# Previsão de Fluxo de Tráfego Rodoviário com *Machine Learning*

**Mestrado em Inteligência Artificial, Universidade do Minho**

13/01/2026

Pedro Reis PG59908
João Azevedo PG61693
Guilherme Pinto PG60225
Luís Silva PG60390

# 1 . Introdução

# Contextualização



**Problema:** Prever o fluxo de tráfego rodoviário na cidade do Porto, lidando com a natureza estocástica e não-linear da mobilidade urbana.

**Cenário:**

- **Fonte:** Dados históricos reais, em 2018 e 2019.
- **Desafio:** Transformar dados bruto em inteligência preditiva.

**Projeto**

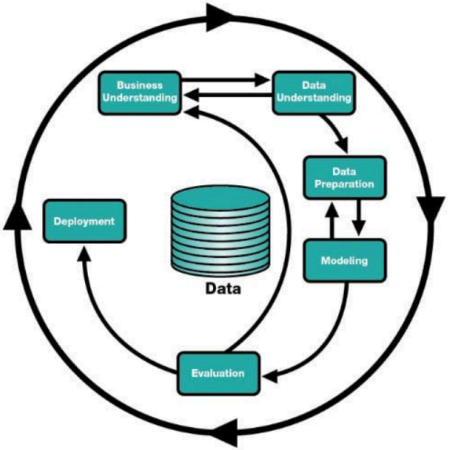
- Análise Exploratória;
- Engenharia de Atributos;
- Comparação de Modelos;
- Validação em Competição.

# Metodologia

Adotou-se o processo iterativo CRISP-DM:

1. Compreensão do Negócio;
2. Compreensão dos Dados;
3. Preparação dos Dados;
4. Modelação;
5. Avaliação.

*Iteração contínua entre a preparação dos dados e a modelação para otimização de performance.*



# 2 . Compreensão do Negócio

# Enquadramento do Problema



O **desafio** central é a previsão precisa do fluxo de tráfego num intervalo temporal específico, classificando a severidade do congestionamento (problema *multiclass*).

A **complexidade** do cenário trata-se de um fenómeno estocástico e não-linear, influenciado por:

- Padrões temporais;
- Condições meteorológicas;
- Comportamento humano imprevisível.

**Valor e impacto social**

A capacidade de perceber quais os fatores que influenciam os congestionamentos permite:

1. **Otimização de rotas:** Crucial para veículos de emergência.
2. **Sustentabilidade:** Redução de emissões poluentes.
3. **Qualidade de vida:** Gestão proativa do tráfego urbano e redução do tempo de viagem.

# Objetivos Estratégicos



**1. Extração de Conhecimento**
Perceber como variáveis observadas definem matematicamente as diferentes classes de tráfego.

**2. Excelência Preditiva**
Maximizar a capacidade do modelo em distinguir corretamente entre categorias vizinhas usando os dados disponíveis.

**3. Robustez e Generalização**
Garantir que o modelo mantém a performance em cenários de tráfego atípicos.

**4. Validação Competitiva**
Prova de conceito através do *ranking* Kaggle: a capacidade de generalizar a classificação para dados nunca vistos.

# 3 . Compreensão dos Dados

# Análise Univariada: Variável Alvo

**Definição:** Diferença entre a velocidade de fluxo livre e a velocidade real.

**Ausência do "None":** A categoria "None" foi interpretada como `NaN` durante o processo de ingestão.

**Análise de Distribuição:**

- **Desequilíbrio severo:** A classe "None" domina com 32.3%.
- **Escassez de casos críticos:** A classe "Very_High" representa apenas 7.1%.

**Consequência:** O modelo terá tendência natural a subestimar o trânsito grave.

| Classe | Frequência |
|---|---|
| `NaN` | 32.3% |
| Low | 20.8% |
| Medium | 24.2% |
| High | 15.6% |
| Very_High | 7.1% |

# Análise Univariada: Temporal

**Dinâmica Temporal (`record_date`)**

- **Formato Bruto:** Timestamp (`MM/DD/YYYY HH:MM`).
- **Desafio:** Não é interpretável diretamente por algoritmos.
- **Potencial:** Esconde padrões cíclicos cruciais.

| ID | record_date |
|---|---|
| 0 | 8/29/2019 7:00 |
| 1 | 8/10/2018 14:00 |
| 2 | 9/1/2019 16:00 |
| 3 | 2/26/2019 11:00 |
| 4 | 6/6/2019 12:00 |

# Análise Univariada: Cloudiness e Rain



**Problemas críticos de qualidade** Identificaram-se falhas graves nas variáveis `AVERAGE_RAIN` e `AVERAGE_CLOUDINESS` que exigem intervenção imediata.

**`AVERAGE_RAIN`**

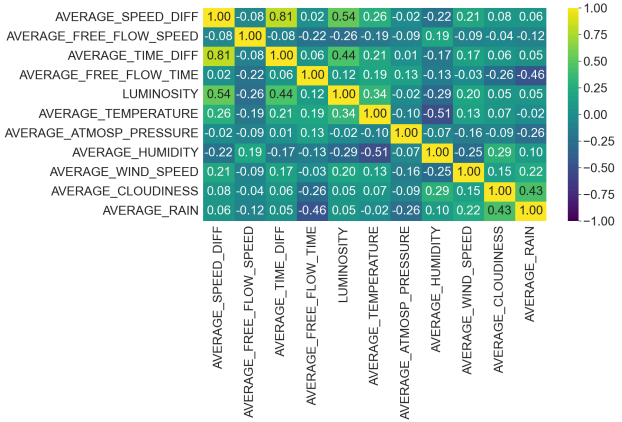
1. **Inconsistência de nulos**
   Valores em falta representados de múltiplas formas: `NULL` e strings vazias.
2. **Ausência de "zero"**
   Não existe uma categoria explícita para "ausência de chuva", obrigando a deduções lógicas.
3. **Redundância**
   Categorias sobrepostas que dificultam a distinção da intensidade.

**`AVERAGE_CLOUDINESS`**

1. **Lacunas de Informação**
   Ao contrário da chuva, os valores em falta representam uma perda real de dados e não apenas "céu limpo".
2. **Ruído Categórico**
   Existência de termos sinónimos e distinções demasiado subtis que diluem a informação.

# Análise Multivariada: Matriz de Correlação

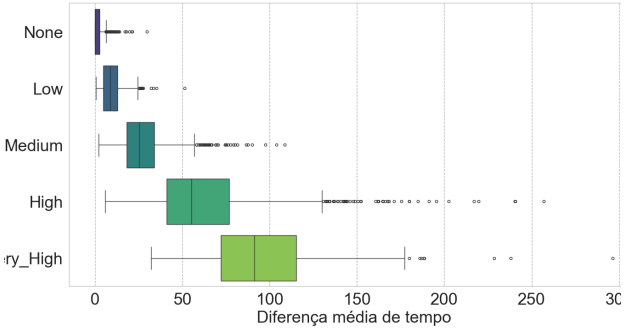
- **`AVERAGE_SPEED_DIFF`**
  - Correlação de **0.81** com `AVERAGE_TIME_DIFF`.
  - Correlação de **0.54** com `LUMINOSITY`.
- **Multicolinearidade**
  - Correlação inversa de **−0.51** entre `TEMPERATURE` e `HUMIDITY`.
  - Correlação de **0.43** entre `AVERAGE_CLOUDINESS` e `AVERAGE_RAIN`.



# Análise Multivariada: Separabilidade e Padrões

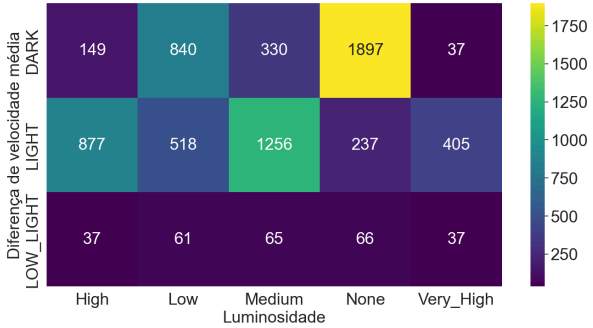
## AVERAGE_TIME_DIFF

Observa-se uma progressão estritamente crescente: quanto maior o atraso, mais grave a classe.




## LUMINOSITY

- "DARK": Predominância de "None", por ser durante a noite.
- "LIGHT": Concentração de transito.



# 4 . Preparação dos Dados

# Tratamento de *Missing Values*: Correção e Lógica



A análise exploratória identificou erros de ingestão e definições de negócio em falta que exigiram intervenção manual antes de qualquer imputação.

**Correção de *Parsing* , `AVERAGE_SPEED_DIFF`**

**Problema:** A categoria "None" foi erradamente interpretada como `NaN`.

**Impacto:** A exclusão destes registos eliminaria **32.3%** dos dados, enviesando o modelo para cenários de tráfego intenso.

**Solução:** Substituir os *missing values* por a categoria "None".

**Inferência de Negócio , `AVERAGE_RAIN`**

**Problema:** Elevada taxa de `NULL` e inexistência da categoria "Sem Chuva".

**Análise:** A inspeção dos dados brutos revelou que `NULL` correspondia a ausência de registo pluviométrico.

**Solução:** Conversão de `NULL` $\rightarrow$ "Sem Chuva".

# Tratamento de *Missing Values*: *Binning*

Para lidar com a redundância semântica e o desequilíbrio de classes nas variáveis meteorológicas, aplicou-se redução de dimensionalidade.

| Variável | Problema Identificado | Transformação |
|---|---|---|
| `AVERAGE_RAIN` | 13 categorias redundantes e ruído. | **Binária:** Agrupamento em "Sem Chuva" e "Com Chuva" para focar no impacto macroscópico. |
| `AVERAGE_CLOUDINESS` | 9 categorias com distinções subtis e classes raras. | **Ordinal:** Redução para "Céu Limpo", "Parcialmente Nublado" e "Nublado". |

# Tratamento de *Missing Values*: Imputação

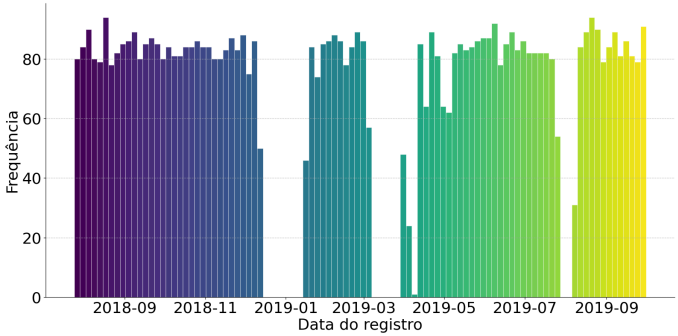
Para preencher lacunas reais de informação, realizou-se um estudo comparativo de métodos.

**Métodos Rejeitados:**

- ***Machine Learning*:** Introduziu um viés considerável, distorcendo a distribuição estatística original.
- **Moda:** Demasiado estática para séries temporais.

**Método Selecionado: *Forward Fill***

- **Lógica:** Propaga a última observação válida.
- **Justificação:** Respeita a inércia temporal dos fenómenos meteorológicos.
- **Validação:** Apesar das descontinuidades temporais observadas na figura, assume-se a persistência do estado como a aproximação mais conservadora.



# Remoção e Codificação de Atributos

**Remoção de variáveis**

Atributos sem variância (informação discriminativa) foram eliminados:

- `city_name`
- `AVERAGE_PRECIPITATION`

**Label Encoding**

Dada a natureza ordinal das variáveis categóricas, optou-se por *Label Encoding* em vez de *One-Hot*.

- `AVERAGE_SPEED_DIFF`
- `LUMINOSITY`
- `AVERAGE_RAIN`
- `AVERAGE_CLOUDINESS`

# *Feature Engineering*: Tempo Cíclico

**Problema:** A representação linear da hora (0-23) cria uma descontinuidade artificial (23h e 00h parecem distantes).

**Solução:** Transformação trigonométrica com a projeção da hora num círculo unitário para preservar a proximidade temporal matemática.

$$\text{'hour\_sin'} = \sin\left(\frac{2\pi \cdot \text{'hour'}}{24}\right)$$

$$\text{'hour\_cos'} = \cos\left(\frac{2\pi \cdot \text{'hour'}}{24}\right)$$


**Outras Extrações:** Decomposição de `record_date` em: Dia da semana, Semana, Mês, Dia do ano e Ano.

# Normalização e Estratégia de Balanceamento



**Normalização e tratamento de *outliers***

- **Min-Max *Scaling*:** Todas as variáveis numéricas escaladas para $[0, 1]$ para ajudar na convergência.
- ***Outliers*:** Em tráfego, os extremos não são ruído, são os eventos de interesse (acidentes, congestionamento).

**Balanceamento dos dados de treino**

Combinação de duas técnicas robustas:

1. **Stratified K-Fold:** Mantém a proporção das classes na validação.
2. **Oversampling com SMOTE:** Geração de dados sintéticos para as classes minoritárias.

# Seleção de Atributos

Comparativo de três abordagens para reduzir a dimensionalidade e evitar *overfitting*:

| Método | Análise | Decisão |
|---|---|---|
| PCA<br>*(Principal Component Analysis)* | Reduz a dimensionalidade eficazmente, mas sacrifica a **interpretabilidade semântica** das variáveis (perda de significado físico). | **Rejeitado** |
| PSO<br>*(Particle Swarm Optimization)* | Inviável devido à **explosão combinatória** do tempo de execução quando acoplado à otimização de hiperparâmetros. | **Rejeitado** |
| Eliminação Iterativa<br>*(Manual Backward Elimination)* | Baseada na **Feature Importance**. Permitiu reduzir a complexidade e validar os preditores dominantes face à realidade do negócio. | **Selecionado** |

# 5 . Modelação

# Estratégia de Modelação



A abordagem seguiu uma evolução de complexidade incremental para equilibrar o enviesamento (*bias*) e a variância.

**1. *Baselines***

Estabelecer o chão de performance com modelos interpretáveis.

*Decision Tree, Logistic Regression*

→

**2. Redução de Variância**

Capturar não--linearidade e reduzir ruído.

*SVM, Random Forest, Boosting*

→

**3. Meta-Learning**

Generalização máxima via arquiteturas complexas.

*Deep Learning, Stacking*

**Nota:** Otimização transversal via *Grid Search* em todos os modelos (em exceto o *Deep Learning*).

# Modelos: Referência

O ponto de partida para validar a complexidade futura.

***Decision Tree***

Segmentação não-linear via regras hierárquicas.

| | |
|---|---|
| `criterion` | Gini |
| `max_depth` | 7 |
| `min_samples_split` | 3 |
| `min_samples_leaf` | 1 |

***accuracy*: 0.77**

***Logistic Regression***

*Baseline* linear paramétrica com regularização L1 (*sparsity*).

| | |
|---|---|
| `penalty` | L1 (lasso) |
| `solver` | saga |
| `C` | 1.0 |
| `max_iter` | 2000 |

***accuracy*: 0.77**

# Modelos: Redução de Variância

Aumentar a complexidade para capturar fronteiras não-lineares e melhorar a generalização.

***Support Vector Machine***

Maximização de margem via *Kernel Trick*.

| | |
|---|---|
| `kernel` | poly |
| `C` | 10 |
| `gamma` | scale |

***accuracy*: 0.79**

***Random Forest***

*Ensemble* (*Bagging*) para robustez.

| | |
|---|---|
| `n_estimators` | 120 |
| `max_depth` | 18 |
| `criterion` | entropy |
| `min_samples_split` | 3 |
| `min_samples_leaf` | 1 |
| `bootstrap` | true |

***accuracy*: 0.81**

# Modelos: Boosting

Estratégia sequencial: cada novo modelo foca-se em corrigir os erros residuais do anterior.

***Gradient Boosting***

Otimização direta da função de perda (*loss*).

| | |
|---|---|
| `n_estimators` | 150 |

***accuracy*: 0.81**

***XGBoost***

Versão otimizada com regularização na função objetivo.

| | |
|---|---|
| `max_depth` | 6 |
| `n_estimators` | 120 |
| `colsample_bytree` | 0.8 |

***accuracy*: 0.81**

# Modelos: Arquitetura ANN

Rede *Feedforward* (MLP) desenhada para capturar relações não-lineares de alta ordem.

***Input Layer***
Normalização *StandardScaler*

*Ajustada à dimensionalidade*

↓

***Hidden Layer* 1**
128 Neurónios, *ReLU*

*+ BatchNormalization + Dropout (0.3)*

↓

***Hidden Layer* 2**
64 Neurónios, *ReLU*

*+ Dropout (0.2)*

↓

***Output Layer***
5 Neurónios, *Softmax*

*Distribuição de probabilidades*

# Modelos: Estratégia e Validação ANN

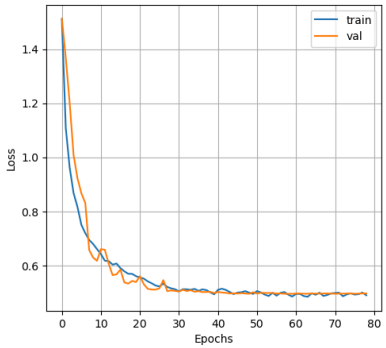
## Configuração

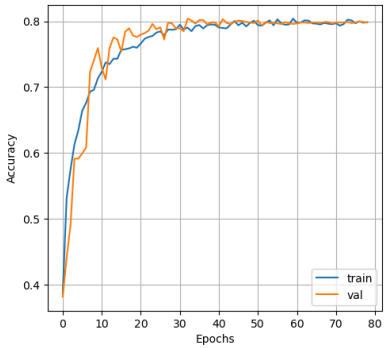
**Otimizador:Adam (`lr` = 0.001)**

***Loss Fn*** *Categorical Crossentropy*

***Callbacks***
- *EarlyStopping* (P: 20)
- *ReduceLROnPlateau*

***accuracy*: 0.80**



# Modelos: Arquiteturas de Especialistas

**Motivação:** Mitigar o viés do desequilíbrio de classes, forçando o sistema a especializar-se em categorias específicas em vez de otimizar uma função de perda global.

**Estratégia One-vs-Rest:** Decomposição do problema *multiclass* em 5 sub-problemas binários.

- **Inovação:** Aplicação de SMOTE. O *oversampling* ocorre apenas dentro do problema binário, evitando ruído nas fronteiras globais.
- **Decisão:** Competição direta, ganha o agente que apresentar maior grau de confiança.

***accuracy*: 0.80**

**Mixture of Experts (MoE):** Implementação de computação condicional dinâmica.

- **Experts:** Classificadores treinados em diferentes perspetivas dos dados.
- **Gating Network:** Um "Supervisor" que avalia o contexto e decide qual o perito mais competente para aquele cenário.
- **Decisão:** Soma ponderada dinâmica (não é uma votação democrática estática).

***accuracy*: 0.79**

# Modelos: *Stacking Ensemble*

**Conceito:** Arquitetura onde um "meta-modelo" aprende a corrigir os erros dos modelos base, ponderando a fiabilidade de cada um em diferentes cenários.

**Arquitetura Heterogénea**
**Nível 1 (*Base Learners*):**
- *Random Forest*
- *XGBoost*
- *Gradient Boosting*
- *SVM (Poly)*
- *Logistic Regression*

↓

**Nível 2 (Meta-Learner):**
- *Logistic Regression*

**Integração de Fronteiras de Decisão:** Ao contrário dos *ensembles* homogéneos, o *Stacking* combina visões geométricas distintas:
- **Linearidade** (*Logistic Regression*);
- **Hiperplanos Multidimensionais** (SVM);
- **Cortes Ortogonais** (Modelos baseados em árvores).

**Conclusão:** O modelo capturou a complexidade onde algoritmos individuais falharam.

***accuracy*: 0.82**

# 6 . Avaliação

# Performance Global

**Destaques**

- ***Stacking*:** Melhor desempenho global de **82%**, beneficiando da heterogeneidade dos modelos base.
- ***Ensembles* (*Tree-based*):** Convergência num teto de 81% (RF, *XGBoost*, GBM).
- ***Baselines*:** Estagnaram nos 77%, limitados pela linearidade ou simplicidade.

| Modelo | *Accuracy* | *F1-Score* |
|---|---|---|
| ***Stacking*** | **0.82** | **0.82** |
| *XGBoost* | 0.81 | 0.81 |
| *Random Forest* | 0.81 | 0.81 |
| *Gradient Boosting* | 0.81 | 0.81 |
| ANN | 0.80 | 0.80 |
| *One-vs-Rest* | 0.80 | 0.80 |
| SVM | 0.79 | 0.79 |
| *Logistic Regression* | 0.77 | 0.77 |
| *Decision Tree* | 0.77 | 0.78 |

# Validação Externa e Robustez

A consistência entre a validação local e o *ranking* na competição Kaggle confirma a ausência de *overfitting* significativo.

**Métricas de Validação:**

| | |
|---|---|
| Estimativa Local: | **0.820** |
| Public Score (30%): | **0.835** |
| Private Score (70%): | **0.813** |

**Análise de Erros:**

- O modelo demonstra um comportamento **conservador**.
- A maioria dos erros ocorre entre **classes adjacentes**, o que é menos penalizador num contexto real de tráfego.

# Interpretabilidade do Modelo (SHAP)

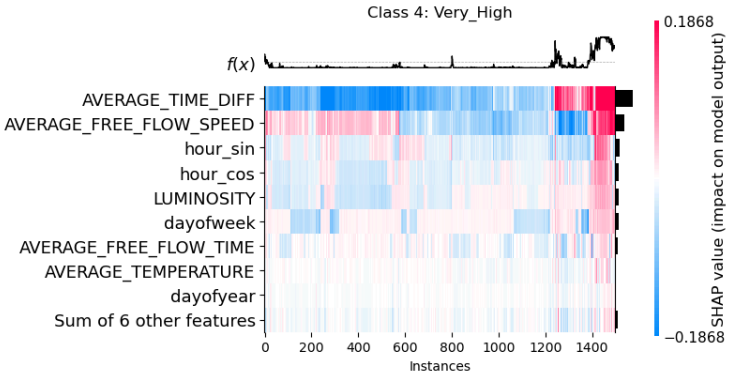
Apesar da complexidade, a análise SHAP revela que o modelo "aprendeu" os pontos chave do problema.

**1. Dominância Causal**

- `AVERAGE_TIME_DIFF` é o preditor transversalmente dominante. O modelo entende que o atraso é a definição direta de congestionamento.

**2. Modulação Contextual**

- Nas classes intermédias, o modelo usa a `LUMINOSITY` e a `hour` para refinar a probabilidade.



# 7 . Conclusão