

**Universidade do Minho**  
Escola de Engenharia

# **Dados e Aprendizagem Automática**

## **Trabalho Prático**

Grupo 3

pg60242

pg59788

pg59789

pg61005

Daniel Andrade   José Diogo Martins   Luis Enrique Diaz   Pedro Malainho

# Índice

<b>1. Introdução .....</b>	<b>1</b>
<b>2. Metodologia .....</b>	<b>1</b>
<b>3. Estudo do problema .....</b>	<b>2</b>
3.1. Estudo do Negócio .....	2
<b>4. Estudo dos dados .....</b>	<b>2</b>
4.1. average_speed_diff (Target Variable) .....	3
4.2. city_name .....	4
4.3. record_data .....	4
4.4. average_free_flow_speed .....	4
4.5. average_time_diff .....	4
4.6. average_free_flow_time .....	5
4.7. luminosity .....	5
4.8. average_temperature .....	6
4.9. average_atmosp_pressure .....	6
4.10. average_humidity .....	7
4.11. average_wind_speed .....	7
4.12. average_cloudiness .....	8
4.13. average_precipitation .....	9
4.14. average_rain .....	9
4.15. Análise da correlação .....	10
<b>5. Preparação dos dados .....</b>	<b>10</b>
5.1. Tratamento de Dados 1 .....	11
5.2. Tratamento de Dados 2 .....	12
5.3. Tratamento de Dados 3 .....	13
<b>6. Modelação .....</b>	<b>15</b>
6.1. Metodologia de Avaliação .....	15
6.2. Seleção e Análise de Modelos Base .....	16
6.3. Otimização de Hiperparâmetros .....	16
6.4. Estratégias Avançadas: Deep Learning e Ensembles .....	17
6.4.1. Redes Neuronais (Deep Learning) .....	17
6.4.2. Ensemble Learning (Stacking & Voting) .....	17
<b>7. Avaliação e Interpretação dos Resultados .....</b>	<b>18</b>
<b>8. Conclusão .....</b>	<b>20</b>

# 1. Introdução

Este relatório diz respeito ao projeto desenvolvido no âmbito da unidade curricular de Dados e Aprendizagem Automática. O principal objetivo do projeto é aplicar diferentes paradigmas de machine learning na conceção e desenvolvimento de modelos de aprendizagem e decisão, reforçando a ligação entre os conceitos teóricos abordados na disciplina e a sua aplicação prática na resolução de problemas com dados reais.

# 2. Metodologia

O desenvolvimento deste projeto de Aprendizagem por Máquinas (Machine Learning) seguiu a metodologia **CRISP-DM**, que orientou o planeamento e a gestão das tarefas, garantindo maior robustez e facilitando a sua compreensão, implementação e evolução. Esta metodologia é composta por seis fases: (1) **Business Understanding** – definição dos objetivos e do problema analítico; (2) **Data Understanding** – exploração inicial dos dados, com avaliação da sua qualidade e relevância; (3) **Data Preparation** – transformação, limpeza e seleção de variáveis para a modelação; (4) **Modeling** – aplicação de algoritmos, afinação de metaparâmetros e testes; (5) **Evaluation** – medição da performance dos modelos obtidos; e (6) **Deployment** – implementação do modelo em ambiente real. Esta última fase não foi aplicada, uma vez que o projeto decorreu num ambiente académico. A aplicação desta metodologia contribuiu para uma abordagem mais sistemática, madura e orientada a resultados ao longo de todo o projeto.

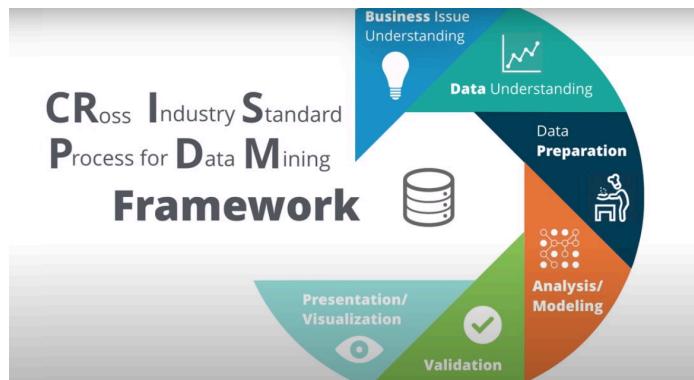


Figura 1: Metodologia CRISP-DM

### 3. Estudo do problema

O dataset do projeto abordou o problema relacionado com o nível de trânsito, para o qual nos foi fornecido um dataset com informação relevante sobre o tempo que se demora a percorrer as ruas da cidade do Porto.

#### 3.1. Estudo do Negócio

Neste problema, o nosso objetivo é desenvolver um modelo capaz de prever a diferença de velocidade média. Trata-se de um problema de classificação, uma vez que os estados possíveis são representados por classes distintas. O dataset fornecido contém dados sobre diversos registo ao nível do trânsito. Pretendemos criar um modelo que consiga analisar esses dados e determinar, com base nas características apresentadas, qual será a diferença de velocidade média. Os possíveis estados de previsão são:

- **None:** Trânsito Livre. A velocidade real é a máxima permitida (ou próxima).
- **Low:** Trânsito Ligeiro. Pequeno abrandamento, impacto mínimo no tempo de viagem.
- **Medium:** Trânsito Moderado. Abrandamento notável, fluxo denso e tempo de viagem afetado.
- **High:** Trânsito Intenso. Congestionamento significativo, grande redução da velocidade.
- **Very\_High:** Congestionamento Severo. Quase paralisação do trânsito.

Para prever o estado com o modelo desenvolvido, será realizada uma análise completa do dataset, com o objetivo de corrigir inconsistências, extraír conhecimento e visualização de dados e, por fim, construir vários modelos de previsão recorrendo a diferentes técnicas.

### 4. Estudo dos dados

O dataset é constituído por 6812 linhas e 14 colunas/atributos. Os atributos são os seguintes:

- **city\_name** - nome da cidade em causa;
- **record\_date** - o timestamp associado ao registo;
- **average\_speed\_diff** - a diferença de velocidade corresponde à diferença entre (1.) a velocidade máxima que os carros podem atingir em cenários sem trânsito e (2.) a velocidade que realmente se verifica. Quanto mais alto o valor, maior é a diferença entre o que se está a andar no momento e o que se deveria estar a andar sem trânsito, i.e., valores altos deste atributo implicam que se está a andar mais devagar;

- **average\_free\_flow\_speed** - o valor médio da velocidade máxima que os carros podem atingir em cenários sem trânsito;
- **average\_time\_diff** - o valor médio da diferença do tempo que se demora a percorrer um determinado conjunto de ruas. Quanto mais alto o valor, maior é a diferença entre o tempo que demora para se percorrer as ruas e o que se deveria demorar sem trânsito, i.e., valores altos implicam que se está a demorar mais tempo a atravessar o conjunto de ruas;
- **average\_free\_flow\_time** - o valor médio do tempo que demora a percorrer um determinado conjunto de ruas quando não há trânsito;
- **luminosity** - o nível de luminosidade que se verificava na cidade do Porto;
- **average\_temperature** - valor médio da temperatura para o record\_date na cidade do Porto;
- **average\_atmosp\_pressure** - valor médio da pressão atmosférica para o record\_date na cidade do Porto;
- **average\_humidity** - valor médio de humidade para o record\_date na cidade do Porto;
- **average\_wind\_speed** - valor médio da velocidade do vento para o record\_date na cidade do Porto;
- **average\_cloudiness** - o valor médio da percentagem de nuvens para o record\_date na cidade do Porto;
- **average\_precipitation** - valor médio de precipitação para o record\_date na cidade do Porto;
- **average\_rain** - avaliação qualitativa do nível de precipitação para o record\_date na cidade do Porto.

De seguida, vamos analisar cada um dos atributos e extrair algumas conclusões.

#### 4.1. average\_speed\_diff (Target Variable)

A variável **average\_speed\_diff** é a **variável objetivo**. As categorias existentes são **None, Low, Medium, High, Very High** e a categoria mais representada é a categoria **None** com cerca de 32.3% de frequência. As categorias Medium, High, Low, Very\_High têm frequência relativa de 24.2%, 15.6%, 20.8% e 7.0% respetivamente.

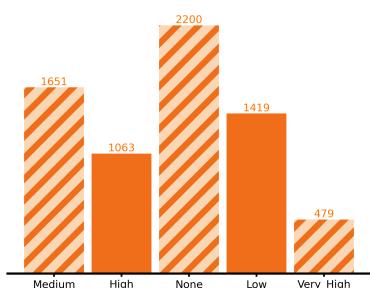


Figura 2: Bar Chart  
average\_speed\_diff

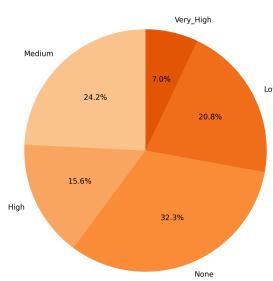


Figura 3: Pie Chart  
average\_speed\_diff

## 4.2. city\_name

A variável **CITY\_NAME** é uma variável categórica e indica a cidade onde cada registo foi recolhido. No dataset disponível, todos os registos correspondem à cidade do Porto, representando 100% dos dados.

Como apenas existe uma cidade esta variável não contribui para a distinção entre diferentes localizações neste dataset.

## 4.3. record\_data

A variável **RECORD\_DATA** é uma variável temporal no formato AAAA-MM-DD HH:MM:SS composta por **6812 valores únicos** e sem registo de missing values , servindo como um identificador único para cada observação. Ao extrair características como a hora do dia e o dia da semana, podemos analisar padrões temporais, como o comportamento do tráfego durante as horas de pico ou nos fins de semana.

## 4.4. average\_free\_flow\_speed

A variável numérica **AVERAGE\_FREE\_FLOW\_SPEED**, composta por **225 valores únicos** e sem registo de missing values, apresenta uma **média de 40.6610** e uma **mediana de 40.7000**, cuja proximidade sugere uma distribuição central equilibrada. Esta simetria é corroborada por uma **skewness reduzida (0.1097)**, indicando um comportamento próximo da distribuição normal, enquanto a **kurtosis negativa (-0.3244)** revela um perfil platicúrtico de caudas leves, denotando uma dispersão mais uniforme dos dados e uma **ausência de outliers extremos significativos**.

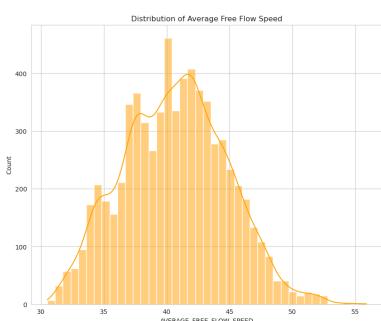


Figura 4: Bar Chart  
AVERAGE\_FREE\_FLOW\_SPEED

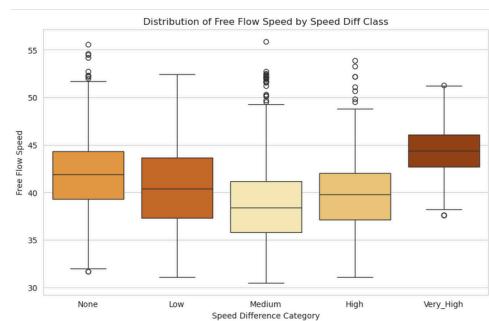


Figura 5: Box Plot  
AVERAGE\_FREE\_FLOW\_SPEED

## 4.5. average\_time\_diff

A variável **AVERAGE\_TIME\_DIFF** é numérica e mede o atraso médio de tráfego. A análise visual do gráfico de Densidade mostra que os valores oscilam entre **0 (mínimo)** e **296.5 (máximo)**. A distribuição é fortemente assimétrica. A grande maioria dos dados está concentrada nos valores mais baixos (próximo de zero), indicando que as viagens,

na maioria das vezes, sofrem atrasos mínimos. No entanto, o gráfico estende-se significativamente para a direita. Estes valores altos são cruciais, pois identificam os **atrasos extremos** e os **gargalos críticos do tráfego**.

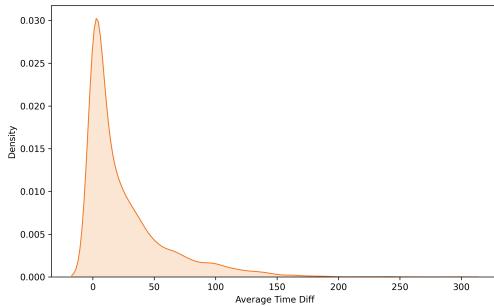


Figura 6: Histogram AVERAGE\_TIME\_DIFF

#### 4.6. average\_free\_flow\_time

A variável AVERAGE\_FREE\_FLOW\_TIME é uma variável numérica contínua e indica o tempo médio necessário para percorrer um conjunto de ruas em condições de free flow (sem trânsito). Esta variável oscila entre um **mínimo de 46.4** e um **máximo de 112.0**, registando uma **média de 81.1440** e uma **mediana de 82.4000**. A diferença entre a média e a mediana indica uma ligeira assimetria negativa, confirmada para uma skewness de -0.3658. Adicionalmente, a kurtosis ligeiramente negativa (-0.2096) sugere um comportamento platicúrtico, indicando um pico ligeiramente mais achatado e caudas ligeiramente mais leves do que uma distribuição normal, sem a presença significativa de outliers ou caudas pesadas.

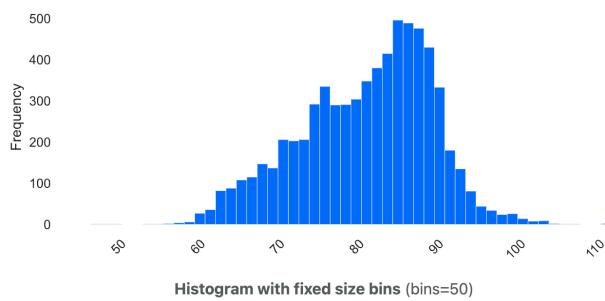


Figura 7: Histogram AVERAGE\_FREE\_FLOW\_TIME

#### 4.7. luminosity

A variável LUMINOSITY é uma variável categórica ordinal, pois representa o nível de luz. Após a verificação, a variável não tem valores nulos. A categoria LIGHT é a mais frequente, representando 48.34% do total das observações, enquanto a categoria LOW\_LIGHT é a menos representada, representando 3.90% do total das observações.

Esta distribuição é crucial para analisar como as condições de luz podem influenciar o fenómeno em estudo.

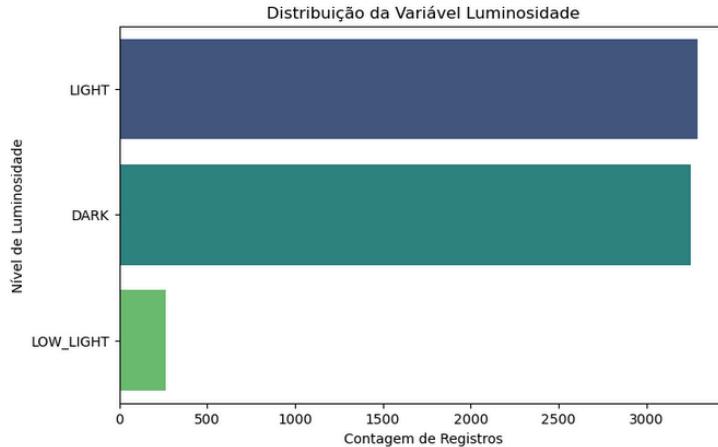


Figura 8: Distribuição da Variável LUMINOSITY

#### 4.8. average\_temperature

A variável numérica **AVERAGE\_TEMPERATURE** oscila entre um **mínimo de 0.0 e um máximo de 35.0**, registando uma **média de 16.1935** e uma **mediana de 16.0000**. A notável proximidade entre a média e a mediana, reforçada por uma skewness baixa (0.1818), evidencia uma distribuição simétrica. Adicionalmente, a **kurtosis próxima de zero (0.2865)** sugere um comportamento mesocúrtico, indicando que os dados seguem uma distribuição muito próxima da normal, **sem a presença significativa de outliers ou caudas pesadas**.

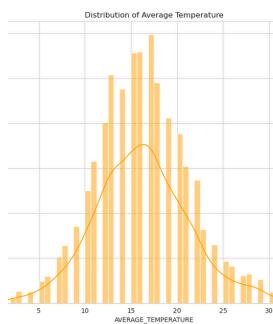


Figura 9: Bar Chart AVERAGE\_TEMPERATURE

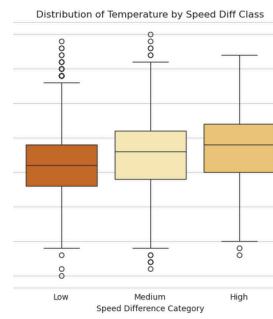


Figura 10: Box Plot AVERAGE\_TEMPERATURE

#### 4.9. average\_atmosp\_pressure

A variável **AVERAGE\_ATMOSP\_PRESSURE** é numérica e mede a pressão atmosférica média. A análise visual do gráfico de Densidade mostra que os valores oscilam entre **985 (mínimo)** e **1033 (máximo)**. A distribuição é fortemente assimétrica, com a grande maioria dos **dados concentrada em torno de 1017**. O pico elevado indica que esta é a pressão atmosférica mais comum. A distribuição apresenta uma **assimetria negativa**.

(ou cauda à esquerda), o que significa que, embora a pressão seja geralmente alta, existem eventos de pressão significativamente baixa que são menos frequentes.

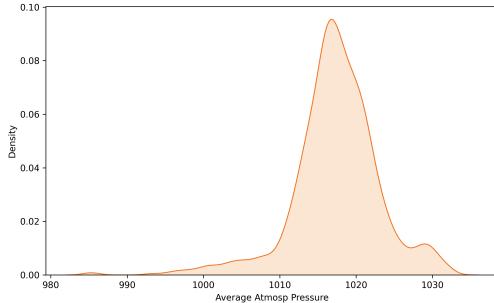


Figura 11: Histogram AVERAGE\_ATMOSP\_PRESSURE

#### 4.10. average\_humidity

A variável numérica **AVERAGE\_HUMIDITY** oscila entre um **mínimo de 14.0** e um **máximo de 100.0**, registando uma **média de 80.0842** e uma **mediana de 83.0000**. A diferença entre a média e a mediana, juntamente com uma skewness de  $-0.9660$ , evidencia uma assimetria negativa significativa. Adicionalmente, a **kurtosis próxima de zero (0.3497)** sugere um comportamento ligeiramente mesocúrtico, indicando que os dados seguem uma distribuição muito próxima da normal, **sem a presença significativa de outliers ou caudas pesadas**.

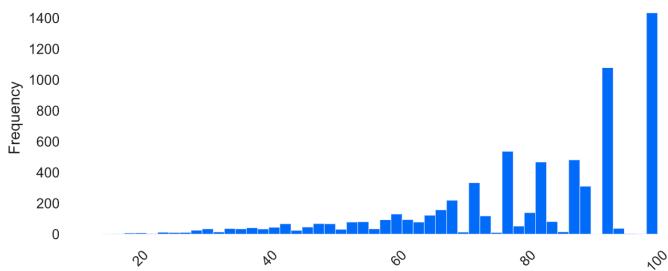


Figura 12: Histogram AVERAGE\_HUMIDITY

#### 4.11. average\_wind\_speed

A variável numérica **AVERAGE\_WIND\_SPEED**, composta por 15 valores únicos e sem registo de valores em falta, oscila entre um mínimo de 0,0000 e um máximo de 14,0000, registando uma média de 3,0586 e uma mediana de 3,0000. A notável proximidade entre a média e a mediana sugere uma distribuição central equilibrada. Esta simetria é ligeiramente comprometida por uma assimetria de 0,8735, que é positiva e indica um viés para a direita (ou cauda longa para valores elevados de velocidade do vento). Além disso, uma **kurtosis positiva de 0,8871** sugere um comportamento leptocúrtico (pico mais acentuado do que a distribuição normal), indicando que os dados, embora

sejam relativamente simétricos no seu centro, têm caudas um pouco mais pesadas, o que poderia sugerir a presença de valores de vento alto mais frequentes do que seria de esperar numa distribuição normal perfeita.

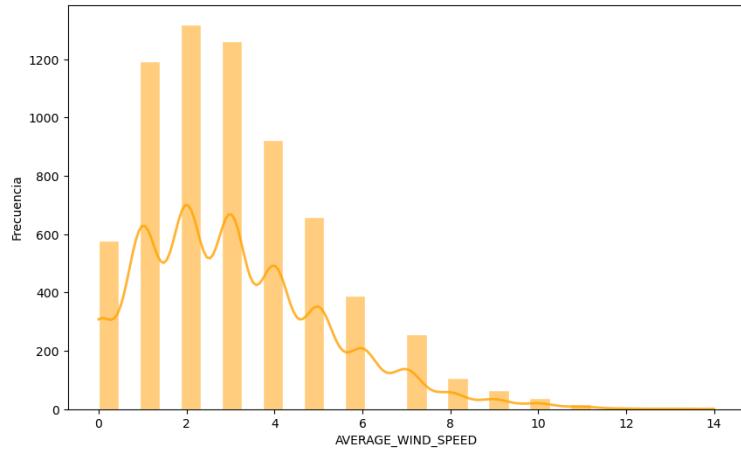


Figura 13: Bar Chart AVERAGE\_WIND\_SPEED

#### 4.12. average\_cloudiness

A variável **AVERAGE\_CLOUDINESS** é uma variável categórica ordinal que descreve o nível de cobertura de nuvens, apresentando várias categorias distintas.

As categorias mais frequentes são “céu claro” (1.582 registos) e “céu pouco nublado” (516 registos), indicando que a maior parte dos registos corresponde a condições de céu predominantemente limpo. Outras categorias registadas incluem “nuvens dispersas” (459), “nuvens quebrados” (448), “algumas nuvens” (422), “nuvens quebradas” (416), “céu limpo” (153), “tempo nublado” (67) e “nublado” (67).

A presença de várias categorias permite analisar o efeito da cobertura de nuvens na diferença média de velocidade (average\_speed\_diff) e avaliar se certas condições atmosféricas estão associadas a maior ou menor tráfego na cidade.

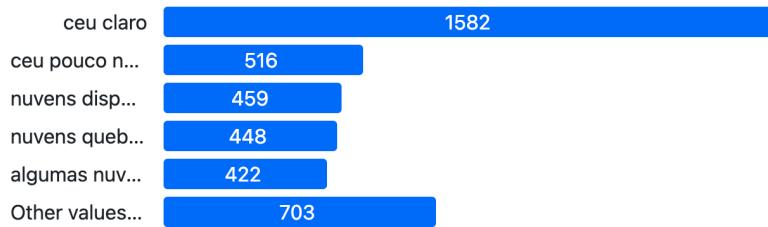


Figura 14: Histogram AVERAGE\_CLOUDINESS

#### 4.13. average\_precipitation

A variável **AVERAGE\_PRECIPITATION** apresenta-se, neste conjunto de dados, como **uma constante (valor único de 0.0)**. Dado que uma variável com variância zero não fornece informação discriminativa, esta coluna não possui valor preditivo para a modelação.

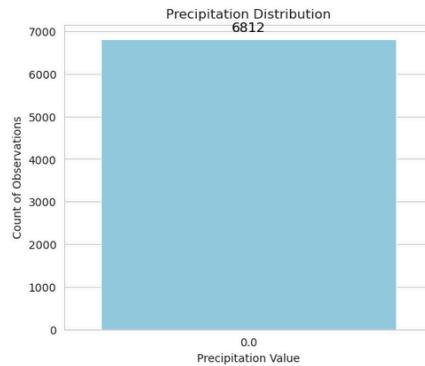


Figura 15: Bar Chart AVERAGE\_PRECIPITATION

#### 4.14. average\_rain

A variável **AVERAGE\_RAIN** é uma variável categórica ordinal, pois representa o nível da chuva. Após a verificação, a variável tem uma percentagem de 91.735173% de valores nulos. A categoria **CHUVA FRACA** é a mais frequente, representando 46.36% do total das observações, enquanto a categoria **CHUVISCO E CHUVA FRACA** é a menos representada, representando 0.18% do total das observações. Esta distribuição é crucial para analisar como as condições da chuva podem influenciar o fenômeno em estudo.

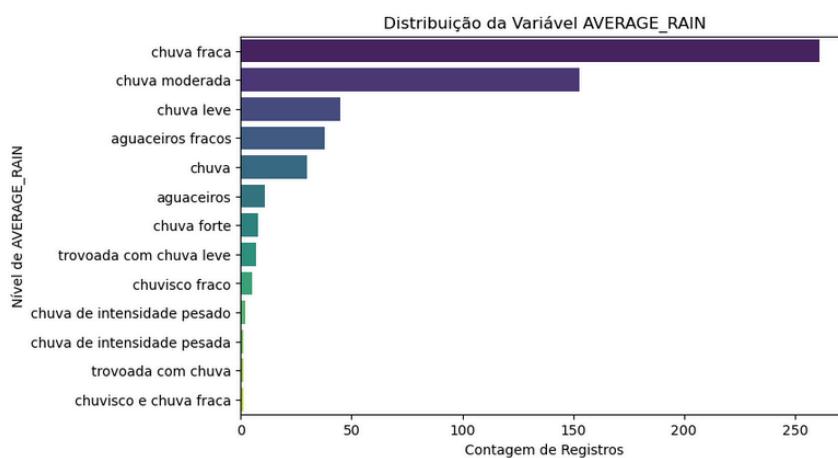


Figura 16: Distribuição da Variável AVERAGE\_RAIN

## 4.15. Análise da correlação

Após a análise individual das variáveis do dataset, foi avaliada a correlação entre elas. Para tal foram identificadas quais variáveis estão fortemente relacionadas. Esta análise permite perceber que variáveis podem ser redundantes, influentes ou merecer tratamento específico na preparação dos dados.

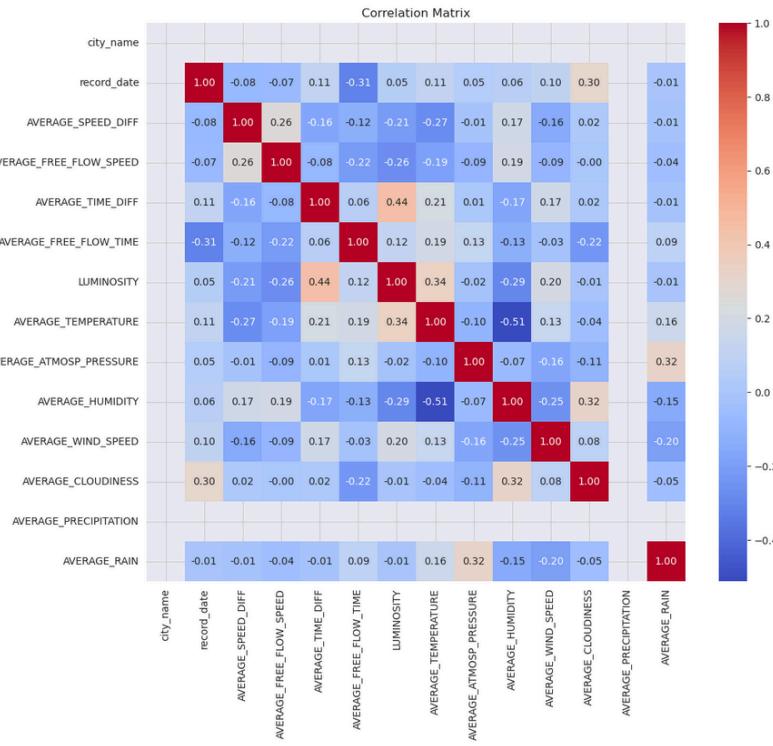


Figura 17: Tabela de Rank de Correlação

A análise da Figura 17 mostra que não existem pares de variáveis com **correlação forte**. O único par que se destaca é AVERAGE\_HUMIDITY e AVERAGE\_TEMPERATURE com uma correlação de **-0.51**, não sendo um valor elevado.

Os valores indicam que não se justificará realizar uma **seleção de features** fortes. Esta sugestão de menor redundância direta pode ainda assim justificar o uso de técnicas de **feature engineering**.

## 5. Preparação dos dados

Após o processo de análise dos dados procedeu-se à preparação e tratamento de dados que posteriormente serão usados para treinar o modelo. Ao longo do processo de maturing do projeto foram usados vários tipos de tratamentos de dados que foram de forma **progressivamente aumentando em complexidade e raciocínio**. Neste projeto, dividimos a fase de preparação em três etapas distintas: o **pré-tratamento (limpeza)**, o **tratamento (engenharia de atributos)** e, em alguns casos, o **tratamento específico para modelos** mais sensíveis, como as redes neurais.

## 5.1. Tratamento de Dados 1

O primeiro tratamento pensado passou por uma análise dos problemas mais superficiais do dataset.

### 1. Pré-Tratamento

O objetivo desta fase foi padronizar, agrupar e garantir a qualidade das variáveis relacionadas ao clima e ao tráfego para que o modelo de Machine Learning pudesse interpretá-las corretamente.

#### **Padronização e Uniformização de Categorias**

Identificámos que as variáveis categóricas AVERAGE\_CLOUDINESS e AVERAGE\_RAIN apresentavam uma granularidade excessiva, contendo múltiplas descrições textuais para representar estados meteorológicos semelhantes (ex: várias designações para chuva fraca). Para reduzir o ruído e a dimensionalidade, aplicámos uma estratégia de agregação semântica:

- **Nebulosidade:** As diversas descrições do céu foram condensadas em três níveis hierárquicos de obstrução: *Céu Claro* (para condições de visibilidade total), *Nuvens Parciais* (para céu fragmentado ou disperso) e *Nublado* (para cobertura total). Esta simplificação facilita a deteção de padrões pelo modelo, que deixa de distinguir variações irrelevantes de nomenclatura.
- **Precipitação:** As descrições de chuva, que variavam em detalhe e inconsistência, foram agrupadas estritamente por intensidade. Criámos categorias ordinais – *Chuva Leve*, *Moderada*, *Pesada* e *Tempestade* – garantindo que eventos como “aguaceiros ligeiros” e “chuva fraca” fossem tratados como a mesma entidade estatística.
- **Correção da Variável Alvo:** Detetámos ainda inconsistências na formatação da variável AVERAGE\_SPEED\_DIFF (o *target*) onde a classe None aparecia como *missing value*. Procedemos à correção para o valor None

### 2. Tratamento

#### **Tratamento de Missing Values**

Para abordar o desafio dos valores em falta nas variáveis climatéricas (AVERAGE\_CLOUDINESS e AVERAGE\_RAIN), descartámos abordagens simplistas, como a eliminação de registos ou a imputação pela média/moda, que poderiam distorcer as correlações naturais entre fenómenos (ex: a relação intrínseca entre nebulosidade e luminosidade).

Em alternativa, implementámos uma estratégia de imputação baseada em *Machine Learning*, utilizando um **Random Forest Classifier**. Este algoritmo foi treinado para prever as condições meteorológicas em falta com base em variáveis numéricas estáveis (como temperatura, humidade e pressão atmosférica). O processo incluiu ainda um filtro de “etiquetas limpas”, garantindo que o modelo aprendesse apenas com categorias

padronizadas e isentas de ruído, assegurando assim uma reconstrução dos dados fiel ao contexto de cada amostra.

### Preparação e Codificação de Dados

- Resolvemos a questão crítica dos valores NaN na coluna AVERAGE\_SPEED\_DIFF. Seguindo a descrição do problema, mapeámos estes nulos para a categoria “None” (Classe 0), garantindo que o modelo aprenda situações de trânsito fluido.
- Extraímos componentes temporais (ano, mês, dia, hora e dia da semana) da coluna record\_date para capturar padrões cíclicos de trânsito.
- Utilizámos a técnica de One-Hot Encoding (get\_dummies) para as variáveis qualitativas, permitindo que o modelo processe informações como luminosidade e estado do tempo sem assumir uma ordem hierárquica falsa.

### Tratamento de Outliers

Para evitar que valores extremos distorcessem a aprendizagem dos modelos, aplicámos o método estatístico do Intervalo Interquartil (IQR).

Calculámos os limites inferior e superior para cada variável numérica. Valores fora deste intervalo foram substituídos (Capping) pelos valores dos limites. Esta técnica garante que o modelo seja robusto, não sendo excessivamente influenciado por leituras anómalas de sensores, mantendo a integridade estatística do conjunto de dados.

### Normalização de Dados

Dado que as variáveis numéricas possuem escalas muito diferentes (ex: a pressão atmosférica está na casa dos milhares, enquanto a velocidade do vento é baixa), a normalização é essencial.

Utilizámos o MinMaxScaler, que transforma todos os valores numéricos para um intervalo entre 0 e 1. Isto impede que variáveis com números maiores dominem injustamente o cálculo de modelos baseados em distância (como KNN) ou gradientes, garantindo uma convergência mais rápida e equilibrada.

## 5.2. Tratamento de Dados 2

Esta segunda abordagem de tratamento de dados procurou simplificar o pipeline de processamento, focando-se na preservação da escala original das variáveis e numa codificação que mantivesse a relação de ordem entre as categorias. Ao contrário do tratamento anterior, optou-se por não aplicar normalização nem o tratamento de outliers, testando a robustez dos modelos a valores extremos.

### 1. Pré-Tratamento

Foi utilizado o pré-tratamento de dados utilizado no **Tratamento de Dados 1** visto que os casos de pré-tratamento que pensamos relevantes são tratados com as decisões aqui tomadas.

## **2. Tratamento**

### **Codificação Ordinal**

Para o tratamento de dados só vão ser registados as diferenças relativamente ao tratamento de dados 1.

Enquanto o tratamento de Dados 1 utiliza *One-Hot Encoding*, nesta novo tratamento utilizamos o ***Ordinal Encoding***. Esta escolha foi aplicada às seguinte variáveis:

- Luminosity, day\_of\_week, AVERAGE\_CLOUDINESS e AVERAGE\_RAIN
- A vantagem desta técnica é que ela converte as categorias em números numa única coluna, preservando uma hierarquia implícita e mantendo o *dataset* mais compacto, o que facilita a aprendizagem de modelos baseados em árvores.

### **Simplificação e Limpeza Final**

- **Mapeamento da Variável Alvo:** A variável de saída foi convertida manualmente para uma escala numérica de 0 a 4 (*None* a *Very\_High*).
- **Remoção de Redundância:** Identificou-se que a coluna *city\_name* continha apenas um valor único assim como *AVERAGE\_PRECIPITATION*, pelo que ambas foram removidas para reduzir o ruído.
- **Ausência de Normalização:** Mantivemos as variáveis numéricas na sua escala original, confiando na capacidade dos modelos de *ensemble* em lidar com diferentes magnitudes sem a necessidade de um *MinMaxScaler*

## **5.3. Tratamento de Dados 3**

### **Pré-Tratamento**

Foi utilizado o pré-tratamento de dados utilizado no **Tratamento de Dados 1** visto que os casos de pré-tratamento que pensamos relevantes são tratados com as decisões aqui tomadas.

### **2. Tratamento**

Para o tratamento de dados só vão ser registados as diferenças relativamente ao tratamento de dados 1 e 2.

### **Tratamento de Missing Values**

Foi utilizada a mesma técnica com o modelo *Random Forest* mas foram utilizados os dados do dataset de teste. Deste modo, aumentou-se a informação para a previsão dos valores em falta.

### **Tratamento de Outliers**

Para a gestão de valores atípicos, optou-se por uma abordagem estatística robusta baseada no **Intervalo Interquartil (IQR)**, em detrimento de cortes cegos baseados em percentis fixos. Esta técnica define limites dinâmicos para cada variável, calculados

a partir da sua dispersão central. Implementámos uma estratégia de “Capping”, onde os valores que excedem os limites não são removidos, mas sim substituídos por esses mesmos limites.

Aplicámos fatores multiplicativos  $k$  distintos consoante a sensibilidade da variável:

- $k = 1.5$  (Limites Standard): Para variáveis com distribuições mais comportadas (ex: Temperatura, Humidade), captando *outliers* moderados.
- $k = 3.0$  (Limites Conservadores): Para variáveis com elevada variância natural (ex: Pressão Atmosférica, Velocidade do Vento), garantindo que apenas os valores extremos severos fossem truncados, preservando a variabilidade necessária para a modelação.

### Extração de Features Temporais

Adicionalmente ao tratamento passado, criámos uma variável binária `is_weekend`, dado que o comportamento do tráfego aos fins de semana é estruturalmente diferente dos dias úteis.

### Transformação Cíclica

As variáveis temporais como a “hora” apresentam um desafio: matematicamente, a hora 23 e a hora 0 estão distantes, mas temporalmente são contíguas. Para resolver este problema de descontinuidade, transformámos as horas e os meses em coordenadas cíclicas (Seno e Cosseno). Isto permite que o modelo compreenda a natureza circular do tempo.

### Codificação de Variáveis (Encoding)

Adotámos estratégias diferentes consoante a natureza da variável:

- **Ordinal Encoding:** Para variáveis com hierarquia intrínseca (como a LUMINOSITY ou o Target), onde a ordem importa.
- **One-Hot Encoding:** Para variáveis nominais, onde não existe ordem matemática, criando colunas binárias para evitar que o modelo assuma falsas hierarquias.

### Normalização (Scaling)

Como as variáveis possuem escalas muito díspares (ex: pressão atmosférica vs. velocidade do vento), aplicámos a *Standardization* (Z-Score). Isto coloca todas as variáveis na mesma escala (média 0 e desvio padrão 1), impedindo que variáveis com magnitudes maiores dominem o processo de aprendizagem dos modelos.

### Tratamento Específico para Modelos

Para a implementação das Redes Neuronais, que são mais propensas ao *overfitting* e sensíveis ao ruído, realizámos passos adicionais.

### Seleção de Features (Feature Selection)

Utilizámos a importância das variáveis calculada pelo nosso melhor modelo inicial de árvore (XGBoost) para filtrar o *dataset*. Selecioneámos apenas as *features* com maior poder preditivo, eliminando variáveis irrelevantes que introduziam ruído e dificultavam a convergência da rede neuronal.

## Adaptação do Target

Enquanto os modelos de árvore aceitam classes numéricas, a rede neuronal beneficia de um formato probabilístico. Convertemos a variável alvo para *One-Hot Encoding*, permitindo que a camada de saída da rede calcule a probabilidade de pertença a cada uma das 5 classes de tráfego independentemente.

# 6. Modelação

A fase de modelação teve como principal objetivo identificar os algoritmos com maior capacidade para captar padrões complexos no tráfego rodoviário, assegurando simultaneamente uma adequada capacidade de generalização a dados não observados. Este processo foi conduzido de forma iterativa, iniciando-se com modelos lineares simples e evoluindo progressivamente para arquiteturas de ensemble mais complexas.

A avaliação comparativa apresentada neste capítulo baseia-se na métrica de accuracy e incide sobre diversos algoritmos de aprendizagem automática, tais como Logistic Regression, K-Nearest Neighbors (KNN), Support Vector Machines (SVM), Random Forest e XGBoost.

A análise comparativa dos resultados obtidos possibilita identificar os algoritmos e as configurações de hiperparâmetros que melhor se adequam ao conjunto de dados em estudo e aos respetivos tratamentos aplicados.

Por fim, os resultados alcançados contribuem para a identificação das abordagens mais eficazes na previsão da variável alvo, permitindo selecionar o modelo com maior precisão global.

## 6.1. Metodologia de Avaliação

Para garantir a robustez dos resultados e mitigar o risco de conclusões baseadas em subconjuntos de dados favoráveis, adotámos uma estratégia de validação rigorosa. A métrica primária de avaliação foi a **Accuracy**, complementada pelo **F1-Score (Weighted)** para monitorizar o desempenho nas classes desequilibradas.

Os testes experimentais foram conduzidos recorrendo a diferentes estratégias de divisão dos dados, incluindo validação cruzada (*Cross-Validation*), concretamente **Stratified K-Fold**, e uma abordagem de **Holdout** com particionamento 70/30, utilizada para simular um cenário de teste final. Esta metodologia foi aplicada de forma transversal aos três conjuntos de dados preparados (TD1, TD2 e TD3), garantindo uma avaliação consistente entre experiências.

A comparação entre o desempenho obtido durante a validação local e os resultados apresentados no *Public Leaderboard* do Kaggle foi utilizada como principal indicador para a deteção de fenómenos de *overfitting*. Estas abordagens permitiram avaliar não só a capacidade de generalização dos modelos, mas também a sua robustez face a dados não observados durante a fase de treino.

## 6.2. Seleção e Análise de Modelos Base

Numa primeira etapa, submetemos os três tratamentos de dados a um leque diversificado de algoritmos com os seus hiperparâmetros por defeito (ou configurações padrão ligeiras). O objetivo foi perceber a afinidade de cada família de algoritmos com as diferentes estratégias de engenharia de atributos.

Modelo Base	TD1 (Local)	TD1 (Kaggle Público)	TD3 (Local)	TD3 (Kaggle Público)
Decision Tree	76.30%	X	X	X
Logistic Regression	64.64%	X	76.45%	X
K-Nearest Neighbors	54.27%	X	64.00%	X
Support Vector Machine	X	X	75.94%	X
Random Forest	78.41%	X	80.04%	83.33%
XGBoost	80.34%	83.11%	81.00%	82.00%

A análise desta tabela comparativa permite retirar conclusões importantes sobre a natureza dos dados:

- **Dominância de Árvores de Decisão:** Os modelos baseados em árvores (Random Forest e XGBoost) superaram consistentemente as abordagens baseadas em distância (KNN) ou margem (SVM). Isto sugere que as fronteiras de decisão do tráfego são não-lineares e beneficiam da capacidade das árvores de segmentar o espaço de características.

Como as submissões no Kaggle eram limitadas por dia, só alguns modelos foram submetidos, não havendo necessidade de submeter todos, especialmente os que já tinham uma accuracy baixa localmente.

**Nota:** Como no tratamento 2 se procedeu logo para modelos com *tunning* não foi feita uma avaliação basilar.

## 6.3. Otimização de Hiperparâmetros

Com base na etapa anterior, selecionámos os candidatos mais promissores, **XGBoost** e **Random Forest**, para um processo de **Fine-Tuning**, utilizando **RandomizedSearchCV** e **GridSearchCV**.

Modelo Otimizado	TD2 (Local)	TD2 (Kaggle Público)	TD3 (Local)	TD3 (Kaggle Público)
Random Forest	80.38%	82.66%	80.63%	81.55%
XGBoost	80.47%	81.11%	81.29%	83.11%

## 6.4. Estratégias Avançadas: Deep Learning e Ensembles

Para superar os limites dos modelos baseados em árvores, explorámos arquiteturas de Redes Neuronais e técnicas de combinação de modelos (*Ensembles*).

### 6.4.1. Redes Neuronais (Deep Learning)

A implementação de *Deep Learning* foi iterativa, variando entre bibliotecas e estratégias de otimização nos diferentes tratamentos:

- **MLP (Tratamento 2 - PyTorch):** Arquitetura profunda em “funil” composta por 5 camadas densas (iniciando em 1024 até 64 neurónios) com ativação LeakyReLU (0.01) e regularização via Batch Normalization e Dropout estratificado. O modelo foca-se na generalização através da SmoothedCrossEntropyLoss (fator 0.15) para mitigar o excesso de confiança (overconfidence), utilizando o otimizador AdamW com Weight Decay. A entrada de dados foi enriquecida com uma engenharia de atributos robusta, incluindo termos cúbicos, quadráticos e interações polinomiais das variáveis críticas.
- **Rede Neuronal Inicial (Tratamento 3):** Priorizou a regularização com **Batch Normalization** e **Dropout** (20-30%) após cada camada densa. Utilizou **Class Weights** na função de perda para combater o desequilíbrio das classes e *callbacks* (*ReduceLROnPlateau*) para ajuste dinâmico da *learning rate*.
- **Rede Neuronal Refinada (Tratamento 3):** Evolução focada na maximização da **Accuracy**. Substituiu-se a ativação pela função **Swish** (não-monotónica), aumentou-se a capacidade da rede (início com 256 neurónios) e removeram-se os pesos de classe para reduzir o viés a favor das minoritárias. O critério de paragem (**EarlyStopping**) passou a monitorizar estritamente a **val\_accuracy**.

### 6.4.2. Ensemble Learning (Stacking & Voting)

Combinámos os melhores modelos individuais para reduzir a variância e melhorar a generalização:

- **Weighted Soft Voting:** Média ponderada das probabilidades, atribuindo 40% ao XGBoost, 40% à Rede Neuronal Refinada e 20% ao Random Forest. Esta abordagem mitiga erros individuais sem necessitar de treino adicional.

- **Stacking Classifier:** Meta-arquitetura onde um **XGBoost** aprende a corrigir as previsões dos modelos base. Testámos duas variantes:
  1. **Stacking de Árvores:** Combinação de Random Forest, XGBoost, LightGBM, CatBoost e ExtraTrees.
  2. **Stacking Híbrido:** Adição da Rede Neuronal ao conjunto anterior. A diversidade matemática (Árvores vs. Redes Neuronais) permitiu captar padrões distintos nos dados, resultando na arquitetura mais robusta do projeto.

Modelo	TD2 (Local)	TD2 (Kaggle Público)	TD3 (Local)	TD3 (Kaggle Público)
MLP	77.50%	80.06%	X	X
RNI	X	X	81.14%	X
RNR	X	X	79.82%	81.33%
Soft Voting	X	X	81.37%	82.67%
Stacking 1	81.11%	80.95%	X	X
Stacking 2	81.50%	82.00%	X	X
Stacking 3	X	X	81.88%	82.22%

## 7. Avaliação e Interpretação dos Resultados

A validação final do projeto foi realizada através da submissão das previsões na plataforma Kaggle. A tabela abaixo resume o desempenho dos nossos melhores modelos em ambiente “real”. Após a análise de todos os modelos foram recolhidos os 3 melhores conjuntos de decisões para destaque.

Modelo	Tra-tamento	Valida-ção Local	Kaggle Public	Kaggle Private	Classifi-cação média	Dife-rença média
Stacking 3	TD3	81.88%	82.22%	80.95%	81.68%	0.85%
XG Boost	TD3	80.99%	82.00%	80.99%	81.33%	0.67%
Stacking 2	TD2	81.50%	82.00%	79.14%	80.88%	1.91%

1. Como modelo com melhor desempenho global foi selecionado o modelo de *stacking* 3, aplicado ao conjunto de dados resultante do Tratamento de Dados 3. Este modelo apresentou a *accuracy* mais elevada na fase de validação local (81,88%), mantendo um desempenho consistente no Kaggle Privado (80,95%), o que evidencia uma boa capacidade de generalização. A reduzida diferença entre os resultados obtidos em

validação local e no teste final sugere a ausência de fenómenos significativos de *overfitting*, reforçando a robustez da abordagem adotada.

2. O modelo XGBoost, aplicado ao conjunto de dados do Tratamento de Dados 3, destacou-se pela sua elevada estabilidade entre as diferentes fases de avaliação. Este modelo registou uma accuracy de 80,99% na validação local, 82,00% no Kaggle Public e 80,99% no Kaggle Privado. A inexistência de diferença significativa entre a accuracy da validação local e o resultado do Kaggle Privado (diferença de 0%) demonstra uma capacidade de generalização excepcional. Este comportamento sugere que o modelo aprendeu os padrões fundamentais dos dados sem memorizar ruído, garantindo a máxima robustez na previsão de novos dados.
3. O Modelo Stacking 2, treinado com o conjunto do Tratamento de Dados 2, apresentou inicialmente resultados promissores, com uma accuracy de 81,50% na validação local e atingindo 82,00% no Kaggle Public. Contudo, observou-se uma quebra de desempenho no Kaggle Privado, onde o valor desceu para 79,14%. A diferença mais acentuada entre as métricas de validação/públicas e o resultado final sugere a ocorrência de fenómenos de *overfitting* aos dados de treino ou à leaderboard pública. Isto indica que, apesar das boas métricas iniciais, esta abordagem revelou-se menos eficaz na generalização para dados nunca vistos em comparação com o modelo usado com TD3.

## **8. Conclusão**

O projeto permitiu explorar diferentes abordagens de tratamento de dados e modelação para o problema da previsão de tráfego. Neste caso, foram estudados vários modelos e os seus meta-parâmetros tal como diferentes modos de preparação de dados.