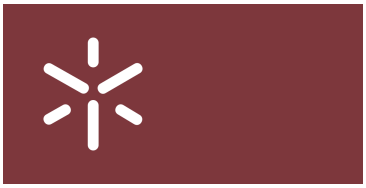


Universidade do Minho
Escola de Engenharia
Licenciatura em Engenharia Informática

Unidade Curricular de Aprendizagem e Decisão Inteligentes

Ano Letivo de 2024/2025

André Carvalho a100818
Enzo Vieira a98352
Gustavo Barros a100656

Maio, 2025

ADI

Índice

1. Introdução

2. Dataset Empréstimos

2.1. Análise dos dados

O dataset de empréstimos consiste num ficheiro CSV com várias informações relativas a pedidos de empréstimos e dados dos solicitantes.

Coluna	Descrição
date	Data de nascimento
person_name	Nome da pessoa
person_age	Idade da pessoa
person_gender	Gênero da pessoa
person_education	Nível de escolaridade
person_income	Rendimento anual
person_emp_exp	Anos de experiência profissional
person_home_ownership	Estado de moradia (ex.: aluguel, próprio, hipoteca)
loan_amnt	Valor do empréstimo solicitado
loan_intent	Finalidade do empréstimo
loan_int_rate	Taxa de juro do empréstimo
loan_percent_income	Empréstimo como percentual do rendimento anual
cb_person_cred_hist_length	Duração do histórico de crédito (anos)
credit_score	Pontuação de crédito
previous_loan_defaults_on_file	Indicador de inadimplências anteriores
tax	Indicador fiscal do solicitante
loan_status	Status do empréstimo: Aprovado / Pendente / Rejeitado

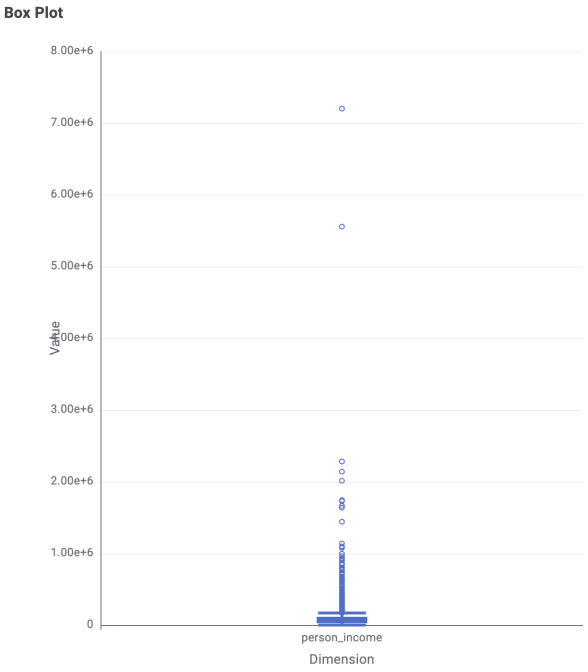
2.2. Visualização

Antes de iniciar o pré-processamento, realizámos um estudo exploratório do dataset de Empréstimos, usando nodos como Box Plot, Data Explorer, Scatter Plot e Bar Chart para visualização dos dados. A seguir apresentamos os principais achados:

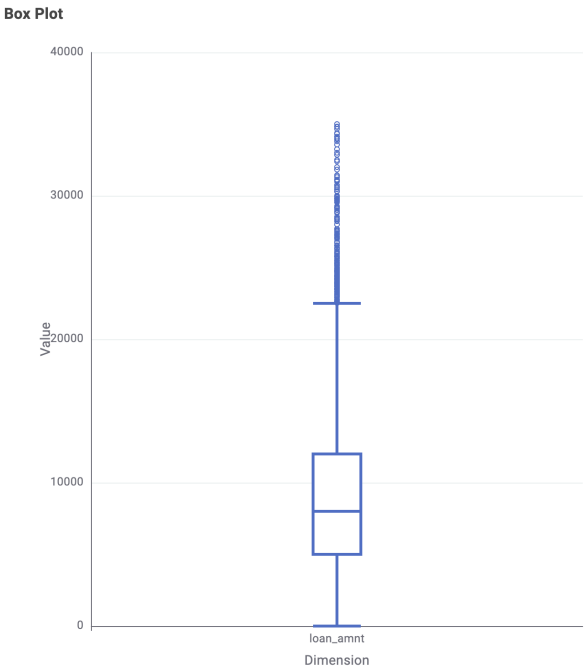
2.2.1. Box Plot

Por meio do Box Plot, foi possível examinar a distribuição de cada variável do conjunto de dados, evidenciar valores atípicos (outliers) e revelar padrões específicos em distintos grupos ou categorias.

Observamos que a variável person_income apresenta outliers em ampla amplitude, com valores extremamente discrepantes entre si.

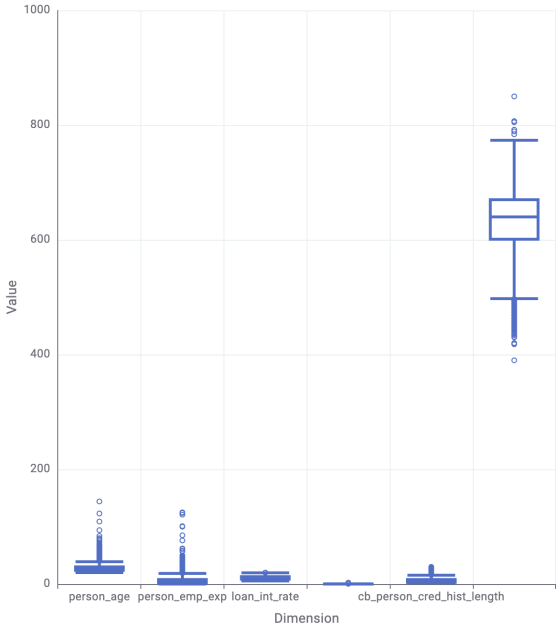


Similarmente acontece com o *loan_amount*, onde há uma grande quantidade de outliers.



Outros dados também apresentavam outliers que devem ser tratados adequadamente.

Box Plot



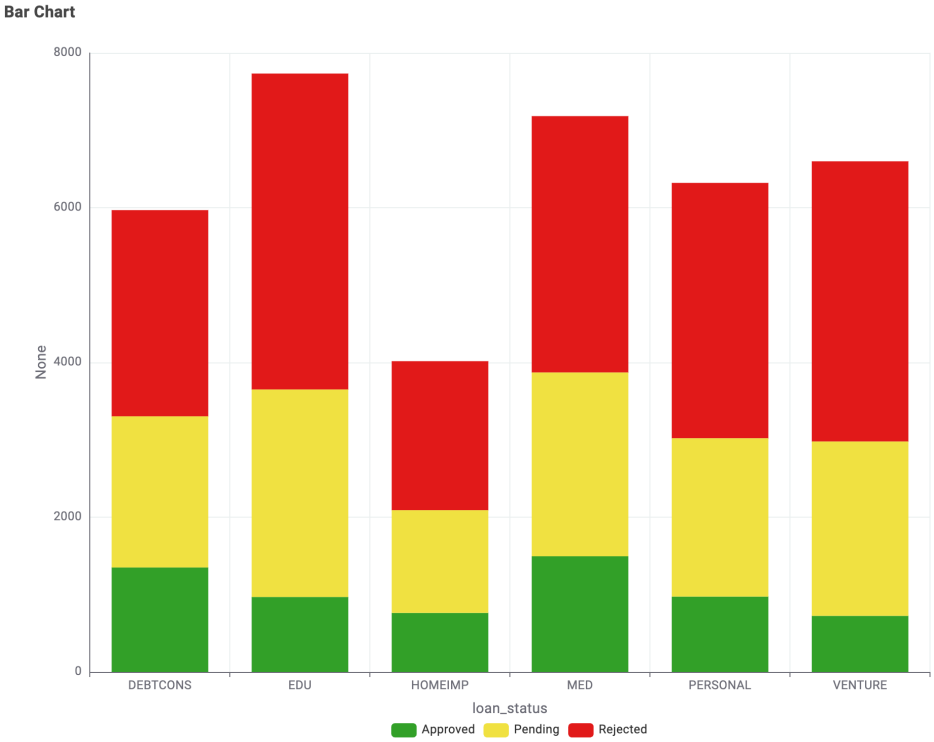
2.2.2. Scatter Plot

Uma das hipóteses sobre os dados era que o status de um empréstimo estava relacionado com a idade do solicitante, o que não parece ser verdade com base na seguinte visualização:

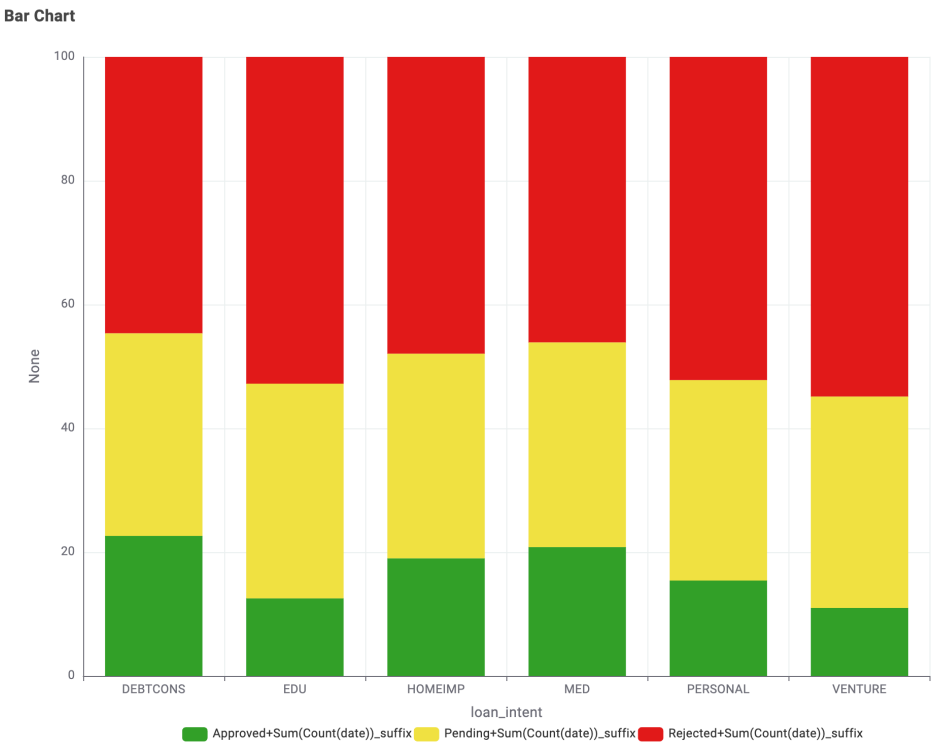
Scatter Plot

2.2.3. Bar Chart

No primeiro bar chart, plotámos a contagem absoluta de cada status de empréstimo (Aprovado, Pendente, Rejeitado) para cada categoria de finalidade do empréstimo (*loan_intent*). Esse gráfico evidencia o volume bruto de empréstimos em cada combinação de intenção e resultado, permitindo comparações diretas de quantos casos – por exemplo – de "EDUCATION" foram aprovados, pendentes ou rejeitados.



Em seguida, apresentámos um bar chart de contagem relativa, em que cada coluna mostra a proporção percentual de cada status dentro do seu próprio grupo de *loan_intent*. Assim, conseguimos perceber não apenas o tamanho absoluto de cada categoria, mas também como se distribuem internamente as taxas de aprovação, pendência e rejeição em cada finalidade de empréstimo.



2.3. Processamento dos dados, técnicas e feature engineering

Antes de treinar os modelos, aplicamos uma pipeline de pré-processamento e criação de novas features para maximizar a informação disponível.

2.3.1. Limpeza

Iniciamos pela limpeza de valores ausentes, removendo todos os valores numéricos que estavam em falta, visto que a sua quantidade em relação aos dados era pequena.

2.3.2. Outliers

O tratamento de outliers seguiu com a remoção das linhas que tivessem valores atípicos ou a substituição desses valores pela média para àqueles que tivessem desvio padrão baixo.

2.3.3. Normalização

Alguns dados foram normalizados segundo o padrão min-max. Outros foram padronizados, visto que seguiam uma distribuição próxima à normal.

2.3.4. One-hot-encoding

A fim de melhorar o entendimento do modelo sobre valores categóricos, o one-hot-encoding foi útil para separar categorias de, por exemplo, género, *loan_intent*, *person_education*, entre outros.

2.3.5. Novos atributos

Implementamos etapas de construção de novos atributos, como calcular a idade dos solicitantes e ainda o credit quality score, que é uma multiplicação dos valores normalizados do credit score e person credit history length. Foram usados estes dois valores pois o credit score é um indicador direto do risco do crédito (quanto mais alto melhor chances de obter um crédito) e o person credit history length é também importante pois quanto mais longo maior confiança se pode ter no credit score.

2.4. O Pending

A previsão final incluía 3 valores: Accepted, Rejected e Pending.
O caso do “Pending” que piorava bastante a accuracy dos modelos, pois é um estado que depende muito de questões extra ao modelo, e não faria sentido este prever se um crédito se manteria em “Pending” ou não, contudo testámos mesmo com essa hipótese, e foi de notar que quando esta foi removida os modelos tiveram um aumento de accuracy enorme.
Para remover todas as linhas com Pending usámos um modelo com alta accuracy auxiliar treinado com todos os outros dados Accepted e Rejected para prever os Pending em Accepted ou Rejected. Após isto voltámos a juntar o dataset e só depois usámos os modelos principais.

2.5. Fluxos alternativos

Foram feitos vários fluxos alternativos com diferentes estratégias de tratamento de dados (outliers, a criação das novas variáveis, etc…) que foram testados numa primeira fase para ver quais seriam os que obteriam melhores resultados para serem usados no fluxo de dados final para testar com os modelos.

2.6. Modelos

Para o primeiro dataset usámos os seguintes modelos: Decision Tree, Gradient Boosted, Random Forest, Tree Ensemble, Logistic Regression, XGBoost Linear, k-Means, RProp MLP e Keras Network.

MODEL WITHOUT PENDING	COM OUTLIERS E PARTITIONING	SEM OUTLIERS E PARTITIONING	COM OUTLIERS E X-PARTITIONING	SEM OUTLIERS E X-PARTITIONING
Decision Tree	93.345%	93.632%	93.374%	93.376%
Gradient Boosted	94.136%	94.028%	94.036%	93.964%
Random Forest	94.565%	94.502%	94.407%	94.241%
Tree Ensemble	94.328%	94.41%	94.441%	94.349%
Logistic Regression	90.508%	89.888%	90.456%	90.674%
XGBoost Linear	90.384%	89.703%	90.366%	90.502%
k-Means	46.644%	73.54%	31.659%	37.477%
MLP	91.198%	91.259%	91.439%	91.43%
Keras Network	79.017%	78.51%	78.282%	79.916%

Figura 1: Tabela resultados Dataset loans

O modelo com o melhor resultado foi o Random Forest que obteve acima de 94% em todos os testes.
O pior modelo foi o k-Means ainda de notar que provavelmente contém overfitting quando testado sem outliers e apenas com partitioning pois apresenta uma percentagem de 73.54% que baixa para 37.477 quando testado com cross validation.

MODEL WITH PENDING	COM OUTLIERS E PARTITIONING	SEM OUTLIERS E PARTITIONING	COM OUTLIERS E X-PARTITIONING	SEM OUTLIERS E X-PARTITIONING
Decision Tree	57.74%	57.601%	57.389%	57.047%
Gradient Boosted	61.367%	59.895%	60.876%	60.543%
Random Forest	59.65%	58.787%	59.719%	59.18%
Tree Ensemble	60.079%	59.038%	59.52%	59.372%
Logistic Regression	59.751%	58.655%	59.05%	58.9%
XGBoost Linear	59.661%	58.629%	58.973%	58.871%
k-Means	32.407%	42.716%	29.718%	31.185%
MLP	59.412%	59.736%	59.574%	59.283%
Keras Network	Não avaliado	Não avaliado	Não avaliado	Não avaliado

Figura 2: Tabela resultados Dataset loans

O pior modelo foi novamente o k-Means que obteve um resultado de 29.718%, contudo é que a configuração em que possivelmente existiria overfitting deste modelo, parece não existir mais agora que se tem que prever o valor de Pending, pois a queda para a utilização de cross validation foi de 10%.
O melhor modelo foi o Gradient Boosted.

Como é possível reparar há uma queda muito significativa de performance dos modelos quando estes têm que prever algo incerto como o estado “Pending” de uma loan, pois este é um estado que muitas vezes não está dependente dos dados fornecidos mas sim de eventos extra registo que não podem ser representados.

2.6.1. Avaliação

Como apresentado na sessão anterior, apresentamos a comparação de desempenho dos modelos treinados, identificando o melhor e o pior em termos de métricas como Accuracy, Precision, Recall e AUC-ROC.

Analisamos também a curva ROC para avaliar a trade-off entre sensibilidade e especificidade de um dos modelos, além de indicar o limiar de decisão mais adequado.

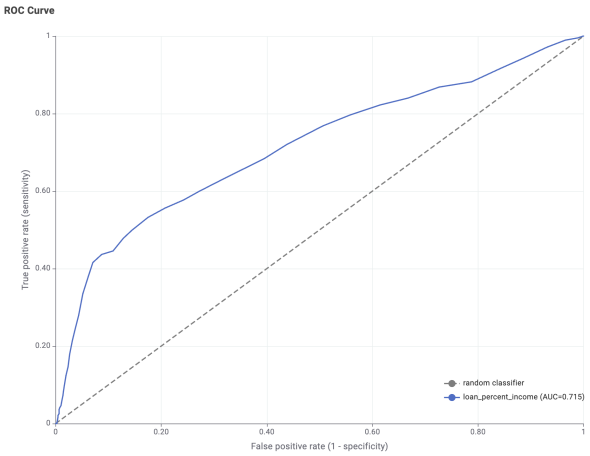


Figura 3: Curva ROC da árvore de decisão

3. Dataset de previsão do preço de carros

3.1. Modelo

3.2. Análise dos dados

O dataset de previsão do preço de carros consiste num ficheiro CSV com várias informações relativas a carros e contém as seguintes colunas:

Coluna	Descrição
Id	Identificador de cada carro
Price	Preço do carro
Levy	Imposto sobre o carro
Manufacturer	Marca do carro
Model	Modelo do carro
Production Year	Ano de produção do carro
Engine volume	Volume do motor
Category	Categoria do carro (ex.: Jeep, Sedan, etc)
Leather Interior	Se o interior do carro é em couro
Fuel Type	Tipo de combustível do carro
Mileage	A quantidade de quilómetros que o carro tem

Cylinders	O número de cilindros
Gear Box	O tipo de caixa de velocidades (ex.: Manual, Automático, etc)
Drive Wheel	Tração do carro
Doors	Número de portas
Color	Cor do carro
Airbags	Quantidade de airbags

Este dataset não apresentava missing values, então, através de um script em python foram gerados missing values aleatórios ao longo do dataset, totalizando uma percentagem de cerca de 5% do dataset total.

3.3. Visualização

Tal como no dataset anterior, começamos por explorar os nossos dados em busca de anomalias utilizando métodos semelhantes. Encontramos então as seguintes peculiaridades.

3.4. Box Plot

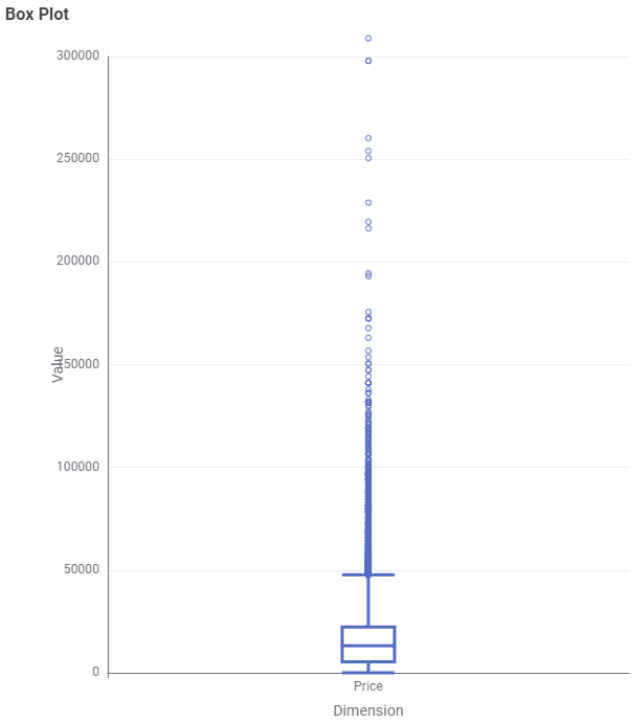


Figura 4: Box Plot do preço dos carros

Ao analisar o box plot do preço percebemos que existem bastantes outliers extremos, sendo que, mesmo com uma janela de visualização entre 0 e 300 000 não eram observáveis. Nenhuma das outras variáveis numéricas apresentava outliers que necessitassem de ser tratados

Contudo, ao olhar para as restantes colunas, percebemos que o levy (imposto), o Engine Volume (Volume do motor) e a quilometragem do carro eram ambos strings.

Fomos então forçados a tratar esses dados antes de continuar com a análise e deparamo-nos com os seguintes resultados:

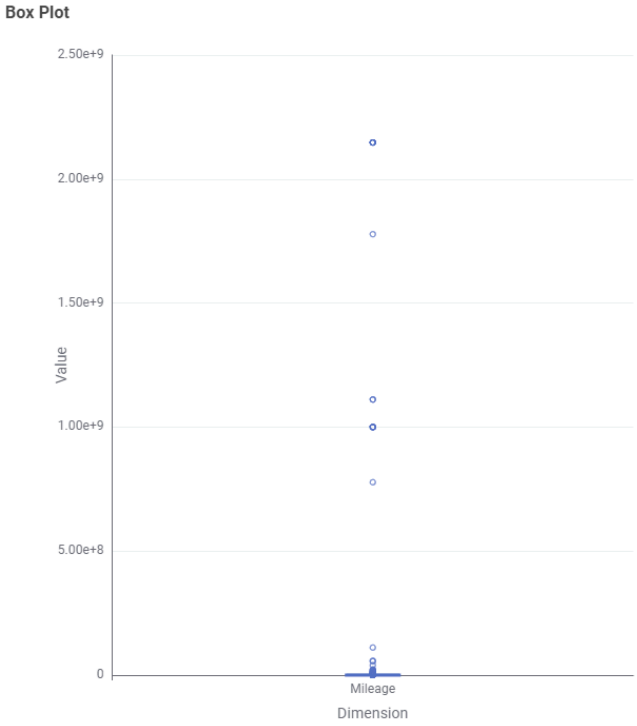


Figura 5: Box Plot da quilometragem dos carros

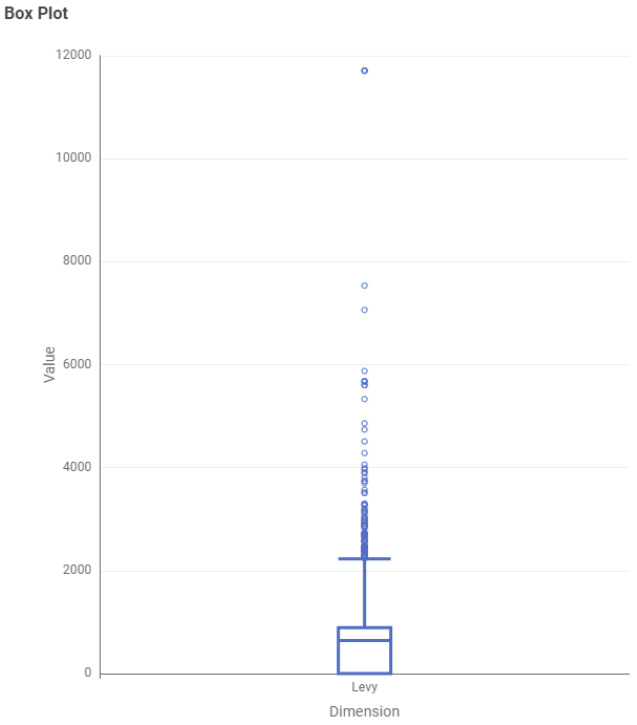


Figura 6: Box Plot do imposto dos carros

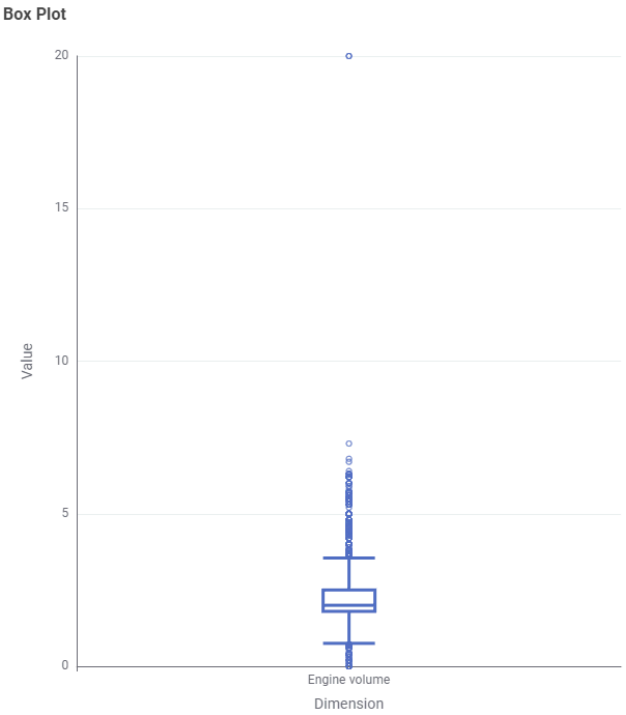


Figura 7: Box Plot do volume do motor dos carros

Tanto a quilometragem como o imposto apresentam alguns outliers extremos e alguns bastante próximos a valores reais que teriam de ser tratados mais tarde.

Para além disso foram encontradas entradas de carros com 20 cilindros, coisa que, é bastante irrealista e por isso também sofrerá tratamento.

3.5. Processamento dos dados, técnicas e feature engineering

3.5.1. Limpeza

Tal como foi referido anteriormente o Levy e a Mileage eram strings, o levy ocasionalmente apresentava um carácter "-" para representar 0, então utilizamos uma rule engine para alterar todos os caracteres para um 0. A mileage tinha um "km" depois do número de quilómetros que foi também removido utilizando uma rule engine. Por fim o Engine Volume, por vezes, apresentava um "Turbo" à frente do volume. Foi utilizada uma outra rule engine para criar uma nova coluna "isTurbo" que apresentava 1 caso o motor fosse turbo. Depois através da String Manipulation foi retirado o turbo da coluna Engine Volume. Todas as entradas que continham mais do que 8 cilindros tiveram este valor substituído por 8 para que estas tenham um valor mais realista.
Em seguida todas estas colunas foram convertidas para números e passou-se ao tratamento dos *Missing Values*.
Primeiramente removeu-se todas as colunas que não apresentavam o preço e o id. Os restantes valores foram substituídos pela mediana, caso fossem valores numéricos ou pelo valor mais comum caso fossem strings. Todos os Manufacturers com menos de 30 entradas foram convertidos em "Other".
Por fim converteu-se o ano do carro na sua idade para que o modelo a conseguisse processar melhor e passou-se à remoção de duplicados.

3.5.2. Outliers

O tratamento de outliers baseou-se na remoção de todos os outliers acima e abaixo dos seus respetivos limites uma vez que estes não representavam uma parte significativa do nosso dataset. Para além disto utilizou-se um filtro de linhas cujo preço fosse abaixo de 500, uma vez que, todos estes valores são extremamente irrealistas.

3.5.3. One-hot-encoding

Foi utilizado o one-hot-encoding para quatro colunas, sendo elas a Category, Fuel Type, Drive Wheel e Gear Box type uma vez que não apresentam uma cardinalidade alta.

3.5.4. Target Encoding

Para a coluna do Modelo que apresenta uma cardinalidade alta, foi utilizado o Target encoding que substitui o modelo pelo preço médio reduzindo assim a dimensionalidade e mantendo a relação com a variável alvo.

3.5.5. Leave one out encoding

Num outro tratamento foi utilizado este encoding tanto para o modelo como para a marca. Esta técnica é uma variação do Target Encoding porém, ao calcular a média, exclui a própria linha, evitando assim que o modelo aprenda com o seu próprio valor.

3.5.6. Transformação de colunas

No segundo tratamento a coluna de Mileage foi transformada em Preço por quilómetro de forma a tornar esse valor mais simples de interpretar.

3.6. Modelos

Para este dataset foram utilizados seguintes modelos: Linear Regression, Gradient Boosted Tree e Random Forest.

3.6.1. Avaliação

A seguir, apresentamos os modelos com melhor e pior desempenho com base na métrica R^2, utilizada para avaliar a eficácia dos modelos na tarefa de regressão.

De entre os 12 modelos testados — sendo que 6 receberam o tratamento com Target Encoding e os outros 6 com Leave-One-Out Encoding — o modelo que obteve melhor desempenho foi o Gradient Boosted Tree Learner com X-Partitioning, utilizando o segundo tratamento, alcançando um R^2 de 0.968.

Por outro lado, o modelo com pior desempenho foi a Regressão Linear com Target Encoding (primeiro tratamento), que obteve um R^2 de 0.467.

Dois modelos apresentaram valores negativos de R^2, indicando desempenho inferior ao de um modelo base (média), e por isso não foram considerados na comparação final.

DATASET CARROS				
	Model	Tipo de Limpeza	Tipo de Tratamento	Erro ou Accuracy
1	Linear Regression	Missing Values: Numéricos -> most frequent value string -> most frequent value double -> median string to number: Levy; Engine volume; Mileage Row Filter: remover linhas sem id ou price; remover price < 500 Column Filter: remover Leather interior; Doors; Wheel; Color Duplicate row filter Outliers:	tratar inconsistências one-hot-encoding groupby: Preço médio por tipo de modelo Normalizer: min-max Levy; Engine Volume; Mileage; Cylinders; Airbags; Age	R^2 = 0.467
2	Gradient Boosted Tree Learner			R^2 = 0.666
3	Random Forest			R^2 = 0.727
4	Linear Regression (x-partitioner)			R^2 = 0.472
5	Gradient Booted Tree Learner (x-partitioner)			R^2 = 0.673
6	Random Forest (x-partitioner)			R^2 = 0.739
7	Linear Regression	Missing Values: Numéricos -> most frequent value string -> most frequent value double -> median string to number: Levy; Engine volume; Mileage Row Filter: remover linhas sem id ou price; remover price < 500 Column Filter: remover Leather interior; Doors; Wheel; Color Duplicate row filter Outliers: remover extremos de Price; Levy; Mileage	tratar inconsistências one-hot-encoding groupby: Preço médio por tipo de modelo Sum por tipo de Manufacturer Count id por tipo de Manufacturer Normalizer: min-max Levy; Engine Volume; Mileage; Cylinders; Airbags; Age	R^2 = -0.53
8	Gradient Boosted Tree Learner			R^2 = 0.957
9	Random Forest			R^2 = 0.908
10	Linear Regression (x-partitioner)			R^2 = -0.086
11	Gradient Booted Tree Learner (x-partitioner)			R^2 = 0.968
12	Random Forest (x-partitioner)			R^2 = 0.925

Figura 8: Resultados dos modelos

4. Conclusão

Ao longo deste trabalho, aplicamos um fluxo completo de pré-processamento, tratamento e visualização de dados — incluindo limpeza de missing values, tratamento de outliers, feature encoding e feature engineering — e treinamos diversos modelos (árvores de decisão, florestas aleatórias, regressão linear, redes neuronais, k-means). Identificamos o melhor modelo e o pior modelo dos dois tipos de datasets, cujo os resultados foram positivos, obtendo +90% de Accuracy e R^2 superior a 0.9 nos modelos de regressão lineares.

Como planos futuros, propomos transformar o modelo de regressão linear (preço de carros) em um classificados. Uma estratégia é discretizar a feature alvo em faixas de preço (preço baixo, preço médio e preço alto) e aplicar thresholds às previsões.