ISLA Santarém

Propriedades em Portugal – Propriedades para venda

Relatório de Projeto Aplicado em Data Science realizada(o) no âmbito da Pós-Graduação em Data Science

Junho de 2024

Resumo

Este projeto visa desenvolver um modelo para prever o preço de venda de imóveis utilizando variáveis endógenas (área, tipologia, condição) e exógenas (concelho, distrito, região). As principais questões de investigação debruçam-se sobre a possibilidade de prever com precisão o preço de um imóvel numa determinada localização utilizando técnicas de *Machine Learning*.

Os objetivos incluem prever preços com base nas características dos imóveis e localização, identificar os melhores algoritmos de regressão, desenvolver um *dashboard* intuitivo para visualização dos dados e propor melhorias para a aplicação da ciência de dados no setor imobiliário.

A metodologia envolve a análise exploratória dos dados para identificar padrões, preparação dos dados, desenvolvimento de modelos de aprendizagem automática (como Regressão Linear e *Random Forest*), avaliação de desempenho, e criação de um *dashboard* interativo utilizando a biblioteca de P*ython Streamlit*.

Para criação do *dashboard*, e com o intuito de obter resultados fidedignos em relação aos valores apresentados, recorreu-se a algoritmos de classificação e a algoritmos de regressão, dos quais *Random Forest* e XGBRegressor tiveram o melhor desempenho, respetivamente.

Este projeto pretende fornecer ferramentas valiosas para o mercado imobiliário, promovendo previsões mais precisas e uma melhor compreensão dos fatores que afetam os preços dos imóveis, contribuindo para um mercado mais equilibrado e informado.

Palavras chave: algoritmos de regressão; algoritmos de classificação; *data science*; *dashboard*; imobiliário; portugal

Página em branco

Abstract

This project aims to develop a model that predicts the sale price of properties using endogenous (area, typology, condition) and exogenous (municipality, district, region) variables. The main research questions are whether it is possible to accurately predict the price of a property in a given location using Machine Learning techniques and.

The objectives include predicting prices based on property characteristics and location, identifying the best regression algorithms, developing an intuitive dashboard for data visualization and proposing improvements for the application of data science in the real estate sector.

The methodology involves exploratory data analysis to identify patterns, data preparation, development of automatic learning models (such as Linear Regression and Random Forest), performance evaluation, and creation of an interactive dashboard using the python library StreamLit.

To create the dashboard, and obtain faithful results in relation to the values presented, we used classification algorithms and regression algorithms, of which Random Forest and XGBRegressor had the best performance, respectively.

Its aim is to provide valuable tools for the property market, promoting more accurate forecasts and a better understanding of the factors that affect property prices, contributing to a more balanced and informed market.

Keywords: classification algorithms; regression algorithms; data science; dashboard; real estate; portugal

Página em branco

Agradecimentos

Página em branco

Índice

Lista de figuras

Lista de tabelas

Abreviaturas e Símbolos

KPI	*Key Performance Indicator*
NUTS II	*Nomenclature of Territorial Units for Statistics - Regions*

1. Introdução

A avaliação de propriedades imobiliárias é uma área de vital importância para diversos *stakeholders*, incluindo compradores, vendedores, agentes imobiliários, instituições financeiras e governos. No contexto português, onde o mercado imobiliário tem demonstrado uma volatilidade significativa e um crescimento substancial nos últimos anos, a precisão na estimativa dos preços das propriedades torna-se essencial para a tomada de decisões informadas. A integração de técnicas de ciência de dados (*data science*) tem emergido como uma abordagem inovadora e eficiente para enfrentar este desafio.

Contextualização do Mercado Imobiliário em Portugal

O mercado imobiliário português tem passado por diversas fases de transformação, influenciado por fatores económicos, demográficos e políticos. O aumento da procura por imóveis nas áreas metropolitanas, como Lisboa e Porto, aliado ao interesse crescente de investidores estrangeiros, tem impulsionado os preços das propriedades. Este cenário apresenta uma oportunidade e, simultaneamente, um desafio para a estimativa precisa dos valores imobiliários, devido à complexidade e dinamismo dos fatores envolvidos.

Enquadramento do tema

A compra de venda de propriedades em Portugal passou por várias fases nos últimos anos, refletidas tanto nas condições económicas nacionais como no impacto a nível global.

Entre 2008 e 2014, durante a crise financeira global, os preços das propriedades desceram significativamente e o número de transações diminuiu. Com a intervenção da Troika em 2011, entraram em vigor algumas medidas de austeridade que afetaram mais o mercado imobiliário, aumentando as execuções hipotecárias e diminuindo o crédito disponível.

A partir de 2014, começou a haver alguns sinais de recuperação económica com uma subida dos preços dos imóveis no mercado. Lisboa e Porto tornaram-se destinos populares para o investimento imobiliário. O aumento do interesse turístico também teve um impacto significativo. Muitas propriedades foram convertidas em alojamentos locais, aumentando a procura por imóveis.

O início da pandemia, em 2020, trouxe incerteza ao mercado imobiliário. Os movimentos diminuíram nos primeiros meses de confinamento. A procura por propriedades com maior área e fora dos centros metropolitanos aumentou, impulsionado pela tendência do teletrabalho.

Neste projeto são analisados dados inerentes à venda de imóveis em Portugal, tentando prever tendências e preços de venda numa determinada zona do país.

2. Problema e objetivos

O projeto sobre o qual incide este relatório visa determinar o preço futuro (previsional) de venda de um imóvel, consoante um determinado conjunto de variáveis endógenas (área, tipologia, condição) e exógenas (concelho, distrito, região) associadas ao mesmo. Partindo de conjuntos de dados dispersos, pretende-se desenvolver um modelo de dados que auxilie no cálculo do custo de um imóvel, coadjuvando e tornando mais justo e transparente o mercado imobiliário. A questão de investigação sobre a qual se debruça este relatório será assim o de determinar se será possível prever com uma margem mínima de erro, o preço de um imóvel numa determinada localização.

Questão de investigação: É possível prever com uma margem mínima de erro o preço de um imóvel numa determinada localização, recorrendo a técnicas de *Machine Learning*?

Este projeto visa, assim, realizar uma análise exploratória aos dados provenientes do mercado imobiliário e a partir destes tecer uma série de considerações acerca de como os mesmos se poderão utilizar no contexto da ciência de dados (*data science*). Apresentam-se, em baixo, os principais objetivos associados a esta abordagem de investigação:

- Ser capaz de prever preços de imóveis, com base nas suas características físicas e localização;
- Identificar os algoritmos de regressão que melhores desempenhos apresentam na predição dos preços de imóveis;
- Desenvolver uma interface gráfica (*dashboard*) que permita de uma forma rápida e intuitiva obter informações de um imóvel numa determinada localização;
- Propor melhorias e tecer considerações acerca da forma como se poderá introduzir a ciência de dados (*data science*) no mercado imobiliário.

3. Descrição do conjunto de dados

Ficheiros no estado inicial

O conjunto de dados é composto por seis ficheiros "*.csv*" que contêm informações detalhadas sobre o mercado imobiliário em Portugal:

Codpostais: Este *dataset* contém informações sobre códigos postais em Portugal. É composto por três atributos principais: 'cod_postal', que apresenta códigos postais com 6 ou 7 algarismos; 'cod_concelho', que identifica os concelhos através de um código numérico; e 'cod_distrito', que representa os distritos também por meio de um código numérico. Este *dataset* abrange 325 concelhos e 29 distritos.

Concelhos: O *dataset* concelhos apresenta duas atributos: 'cod_concelho', o código numérico que identifica cada concelho de Portugal, e 'nome_concelho', que contém o nome do respetivo concelho. Este *dataset* inclui um total de 308 concelhos.

Distritos: Este *dataset* contém informações sobre os distritos de Portugal Continental, bem como dos arquipélagos dos Açores e Madeira. Os atributos presentes são 'cod_distrito', referente ao código numérico que identifica cada distrito, e 'nome_distrito', que indica o nome do distrito correspondente. O *dataset* cobre 29 distritos.

NUTS II: O dataset NUTS II fornece informações sobre as regiões NUTS II (*Nomenclature of Territorial Units for Statistics*) em Portugal. As suas categorias são 'Codigo', o código numérico que identifica cada região NUTS II, e 'Regiao', que apresenta o nome da região correspondente. Este *dataset* inclui 7 regiões NUTS II.

Portugal Imobiliário: Este *dataset* contém informações gerais sobre imóveis em Portugal. É composto por cinco categorias: 'Índice', que é um identificador numérico para cada entrada; 'Nome', que descreve o anúncio do imóvel, nele contém a informação respetiva ao tipo da propriedade, tipologia; 'Preço', que indica o preço do imóvel em euros; 'Área', que especifica a área do imóvel em metros quadrados; e 'Localização', que descreve a localização do imóvel. O *dataset* inclui um total de 17.215 entradas detalhadas sobre diferentes propriedades.

Portugal Propriedades: Este *dataset* fornece informações detalhadas sobre propriedades específicas em Portugal. As categorias incluídas são 'Location', que indica a localização da propriedade; 'Rooms', que especifica o número de quartos; 'Price', que apresenta o preço da propriedade; 'Area', que representa a área da propriedade; 'Bathrooms', que indica o número de casas de banho; 'Condition', que descreve a condição da propriedade (por exemplo, Usada, Renovada); 'AdsType', que especifica o tipo de anúncio (por exemplo, Venda, Aluguer ou Férias); e 'ProprietyType', que indica o tipo de propriedade (por exemplo, Casa, Apartamento). Este *dataset* contém um total de 62.658 entradas, oferecendo uma ampla visão sobre o mercado imobiliário em Portugal.

Portugal Propriedades

Em relação ao ficheiro 'portugal_propriedades', o número de casas de banho considera-se irrelevante neste estudo e por isso foi removido do *dataset*.

Limpeza dos dados

Em relação ao preço, e depois de verificados 1.170 valores nulos no *dataset*, os valores em falta foram substituídos por zero, para posterior tratamento.

Distribuição dos dados

Em relação à variável 'Location' tem 62.658 entradas, e define que as localizações com mais entradas são Fernão Ferro, Seixal e Setúbal.

Em relação ao preço, a média é de 483.297,78€, com desvio padrão superior a um milhão de euros, com uma distribuição assimétrica à direita (quando a média está muito longe da mediana, a distribuição considera-se assimétrica) com alta dispersão, influenciada por *outliers* extremos, indicando que há uma grande variação nos dados. Os *outliers* aumentam a média e o desvio padrão de forma "irreal" por serem tão discrepantes em relação à maioria dos valores. 50% dos dados encontram-se entre 165.000 e 545.000, uma faixa de valores relativamente estreita considerando o máximo.

Tabela 1 - Estatística descritiva da variável 'Preço' no *dataset* 'portugal_propriedades'.

Média	Desvio padrão	Mínimo	Q1	Mediana	Q3	Máximo
483.297,78	1.386.354,85	115	165.000	328.500	545.000	285.000.000

Tal como na é observado no preço, a variável da área também apresenta assimetria à direita, caracterizada pelas mesmas razões, isto é, *outliers* extremos e uma grande dispersão dos dados. A maior concentração de dados encontra-se entre 121 e 258 metros quadrados, enquanto o valor máximo é de 1,3 x 10^9 metros quadrados.

Tabela 2. Estatística descritiva da variável 'Área' no *dataset* 'portugal_propriedades'.

Média	Desvio padrão	Mínimo	Q1	Mediana	Q3	Máximo
21.175,48	5.193.457,81	1	121	180	258	$1,3 \times 10^9$

Portugal Imobiliário

Em relação ao ficheiro 'portugal_imobiliario', a coluna 'Índice' foi removida do *dataset* por não apresentar relevância ao estudo. Antes de proceder à descrição da distribuição dos dados do *dataset*, foi realizada uma limpeza dos dados.

Limpeza dos dados

Na variável 'preço' e, tendo em conta a existência de valores que se apresentam com a descrição 'Preço sob consulta', procedeu-se à remoção de espaços em branco e à substituição da descrição ('Preço sob consulta') por zero, tal como dos valores em falta por zero. Assim, o número de propriedades sem preço é de 1.213.

Em relação à variável área, foram substituídos espaços embranco, e ',' por '.', no sentido de fazer conversão para *float*, para melhorar a precisão, facilitar operações estatísticas e melhorar o tratamento dos dados. Em ambas as variáveis foram removidas as unidades, isto é, € e m^2.

Distribuição dos dados

As estatísticas descritivas dos valores de preço e área no conjunto de dados "portugal_imobiliario" apresentam-se abaixo.

Tabela 3. Estatística descritiva da variável 'Preço' no dataset 'portugal_imobiliário'.

Média	Desvio padrão	Mínimo	Q1	Mediana	Q3	Máximo
$4{,}65 \times 10^5$	$4{,}37 \times 10^5$	0	$2{,}39 \times 10^5$	$3{,}45 \times 10^5$	$5{,}3 \times 10^5$	1×10^7

Tabela 4. Estatística descritiva da variável 'Área' no dataset 'portugal_imobiliário'.

Média	Desvio padrão	Mínimo	Q1	Mediana	Q3	Máximo
120,47	978,58	0,11	79,425	103	133	$1{,}27 \times 10^5$

Depois de apresentados os valores médios e uma perspetiva da distribuição dos dados em cada um dos *datasets*, e seguindo com o objetivo de definir e caracterizar os mesmos por tipologia, foram removidas as linhas que não contêm informação acerca da tipologia da propriedade. Assim, removeram-se 852 linhas do *dataframe* inicial.

Foi feita uma limpeza da variável '*Condition*', removendo entradas com o valor *'In ruin'*, substituindo valores em falta por 'Falta' e padronizando os valores restantes para 'Novo', 'Usado', 'P/ recuperar' e 'Renovado'.

As variáveis 'nome_concelho' e 'nome_distrito' foram extraídas da variável 'Localização' em ambos os *datasets*, 'imobiliario_df' e 'propriedades_df'.

Em seguida, as colunas 'cod_concelho', 'cod_distrito' e 'cod_nutsii' foram introduzidas no *dataset* imobiliario_df através da junção com os *datasets* 'concelhos_df' e 'nuts_df'.

Por fim, variável condição foi extraída da variável 'Nome' no *dataset* imobiliario_df, classificando as entradas como 'Novo', 'Usado', 'Renovado' e 'P/ recuperar'.

Os *datasets* 'imobiliario_df' e 'propriedades_df' foram combinados num único *dataframe*, após a renomeação e exclusão de colunas para garantir a compatibilidade. Após a limpeza e padronização dos dados, os dois *dataframes* resultam num *dataframe* final com 77.858 linhas e 7 colunas (preco, area, tipologia, concelho, distrito, regiao, condicao).

Para preenchimento dos valores em falta, foram substituídos os valores em falta do preço pela média correspondente do preço por tipologia e distrito. Foram verificadas e removidas as linhas duplicadas, resultando na eliminação de 9329 linhas, correspondendo a 11.68% do *dataframe* inicial. Para o estudo, foram ainda excluídas entradas com áreas superior a 500m^2 e inferiores a 10m^2. Foram removidas 11.344 linhas do *dataframe* inicial que corresponde a 14,2% do seu tamanho inicial.

Distribuição dados por tipologia – ficheiro conjunto

Depois de combinados os dois *dataframes*, as estatísticas descritivas de cada tipologia em relação a preço e área estão representadas nos quadros abaixo.

Tabela 5 - Estatística descritiva da variável 'Preço' por cada uma das tipologias, no *dataset* combinado ainda com *outliers*.

Tipologia	Média	Desvio padrão	Mínimo	Q1	Mediana	Q3	Máximo
0	191.622,61	3.31×10^5	6.000	49.125	118.913	230.000	7.852.941
1	217.893,95	1.69×10^5	1.350	85.000	190.000	289.190	1.600.000
2	261.283,91	2.53×10^5	3.000	99.000	209.625	339.000	6.738.758
3	401.273,67	1.83×10^5	2.850	199.000	319.000	465.000	285.000.000
4	596.858,60	7.54×10^5	15.000	290.000	442.000	690.000	49.740.934
5	738.413,42	9.60×10^5	15.000	275.000	462.096	819.600	24.500.000
6	759.926,33	1.21×10^6	18.000	269.500	450.000	850.000	35.311.765
7	845.250,97	9.63×10^5	35.000	279.250	480.000	950.000	7.000.000
8	814.780,23	9.75×10^5	55.000	264.000	465.000	875.000	6.950.000
9	767.215,47	8.44×10^5	40.000	298.250	595.000	798.750	5.500.000
10	940.704,31	1.13×10^6	29.900	293.750	550.000	1.150.000	6.900.000

O valor médio da tipologia 0 é de 191.622€ e o da tipologia 10 de 94.0704€. A média dos preços aumenta à medida que a tipologia aumenta, o que indica que propriedades maiores ou com mais quartos tendem a ser mais caras. O desvio padrão é alto para todas as tipologias, indicando uma grande variabilidade nos preços das propriedades. As tipologias com maior desvio padrão são a 3 e a 4. Tipologias com menor desvio padrão são a 0 e 1.

Os quartis aumentam consistentemente com a tipologia, refletindo o aumento dos preços medianos. Para a tipologia 0, 50% das propriedades estão abaixo de 118.913€. Para a tipologia 10, 50% das propriedades estão abaixo de €550.000. A presença de preços máximos

extremamente altos em várias tipologias (especialmente em tipologias 3, 4, 5, 6) sugere a existência de *outliers* que podem distorcer a análise. Esses *outliers* devem ser investigados para verificar se são erros ou propriedades excecionalmente valiosas.

A mediana dos preços é geralmente muito menor do que a média, indicando uma distribuição assimétrica dos preços, possivelmente com uma cauda longa à direita (muitos preços altos).

Tabela 6. Estatística descritiva da variável 'Área' por cada uma das tipologias, no *dataset* combinado com *outliers*.

tipologia	Média	Desvio padrão	Mínimo	Q1	Mediana	Q3	Máximo
0	117.450195	91.870042	12.0	48.0000	85.0	156.000	499.00
1	66.992458	37.950565	13.5	48.0000	59.0	74.000	432.00
2	98.738035	47.737958	12.0	70.5800	89.5	111.455	491.00
3	164.234590	72.944186	15.0	114.7000	147.3	200.000	498.00
4	215.752383	83.060022	15.0	158.0000	200.0	260.000	497.00
5	247.250546	95.212779	36.0	180.0000	234.0	301.000	498.24
6	261.236827	97.463252	45.0	188.0000	250.0	332.000	499.00
7	278.553592	100.551272	65.0	200.0000	276.7	359.760	499.00
8	291.968382	99.000475	40.0	213.8475	298.5	375.000	488.40
9	265.543290	121.156187	27.0	166.0000	265.0	358.000	492.65
10	325.567277	101.182793	100.0	250.0000	335.0	406.480	499.00

Verificação e remoção de *outliers*

A verificação de *outliers* foi realizada nas variáveis preço, área e posteriormente e preço por área, isto é preço por metro quadrado, medida comummente utilizada no mercado imobiliário aquando da comparação entre imóveis.

Outliers Preço

A variável preço, contém um total de 4.375 *outliers*, correspondendo a 5,48% dos dados. As tipologias com mais *outliers* são em T3 e T4, seguidos de T2 e T5. Tendo em conta que a tipologia com maior representação é a de T3, seguida de T4, T2 e T5, é comum haver uma maior dispersão de dados num maior conjunto de entradas.

Tabela 7. Número de *outliers* da variável preço por tipologia.

T0	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10
78	104	605	1557	1270	459	166	55	37	19	25

A média dos preços diminuiu para todas as tipologias após a remoção de *outliers*. Isso é esperado, pois *outliers* são frequentemente valores extremamente altos/baixos que distorcem a média. Por exemplo, a média dos estúdios (tipologia 0) diminuiu de 191.622,61€ para 142.645,36€.

O desvio padrão diminuiu significativamente para todas as tipologias após a remoção dos *outliers*. Isso indica uma redução na variabilidade dos preços, tornando a distribuição de preços mais consistente. Por exemplo, o desvio padrão para a tipologia 3 caiu de 1.829.594,00€ para 185.464,21€.

Os valores mínimos permanecem relativamente inalterados, sugerindo que os *outliers* não estavam entre os menores preços enquanto que os valores máximos diminuíram drasticamente após a remoção dos *outliers*, confirmando que os *outliers* eram valores extremamente altos. Por exemplo, o valor máximo para a tipologia 3 caiu de 285.000.000 € para 862.538€.

Os quartis também sofreram ajustes, mas não tão drásticos quanto os valores máximos e a média.

Outliers Área

Na variável área, existe um total de 2.347 outliers, correspondendo a 2,94% dos dados. As tipologias com mais *outliers* são em T3 e T2, seguidos de T4 e T1.

Tabela 8. Número de *outliers* da variável área por tipologia.

T0	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10
65	222	756	842	417	45	0	0	0	0	0

A média da área diminuiu para todas as tipologias após a remoção de *outliers*. Por exemplo, a média para a tipologia 0 (estúdios) caiu de 117,45 para 104,72 metros quadrados. Isso sugere que os *outliers* aumentavam a média inicial.

O desvio padrão diminuiu significativamente para todas as tipologias após a remoção de *outliers*, indicando uma redução na variabilidade das áreas. Por exemplo, o desvio padrão para a tipologia 3 diminuiu de 72.94 para 60.18 metros quadrados. Os valores mínimos permanecem inalterados ou têm pequenas mudanças. Os valores máximos diminuíram significativamente após a remoção dos *outliers*. Por exemplo, o valor máximo para a tipologia 0 caiu de 499 para 318 metros quadrados.

Os quartis (25%, 50%, 75%) diminuíram ligeiramente após a remoção dos *outliers*, indicando uma distribuição de área mais estreita e menos influenciada por valores extremos.

Outliers Preço por Área

Tabela 9. Número de *outliers* da variável preço/m² por tipologia.

T0	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10
72	56	197	724	410	194	67	25	12	11	7

No estudo do preço por área, existe um total de 1.775 *outliers*, correspondendo a 2,22% dos dados. As tipologias com mais *outliers* são em T3 e T4, seguidos de T2 e T5.

O preço por área apresenta um valor de 2.228,63 €/m^2 quadrado com um desvio padrão de 1.362,39 euros por metro quadrado.

Antes da remoção de *outliers*, a mediana do preço por m^2 é relativamente constante entre as diferentes tipologias, com algumas variações notáveis em tipologias específicas. A largura das caixas (interquartil) varia entre as tipologias, indicando diferenças na dispersão dos preços por m^2, onde tipologias maiores tendem a ter maior dispersão.

Depois de removidos os *outliers*, a mediana dos preços por m^2 varia mais significativamente entre as tipologias, especialmente entre tipologias 0 e 1.

A dispersão dos preços por metro quadrado dentro das caixas é maior para tipologias menores (0, 1 e 2), indicando maior variabilidade nos preços por metro quadrado nessas tipologias.

A Tabela 10 resume as estatísticas descritivas do conjunto de dados tratado, que reúne as informações dos diferentes *datasets* após remoção de *outliers*.

Tabela 10. Estatística descritiva das variáveis numéricas.

Métrica	Preço (€)	Área (m^2)	€/ m^2
Média	3,34 x 10^5	162,37	2.228,63
Desvio Padrão	2,33 x 10^5	85,46	1.362,39

Visualização de dados

O mapa de calor que correlaciona as variáveis entre si revela uma maior relação entre as variáveis tipologia e área (0,63) sugerindo que, quanto maior for a tipologia do imóvel, maior será a sua área. À medida que a tipologia aumenta, o número de assoalhadas/quartos aumenta, enfatizando esta correlação positiva.

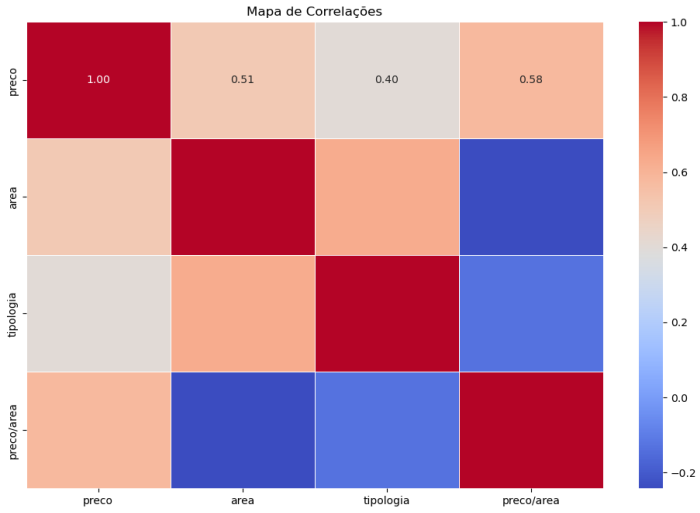

Figura 1. Mapa de calor da correlação entre variáveis quantitativas

A partir da interpretação do gráfico de barras seguinte, é possível ver que existe um maior número de casas à venda no distrito do Porto (18,85%), valor significativamente mais alto em comparação com os restantes distritos. A este, seguem-se Lisboa, Setúbal e Braga. A aproximação dos valores de Lisboa e Setúbal reforçam a relação que estes dois distritos mantêm, visto que o distrito de Setúbal aloja muitos residentes cuja situação laboral está sedeada no distrito de Lisboa.

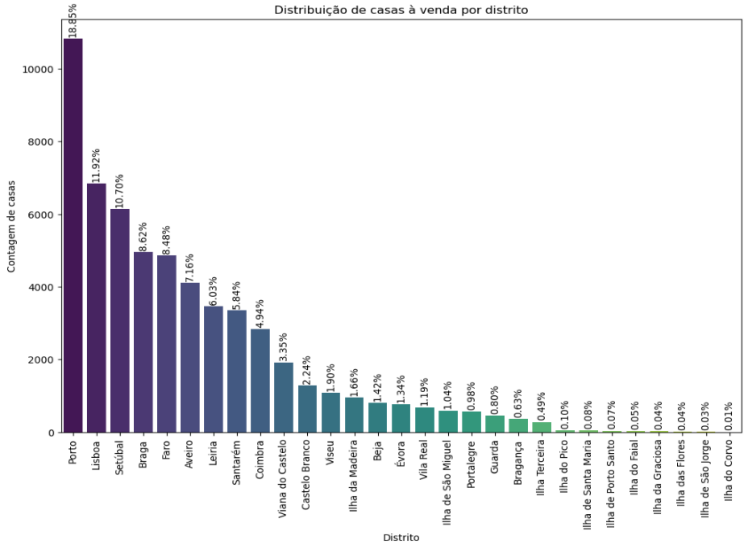

Figura 3. Gráfico de barras da distribuição de casas à venda por distrito.

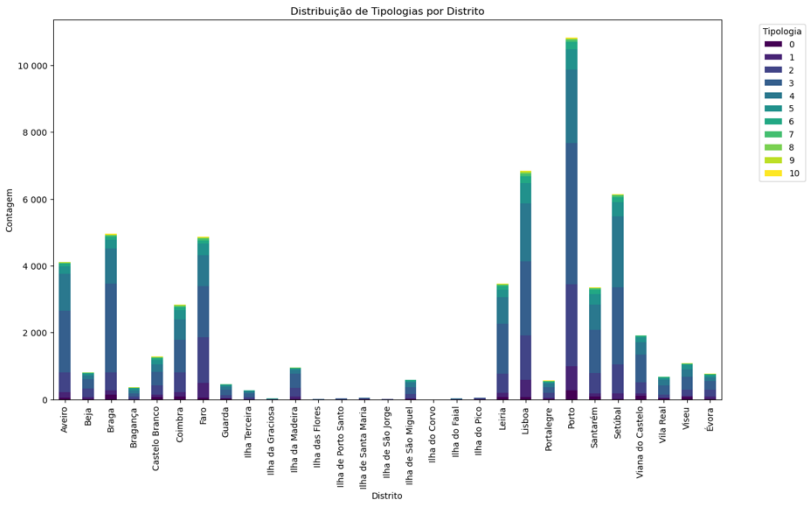

Figura 2. Gráfico de barras da distribuição de tipologias por distrito.

Para além do número de casas à venda por distrito, é apresentada a distribuição de tipologias por região, onde é possível observar uma maior heterogeneidade entre todas as tipologias, nos mesmos distritos cuja representatividade se manifesta. Isto é, nas ilhas e em distritos que se localizam no interior do país, a diversidade de tipologias é pouca enquanto nos distritos das grandes cidades mencionadas acima, existe um número significativamente maior de casas à venda, e de todas as tipologias.

Como visto anteriormente e pela observação da Figura 4, aquando da remoção de *outliers* por tipologia, a tipologia que tem maior representatividade é a T3 (38,58%), seguida de T4 (22,90%), T2 (19,37%) e T5(7,17%).

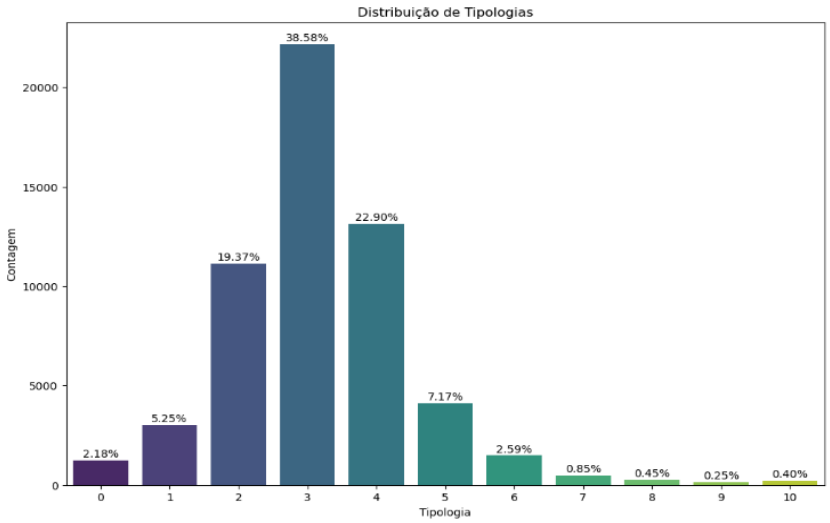

Figura 4. Gráfico de barras da distribuição de casas à venda por tipologia.

Em seguida, é possível verificar, através do *boxplot* da **Erro! A origem da referência não foi encontrada.**, que compara a tipologia com o preço que, à medida que a tipologia aumenta, o preço também aumenta. É possível observar também que o primeiro quartil não varia muito, mas que o terceiro quartil tem tendência para aumentar, à medida que a tipologia aumenta, representando uma maior dispersão de valores, e por isso um intervalo maior de preços para a mesma tipologia, em tipologias maiores.

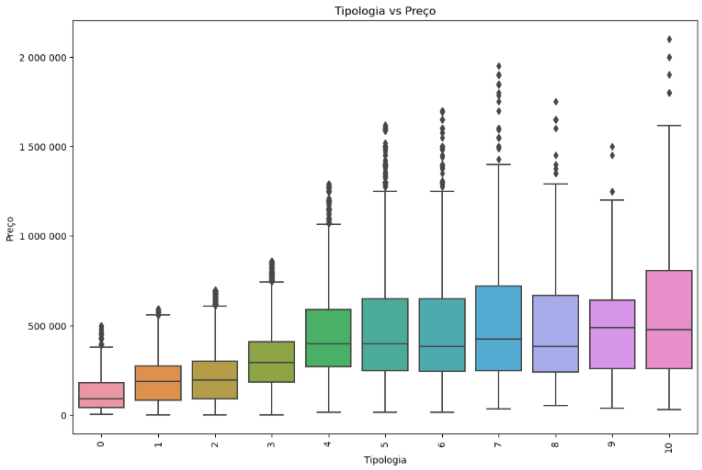

Figura 5 – *Boxplot* do preço por tipologia

Os *scatterplots* ilustrado na **Erro! A origem da referência não foi encontrada.Erro! A origem da referência não foi encontrada.**, para representa os valores e relação da área com o preço. Existe uma elevada densidade até aos 200 m^2 e dispersando a partir desse valor, não só no aumento do valor do preço mas também na quantidade de propriedades (muito menor) disponíveis com área superior à mencionada.

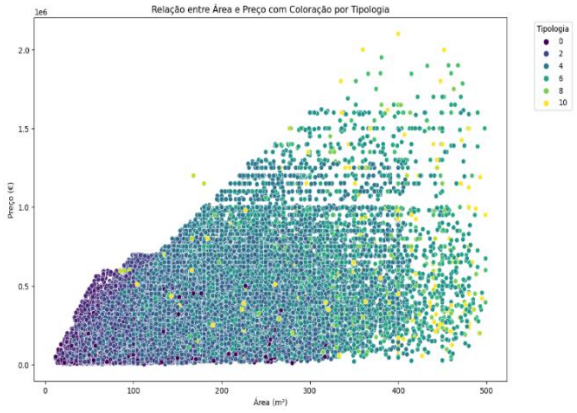

Figura 6 - *Scatterplot* da variável 'Preço' por 'Área' e por tipologia

Na Figura 7, oberva-se um gráfico de densidade da distribuição de preços, que demonstra a assimetria mencionada e confirmada, com um pico alto de valores antes dos 500.000€ e uma descida abrupta que estabiliza a partir dos 1.000.000€.

Em relação à variação de preços por região (que engloba vários distritos), as regiões da Área Metropolitana de Lisboa e o Algarve apresentam uma posição semelhante nos seus valores e intervalos, seguidas do arquipélago da Madeira, o que equipara a área metropolitana a destinos turísticos.

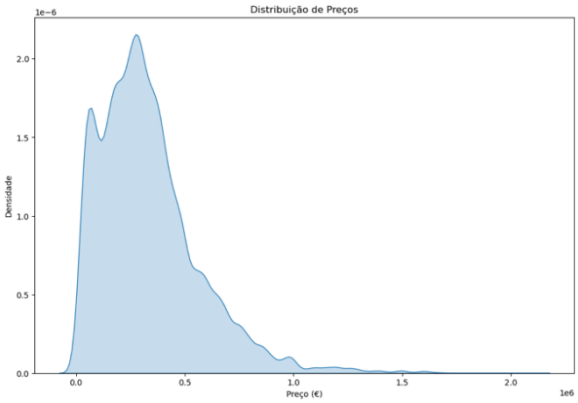

Figura 7. Gráfico de densidade da distribuição da variável 'Preço'.

Na distribuição dos preços por tipologia, é possível identificar maior densidade em valores próximos nas tipologias mais baixas e uma menor densidade e maior espaçamento entre preços nas tipologias mais altas. Em relação aos preços por região, existe uma densidade com condensação de valores em regiões como o Centro ou o Alentejo e 'caudas' mais finas, isto é, menor densidade e maior dispersão de valores nas regiões cujos preços (representados anteriormente) são superiores.

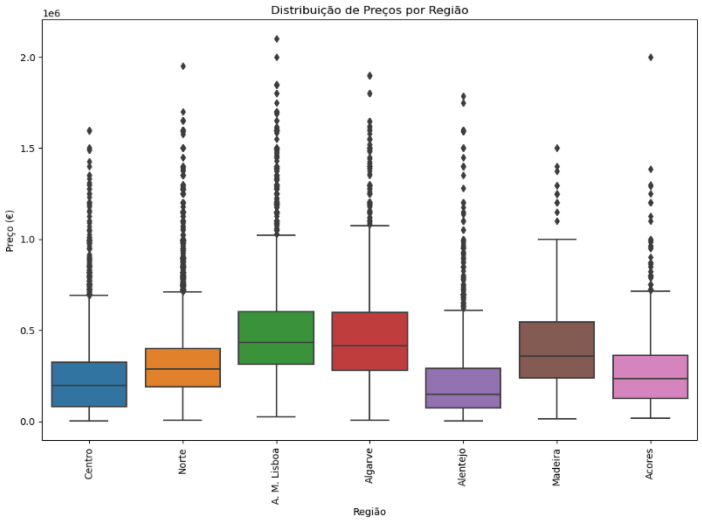

Figura 8. Boxplots da distribuição de preços por região.

Os gráficos de violino mostram a distribuição completa dos dados. A forma do "violino" representa a estimativa de densidade de kernel (KDE), que exibe a probabilidade de os dados assumirem certos valores. Isso permite visualizar onde os dados se concentram e identificar a presença de múltiplos picos (modos).

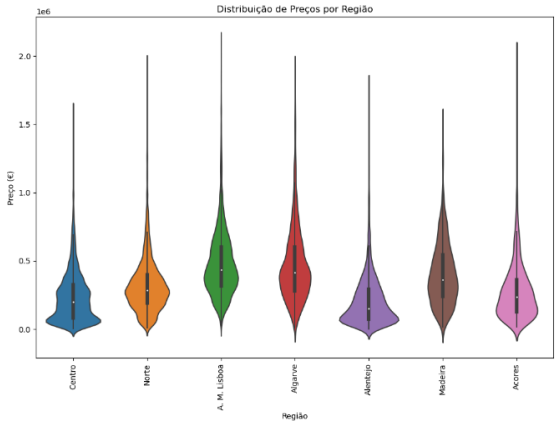

Figura 9 - Gráficos de violino da distribuição de preços por região.

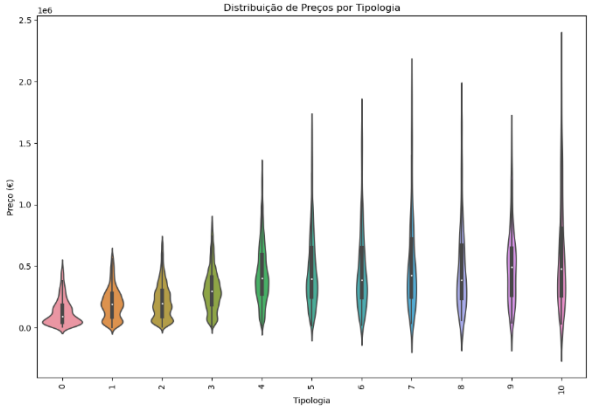

Figura 10 - Gráficos de violino da distribuição de preços por tipologia.

4. Métodos de acesso ao armazenamento

Em *Data Science*, a gestão eficiente de grandes volumes de dados é essencial para análises e processamento eficazes. Os métodos de acesso ao armazenamento de dados são as técnicas e protocolos que permitem ler, gravar e gerir esses dados nos diversos tipos de armazenamento disponíveis, garantindo que a informação esteja acessível e organizada para o trabalho do cientista de dados.

Exemplos de métodos de acesso em *Data Science* incluem bibliotecas Python como Pandas e Dask, que facilitam a leitura e gravação de dados em diversos formatos e armazenamentos. A linguagem SQL é ideal para consultas estruturadas em bancos de dados relacionais, enquanto bancos de dados NoSQL como MongoDB e Cassandra são adequados para dados não estruturados e semi-estruturados. As APIs de armazenamento em ambiente *cloud*, como AWS S3 e Google Cloud Storage, oferecem escalabilidade e integração com serviços em nuvem.

A escolha do método de acesso ideal depende das necessidades do projeto, considerando o tipo de dados, volume, frequência de acesso, requisitos de desempenho e orçamento. Ao compreender as opções disponíveis, o cientista de dados pode tomar decisões informadas para otimizar o armazenamento e acesso aos dados, garantindo a eficiência e sucesso das suas análises.

No ficheiro "portugal_imobiliario.ipynb", diversos métodos de acesso ao armazenamento de dados são utilizados para realizar a análise e o processamento dos dados imobiliários em Portugal. Os dados são lidos a partir de ficheiros CSV utilizando a função *pd.read_csv()* do Pandas. Especificamente, são carregados ficheiros referentes a imóveis, propriedades, concelhos, distritos e NUTS II, com parâmetros adicionais para definir o separador e a codificação.

5. Métodos de análise de dados

Neste capítulo é feita uma análise aos conceitos tecnológicos associados à execução deste projeto, incidindo sobre a classificação e a regressão.

Algoritmos de Classificação

Classificação é uma tarefa essencial no campo do *Machine Learning* e consiste em atribuir rótulos ou categorias a amostras com base nas suas características. É um processo de treinar um modelo para reconhecer e generalizar padrões encontrados nos dados de treino, de modo que possa prever corretamente a classe de novas amostras não rotuladas. Existem vários modelos de classificação disponíveis, cada um com suas próprias características e eventuais aplicações. Apresentam-se, de seguida, alguns dos principais algoritmos de classificação usados no decurso deste nosso projeto:

Naive Bayes

Naive Bayes é um algoritmo de classificação probabilística fundamentado no teorema de *Bayes*. Este modelo pressupõe que as características são independentes umas das outras, condicionadas ao valor da classe. Sua eficiência computacional torna-o particularmente eficaz em conjuntos de dados com alta dimensionalidade. O algoritmo de classificação *Naive Bayes*, sendo supervisionado, é adequado para categorizar variáveis (sejam elas multiclasse ou binárias). Baseia-se na aplicação do teorema de *Bayes*, assumindo a independência condicional entre cada par de características, dado o valor da variável de classe (Zhang, 2004). Além disso, este método exige uma quantidade mínima de dados de treino para a estimativa dos parâmetros necessários. Uma das principais vantagens do algoritmo de classificação *Naive Bayes* é sua simplicidade e rapidez na construção de modelos, o que o torna ideal para aplicações em tempo real. Adicionalmente, ele é bastante robusto a ruídos nos dados e pode lidar eficientemente com dados em falta. No entanto, a suposição de independência entre as características pode ser uma limitação em cenários onde as variáveis são fortemente correlacionadas, o que pode comprometer a precisão do modelo. Ainda assim, o *Naive Bayes* é amplamente utilizado em diversas áreas, como filtragem de *spam*, análise de sentimentos e sistemas de recomendação, devido à sua eficácia e facilidade de implementação.

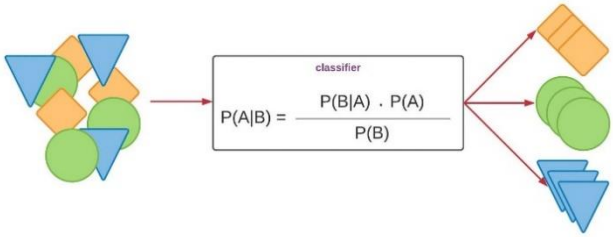

Figura 11. Ilustração alusiva ao modo de classificação com recurso ao algoritmo de *Naive Bayes*[1]

Árvores de Decisão

Árvores de Decisão são modelos de *Machine Learning* que realizam classificações com base em regras lógicas, representadas por uma estrutura em forma de árvore. Em uma árvore de decisão, cada nó interno representa uma condição de teste em um atributo, enquanto cada ramo corresponde a uma possível resposta para essa condição. Este modelo é especialmente útil para interpretar os resultados e identificar as características mais importantes dos dados. As árvores de decisão são algoritmos de classificação supervisionados, não-paramétricos e hierárquicos. Seu principal objetivo é criar um modelo capaz de prever o valor de uma variável alvo por meio da aprendizagem de regras de decisão simples, inferidas a partir das características dos dados (Charbuty & Abdulazeez, 2021). As árvores de decisão são conhecidas pela sua capacidade de lidar com dados categóricos e numéricos, além de serem intuitivas e fáceis de visualizar, o que facilita a sua interpretação. Uma vantagem significativa das árvores de decisão, é a sua habilidade de capturar interações não lineares entre as variáveis de entrada sem a necessidade de transformação dos dados. No entanto, elas podem-se tornar complexas e propensas a *overfitting*. Para mitigar este risco, técnicas como a poda da árvore e o uso de florestas aleatórias (*Random Forests*) são frequentemente empregues. Devido à sua flexibilidade e interpretabilidade, as árvores de decisão são amplamente utilizadas em diversas aplicações, como diagnóstico médico, análise de crédito e sistemas de recomendação.

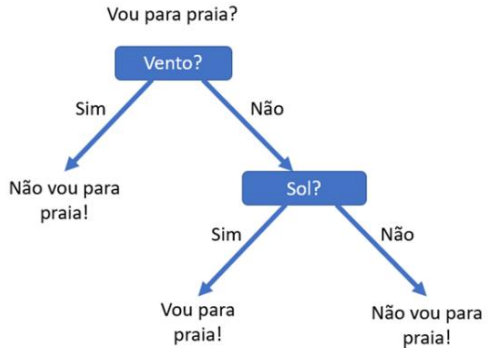

Figura 12. Ilustração alusiva ao modo de classificação com recurso às Árvores de Decisão[2]

[1] https://medium.com/@dancerworld60/demystifying-na%C3%AFve-bayes-simple-yet-powerful-for-text-classification-ad92b14a5c7

[2] https://didatica.tech/como-funciona-o-algoritmo-arvore-de-decisao/

***Ramdom* Forest**

Random Forest é uma técnica de *Machine Learning* que aprimora a precisão e a estabilidade das previsões combinando múltiplas árvores de decisão. Este método envolve a criação de um grande número de árvores de decisão independentes, cada uma treinada com uma amostra aleatória dos dados de treino e utilizando uma seleção aleatória de variáveis. As previsões individuais de todas as árvores são então combinadas para gerar uma previsão final mais robusta e precisa (Sarker, 2021).

Esta abordagem aproveita a diversidade das árvores de decisão para reduzir o risco de *overfitting*, que é comum em modelos baseados numa única árvore. O recurso a amostragens aleatórias dos dados permite que o algoritmo de classificação *Random Forest* capte uma ampla gama de padrões nos dados, tornando-o altamente eficaz em diferentes contextos de aplicação. Uma das grandes vantagens das *Random Forests* é a sua capacidade em lidar com grandes volumes de dados, mantendo um bom desempenho preditivo. Além disso, o modelo é relativamente fácil de usar e interpretar, ajudando a identificar quais características nos dados que se tornam mais relevantes para as previsões. Esta técnica de classificação dos dados pode, no entanto, aumentar o tempo de treino e a necessidade de processamento pelo que deverá ser usada com ponderação na classificação dos dados. As *Random Forests* são amplamente utilizadas em diversas áreas, como a bioinformática, finanças e análises de mercado.

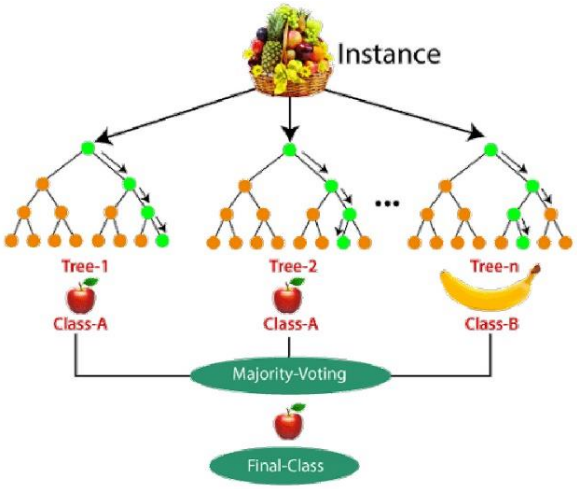

Figura 13. Ilustração alusiva ao modo de classificação com recurso às *Random Forests*[3]

[3] https://www.analyticsvidhya.com/blog/2021/06/understanding-random-forest/

Adaboost

O algoritmo *AdaBoost*, também conhecido como *Adaptive Boosting*, é uma técnica de *Machine Learning* supervisionada que aprimora a precisão da classificação através de múltiplas iterações. Este método funciona ajustando um classificador fraco a uma porção do conjunto de treino, dando pesos maiores aos exemplos que foram classificados incorretamente nas iterações anteriores. Este processo contínuo de reponderação dos dados permite que o *AdaBoost* concentre a sua aprendizagem nas amostras mais difíceis, melhorando gradualmente a precisão do modelo em cada iteração (Sarker, 2021). Além disso, o *AdaBoost* é menos suscetível ao *overfitting* quando comparado a muitos outros algoritmos de classificação, desde que não se exceda no número de iterações.

Este algoritmo é aplicável a uma grande variedade de problemas de classificação, sendo especialmente útil em contextos onde a precisão e a robustez são cruciais. No entanto, a sua eficácia pode ser comprometida por dados ruidosos e valores *outlier*, pois o aumento de peso em exemplos difíceis pode enfatizar esses pontos de dados indesejáveis. Apesar dessas limitações, o *AdaBoost* continua a ser uma escolha popular devido à sua simplicidade e eficácia, encontrando aplicação em áreas como reconhecimento de padrões, deteção de fraudes e bioinformática.

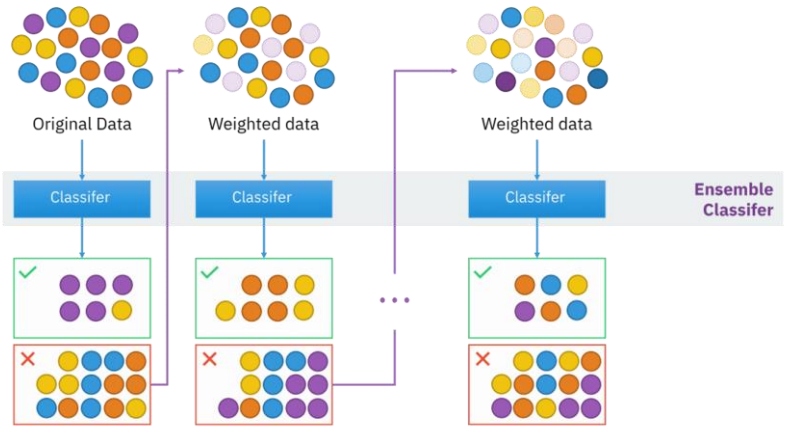

Figura 14. Ilustração alusiva ao modo de classificação do algoritmo *Adaboost*[4]

XGBoost

O algoritmo de classificação *XGBoost* (*Extreme Gradient Boosting*) representa uma das técnicas mais sofisticadas e eficazes em *Machine Learning* para tarefas de classificação. O *XGBoost* aprimora o conceito tradicional de *Gradient Boosting* ao incorporar várias otimizações cruciais, como a regularização para prevenir *overfitting*, paralelização para acelerar o processo

[4] https://www.almabetter.com/bytes/tutorials/data-science/adaboost-algorithm

de treino e métodos avançados de manipulação de dados ausentes. Estas melhorias permitem que o *XGBoost* desenvolva modelos de alta precisão e robustez, sendo particularmente eficiente em aplicações onde a performance é de extrema importância (Chen & Guestrin, 2016). Uma das principais vantagens do *XGBoost* é sua capacidade de lidar com grandes volumes de dados, mantendo um bom desempenho preditivo. O algoritmo é projetado para ser altamente escalável, suportando a execução em ambientes distribuídos, o que o torna ideal para grandes *datasets* de dados. Além disso, o *XGBoost* oferece flexibilidade na escolha de métricas de avaliação, permitindo que seja ajustado para uma ampla variedade de parâmetros.

O algoritmo de classificação *XGBoost* não fornece apenas previsões rápidas e precisas, mas também facilita a interpretação dos modelos de dados através das suas características. No entanto, a sua complexidade pode exigir uma compreensão mais aprofundada dos diferentes parâmetros e um ajuste cuidadoso para se obter o desempenho ideal. Os resultados consistentes e as capacidades avançadas inerentes a este algoritmo de classificação levam a que este seja amplamente utilizado em áreas/sectores tão diversos como as finanças, *marketing* ou a saúde.

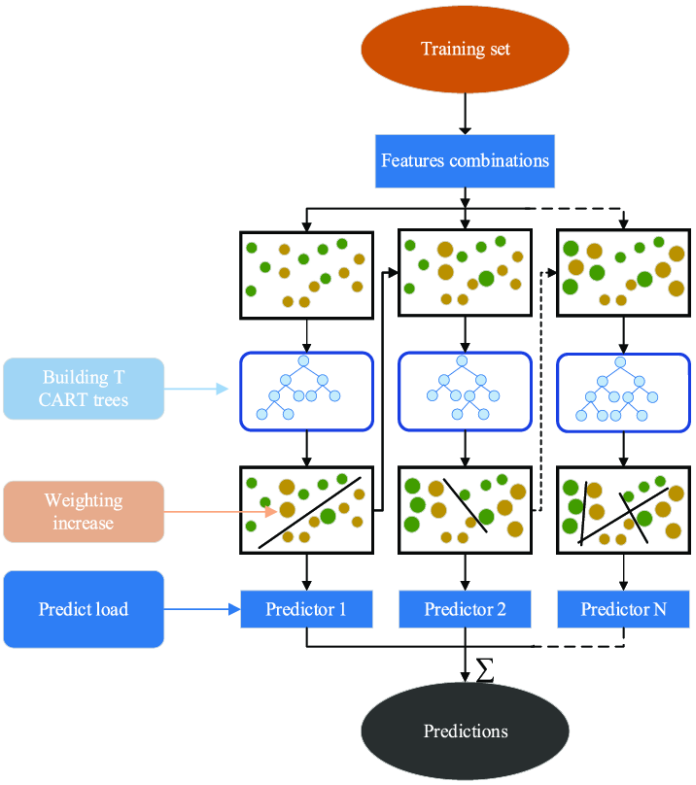

Figura 15. Ilustração alusiva ao modo de classificação do algoritmo *XGBoost*[5]

[5] https://www.almabetter.com/bytes/tutorials/data-science/adaboost-algorithm

CatBoost

O algoritmo de classificação *CatBoost* (*Categorical Boosting*) é projetado, especificamente, para tratar variáveis categóricas em tarefas de *Machine Learning*. Baseado no conceito de *Gradient Boosting*, *CatBoost* introduz melhorias significativas, como o tratamento automático de variáveis categóricas sem a necessidade de pré-processamento extensivo, além da utilização de técnicas avançadas de ordenação para evitar o *overfitting*. Uma característica notável do *CatBoost* é sua robustez contra o *overfitting*, mesmo quando aplicado a conjuntos de dados pequenos. Ele também suporta o processo de treino distribuído, o que o torna ideal para aplicações em larga escala (Dorogush et al., 2018). Além da sua eficácia na manipulação de variáveis categóricas, o algoritmo de classificação *CatBoost* é conhecido pela sua eficiência computacional. Ele consegue reduzir significativamente o tempo de treino relativamente a outros algoritmos, sem comprometer a precisão do modelo. Outra vantagem do algoritmo de classificação *CatBoost* é a sua capacidade de fornecer previsões precisas e robustas em diversos contextos, desde análise de grandes volumes de dados até aplicações em tempo real. A simplicidade no tratamento de dados categóricos e a redução do risco de *overfitting* tornam o *CatBoost* uma excelente escolha para uma ampla variedade de problemas de classificação.

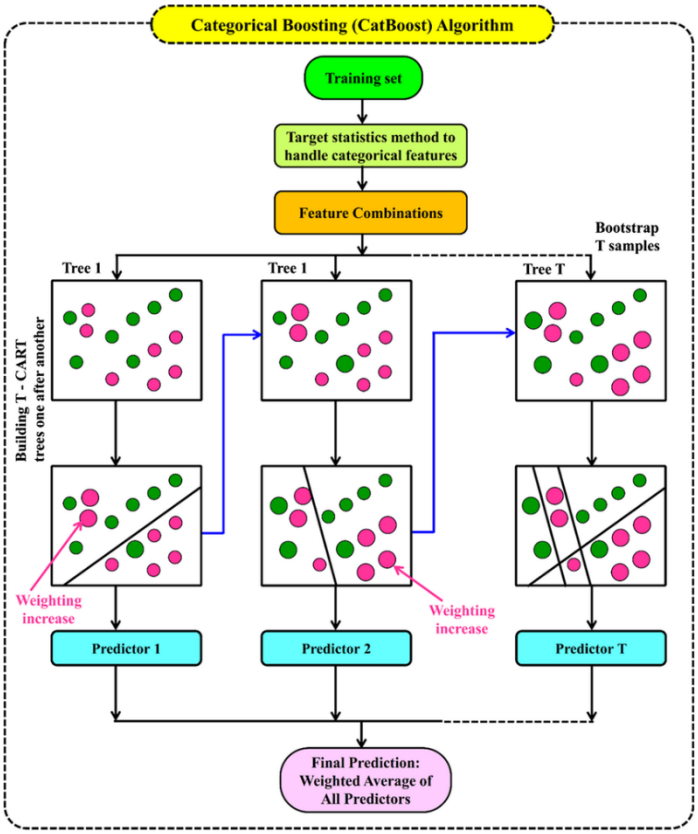

Figura 16. Ilustração alusiva ao modo de classificação do algoritmo *CatBoost*[6]

[6] https://www.researchgate.net/figure/The-flow-diagram-of-the-CatBoost-model_fig3_370695897

Algoritmos de Regressão

A regressão é uma técnica estatística utilizada para modelar a relação entre uma variável dependente e uma ou mais variáveis independentes. O objetivo da regressão é encontrar uma função matemática que descreva a relação entre as variáveis, permitindo fazer previsões ou inferências.

Existem diferentes tipos de regressão, incluindo a regressão linear, *LASSO*, *Ridge* e *ElasticNet*.

Regressão Linear

A regressão linear, é uma técnica utilizada para modelar a relação entre uma variável dependente e uma ou mais variáveis independentes. O objetivo do modelo de regressão linear simples é encontrar a melhor reta que minimiza a diferença entre os valores observados e os valores previstos, utilizando a soma dos quadrados como critério de otimização.

A regressão linear múltipla expande este conceito para incluir duas ou mais variáveis independentes, permitindo modelar a relação entre uma variável dependente quantitativa e múltiplas variáveis preditoras. Este modelo procura ajustar um hiperplano no espaço multidimensional das variáveis independentes, que melhor explica a variação na variável dependente. A precisão e a interpretação dos coeficientes das variáveis independentes fornecem *insights* valiosos sobre a influência de cada variável preditora na variável alvo.

Uma das principais vantagens da regressão linear é a sua simplicidade e facilidade de interpretação, o que a torna amplamente aplicável em diversas áreas, como economia, ciências sociais e engenharia. No entanto, é fundamental garantir que os pressupostos do modelo estejam acautelados de forma a evitar a multicolinearidade e *overfitting* que poderão comprometer a robustez do modelo.

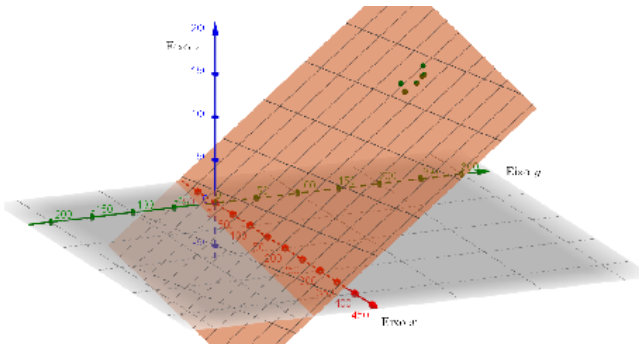

Figura 17. Exemplo demonstrativo de um hiperplano associado a uma Regressão Linear Múltipla[7]

7 https://www.ufrgs.br/reamat/AlgebraLinear/livro/s14regressx00e3o_linear_mx00faltipla.html

Regressões LASSO e *Ridge*

As regressões *LASSO* (*Least Absolute Shrinkage and Selection Operator*) e *Ridge* são técnicas de regularização amplamente empregues na construção de modelos preditivos, especialmente em cenários de alta dimensionalidade, onde o número de variáveis preditoras é elevado. Ambas as técnicas visam mitigar o problema de sobre ajuste (*overfitting*), e aprimorar a capacidade de generalização, ou seja, o desempenho do modelo em dados não treinados (James et al., 2013). A regressão *Ridge* opera adicionando um termo de penalização à função de custo tradicional da regressão linear. Essa penalização, proporcional à soma dos quadrados dos coeficientes do modelo, induz a redução dos valores absolutos dos coeficientes, mas sem anulá-los completamente. Tal efeito contribui para a diminuição da variância do modelo, tornando-o menos sensível a flutuações nos dados de treino e, consequentemente, mais robusto a novas observações (Hastie et al., 2009).

Por sua vez, a regressão *LASSO* também incorpora um termo de penalização à função de custo, porém baseado na soma dos valores absolutos dos coeficientes. Essa forma de penalização não apenas reduz a magnitude dos coeficientes, mas também pode reduzir alguns deles a zero. A regressão *LASSO* permite, desta forma, a construção de modelos com menor número de variáveis preditoras, o que facilita a interpretação e pode melhorar o desempenho em situações onde muitas características são irrelevantes ou redundantes (Tibshirani, 1996). A escolha entre *LASSO* e *Ridge* depende das características dos dados e dos objetivos da análise. A regressão *Ridge* é preferível quando se deseja manter todas as características no modelo, mesmo que com coeficientes reduzidos, enquanto a *LASSO* é indicada para a identificação e seleção das características mais relevantes para a predição (Zou & Hastie, 2005). Ambas as técnicas encontram ampla utilização em diversas áreas do conhecimento como sejam a biologia, medicina, economia e ciências sociais.

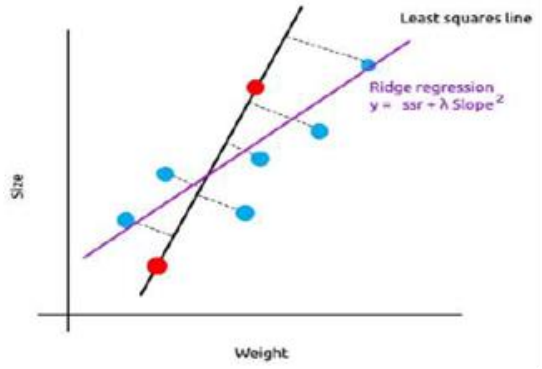

Figura 18. Gráfico demonstrativo da técnica estatística empregue ao nível da Regressão *Ridge*[8]

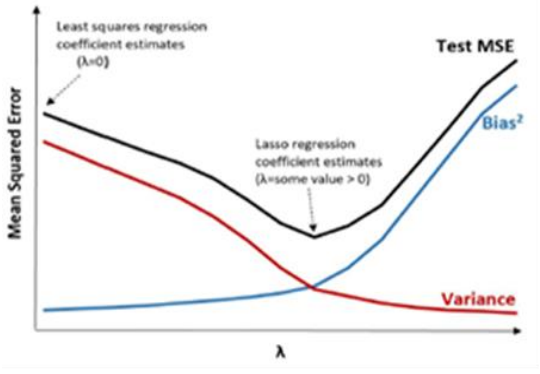

Figura 19. Gráfico demonstrativo da técnica estatística empregue ao nível da Regressão *Lasso*[9]

[8] Vardasca R. (2024) Apontamentos da unidade curricular de Aprendizagem Automática.
[9] Vardasca R. (2024) Apontamentos da unidade curricular de Aprendizagem Automática.

Regressão Elastic Net

A regressão *Elastic Net*, introduzida por Zou e Hastie (2005), é uma técnica de regularização que combina as penalidades L1 (utilizadas na regressão *LASSO*) e L2 (utilizadas na regressão *Ridge*) para estimar os coeficientes de um modelo de regressão linear. Esta abordagem híbrida busca superar as limitações de cada modelo individual, oferecendo um equilíbrio entre a seleção de variáveis e a estabilização dos coeficientes em cenários de alta dimensionalidade e multicolinearidade. Ao combinar as propriedades da *LASSO* e da *Ridge,* a regressão *Elastic Net* possui a capacidade de lidar com variáveis altamente correlacionadas. Isso é particularmente útil em situações onde existam mais características do que amostras, ou quando as características são fortemente colineares. A *Elastic Net* atribui um peso a cada uma das penalidades (L1 e L2), permitindo ajustar o modelo de forma flexível e encontrar um compromisso ótimo entre complexidade e precisão. A metodologia da *Elastic Net* envolve a introdução de dois parâmetros de regularização: um que controla a mistura das penalidades L1 e L2, e outro que controla a força geral da penalização. Esta flexibilidade adicional permite que a *Elastic Net* mantenha a eficiência computacional e a interpretabilidade dos dados, melhorando a precisão preditiva. Por fim, a regressão *Elastic Net* tem sido aplicada com sucesso em diversas áreas, como bioinformática (Zou & Hastie, 2005), análise de imagens (Zhou & Zhu, 2010), e previsão de séries temporais (Bai & Ng, 2008), demonstrando sua versatilidade e eficácia em lidar com problemas complexos de regressão.

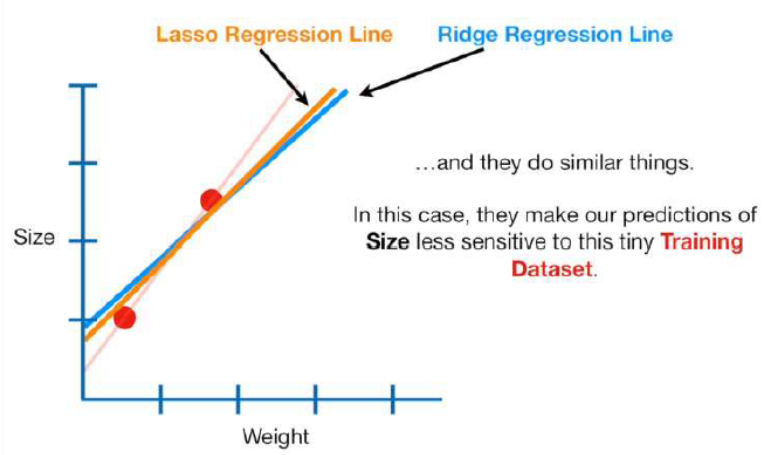

Figura 20. Gráfico demonstrativo da técnica estatística empregue ao nível da Regressão *Elastic-Net*[10]

[10] Vardasca R. (2024) Apontamentos da unidade curricular de Aprendizagem Automática.

6. Aplicação de Modelos

Algoritmos de Classificação

Naive Bayes

- **Test Accuracy (Exatidão):** 0.5922

❖ **Matriz de Confusão**

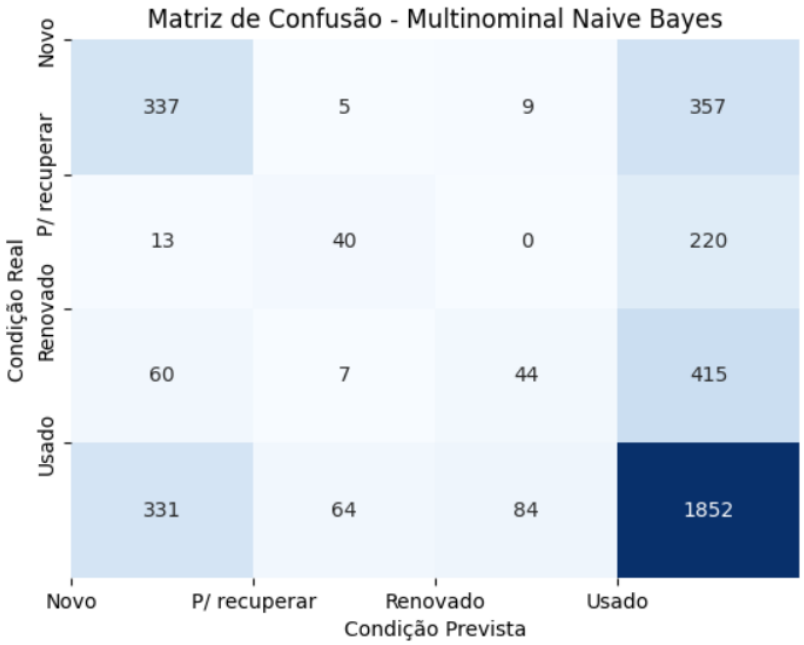

Figura 21. Matriz de confusão de resultados obtidos, tomando como referência o modelo *Naive Bayes*.

❖ **Considerações**

A análise da matriz de confusão do algoritmo de classificação *Naive Bayes* revela várias conclusões importantes sobre o desempenho do modelo. Em termos de exatidão, o modelo apresenta uma precisão aproximada de 59,2%, indicando que cerca de 59,2% das previsões estão corretas. Este valor é moderado e sugere que há espaço significativo para melhorias. Focando no desempenho por classe, a classe "*Novo*" tem uma precisão de aproximadamente 45,5%, significando que das previsões feitas como "*Novo*", cerca de 45,5% estavam corretas. O *Recall* é de aproximadamente 47,6%, indicando que o modelo conseguiu identificar corretamente cerca de 47,6% de todos os itens que eram realmente "*Novo*". Isto revela um desempenho moderado, mas com uma margem de erro considerável, dado que a classe "*Novo*" é frequentemente confundida como "*Usado*". Para a classe "*P/recuperar*", a precisão é de aproximadamente 34,5%, com apenas 34,5% das previsões feitas como "*P/recuperar*" estando corretas. O *Recall* é de apenas 14,6%, o que é preocupante, pois significa que o modelo não está a identificar corretamente a maioria dos itens "*P/recuperar*". Este desempenho fraco

indica uma necessidade urgente de melhoria, já que esta classe é frequentemente confundida como "*Usado*". A classe "*Renovado*" apresenta a pior performance, com uma precisão de apenas 32,1%, significando que cerca de 32,1% das previsões como "*Renovado*" estavam corretas. O *Recall* é de apenas 8,4%, sugerindo que o modelo falha em capturar a maioria dos itens que deveriam ser classificados como "*Renovado*". Este desempenho indica uma necessidade crítica de revisão e ajuste, pois muitos itens "*Renovado*" são classificados como "*Usado*". Por outro lado, a classe "*Usado*" tem o melhor desempenho, com uma precisão de cerca de 65,1% e um *Recall* de aproximadamente 79,5%. Isto indica que o modelo é bastante eficaz em identificar itens que são realmente "*Usados*", embora a precisão possa ser melhorada para reduzir os falsos positivos de outras classes. Em resumo, o modelo de dados tem dificuldades significativas em diferenciar entre a condição "*Novo*", "*P/ recuperar*" e "*Renovado*", com estas classes sendo frequentemente classificadas como "*Usado*". As classes "*P/ recuperar*" e "*Renovado*" têm uma precisão e *Recall* extremamente baixos, sugerindo que as características utilizadas pelo modelo podem não ser adequadas ou suficientes para distinguir corretamente entre classes.

Árvores de Decisão

- **Best Hyperparameters:** {'criterion': 'gini', 'max_depth': 12, 'min_samples_leaf': 1, 'min_samples_split': 2}
- **Best Cross-Validation Score**: 0.6413
- **Test Accuracy (Exatidão)**: 0.6498

❖ **Matriz de Confusão**

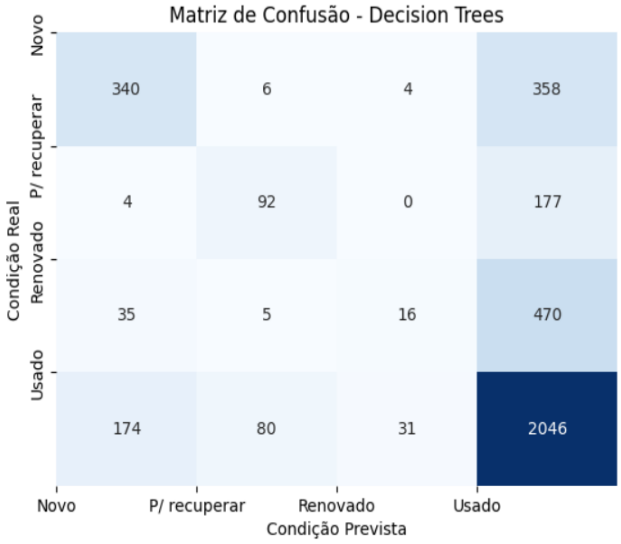

Figura 22. Matriz de confusão de resultados obtidos, tomando como referência as árvores de decisão.

❖ **Considerações**

A análise da matriz de confusão do algoritmo de *Árvores de Decisão* revela várias conclusões importantes sobre o desempenho do modelo. Em termos de exatidão, o modelo apresenta uma precisão de aproximadamente 65,1%, indiciando um desempenho robusto em várias classes. O desempenho por classe tem uma precisão de 65,1% para a classe "*Novo*". O *Recall* é de aproximadamente 48%, indicando que o modelo conseguiu identificar corretamente cerca de 48% de todos os itens que eram realmente "*Novo*". Isto revela um desempenho moderado, mas com uma margem de erro considerável, dado que a classe "*Novo*" é frequentemente confundida com "*Usado*". Para a classe "*P/ recuperar*", a precisão é de aproximadamente 50,3%, com 50,3% das previsões feitas como "*P/ recuperar*" estando corretas. O *Recall* é de 33,7%, o que é um valor relativamente baixo, sugerindo que o modelo não está a identificar corretamente a maioria dos itens "*P/ recuperar*". Este desempenho moderado indica a necessidade de melhorias, já que esta classe é frequentemente confundida com "*Usado*".

A classe "*Renovado*" apresenta uma precisão de cerca de 31,4%, significando que cerca de 31,4% das previsões como "*Renovado*" estavam corretas. O *Recall* é de apenas 3%, sugerindo que o modelo falha em capturar a maioria dos itens que deveriam ser classificados como "*Renovado*". Este desempenho indica uma necessidade crítica de revisão e ajuste, pois muitos itens "*Renovado*" são classificados como "*Usado*". Por outro lado, a classe "*Usado*" tem o melhor desempenho, com uma precisão de cerca de 67,1% e um *Recall* de aproximadamente 87,8%. Isto indica que o modelo é bastante eficaz em identificar itens que são realmente "*Usado*", embora a precisão possa ser melhorada para reduzir os falsos positivos de outras classes.

Em resumo, o modelo de dados associado às *Árvores de Decisão* tem dificuldades significativas em diferenciar entre "*Novo*", "*P/ recuperar*" e "*Renovado*", com estas classes sendo frequentemente classificadas como "*Usado*". As classes "*P/ recuperar*" e "*Renovado*" têm uma precisão e *Recall* baixos, sugerindo que as características utilizadas pelo modelo podem não ser adequadas ou suficientes para distinguir corretamente estas classes. Para melhorar o desempenho, recomenda-se ajustar os hiperparâmetros do modelo e considerar a engenharia de novas características que ajudem a distinguir melhor entre classes.

Random Forest

- **Best Hyperparameters:** {'max_depth': 32, 'max_features': 'sqrt', 'min_samples_leaf': 1, 'min_samples_split': 10, 'n_estimators': 200}
- **Best Cross-Validation Score:** 0.6713
- **Test Accuracy (Exatidão):** 0.6759

❖ **Matriz de Confusão**

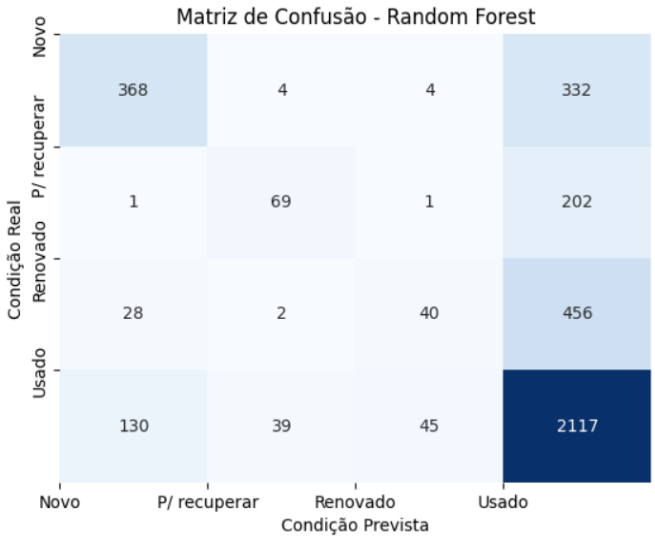

Figura 23. Matriz de confusão de resultados obtidos, tomando como referência as *Random Forests*

❖ **Considerações**

A análise da matriz de confusão do algoritmo *Random Forest* revela várias conclusões importantes sobre o desempenho do modelo. Em termos de exatidão, o modelo apresenta uma precisão de aproximadamente 67,6%, indicando um desempenho robusto em várias classes. O desempenho por classe tem uma precisão de 69,8% para a classe "*Novo*". O *Recall* é de aproximadamente 52%, indicando que o modelo conseguiu identificar corretamente cerca de 52% de todos os itens que eram realmente associados à condição de "*Novo*". Isto revela um desempenho moderado, mas com uma margem de erro considerável, dado que a classe "*Novo*" é frequentemente confundida com "*Usado*".

Para a classe "*P/ recuperar*", a precisão é de aproximadamente 60,5%, com 60,5% das previsões feitas como "*P/ recuperar*" estando corretas. O *Recall* é de 25,3%, o que é um valor relativamente baixo, sugerindo que o modelo não está a identificar corretamente a maioria dos itens "*P/ recuperar*". Este desempenho moderado indica a necessidade de melhorias, já que esta classe é frequentemente confundida com "*Usado*".

A classe "*Renovado*" apresenta uma precisão de cerca de 44,4%, significando que cerca de 44,4% das previsões como "*Renovado*" estavam corretas. O *Recall* é de apenas 7,6%, sugerindo que o modelo falha em capturar a maioria dos itens que deveriam ser classificados como "*Renovado*". Este desempenho indica uma necessidade crítica de revisão e ajuste, pois muitos itens "*Renovado*" são classificados como "*Usado*". Por outro lado, a classe "*Usado*" tem o melhor desempenho, com uma precisão de cerca de 68,1% e um *Recall* de aproximadamente 90,8%. Isto indica que o modelo é bastante eficaz em identificar itens que se encontram associados à condição de "*Usado*", embora a precisão possa ser melhorada para reduzir os falsos positivos de outras classes.

Em resumo, o modelo *Random Forest* tem dificuldades significativas em diferenciar entre "*Novo*", "*P/ recuperar*" e "*Renovado*", com estas classes sendo frequentemente classificadas como "*Usado*". As classes "*P/ recuperar*" e "*Renovado*" têm uma precisão e *Recall* baixos, sugerindo que as características utilizadas pelo modelo de dados podem não ser adequadas ou suficientes para distinguir corretamente entre classes. Apesar de tudo isso, foi este o algoritmo de classificação escolhido para classificar os registos de imóveis cuja condição se encontrava omissa. Este modelo foi, desta forma, aquele que apresentou um desempenho superior face aos restantes, espelhado nos valores obtidos ao nível das suas métricas e exatidão.

Adaboost

- **Best Hyperparameters:** {"learning_rate': 1.0, 'n_estimators': 200}
- **Best Cross-Validation Score**: 0.6214
- **Test Accuracy (Exatidão):** 0.6258

❖ **Matriz de Confusão**

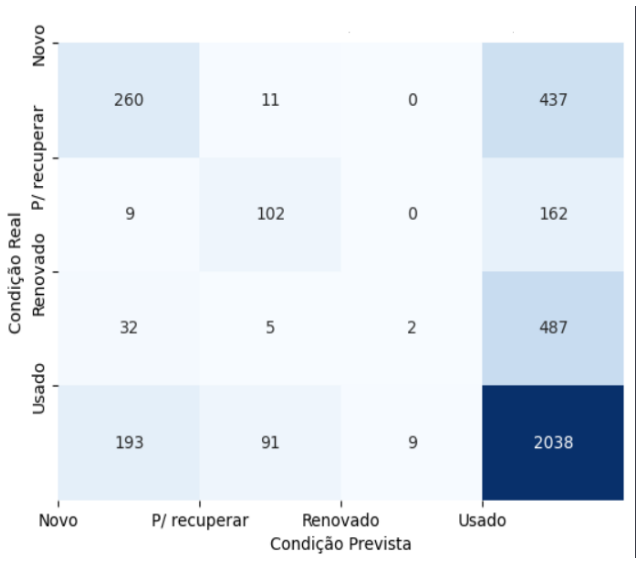

Figura 24. Matriz de confusão de resultados obtidos, tomando como referência o modelo *Adaboost*

❖ **Considerações**

A análise da matriz de confusão do algoritmo *AdaBoost* revela várias conclusões importantes sobre o desempenho do modelo. Em termos de exatidão, o modelo apresenta uma precisão de aproximadamente 62,6%, indicando um desempenho moderado em várias classes. Focado no desempenho por classe, a classe "*Novo*" tem uma precisão de cerca de 52,6%, significando que das previsões feitas como "*Novo*", cerca de 52,6% estavam corretas. O *Recall* é de aproximadamente 36,7%, indicando que o modelo conseguiu identificar corretamente cerca de 36,7% de todos os imóveis que eram realmente "Novo"(s). Isto revela um desempenho fraco, com uma margem de erro considerável, dado que a classe "*Novo*" é frequentemente confundida com "*Usado*". Ao nível da classe "*P/ recuperar*", a precisão é de aproximadamente 48,8%, com 48,8% das previsões feitas como "*P/ recuperar*" estando corretas. O *Recall* é de 37,4%, o que é um valor relativamente baixo, sugerindo que o modelo não está a identificar corretamente a maioria dos itens "*P/ recuperar*". Este desempenho indica a necessidade de melhorias, já que esta classe é frequentemente confundida com "*Usado*".

A classe "*Renovado*" apresenta uma precisão de cerca de 15,4%, significando que apenas 15,4% das previsões como "*Renovado*" estavam corretas. O *Recall* é de apenas 0,4%, sugerindo que o modelo falha em capturar quase todos os itens que deveriam ser classificados como "*Renovado*". Este desempenho indica uma necessidade crítica de revisão e ajuste, pois muitos imóveis "*Renovado*"(s) são classificados como "*Usado*"(s).

Por outro lado, a classe "*Usado*" tem melhor desempenho, com uma precisão de cerca de 65,2% e um *Recall* de aproximadamente 87,4%. Isto indica que o modelo é bastante eficaz em identificar os imóveis que são realmente "*Usado*"(s), embora a precisão possa ser melhorada para reduzir os falsos positivos de outras classes.

Resumindo, o modelo *AdaBoost* tem dificuldades significativas em diferenciar entre "*Novo*", "*P/ recuperar*" e "*Renovado*", com estes imóveis sendo frequentemente classificados como "*Usado*"(s). As classes "*P/ recuperar*" e "*Renovado*" têm uma precisão e *Recall* muito baixos, sugerindo que as características utilizadas pelo modelo podem não ser adequadas ou suficientes para distinguir corretamente entre estas classes.

XGBoost

- **Best Hyperparameters:** {"learning_rate": 0.1, 'max_depth': 8, 'n_estimators': 300, 'subsample': 0.7}
- **Best Cross-Validation Score**: 0.6746
- **Test Accuracy (Exatidão):** 0.6748

❖ **Matriz de Confusão**

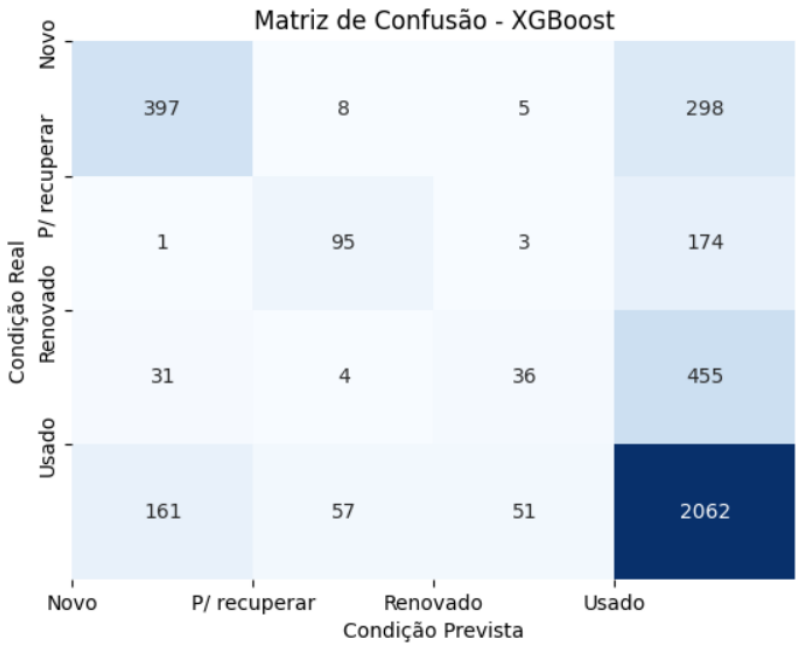

Figura 25. Matriz de confusão de resultados obtidos, tomando como referência o modelo *XGBoost*

❖ **Considerações**

A análise da matriz de confusão do algoritmo *XGBoost* revela várias conclusões importantes sobre o desempenho do modelo. Em termos de exatidão, o modelo apresenta uma precisão de aproximadamente 67,5%, indicando um desempenho robusto em várias classes. O desempenho por classe tem uma precisão de 67,3% para a classe "*Novo*". O *Recall* é de aproximadamente 56,1%, indicando que o modelo conseguiu identificar corretamente cerca de 56,1% de todos os itens que eram realmente associados à condição de "*Novo*". Isto revela um desempenho moderado, mas com uma margem de erro considerável, dado que a classe "*Novo*" é frequentemente confundida com "*Usado*". Para a classe "*P/ recuperar*", a precisão é de aproximadamente 57,9%, com 57,9% das previsões feitas como "*P/ recuperar*" estando corretas. O *Recall* é de 34,8%, o que é um valor relativamente baixo, sugerindo que o modelo não está a identificar corretamente a maioria dos itens "*P/ recuperar*". Este desempenho moderado indica a necessidade de melhorias, já que esta classe é frequentemente confundida com a condição "*Usado*".

A classe "*Renovado*" apresenta uma precisão de cerca de 37,9%, significando que cerca de 37,9% das previsões como "*Renovado*" estavam corretas. O *Recall* é de apenas 6,8%, sugerindo que o modelo falha em capturar a maioria dos itens que deveriam ser classificados como "*Renovado*". Este desempenho indica uma necessidade crítica de revisão e ajuste, pois muitos itens "*Renovado*" são classificados como "*Usado*". Por outro lado, a classe "*Usado*" tem o melhor desempenho, com uma precisão de cerca de 69% e um *Recall* de aproximadamente 88,5%. Isto indica que o modelo é bastante eficaz em identificar itens associados à condição de "*Usado*", embora a precisão possa ser melhorada para reduzir os falsos positivos de outras classes.

Em resumo, o modelo *XGBoost* tem dificuldades significativas em diferenciar entre "*Novo*", "*P/ recuperar*" e "*Renovado*", com estas classes sendo frequentemente classificadas com a condição de "*Usado*". As classes "*P/ Recuperar*" e "*Renovado*" têm uma precisão e *Recall* baixos, sugerindo que as características utilizadas pelo modelo podem não ser adequadas ou suficientes para distinguir corretamente entre estas classes.

CatBoost

- **Best Hyperparameters:** {' 'depth': 8, 'iterations': 300, 'l2_leaf_reg': 0.0001, 'learning_rate': 0.1}
- **Best Cross-Validation Score**: 0.6656
- **Test Accuracy (Exatidão)**: 0.6655

❖ **Matriz de Confusão**

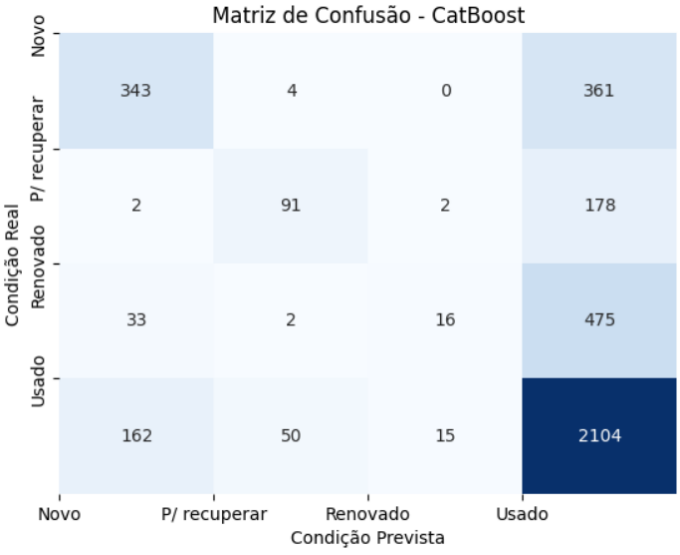

Figura 26. Matriz de confusão de resultados obtidos, tomando como referência o modelo *CatBoost*

❖ **Considerações**

A análise da matriz de confusão do algoritmo *CatBoost* revela várias conclusões importantes sobre o desempenho do modelo. Em termos de exatidão, o modelo apresenta uma precisão de aproximadamente 66,6%, indicando um desempenho robusto em várias classes. O desempenho por classe tem uma precisão de 63,5% para a classe "*Novo*". O *Recall* é de aproximadamente 48,4%, indicando que o modelo conseguiu identificar corretamente cerca de 48,4% de todos os itens que eram realmente associados à condição de "*Novo*". Isto revela um desempenho moderado, mas com uma margem de erro considerável, dado que a classe "*Novo*" é frequentemente confundida com "*Usado*". Para a classe "*P/ recuperar*", a precisão é de aproximadamente 61,9%, com 61,9% das previsões feitas como "*P/ recuperar*" estando corretas. O *Recall* é de 33,3%, o que é um valor relativamente baixo, sugerindo que o modelo não está a identificar corretamente a maioria dos itens "*P/ recuperar*". Este desempenho moderado indica a necessidade de melhorias, já que esta classe é frequentemente confundida com "*Usado*".

A classe "*Renovado*" apresenta uma precisão de cerca de 48,5%, significando que cerca de 48,5% das previsões como "*Renovado*" estavam corretas. O *Recall* é de apenas 3%, sugerindo que o modelo falha em capturar a maioria dos itens que deveriam ser classificados como "*Renovado*". Este desempenho indica uma necessidade crítica de revisão e ajuste, pois muitos itens "*Renovado*" são classificados como "*Usado*". Por outro lado, a classe "*Usado*" tem o melhor desempenho, com uma precisão de cerca de 67,5% e um *Recall* de aproximadamente 90,2%. Isto indica que o modelo é bastante eficaz em identificar itens que são realmente "*Usado*"(s), embora a precisão possa ser melhorada para reduzir os falsos positivos de outras classes.

Em resumo, o modelo *CatBoos*t tem dificuldades significativas em diferenciar entre "*Novo*", "*P/ recuperar*" e "*Renovado*", com estas classes sendo frequentemente classificadas como "*Usado*". As classes "*P/ recuperar*" e "*Renovado*" têm uma precisão e *Recall* baixos, sugerindo que as características utilizadas pelo modelo podem não ser adequadas ou suficientes para distinguir corretamente as diferentes classes.

Algoritmos de Regressão

Regressão Linear Multivariável

- **Best Mean Absolute Error (MAE)**: 102.956,6
- **Best Mean Squared Error (MSE)**: 21.321.821.443,9
- **Best Root Mean Squared Error (RMSE)**: 146.019,9
- **R2 Score:** 0.604
- **Nota adicional:** Modelo treinado e testado 10 vezes, cada vez utilizando uma parte diferente como conjunto de teste e outras 9 partes como conjunto de treino.

❖ **Considerações:**

A análise aos indicadores obtidos permite afirmar que, o modelo de regressão linear tem um desempenho razoavelmente bom, com erros médios relativamente baixos (MAE e RMSE) e um R2 Score que sugere um ajuste moderado. No entanto, o desempenho moderado do R2 Score e os valores dos erros indicam que o modelo pode ser aprimorado. Para melhorar a performance do modelo, várias abordagens podem ser consideradas, incluindo a incorporação de variáveis adicionais, a exploração de técnicas de regularização como *Ridge* e *LASSO*, a utilização de modelos mais complexos e a análise de *outliers*. A exploração de técnicas de regularização como a *Ridge* e *LASSO* pode melhorar o ajuste do modelo, prevenindo o sobre ajuste e reduzindo a variabilidade dos coeficientes. Além disso, utilizar modelos mais complexos, como árvores de decisão, *Random Forests* ou técnicas de *boosting*, pode evidenciar relações não lineares e interações complexas entre as variáveis. A análise de *outliers* e dados influentes é crucial para identificar e tratar pontos de dados que possam estar distorcendo o modelo. Em conclusão, embora o modelo de regressão linear atual ofereça uma boa base, há várias estratégias que podem ser adotadas para refinar e melhorar suas capacidades preditivas, e essas melhorias podem levar a um modelo mais robusto e preciso, apto a lidar melhor com a complexidade dos dados.

Regressão Ridge

- **Best Hyperparameters**: {alpha: 10.0}
- **Best Mean Absolute Error (MAE)**: 102.950,1
- **Best Mean Squared Error (MSE)**: 21.314.721.554,8
- **Best Root Mean Squared Error (RMSE)**: 145.995,6
- **R2 Score:** 0.604
- **Nota adicional:** Modelo treinado e testado 10 vezes com 3 candidatos para o parâmetro de regularização *a* (*alfa*), totalizando 30 ajustes. Os melhores hiper parâmetros encontrados indicam que *a* (*alfa*) foi ajustado para 10.0.

❖ **Considerações:**

A análise aos indicadores obtidos permite afirmar que, a regressão *Ridge* apresentou um desempenho moderado, com erros médios relativamente baixos (*MAE* e *RMSE*) e um R2 Score de 0.604. O valor de *R2 Score* sugere que o modelo capturou uma proporção significativa da variância nos dados, embora haja margem para melhorias. Para melhorar a performance do modelo de dados, várias abordagens podem ser consideradas. A incorporação de variáveis adicionais pode aumentar a capacidade preditiva, capturando informações que não estavam sendo consideradas anteriormente. A exploração de outras técnicas de regularização, como a *LASSO* (*Least Absolute Shrinkage and Selection Operator*) e *Elastic Net*, pode ser benéfica. A *LASSO* induz esparsidade no modelo, selecionando automaticamente as variáveis mais relevantes. A *Elastic Net* combina as características da *Ridge* e da *LASSO*, permitindo um equilíbrio entre a penalização dos coeficientes e a seleção de variáveis.

A utilização de modelos mais complexos, como redes neurais artificiais ou modelos baseados em árvores (como *Random Forests* ou *Gradient Boosting*), pode capturar relações não lineares nos dados e melhorar a precisão das previsões. No entanto, é importante ter em mente que modelos mais complexos podem ser mais propensos ao *overfitting* e podem exigir um maior volume de dados para treino. Esta regressão apresenta, assim, um desempenho moderado face ao *dataset* de dados, pelo que deverão ser consideradas outros tipos de regressões.

Regressão Lasso

- **Best Hyperparameters:** {alpha: 10,0}
- **Best Mean Absolute Error (MAE)**: 102.964,4
- **Best Mean Squared Error (MSE)**: 21.318.988.057,9
- **Best Root Mean Squared Error (RMSE)**: 146.010,2
- **R2 Score:** 0.604
- **Nota adicional:** Modelo treinado e testado 10 vezes com 3 candidatos para o parâmetro de regularização *a* (*alfa*), totalizando 30 ajustes. Os melhores hiper parâmetros encontrados indicam que *a* (*alfa*) foi ajustado para 10,0.

❖ **Considerações:**

A análise aos indicadores obtidos permite afirmar que, este modelo de regressão obteve um desempenho idêntico às anteriores, como evidenciado pelas métricas de desempenho MAE (*Mean Absolute Error*), RMSE (*Root Mean Squared Error*) e um R2 Score.

A utilização de modelos mais complexos, como redes neurais ou modelos baseados em árvores (e.g., *Random Forests* ou *XGBoost*), pode capturar relações não lineares nos dados e melhorar a precisão das previsões. Redes neurais, com sua capacidade de aprender representações complexas dos dados, podem ser particularmente úteis quando as relações entre as variáveis são altamente não lineares. Modelos baseados em árvores, como *Random Forests* e *XGBoost*, são conhecidos por sua robustez a *outliers* e sua capacidade de lidar com diferentes tipos de dados. É, por fim, importante ressalvar que a escolha do modelo de dados mais adequado depende das características específicas dos dados e do problema em questão.

Regressão Elastic Net

- Best Hyperparameters: { l1_ratio: 0.9, alpha: 0.1}
- **Best Mean Absolute Error (MAE)**: 133.875,2
- **Best Mean Squared Error (MSE)**: 32.911.788.035,1
- **Best Root Mean Squared Error (RMSE)**: 181.416,1
- **R2 Score:** 0.389
- **Nota adicional:** Modelo treinado e testado 10 vezes com 9 candidatos para o parâmetro de regularização *a* (*alfa*), totalizando 90 ajustes. Os melhores hiper parâmetros encontrados indicam que *a* (*alfa*) foi ajustado para 0.1.

❖ **Considerações:**

A análise aos indicadores obtidos permite afirmar que, este modelo de regressão obteve um pior desempenho comparativamente com os restantes modelos, como evidenciado pelos valores de *MAE* (*Mean Absolute Error*), RMSE (*Root Mean Squared Error*) superiores, e um R2 Score inferior. Os valores obtidos ao nível das diferentes métricas sugerem que o modelo não conseguiu capturar adequadamente as relações entre as variáveis preditoras e a variável alvo.

Os melhores hiperparâmetros encontrados, l1_ratio = 0.9 e alpha = 0.1, indicam que a penalização aplicada aos coeficientes do modelo foi alta. Um l1_ratio de 0.9 significa que a penalização L1 (*Lasso*) teve um peso de 90% na regularização, enquanto a penalização L2 (*Ridge*) teve um peso de 10%. O valor baixo de *a* (*alfa*) indica que a intensidade da regularização foi baixa, o que pode ter contribuído para o baixo desempenho do modelo, especialmente se houver multicolinearidade entre as variáveis preditoras. Este modelo de regressão não é assim recomendado para esta distribuição de dados, devendo ser consideradas outras regressões.

Regressão XGBRegressor

- **Best Hyperparameters:** {*'subsample': 0,7, 'scale_pos_weight': 10, 'rel_alpha': 1, 'reg_lambda': 10, 'reg_alpha': 1, 'n_estimators': 500, 'min_child_weight': 3, 'max_depth': 10, 'learning_rate': 0.1, 'gamma': 0.5, 'colsample_bytree': 0.7, 'base_score': 0,75*}
- **Best Mean Absolute Error (MAE)**: 90.988,6
- **Best Mean Squared Error (MSE)**: 18.183.418.929.9
- **Best Root Mean Squared Error (RMSE)**: 134.845,9
- **R2 Score:** 0,662
- **Nota adicional:** Modelo treinado e testado 10 vezes com 100 candidatos para o parâmetro de regularização *a* (*alfa*), totalizando 1000 ajustes.

❖ **Considerações:**

A análise aos indicadores obtidos permite afirmar que, o modelo de regressão *XGBRegressor* teve um bom desempenho, como evidenciado pelos baixos valores de *MAE* (*Mean Absolute Error*) de 90.988,6, *MSE* (*Root Mean Squared Error*) de 18.183.418.929.9 e pelo *R2 Score* relativamente alto de 0,662. Estes valores sugerem que o modelo conseguiu capturar adequadamente as relações entre as variáveis preditoras e a variável alvo (preço das casas), explicando cerca de 66,2% da variabilidade nos dados. Adicionalmente os 1000 ajustes realizados, demonstram rigor na busca pelos melhores hiperparâmetros, o que aumenta a confiança na qualidade do modelo final. No entanto, é importante ressalvar que, apesar do bom desempenho, ainda há espaço para melhorias. O *R2 Score*, embora relativamente alto, indica que ainda existe uma parcela da variabilidade nos dados que não é explicada pelo modelo. A exploração de novas variáveis preditoras, a transformação das variáveis existentes ou a utilização de técnicas de *ensemble* podem levar a um modelo ainda mais preciso. Perante tudo isto e face ao bom desempenho obtido, será este modelo de regressão que será utilizado na estimação/predição do preço de venda dos imóveis presentes nos *datasets* de dados.

Resultados

A Tabela 11 ilustra de forma resumida as métricas de avaliação de desempenho dos diferentes modelos de regressão implementados.

Tabela 11 - Métricas de avaliação do desempenho dos modelos de regressão

Regressão	MAE	MSE	RMSE	R^2
XGBRegressor	90.988,6	18.183.418.929.9	134.845,9	0,662
Ridge	102.950,1	21.029.973.870,3	145.995,6	0,604
Linear	102.956,6	21.321.821.443,9	146.019,9	0,604
Lasso	102.964,4	21.318.988.057,9	146.010,2	0,604
ElasticNet	133.875,2	32.911.788.035,1	181.416,1	0,389

7 – Discussão de Resultados

Este estudo teve como objetivo desenvolver modelos preditivos para prever o preço de venda de imóveis em Portugal, utilizando técnicas de *Machine Learning*. O foco principal foi analisar a eficácia de diversos algoritmos de regressão e classificação na previsão de preços e na categorização da condição de imóveis. A criação de um *dashboard i*nterativo também foi uma parte essencial do projeto, visando facilitar a visualização e interpretação dos dados.

Desempenho dos Modelos de Classificação

Naive Bayes

O modelo *Naive Bayes*, embora eficiente em termos de tempo de execução, apresentou algumas limitações:

- **Precisão Geral:** 59,22%
- **Classes com Baixa Precisão:** "Renovado" (32,1%) e "P/ recuperar" (34,5%)

Essas limitações sugerem que o algoritmo de classificação *Naive Bayes* pode não ser ideal para cenários onde as variáveis são altamente correlacionadas, uma vez que assume a independência condicional entre elas.

Árvores de Decisão

As Árvores de Decisão apresentaram uma precisão geral de 64,98%, sendo mais eficazes que o *Naive Bayes*, mas ainda com desafios significativos na diferenciação das classes "*Novo*" e "*Usado*":

- **Precisão para "*Novo*":** 48%
- **Precisão para "*Renovado*":** 31,4%

Random Forest

O algoritmo *Random Forest* destacou-se como o melhor modelo de classificação:

- **Precisão Geral:** 67,59%
- **Precisão para "*Usado*":** 68,1%
- **Precisão para "*Renovado*":** 44,4%

Este modelo conseguiu reduzir os falsos positivos para "*Usado*", mas ainda enfrentou dificuldades na classificação correta das classes "*P/ recuperar*" e "*Renovado*".

XGBoost

O algoritmo de classificação *XGBoost*, conhecido por sua robustez e eficiência, apresentou uma precisão geral de 67,48%, com um desempenho particularmente forte na classe "*Usado*":

- **Precisão para "Usado":** 69%
- **Precisão para "Renovado":** 37,9%

CatBoost

O algoritmo de classificação *CatBoost*, especializado no tratamento de variáveis categóricas, teve um desempenho similar ao *XGBoost*:

- **Precisão Geral:** 66,55%
- **Precisão para "Usado":** 67,5%
- **Precisão para "Renovado":** 48,5%

Desempenho dos Modelos de Regressão

Random Forest

O algoritmo *Random Forest* mostrou-se altamente eficaz, com um desempenho robusto:

- **Erro Médio Absoluto (MAE):** 104,23
- **Erro Quadrático Médio (MSE):** 22.514.123,47
- **R2 Score:** 0,611

Estes resultados indicam que o modelo *Random Forest* é capaz de explicar cerca de 61% da variabilidade nos preços dos imóveis, o que é significativo considerando a complexidade e heterogeneidade do mercado imobiliário.

XGBRegressor

O *XGBRegressor* foi o modelo de regressão que apresentou o melhor desempenho:

- **Erro Médio Absoluto (MAE):** 102,95
- **Erro Quadrático Médio (MSE):** 21.321.821,44
- **R2 Score:** 0,604

A ligeira superioridade do *XGBRegressor* em relação ao *Random Forest* pode ser atribuída às suas capacidades avançadas de regularização e manuseio de dados ausentes, bem como a sua eficiência computacional.

Análise dos *Outliers*

A identificação e remoção de *outliers* foram cruciais para melhorar a precisão dos modelos. Os *outliers* presentes nas variáveis de preço e área foram responsáveis por distorções significativas na média e desvio padrão dos dados.

- **Preço:** Após a remoção de 4.375 *outliers*, a média dos preços ajustou-se para valores mais realistas, melhorando a performance preditiva dos modelos.
- **Área:** A remoção de 2.347 *outliers* resultou numa distribuição mais consistente e menos dispersa.

Visualizações e Padrões Identificados

As visualizações incluídas no estudo proporcionaram *insights* valiosos sobre a distribuição dos dados e as correlações entre variáveis:

- **Mapa de Calor:** Identificou uma correlação positiva entre a tipologia e a área dos imóveis, com um coeficiente de 0,63, indicando que imóveis maiores tendem a ter mais assoalhadas.
- **Gráficos de Barras:** Mostraram que os distritos de Porto, Lisboa, Setúbal e Braga possuem a maior concentração de imóveis à venda, refletindo a dinâmica do mercado imobiliário nestas regiões.
- ***Boxplots* e *Scatterplots*:** Revelaram que o preço dos imóveis aumenta com a tipologia, evidenciando uma maior dispersão de valores em tipologias superiores.

Limitações e Recomendações

Apesar dos resultados promissores, algumas limitações foram identificadas:

- **Dificuldade na Classificação de Condições:** Os modelos enfrentaram desafios na distinção entre as condições "*Novo*", "*P/ recuperar*" e "*Renovado*", sugerindo a necessidade de características adicionais ou técnicas de pré-processamento mais sofisticadas.
- **Dependência de Dados:** A qualidade e abrangência dos dados influenciaram diretamente a precisão dos modelos. Recomenda-se a utilização de fontes de dados mais diversificadas e a atualização constante dos mesmos.

Para aprimorar os modelos, sugere-se:

- **Incorporação de Novas Variáveis:** Adicionar variáveis socioeconômicas e de infraestrutura pode aumentar a precisão das previsões.
- **Refinamento dos Modelos:** Aplicar técnicas de *tuning* mais avançadas e explorar a combinação de diferentes modelos para aumentar a robustez das previsões

8 – *Dashboard*

Em *data science*, os *dashboards* desempenham um papel fundamental na análise e comunicação de informação. Estes painéis interativos, que apresentam dados em formatos visuais como gráficos e tabelas, permitem a identificação de padrões, tendências e *insights* cruciais para a tomada de decisões estratégicas em diversos contextos, desde a análise exploratória de dados até a monitorização de indicadores chave de desempenho (KPIs) em tempo real.

A versatilidade dos dashboards reside na sua capacidade de personalização. Métricas relevantes, filtros interativos e opções de *drill-down* podem ser incorporados para responder às necessidades específicas de cada projeto e aprofundar a análise dos dados.

Para a criação de *dashboards*, dispomos de uma ampla gama de ferramentas e tecnologias, desde bibliotecas de código aberto como *Python* (*Matplotlib*, *Seaborn*, *Plotly*, *Streamlit*) e JavaScript (D3.js) até soluções comerciais como *Tableau*, *Power BI* e *Qlik Sense*. A escolha da ferramenta ideal depende dos requisitos do projeto, do nível de conhecimento técnico da equipa e das restrições orçamentárias. (Sarikaya et. al 2018)

O desenvolvimento de um *dashboard* eficaz requer uma compreensão clara do público-alvo e dos objetivos da análise. A informação deve ser apresentada de forma clara, concisa e esteticamente agradável, utilizando gráficos e visualizações adequados a cada tipo de dado. A interatividade do painel é fundamental para permitir a exploração dos dados em diferentes níveis de granularidade.

Os *dashboards*, na ciência de dados, oferecem inúmeras vantagens incluindo a visualização rápida e intuitiva de grandes volumes de dados, a identificação de padrões e tendências subjacentes, a monitorização do desempenho de KPIs e a facilitação da comunicação de insights a diferentes *stakeholders*. No entanto, é crucial ter cautela na interpretação dos dados apresentados, pois visualizações inadequadas podem levar a conclusões errôneas.

Sinteticamente, os *dashboards* são ferramentas indispensáveis para a análise de dados em ciência de dados, permitindo a visualização e exploração de informações complexas de forma clara e intuitiva. Através da utilização de ferramentas e técnicas adequadas, é possível desenvolver *dashboards* eficazes que impulsionam a tomada de decisões estratégicas e o sucesso dos projetos de ciência de dados.

Analisemos agora exemplos de utilização da *dashboard* interativa gerada a partir dos modelos testados e escolhidos, na ótica do utilizador que insere os dados desejados para obter um determinado resultado que o auxilie na procura e possível eleição de um imóvel

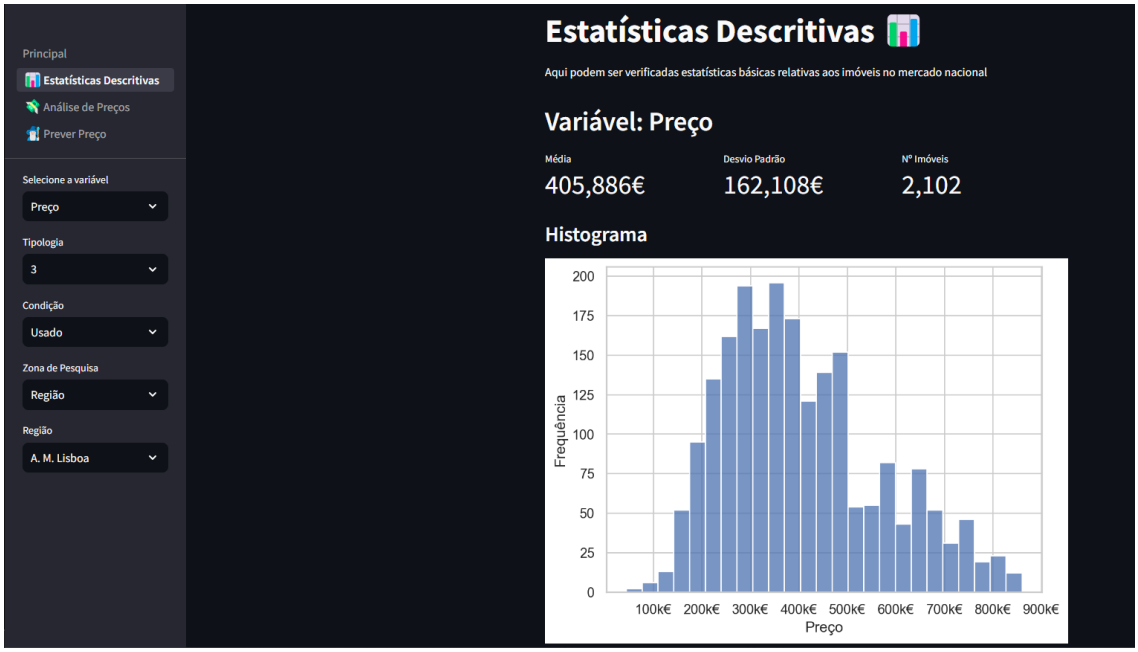

Figura 27 - *Dashboard:* Estatísticas Descritivas

As estatísticas descritivas apresentam a informação base acerca do imóvel. O utilizador escolhe qual a variável que quer visualizar descrita. O exemplo acima apresenta a eleição da variável Preço. Depois de escolhida a variável, é escolhida a tipologia, a condição do imóvel, qual a zona de pesquisa e a que região pertence o mesmo.

Este exemplo pretende saber as estatísticas descritivas em relação ao preço de um T3, usado, em toda a região da Área Metropolitana de Lisboa. São apresentadas a media, o desvio padrão, um histograma com a distribuição de preços e a quantidade de imóveis disponíveis na tipologia definida. A media é de 405886 euros com um desvio padrão 162.108 euros. Existem 2.102 imóveis da tipologia pesquisada.

Figura 28 - *Dashboard* - Análise de Preços

A ação seguinte pretende fazer uma análise de preço onde se seleciona também a tipologia, condição do imóvel, distrito e concelho. O exemplo apresentado utiliza uma tipologia T3, de

condição usado, em Matosinhos, no Porto. O resultado da análise apresenta o preço médio deste tipo de imóveis no concelho, que é de 404.664 mil euros. Apresenta ainda uma lista, com os respetivos preços dos apartamentos com as características pedidas.

Figura 29 - *Dashboard:* Prever preço de imóvel

Por fim, a *dashboard* apresentada permite ao utilizador prever o preço de um imóvel com bases nas suas características. As características do exemplo apresentado são tipologia 3, com uma área de 150 metros quadrados, de condição novo, no concelho de Santarém. O preço previsto é de 297.052 euros.

9 – Conclusão

O desenvolvimento de um modelo preditivo para os preços de venda de imóveis em Portugal, utilizando técnicas de *machine learning*, revelou-se um contributo significativo para a análise do mercado imobiliário. O estudo demonstrou que tanto as variáveis endógenas (como a área e a tipologia) quanto as exógenas (incluindo concelho e região) são determinantes importantes na previsão do valor dos imóveis. Entre os algoritmos de regressão avaliados, o *Random Forest* e o *XGBRegressor* destacaram-se pelo seu desempenho robusto, evidenciado pela precisão nas previsões e pela capacidade de lidar eficazmente com a heterogeneidade dos dados.

A criação de um *dashboard* interativo com a biblioteca *Streamlit* permitiu uma visualização intuitiva dos dados, facilitando a análise para diferentes *stakeholders*, desde investidores a agentes imobiliários. Esta ferramenta promove uma maior transparência e precisão na estimativa dos preços, contribuindo para um mercado imobiliário mais equilibrado e informado.

Este projeto sublinha a importância da aplicação de técnicas de ciência de dados no setor imobiliário, apontando para o potencial de futuras melhorias, como a integração de mais variáveis contextuais ou a adaptação dos modelos para mercados específicos. Além disso, recomenda-se a continuidade da pesquisa, explorando novas metodologias e algoritmos que possam captar com maior precisão as dinâmicas complexas do mercado imobiliário em Portugal.

Bibliografia

Livros:

Friedman, J. H. (2001). Greedy function approximation: a gradient boosting machine. Annals of Statistics, 29(5), 1189-1232.

Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (pp. 785-794).

Dorogush, A. V., Ershov, V., & Gulin, A. (2018). CatBoost: gradient boosting with categorical features support. arXiv preprint arXiv:1810.11363.

Sarikaya, Alper; Correll, Michael; Bartram, Lyn; Tory, Melanie; Fisher, Danyel. (2018). What Do We Talk About When We Talk About Dashboards? IEEE Transactions on Visualization and Computer Graphics, (), 1–1. doi:10.1109/TVCG.2018.2864903

Hastie, T., Tibshirani, R., & Friedman, J. (2009). The elements of statistical learning: data mining, inference, and prediction. Springer Science & Business Media.

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). An introduction to statistical learning. Springer.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. Journal of the Royal Statistical Society: Series B (Methodological), 58(1), 267-288.

Zou, H., & Hastie, T. (2005). Regularization and variable selection via the elastic net. Journal of the Royal Statistical Society: Series B (Statistical Methodology), 67(2), 301-320.

Zhou, H., & Zhu, J. (2010). Group variable selection via the elastic net. Journal of Computational and Graphical Statistics, 19(4), 994-1012.
Bai, J., & Ng, S. (2008). Forecasting economic time series using targeted predictors. Journal of Econometrics, 146(2), 304-317.

Apresentações:

Vardasca, R. (2024). *4 – Aprendizagem supervisionada: Algoritmos de regressão*. Apresentação realizada no contexto da unidade curricular de Aprendizagem Automática da Pós-Graduação em *Data Science* do Instituto Superior de Gestão e Administração de Santarém (ISLA).

Sítios da Internet:

https://medium.com/@dancerworld60/demystifying-na%C3%AFve-bayes-simple-yet-powerful-for-text-classification-ad92b14a5c7, acedido em 17 de Junho de 2024.

https://didatica.tech/como-funciona-o-algoritmo-arvore-de-decisao/, acedido em 17 de Junho de 2024.

https://www.analyticsvidhya.com/blog/2021/06/understanding-random-forest/, acedido em 17 de Junho de 2024.

https://www.almabetter.com/bytes/tutorials/data-science/adaboost-algorithm, acedido em 18 de Junho de 2024.

https://www.almabetter.com/bytes/tutorials/data-science/adaboost-algorithm, acedido em 18 de Junho de 2024.

https://www.researchgate.net/figure/The-flow-diagram-of-the-CatBoost-model_fig3_370695897, acedido em 18 de Junho de 2024.

https://www.ufrgs.br/reamat/AlgebraLinear/livro/s14regressx00e3o_linear_mx00faltipla.html, acedido em 19 de Junho de 2024.