)	Output Shape	Param #
masking (Masking)	(None, 3, 1)	0
gru (GRU)	(None, 100)	30900
repeat_vector (RepeatVector)	(None, 4, 100)	0
gru_1 (GRU)	(None, 4, 64)	31872
dense (Dense)	(None, 4, 64)	4160
dense_1 (Dense)	(None, 4, 1)	65

Total params: 66,997
Trainable params: 66,997
Non-trainable params: 0

GRU Autoencoder optimizado:
LOSS de 0.1805 MAE de0.1043 e **RMSE de 0.1906**

3.5.5 CNN

- Timesteps: 5
- Epochs: 50
- Batch Size: 10
- Missing Values: Masking

Layer (type)	Output Shape	Param #
input_285 (InputLayer)	[(None, 5, 1)]	0
conv1d_284 (Conv1D)	(None, 1, 16)	96
average_pooling1d_284 (Avera	(None, 1, 8)	0
flatten_284 (Flatten)	(None, 8)	0
dense_568 (Dense)	(None, 16)	144
dense_569 (Dense)	(None, 1)	17

Total params: 257
Trainable params: 257
Non-trainable params: 0

CNN Univariante optimizado:
MAE de 0.2308 e **RMSE de 0.3183**

3.6 Modelos Multivariantes

O processo de preparação de dados para os modelos multivariantes foi bastante similar ao processo univariante, nesta abordagem utilizamos 3 features a mais (length in meters, latitude e longitude).

3.6.1 LSTM

- Neurónios: 32
- Timesteps: 3
- Batch Size: 10
- Epochs: 20
- Activation: tanh

Layer (type)	Output Shape	Param #
masking_1 (Masking)	(None, 3, 4)	0
lstm (LSTM)	(None, 32)	4736
dense_2 (Dense)	(None, 32)	1056
dense_3 (Dense)	(None, 1)	33

Total params: 5,825
Trainable params: 5,825
Non-trainable params: 0

LSTM Multivariante optimizado:
LOSS de 0.2889 MAE de 0.2182 e **RMSE de 0.3034**

3.6.2 LSTM Autoencoder

- Neurónios: 32
- Timesteps: 3
- Batch Size: 2
- Epochs: 50
- Activation: tanh

Layer (type)	Output Shape	Param #
masking (Masking)	(None, 3, 4)	0
lstm (LSTM)	(None, 100)	42000
repeat_vector (RepeatVector)	(None, 4, 100)	0
lstm_1 (LSTM)	(None, 4, 32)	17024
dense (Dense)	(None, 4, 32)	1056
dense_1 (Dense)	(None, 4, 1)	33

Total params: 60,113
Trainable params: 60,113
Non-trainable params: 0

LSTM Autoencoder Multivariante optimizado:
LOSS de 0.1971 MAE de 0.1201 e **RMSE de 0.2091**

3.6.3 GRU

- Neurónios: 32
- Timesteps: 3
- Batch Size: 10
- Epochs: 40
- Activation: relu

Layer (type)	Output Shape	Param #
masking (Masking)	(None, 3, 4)	0
gru (GRU)	(None, 32)	3648
dense (Dense)	(None, 32)	1056
dense_1 (Dense)	(None, 1)	33

Total params: 4,737
Trainable params: 4,737
Non-trainable params: 0

GRU Multivariante optimizado:
LOSS de 0.2906 MAE de 0.2164 e **RMSE de 0.3051**

3.6.4 GRU Autoencoder

- Neurónios: 64
- Timesteps: 3
- Batch Size: 10
- Epochs: 50
- Activation: tanh

Layer (type)	Output Shape	Param #
masking (Masking)	(None, 3, 4)	0
gru (GRU)	(None, 100)	31800
repeat_vector (RepeatVector)	(None, 4, 100)	0
gru_1 (GRU)	(None, 4, 64)	31872
dense (Dense)	(None, 4, 64)	4160
dense_1 (Dense)	(None, 4, 1)	65

Total params: 67,897
Trainable params: 67,897
Non-trainable params: 0

GRU Autoencoder Multivariante optimizado:
LOSS de 0.2483 MAE de 0.1869 e **RMSE de 0.2620**

3.6.5 CNN

- Timesteps: 5
- Epochs: 50
- Batch Size: 7
- Missing Values: Interpolation

Layer (type)	Output Shape	Param #
input_284 (InputLayer)	[(None, 5, 4)]	0
conv1d_283 (Conv1D)	(None, 1, 16)	336
average_pooling1d_283 (Avera	(None, 1, 8)	0
flatten_283 (Flatten)	(None, 8)	0
dense_566 (Dense)	(None, 16)	144
dense_567 (Dense)	(None, 1)	17

Total params: 497
Trainable params: 497
Non-trainable params: 0

CNN Multivariante optimizado:
MAE de 0.2314 e **RMSE de 0.3176**

3.7 Previsão Multistep

Após treinar os modelos anteriormente apresentados passamos à implementação de previsões multi-step de forma recursiva. Para esta etapa usamos o modelos CNN univariante para fazer a previsão dos 4 e 50 registos sequintes. Os resultados podem ser vistos nas imagens abaixo, a verde as medições reais e a vermelho a previsão feita pelo nosso modelo.

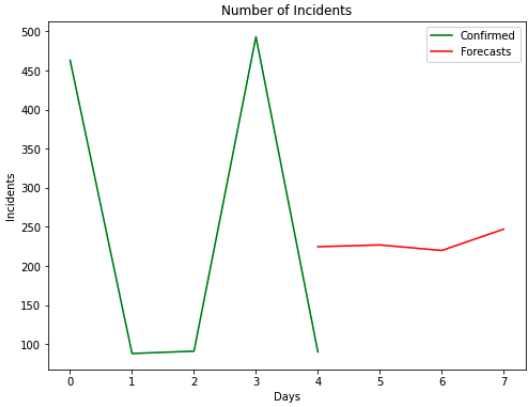


Figura 2: Previsão para 4 registos

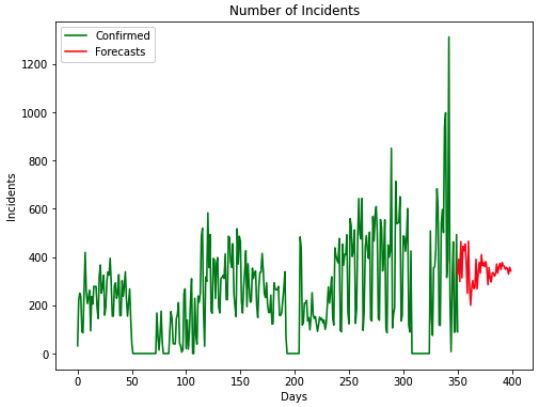


Figura 3: Previsão para 50 registos

4 Volume de passageiros - Metro Minneapolis

4.1 Descrição do Dataset

Para o desenvolvimento do presente ponto do trabalho prático foi necessário encontrar um dataset adequado. Após alguma pesquisa optou-se por utilizar um dataset relativo ao fluxo de passageiros numa linha de metro do estado de Minneapolis, nos Estados Unidos da América. Este dataset foi recolhido entre 2012 e 2018, contendo, antes do processamento e correção dos dados, cerca de 48000 registos, recolhidos de hora em hora.

O conjunto de dados possui, para cada registo efetuado, algumas informações pertinentes. Para cada registo possuimos as seguintes informações

- **holiday**, informa se o registo foi recolhido num feriado
- **temp**, indica a temperatura do momento do registo, em graus Kelvin
- **rain_1h**, precipitação, em mm, ao longo da hora
- **snow_1h**, precipitação(neve), em mm, ao longo da hora
- **clouds_all**, percentagem do céu coberta de nuvens
- **weather_main**, descrição da meteorologia no momento
- **datetime**, data e hora do registo
- **traffic_volume**, número de passageiros

	holiday	temp	rain_1h	snow_1h	clouds_all	weather_main	date_time	traffic_volume
0	None	288.28	0.0	0.0	40	Clouds	2012-10-02 09:00:00	5545
1	None	289.36	0.0	0.0	75	Clouds	2012-10-02 10:00:00	4516
2	None	289.58	0.0	0.0	90	Clouds	2012-10-02 11:00:00	4767
3	None	290.13	0.0	0.0	90	Clouds	2012-10-02 12:00:00	5026
4	None	291.14	0.0	0.0	75	Clouds	2012-10-02 13:00:00	4918

Figura 4: Descrição dos dados.

Após uma análise inicial feita ao dataset conseguimos perceber quais os principais estados meteorológicos encontrados no estado de Minneapolis, prendendo-se entre "nuvens", "limpo", "chuva"e "neblina". Com uma taxa de ocorrência muito mais baixa encontramos situações meteorológicas como "nevoeiro", "trovoada"ou até "neve".

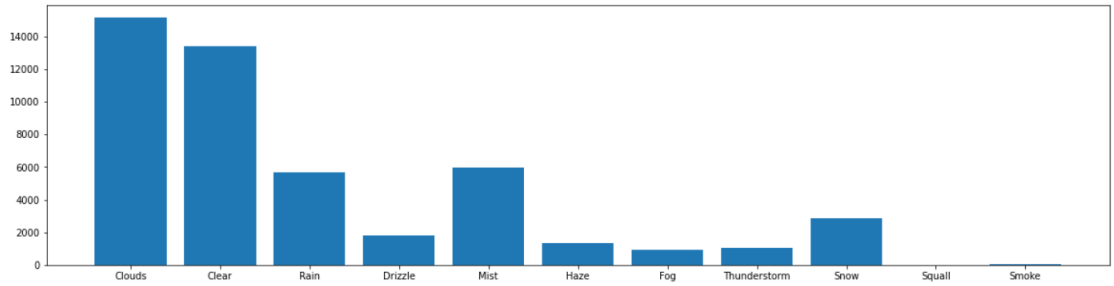


Figura 5: Dados meteorológicos.

Estudamos também a distribuição de passageiros no sistema de Metro ao longo das diversas horas do dia. Registam-se dois picos principais, o primeiro por volta das 7h da manhã e o segundo por volta das 16h, coincidindo com os períodos de começo e término de uma jornada laboral comum. Conseguimos perceber também que o fluxo de passageiros, apesar dos picos, nunca desce tanto durante o dia como durante a noite.

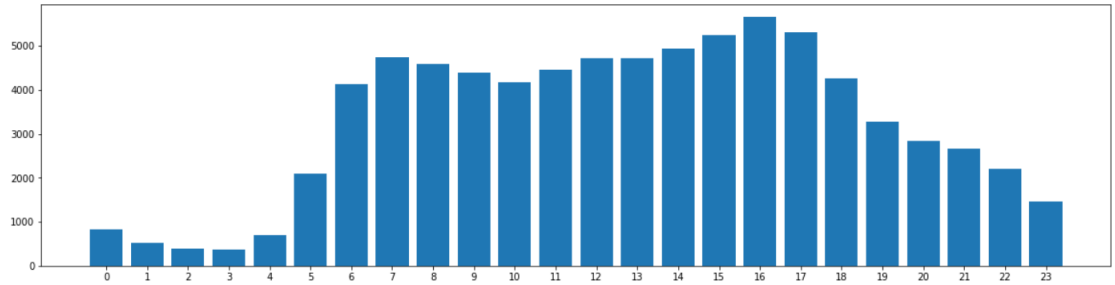


Figura 6: Fluxo por hora.

Num ponto final da compreensão do nosso dataset tentamos compreender se estavamos na presença de outliers nos nossos dados. Para isso obtivemos o valor minímo, médio e máximo para três variáveis, temperatura, chuva e neve.

Temperatura

- Minimo - 0K (-273.15ºC)
- Média - 281K (7.85ºC)
- Máximo - 310K (36.85ºC)

Chuva

- Minimo - 0 mm/h
- Média - 0.33 mm/h
- Máximo - 9831 mm/h

Neve

- Minimo - 0 mm/h
- Média - 0 mm/h
- Máximo - 0.51 mm/h

```
Dados de Temperatura: Min:0.00K Max:310.07K Média:281.21K
Dados de Chuva: Min:0.00 mm/h Max:9831.30 mm/h Média:0.33 mm/h
Dados de Neve: Min:0.00 mm/h Max:0.51 mm/h Média:0.00 mm/h
```

Figura 7: Identificação de outliers.

4.2 Tratamento de dados

4.2.1 Correção de Outliers

Após a analise feita no ponto anterior compreendemos a existência de dois tipos de outliers. O primeiro relativo a inserções incorretas (ou nulas) de temperatura e o segundo relativo a inserções defeituosas dos valores de precipitação. Corrigimos todos os registos com uma temperatura de 0^{o}K com o valor médio da temperatura. Quanto á chuva, os registos que tinham uma precipitação de mais de 100 mm/h foram corrigidos também para o valor médio de precipitação.

4.2.2 Registos repetidos

Algo que também foi possível observar durante a etapa de compreensão dos dados foi a existência de registos duplicados, ou até triplicados. Consideramos como sendo necessário (e boa prática) não utilizarmos dados repetidos, para tal procedemos á limpeza do dataset. Antes deste ponto o nosso conjunto de dados possuia 48204 registos, após esta limpeza ficamos com apenas 40575 registos. 16% dos nossos registos iniciais eram duplicados.

4.2.3 Preencher lacunas nos dados

Não só identificamos dados repetidos como também identificamos lacunas nos dados, alguns conjuntos de horas sem inserção de dados. Para tratarmos o conjunto de dados como uma série temporal é absolutamente necessário que os dados respeitem uma ordem temporal de intervalo igual, o que nos levou a procurar uma solução para preencher estas lacunas.

O método desenvolvido para tal feito passa por identificar uma lacuna e preencher a mesma com um registo constituido pela média calculada entre o registo anterior e o registo seguinte. Com este processo, uma vez preenchidas todas as lacunas, elevamos o nosso dataset a um total de 44454 registos. Aproximadamente 9% do nosso dataset foi então composto de dados calculados, não recolhidos, algo que poderia comprometer os resultados dos nossos modelos.

4.2.4 Feriados

Identificamos uma questão algo caricata relativamente aos feriados. Cada feriado era registado apenas na primeira hora do dia. Ou seja, num feriado, a inserção das 0h seria registada como feriado, no entanto a inserção seguinte, há 1h da manhã não seria registada como tal. Tivemos esta questão em consideração, propagando o estado de "feriado"ás restantes horas do registo, se fosse o caso de esse registo ser realmente um feriado.

4.2.5 Fins de semana

Por fim aplicamos Feature Engeneering para extrairmos mais informação sobre os dados que já tinhamos. Para cada registo verificamos se o mesmo pertencia ou não a um fim-de-semana, atualizado o novo campo de dados inserido de acordo com esta informação.

4.3 Modelos trabalhados

Para desenvolvermos modelos de predição a partir de dados de séries temporais tivemos que implementar algumas arquiteturas de redes neuronais. Estas arquiteturas foram desenvolvidas paralelamente para modelos univariantes e multivariantes, desenvolvendo e adaptando assim cada uma delas para treinar e prever a partir dos dois tipos séries temporais consideradas.

Certos parâmetros referentes ao treino/constituição das redes foram definidos através de um algoritmo de optimização (grid search), assim sendo a apresentação das redes será feita de uma forma mais geral, uma vez que certos parâmetros internos diferem dentro da própria arquitetura (parâmetros óptimos para a versão univariante diferentes de parâmetros óptimos para a versão multivariante)

4.3.1 LSTM

LSTM, ou Long Short Term Memory é um tipo de rede neuronal, mais concretamente uma rede neuronal recorrente. É capaz de processar sequências de informação, indo assim de encontro ao que procuramos com este trabalho. A arquitetura é composta por uma camada LSTM seguida de duas camadas Dense, para obtermos assim um output/previsão.

4.3.2 GRU

GRU, ou Gated Recurrent Unit é um tipo de rede neural recorrente um tanto similar ás LSTMs, possuindo no entanto um "forget gate", responsável por decidir que informações devem ser passadas para o output, e que informações devem ser esquecidas. A nossa implementação deste modelo, um pouco á semelhança do que foi feito com LSTMs, possui uma camada GRU seguida de duas camadas Dense.

4.3.3 CNN

Finalmente temos então a nossa CNN, ou Convolutional Neural Network. O modelo desenvolvido para este ponto foi desenvolvido de modo similar ao usado na primeira parte deste trabalho, fazendo uso então de camadas convolucionais 1D.

4.4 Modelos Univariantes

Antes de podermos considerar o nosso conjunto de dados como uma série temporal adequada temos que o transformar num conjunto de dados supervisionado, de modo a que o modelo possa ter acesso aos n registos anteriores á previsão que desejamos fazer (n sendo o número de timesteps).

Para optimizarmos os modelos implementamos um algoritmo de grid search, iterando sobre os seguintes parâmetros: **neurónios, timesteps, batch size e learning rate**. Os modelos apresentados abaixo são a versão optimizada dos mesmos

Como métricas para compararmos os desempenhos dos diferentes modelos foram utilizadas MAE (Mean Average Error) e RMSE (Root Mean Square Error), dando no entanto um maior peso a RMSE, uma vez que considera o erro ao quadrado, dando maior ênfase a erros maiores.

4.4.1 LSTM

- Neurónios: 32
- Timesteps: 12
- Batch Size: 64
- Learning Rate: 0.01

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
lstm_81 (LSTM)               (None, 32)                4352
_________________________________________________________________
dense_162 (Dense)            (None, 32)                1056
_________________________________________________________________
dense_163 (Dense)            (None, 1)                 33
=================================================================
Total params: 5,441
Trainable params: 5,441
Non-trainable params: 0
```

LSTM Univariante optimizado:
MAE de 0.0335 e **RMSE de 0.0444**

4.4.2 GRU

- Neurónios: 32
- Timesteps: 12
- Batch Size: 64
- Learning Rate: 0.01

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
gru_36 (GRU)                 (None, 32)                3360
_________________________________________________________________
dense_72 (Dense)             (None, 32)                1056
_________________________________________________________________
dense_73 (Dense)             (None, 1)                 33
=================================================================
Total params: 4,449
Trainable params: 4,449
Non-trainable params: 0
```

GRU Univariante optimizado:
MAE de 0.0291 e **RMSE de 0.0401**

4.4.3 CNN

Parâmetros:

- Timesteps: 24
- Batch Size: 64
- Learning Rate: 0.001

Layer (type)	Output Shape	Param #
input_13 (InputLayer)	[(None, 24, 1)]	0
conv1d_12 (Conv1D)	(None, 20, 16)	96
average_pooling1d_12 (Averag	(None, 20, 8)	0
flatten_12 (Flatten)	(None, 160)	0
dense_24 (Dense)	(None, 16)	2576
dense_25 (Dense)	(None, 1)	17

Total params: 2,689
Trainable params: 2,689
Non-trainable params: 0

CNN Univariante optimizado:
MAE de 0.0335 e **RMSE de 0.0463**

4.5 Modelos Multivariantes

O processo de preparação dos dados para os modelos multivariantes foi bastante similar ao processo usado para os modelos univariantes, passando por transformar os mesmos em dados supervisionados para o treino. Da mesma forma os modelos foram optimizados com recurso a grid search.

4.5.1 LSTM

Parâmetros:

- Neurónios: 16
- Timesteps: 24
- Batch Size: 64
- Learning Rate: 0.01

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
lstm_81 (LSTM)               (None, 16)                1408
_________________________________________________________________
dense_162 (Dense)            (None, 16)                272
_________________________________________________________________
dense_163 (Dense)            (None, 1)                 17
=================================================================
Total params: 1,697
Trainable params: 1,697
Non-trainable params: 0
```

CNN Univariante optimizado:
MAE de 0.0344 e **RMSE de 0.0450**

4.5.2 GRU

Parâmetros:

- Neurónios: 64
- Timesteps: 12
- Batch Size: 64
- Learning Rate: 0.01

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
gru_81 (GRU)                 (None, 64)                13632
_________________________________________________________________
dense_326 (Dense)            (None, 64)                4160
_________________________________________________________________
dense_327 (Dense)            (None, 1)                 65
=================================================================
Total params: 17,857
Trainable params: 17,857
Non-trainable params: 0
```

CNN Univariante optimizado:
MAE de 0.0652 e **RMSE de 0.0777**

4.5.3 CNN

Parâmetros:

- Timesteps: 24
- Batch Size: 16
- Learning Rate: 0.001

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_2 (InputLayer)         [(None, 24, 5)]           0
_________________________________________________________________
conv1d_1 (Conv1D)            (None, 20, 16)            416
_________________________________________________________________
average_pooling1d_1 (Average (None, 20, 8)             0
_________________________________________________________________
flatten_1 (Flatten)          (None, 160)               0
_________________________________________________________________
dense_2 (Dense)              (None, 16)                2576
_________________________________________________________________
dense_3 (Dense)              (None, 1)                 17
=================================================================
Total params: 3,009
Trainable params: 3,009
Non-trainable params: 0
```

CNN Univariante optimizado:
MAE de 0.0543 e **RMSE de 0.0704**

4.6 Previsão Multistep

Após treinarmos os modelos previamente apresentados passamos então á implementação de previsões multistep, de forma recursiva.

Para esta etapa usamos o nosso melhor modelo, GRU univariante. Optamos por fazer uma previsão dos 3 registos seguintes a partir dos 12 registos anteriores. Os resultados podem ser vistos nas imagens abaixo. A azul as medições reais e a laranja a previsão feita pelo nosso modelo. As previsões abaixo foram feitas para as 12h, 13h e 14h do dia 30 de Outubro de 2016.

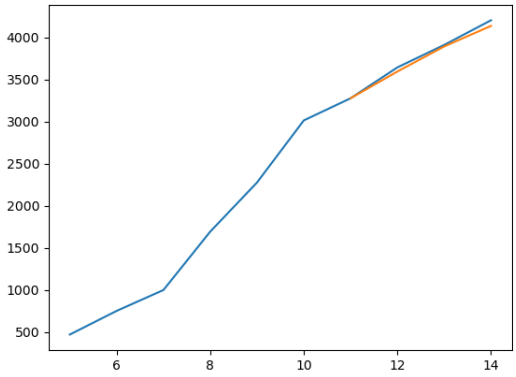


Figura 8: Previsão

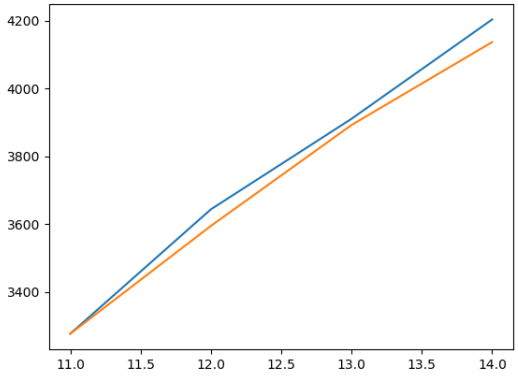


Figura 9: Zoom da previsão

5 Conclusões

Uma vez terminada a exposição do trabalho efectuado podemos então tirar algumas conclusões sobre o mesmo.

Como ponto inicial começamos por ressalvar a utilidade da metodologia CRISP-DM ao longo deste projeto. Serviu não só para garantir a qualidade dos modelos desenvolvidos como também para estruturar e organizar o desenvolvimento do projeto.

Incidindo então concretamente sobre os modelos gerados, relativamente ao *dataset* dos incidentes em Braga, sentimos que os resultados não foram satisfatórios devido aos valores altos das metricas MAE e RMSE. Isto pode ter sido causado pelas abordagens que tomamos no processamento dos dados, nomeadamente o agrupamento por dias. Por outro lado, relativamente ao *dataset* do volume de passageiros no metro, consideramos que obtivemos resultados satisfatórios. Foram tomadas abordagens diferentes no processamentos dos dados mas também os dados iniciais aparentam ser mais constantes. Isto deu para perceber a grande importância e necessidade de um bom tratamento de dados para obtermos bons modelos. De qualquer forma, considerando os processos desenvolvidos e apresentados, consideramos que atingimos os objetivos pedidos para este projeto.