NOVA IMS Information Management School

MDSAA

Mestrado em

Data Science and Advanced Analytics

Previsão do preço ótimo para carros usados com Machine Learning

Trabalho desenvolvido para uma empresa de indústria automóvel

Beatriz Gomes Ferreira Pereira

Relatório de Estágio

apresentado como requisito parcial para obtenção do grau de Mestre em Data Science and Advanced Analytics

NOVA Information Management School

Instituto Superior de Estatística e Gestão de Informação

Universidade Nova de Lisboa

NOVA Information Management School
Instituto Superior de Estatística e Gestão de Informação

PREVISÃO DO PREÇO ÓTIMO PARA CARROS USADOS COM MACHINE LEARNING

por

Beatriz Gomes Ferreira Pereira

Relatório de Estágio apresentado como requisito parcial para obtenção do grau de Mestre em Advanced Analytics, com especialização em Data Science

Orientador: Roberto André Pereira Henriques

Orientador externo*:* Paulo Barbeiro Maurício

Novembro 2023

DECLARAÇÃO DE INTEGRIDADE

Declaro ter realizado o presente trabalho académico com integridade. Confirmo que não recorri à prática de plágio ou de qualquer outra forma de utilização indevida de informação ou de falsificação de resultados durante o processo de elaboração deste trabalho. Declaro ainda que tenho conhecimento das Regras de Conduta e do Código de Honra da NOVA Information Management School.

Beatriz Gomes Ferreira Pereira

Almada, 30 de novembro de 2023

RESUMO

Este documento descreve o trabalho feito durante um estágio de 6 meses na Deloitte. O estágio teve como objetivo desenvolver uma plataforma de *analytics* na *cloud* para um cliente no setor automóvel, que visa proporcionar uma infraestrutura flexível e escalável para a análise preditiva de preços de carros usados, permitindo assim aos comerciantes desta empresa retirarem *insights* importantes sobre os seus dados e, consequentemente, fazer decisões mais estratégicas, otimizando o seu lucro.

É apresentada uma visão geral do projeto, com várias componentes diferentes, embora o foco seja nas tarefas principais da aluna. Sendo esta uma plataforma para ser utilizada por vários mercados diferentes, é apresentada uma metodologia generalista para que a análise consiga ser adaptada a diferentes tipos de dados e comportamentos.

Para avaliar esta metodologia, foi feita uma análise com foco em dados do mercado da Alemanha de vendas diretas, contendo carros retornados à empresa de 3 de maio de 2020 até 30 de junho de 2021, contando com 55 216 carros. Através de uma análise exploratória detalhada, foi possível identificar que, a maioria das variáveis usadas têm uma assimetria positiva muito acentuada, existem variáveis explicativas com dependências elevadas, multicolinearidade, e variáveis categóricas com valores muito elevados de cardinalidade.

Neste contexto, são sugeridas várias técnicas de tratamento de dados e *feature engineering*, tais como *target encoding*, com vista a melhorar o desempenho dos modelos utilizados. Para a seleção de variáveis é aplicado o método RFE, *Recursive Feature Elimination*, com o objetivo de escolher as variáveis que mais contribuem para a previsão do preço.

Procurando por robustez e precisão, optou-se por implementar e comparar modelos *ensemble*, dado que são conhecidos pela sua capacidade de captar comportamentos complexos nos dados e lidar bem com *overfitting*. Os resultados destacam a eficácia do modelo XGBoost, obtendo um R^2 de 0.965 na amostra de teste, com 19 variáveis de input. Esta performance sugere que o modelo é capaz de identificar os comportamentos complexos nos dados, porém precisa de ser analisado com mais cuidado e aplicado a outras amostras para perceber a sua generalidade.

PALAVRAS-CHAVE

Aprendizagem Automática Supervisionada; CRISP-DM; *Target Encoding*; *Overfitting*; Métodos de Ensemble.

ABSTRACT

This document describes the work done during a 6-month internship at Deloitte. The internship aimed to develop a cloud analytics platform for a client in the automotive sector, which is intended to provide a flexible and scalable infrastructure for the predictive analysis of used car prices, thus, enabling the company's dealers to draw important insights from their data and consequently help them on their decision-making process, optimizing their profit.

An overview of the project is presented, with several different components, although with a focus on the student's main tasks. As this is a platform to be used by several different markets, a general methodology is presented so that the analysis can be adapted to different types of data and behaviours.

To evaluate this methodology, an analysis was carried out on data from the German direct sales market, containing used cars returned to the company from May 3, 2020 to June 30, 2021, with 55 216 cars. Through a detailed exploratory analysis, it was possible to identify that most of the variables used have a very pronounced positive asymmetry, some of the explanatory variables have high dependencies, multicollinearity, and that there are categorical variables with a very high cardinality.

In this context, various data processing and feature engineering techniques are suggested, such as target encoding, to improve the performance of the models used. The RFE method, Recursive Feature Elimination, is used to select the variables that contribute most to price prediction.

Seeking for robustness and accuracy, we chose to implement and compare ensemble models, since they are known for their ability to capture complex behaviours in the data and deal well with overfitting. The results highlight the effectiveness of the XGBoost model, obtaining an R^2 0.965 in the test sample, with 19 input variables. This performance suggests that the model is able to identify complex behaviors in the data, but it needs to be analyzed more carefully and applied to other samples to understand its generality.

KEYWORDS

Supervised Machine Learning; CRISP-DM; Target Encoding; Overfitting; Ensemble Methods.

ÍNDICE

ÍNDICE DE FIGURAS

ÍNDICE DE TABELAS

LISTA DE SIGLAS E ABREVIATURAS

B2B	Business to Business
SVM	Support Vector Machine
LASSO	Least Absolute Shrinkage and Selection Operator
ANN	Artificial Neural Network
KNN	K-Nearest Neighbors
MAPE	Mean Absolute Percentage Error
RFE	Recursive Feature Elimination
IUC	Imposto Único de Circulação
ETL	Extract Transform Load
AWS	Amazon Web Services
IaC	Infrastructure as Code
VIN	Vehicle Identification Number
CRISP-DM	Cross-Industry Standard Process for Data Mining
KPI	Key Performance Indicator
S3	Simple Storage Service
IAM	Identity and Access Management
SQL	Structured Query Language
MAE	Mean Absolute Error
RMSE	Root Mean Squared Error
SHAP	SHapley Additive exPlanations
LLM	Large Language Model

1. INTRODUÇÃO

A indústria de automóveis usados registou um crescimento significativo nos últimos anos, uma vez que a competitividade dos preços entre os novos operadores tem sido um ponto fulcral na indústria de automóveis usados. A incapacidade de os clientes comprarem automóveis novos tornou-se uma das razões para o crescente volume de vendas de automóveis usados, que é complementado pelos investimentos efetuados pelos participantes no setor para estabelecerem a sua rede de concessionários no mercado. Estas redes de concessionários ajudaram a comercializar a marca dos participantes e a viabilizar as opções de compra de automóveis usados. («Used Car Market Size, Share & Trend Analysis Report By Vehicle Type (Hybrid, Conventional, Electric), By Vendor Type, By Fuel Type, By Size, By Sales Channel, By Region, And Segment Forecasts, 2023 - 2030», 2022). Dado esta competitividade, antecipar e ajustar proactivamente os preços aplicados é essencial para uma empresa manter o seu lugar no mercado e maximizar o seu lucro.

No entanto, a previsão do preço de carros usados não é uma ciência exata, é uma tarefa bastante desafiante dado o grande número de fatores que podem estar associados com a sua variação. O preço de um veículo pode variar desde variáveis que o caracterizam, tais como a sua idade, popularidade e quilometragem, porém as suas variações podem também estar associadas com o tipo do vendedor, a tendência do mercado e a oferta existente que pode obrigar a exerção de preços mais competitivos. É por isso importante ter um processo de tratamento de dados robusto que consiga evitar possíveis variações atípicas, caracterizadas como *outliers*.

O trabalho que aqui se apresenta foi desenvolvido em âmbito de estágio na Deloitte, com duração de 6 meses, entre 2 de janeiro de 2022 até 1 de julho de 2022. O objetivo do estágio era o desenvolvimento de uma plataforma de *analytics* na *cloud* para um cliente no setor automóvel, que visa proporcionar uma infraestrutura flexível e escalável para a análise preditiva de preços de carros usados, permitindo assim aos comerciantes desta empresa retirarem *insights* importantes sobre os seus dados e, consequentemente, fazer decisões mais estratégicas, otimizando o seu lucro.

Com vista ao objetivo apresentado, a aluna teve a oportunidade de pertencer a duas equipas diferentes, numa primeira fase à equipa de *Data Engineering* e na segunda fase à equipa de *Data Science*. Sendo que os objetivos da primeira foram migrar os dados de várias fontes do cliente para a *cloud* e criar um processo de ETL dinâmico para facilitar a integração dos dados em futuros *dashboards* e também para a sua utilização na segunda fase do projeto, focada na análise preditiva do preço para a venda de carros usados.

Neste contexto, foram exploradas diversas técnicas de *feature engineering* e limpeza de dados, com o objetivo de otimizar a qualidade e relevância das variáveis utilizadas pelo modelo preditivo. Procurando por robustez e precisão, optou-se por implementar e comparar modelos *ensemble*, dado que são conhecidos pela sua capacidade de captar comportamentos complexos nos dados e lidar bem com *overfitting*. Dado a característica destes modelos, de combinar decisões de vários modelos para originar um modelo mais forte, acredita-se que oferecem uma solução promissora para enfrentar os desafios inerentes à previsão do preço de carros usados, onde a variabilidade é uma constante.

A implementação bem-sucedida desta abordagem pode resultar em vantagens competitivas substanciais, proporcionando à empresa uma visão mais proativa e estratégica na gestão do preço de revenda de carros usados.

1.1. Compreensão do Problema

A análise preditiva visa ajudar os comerciantes vendedores da empresa na revenda de carros usados a outros comerciantes compradores, estando perante um cenário de B2B. A figura 1.1, tem como objetivo ajudar o leitor a perceber a *timeline* do processo de revenda da empresa e em que momento a solução apresentada irá atuar.

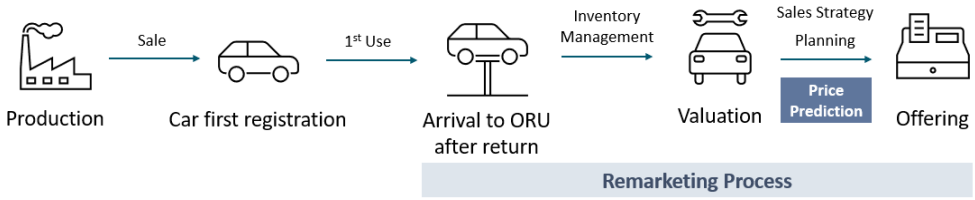


Figura 1.1 – *Timeline* do processo de revenda

Sendo esta empresa também produtora dos carros que revende, este processo pode ser visto com início na produção do carro. É feito o seu primeiro registo no momento da venda do carro, que pode ser feita com vista a diferentes usos, tais como *leasing* ou até para algo mais curto, tais como demonstrações de modelos mais recentes. Depois deste primeiro uso, o carro é retornado à ORU, o armazém que faz a gestão e avaliação do estado do carro retornado. É então, neste momento, que é feita a aplicação do modelo preditivo do preço, com o intuito de ajudar os comerciantes na sua estratégia de vendas, e mais especificamente na fixação do preço para cada carro. Tal como irá ser apresentado na metodologia, o modelo preditivo não atua sozinho, tendo um modelo de otimização a consumir o seu output para fazer a gestão das vendas, ajudando o comerciante na decisão dos carros que deve vender e na escolha do canal de venda que irá otimizar o lucro da venda. Os canais de venda são definidos em: venda direta, leilões fechados (para comerciantes apenas da empresa) e leilões abertos (para comerciantes da empresa e outros comerciantes livres). Sendo o objetivo otimizar o lucro, esta é uma escolha que deve também ser feita com cuidado. Por exemplo, modelos mais exclusivos podem ter maiores ofertas em leilões, mas é preciso ter em conta o custo associado à sua participação no leilão e a incerteza associada ao tipo de compradores em cada leilão.

Estando a escolha do canal de venda feita e o preço previsto, o comerciante começa a sua fase de oferta, podendo esta ser incluída por várias ofertas diferentes. Nesta fase, o comerciante precisa de aplicar o seu conhecimento para a negociação do preço dependendo da tendência e oferta externa do mercado.

1.2. Estrutura do Documento

O seguinte documento é dividido em 4 capítulos. O primeiro capítulo, apresenta a investigação feita sobre o domínio da aprendizagem automática em análises preditivas, identificando os modelos mais relevantes para este tipo de análise, e apresenta também diversos estudos e trabalhos feitos para a previsão do preço de carros usados, podendo assim tirar insights importantes para a definição da metodologia, descrita no capítulo a seguir.

A metodologia, apresenta inicialmente, uma *overview* do projeto, descrevendo todas as suas componentes e a gestão de tarefas entre a equipa. Posteriormente é dada uma introdução ao módulo da previsão do preço que é complementado pelas secções a seguir, detalhando todas as etapas

necessárias para o seu desenvolvimento. Nomeadamente, o levantamento de dados, a análise exploratória, o tratamento de dados e a modelação.

O capítulo 4, apresenta os resultados do estudo, comparando e discutindo os modelos escolhidos e as suas performances. E, por fim, o capítulo 5 apresenta a conclusão deste trabalho, sumarizando os seus desafios e limitações, como também várias recomendações de trabalho futuro e uma breve avaliação do estágio.

2. REVISÃO DA LITERATURA

Na era da explosão de dados, a aprendizagem automática surgiu como uma ferramenta poderosa para a análise preditiva em vários domínios. A análise preditiva tem como objetivo fazer previsões baseadas em padrões de dados históricos e tem um vasto número de aplicações que vão desde os cuidados de saúde às finanças, esta investigação está orientada para um contexto específico - a indústria automóvel. Em particular, pretende-se explorar a investigação existente em torno da previsão de preços de carros usados, que tem uma importância significativa para as empresas automóveis que procuram melhorar as suas estratégias de preços e tomar decisões comerciais mais informadas.

Nesta revisão da literatura, está incluída uma exploração dos modelos de aprendizagem automática mais relevantes para a previsão de preços de carros usados. Ao examinar diversos estudos e projetos relacionados, pretende-se identificar os algoritmos específicos de aprendizagem automática aplicados neste domínio, proporcionando uma análise mais profunda das suas capacidades, vantagens e desafios. Estes conhecimentos irão ajudar a preparar o terreno para esta investigação, que procura contribuir para melhorar a estratégia de definição de preços de revenda da indústria automóvel.

2.1. APRENDIZAGEM AUTOMÁTICA EM ANÁLISES PREDITIVAS

A análise preditiva é o ramo da análise avançada que é utilizada para efetuar previsões sobre eventos futuros. Este processo envolve uma vasta gama de técnicas analíticas, incluindo análise de dados, análise estatística e aprendizagem automática. A aprendizagem automática é um ramo da inteligência artificial, que automatiza a criação de modelos estatísticos e analíticos que permite os sistemas aprenderem com os dados, reconhecerem padrões e a fazerem previsões com pouca intervenção humana (B. Nithya & Dr. V. Ilango, 2017; John D. Kelleher et al., 2020) .

Para construir os modelos usados em análises preditivas, é utilizada a aprendizagem supervisionada. A aprendizagem supervisionada ocorre quando um modelo é treinado com dados já classificados, conseguindo aprender assim a relação entre as variáveis descritivas (dependentes) e a variável *target* (independente). Uma vez o modelo treinado com base nos dados classificados o sistema será capaz de tomar decisões quando receber novos dados de entrada (não classificados) e desta forma atribuir uma saída (classificação) (GeeksforGeeks, 2023). Existem dois tipos de subcategorias dentro da aprendizagem supervisionada, modelos de classificação e modelos de regressão. Os modelos baseados em classificação têm como objetivo identificar a que classe pertence uma determinada amostra do problema, por exemplo se um determinado e-mail é considerado spam ou não. Já os sistemas baseados na regressão focam-se essencialmente em prever um valor numérico, ou seja, o modelo pode aprender uma função para prever o preço de um imóvel. Os modelos baseados em regressão associam-se a problemas com respostas quantitativas e os modelos baseados em classificação a problemas com respostas qualitativas.

2.1.1. Métodos e Aplicações

Esta secção é dedicada à descrição das técnicas mais utilizadas em análises preditivas do preço de carros usados. Concretamente, as seguintes técnicas são descritas: modelos de regressão linear, árvores de decisão, métodos de ensemble – Random Forest (floresta aleatória em português), AdaBoost, Gradient Boosting, e XGBoost – e, por último, as redes neuronais artificiais (ANNs).

2.1.1.1. Regressão Linear

Regressão linear é uma das técnicas mais populares de aprendizagem automática e a base de vários modelos mais complexos. O modelo tenta prever a variável independente com base na sua relação linear com uma ou múltiplas variáveis dependentes. É um modelo com bastante interpretabilidade, no entanto tem alguns pressupostos que podem não ser adequados aos dados do preço de carros usados. Tais como, a relação entre as variáveis ser linear, não conseguindo captar outro tipo de relações mais complexas, e ter o pressuposto de não multicolinearidade, querendo isto dizer que as variáveis independentes não podem ter uma correlação alta entre si (Su et al., 2012), o que não é respeitado por algumas características de um carro, como por exemplo, o modelo do carro e a carroçaria.

2.1.1.2. Árvores de decisão

Árvores de decisão são um modelo de classificação, no entanto podem também ser usadas em problemas de regressão. Tal como o nome indica, este modelo cria um fluxograma de decisões, em que cada ponto é representado por um "nó" de uma árvore e em cada um deles é feita uma pergunta binária, podendo criar dois caminhos diferentes, os "ramos". Cada decisão é feita utilizando o melhor atributo de divisão calculado através de vários critérios, tais como o ganho de informação e o índice de Gini. A variável com menos impurezas é selecionada como "nó" de divisão. Uma vez obtida uma subpartição considerada pura, os "nós" consideram-se como "nós folha" e deixa-se de construir a árvore (Kingsford & Salzberg, 2008).

As vantagens deste método é que tem também um grau elevado de interpretabilidade, dado que segue um processo de regras parecido ao de um humano, mas ao contrário da regressão linear consegue lidar com relações não lineares e com multicolinearidade entre variáveis independentes, sendo que se uma das variáveis for selecionada para a decisão, a segunda não será selecionada no nível seguinte, uma vez que a impureza já é explicada pela primeira variável (Mane, 2021).

Porém, dado que é um método bastante sensível à amostra em que é treinado é fácil de obter resultados com *overfitting*. No entanto, esta desvantagem pode ser controlada ao limitar certos parâmetros do modelo, tais como o número mínimo de observações numa "folha" e o número máximo de "nós" até atingir uma "folha". Desta forma evita-se que sejam feitas demasiadas divisões dos dados e acabar por ter um número muito pequeno de observações numa "folha". Outra solução que pode conseguir controlar este *overfitting* são os métodos de ensemble, dado que estes combinam o output de vários modelos (S. Kumar, 2021).

2.1.1.3. Métodos de ensemble

Métodos de ensemble consistem em combinar decisões de vários modelos com vista a melhorar a sua performance. São baseados no princípio que considerando vários outputs de modelos mais fracos, conhecidos como *base learners*, podem originar um modelo mais forte, assumindo que estes são o mais diversos possível e irão apenas ser treinados num subconjunto da amostra, sendo possível que, ao juntá-los, resulte numa perspetiva mais abrangente dos dados, evitando resultados enviesados (Wang et al., 2011).

Random Forest

Random Forest tem como *base learner* as árvores de decisão. O modelo combina várias árvores de decisão, formando uma floresta, que definem as suas decisões em paralelo baseadas em subconjuntos

dos dados, que são selecionadas aleatoriamente com substituição da amostra inicial, usando o conceito de *bagging* (Breiman, 2001). Neste processo de amostragem, algumas observações podem aparecer mais que uma vez em cada subconjunto, enquanto outras podem nem ser incluídas. No final, o output escolhido é definido pela maioria dos votos de todas as árvores, treinadas nestes subconjuntos de dados, em problemas de classificação, e para problemas de regressão é feita a média de todas as previsões (Zhang & Haghani, 2015). Como o modelo tem esta componente aleatória e junta múltiplas árvores com características diferentes, é particularmente robusto a *outliers*.

AdaBoost

AdaBoost, diminutivo de Adaptive Boosting, pode integrar outros algoritmos como *base learners*, tais como regressões lineares ou SVMs, e ao contrário do Random Forest, os modelos são treinados em sequência e não em paralelo, seguindo o conceito de *boosting*. Este método concentra-se nos dados que são mais difíceis de prever, atribuindo-lhes mais peso, e assim poderem ser processados na próxima iteração de treino com mais representatividade. (Schapire, 2013) refere que este modelo é tipicamente resistente a *overfitting*, no entanto é possível acontecer e, por isso, propõe usar um termo de regularização, tal como o *LASSO* (Tibshirani, 1996), para controlar os pesos da função objetivo.

Gradient Boosting

Gradient Boosting é um modelo bastante conhecido por conseguir excelentes resultados em dados que contêm variáveis heterogéneas, relações complexas e *outliers* (Friedman, 2001). De acordo com o mesmo conceito de *boosting*, do Adaboost, os *base learners* são treinados sequencialmente, no entanto, em vez de ajustar os pesos das observações, o objetivo é minimizar o erro do modelo anterior, por meio de uma função de perda, usando *gradient descent*.

XGBoost

XGBoost, diminutivo de eXtreme Gradient Boost, é uma implementação otimizada do Gradient Boosting (Chen & Guestrin, 2016). Adiciona mais opções de funções de perda, opções de regularização para controlar *overfitting*, como o termo de penalização LASSO na função de perda e o controlo dos parâmetros de cada *base learner*, consegue lidar com dados em falta, e foi também projetado para ser mais eficiente e rápido, usando processamento paralelo (Morde, 2019).

2.1.1.4. Redes Neuronais Artificiais (ANNs)

Redes Neuronais Artificiais são baseadas na capacidade de o sistema nervoso humano processar sinais de input e produzir outputs. Este modelo estruturado numa rede, é composto por várias camadas conectadas entre "nós", os neurónios. A camada de input, consiste no conjunto de neurónios associados às variáveis explicativas, a camada de output corresponde à variável *target* e as camadas intermédias, conhecidas por *hidden layers*, são responsáveis por transformar a entrada de todos os neurónios da camada anterior num valor que a próxima camada possa usar. Esta transformação é feita aplicando um peso a cada input da camada anterior e passando o resultado por uma função de ativação. Se o output de um neurónio estiver acima de um certo *threshold*, esse neurónio é ativado e a sua informação é passada para a próxima camada, caso contrário não são passados dados para a próxima camada. Este processo é tipicamente feito usando o algoritmo de *back-propagation*, cujo objetivo é minimizar a soma do erro quadrático entre o valor do output da rede e o valor real do *target*, ajustando os pesos das conexões entre neurónios (Walczak & Cerpa, 2003).

Este tipo de modelo é conhecido por ter a capacidade de aprender padrões bastante complexos, no entanto exige um maior poder computacional e uma grande amostra de dados.

2.2. TRABALHOS RELACIONADOS

Ao longo dos anos, têm sido feitos diversos estudos com o objetivo de prever o preço de carros usados, usando várias metodologias diferentes de aprendizagem automática, com taxas de sucesso a chegar aos 90%. De seguida são apresentados alguns estudos com potenciais insights para a metodologia utilizada neste trabalho.

Em (Pudaruth, 2014), aplicaram-se várias técnicas de aprendizagem automática, como árvores de decisão, Naïve Bayes, KNN e regressão linear múltipla, para prever o preço de carros usados publicados em jornais na República das Ilhas Maurícias. O autor obteve uma *accuracy* entre 60% e 70% com os algoritmos das árvores de decisão e Naïve Bayes, no entanto sugere o uso de algoritmos mais sofisticados, apontando que a principal desvantagem destes algoritmos é o facto de serem modelos de classificação, e por isso, ser necessário categorizar o preço em diversas classes e assim resultando em piores resultados. Além disso, sugere treinar os modelos com um maior número de dados, uma vez que, após algum processamento, os modelos foram treinados com apenas 97 observações.

Já (Monburinon et al., 2018), optaram por usar modelos de regressão para prever o preço de carros usados. Neste artigo, os autores usaram dados de um site de *e-commerce* alemão com 304 133 observações. Comparando os resultados de vários modelos, usando o erro percentual absoluto médio (MAPE do inglês), atingiram os melhores resultados com a regressão Gradient boosting, tendo o erro um valor de 0.28. Seguidos do modelo de Random Forest com um MAPE de 0.35 e da regressão linear com um MAPE de 0.55. Como melhoria os autores sugerem ajustar os parâmetros utilizados, e dão importância à fase de *data engineering*, mostrando relevância no uso de *one hot enconding* como alternativa ao *label encoding* na transformação de dados categóricos.

Com um *dataset* do Kaggle de 370 000 carros usados, tirados de um site alemão, eBay Kleinanzeigen, (Pal et al., 2018) usaram Random Forest. Os autores tentaram também aplicar uma regressão linear aos dados, porém o coeficiente de determinação, R^2, da amostra de treino foi menos de 75%. Usando uma Random Forest com 500 árvores de decisão, obtiveram um R^2 de 95.82% na amostra de treino e de 83.63% na amostra de teste. Com a ajuda de uma análise exploratória e o cálculo da correlação das variáveis com o preço, identificaram que as variáveis mais relevantes para a previsão foram a quilometragem, a marca e o tipo de veículo (entre eles, limusina, SUV e cabrio são exemplos).

Da International Burch University em Saravejo (Gegic et al., 2019), com 797 observações de carros, pós processamento, tirados de um website da Bósnia e Herzegovina, conseguiram obter uma *accuracy* de 87.38%. Os autores combinaram três técnicas diferentes, utilizaram o Random Forest para categorizar os preços em grupos diferentes, e para cada grupo escolheram o modelo com melhor resultados, entre SVMs ou redes neuronais artificiais.

(Kumar & Samruddhi, 2020) propuseram usar o modelo KNN para prever o preço de carros usados de uma amostra obtida pelo Kaggle incluindo 14 atributos. Depois de experimentarem diferentes valores de K e com uma percentagem de treino de 85% chegaram a uma *accuracy* de 85%, com K=4. Os autores validaram também o modelo com validação cruzada com 5 e 10 *folds*, obtendo respetivamente uma *accuracy* de 80.9% e 82%.

Em (Hankar et al., 2022), os autores usaram dados de um site de *e-commerce* marroquino, Avito, para analisar diversos modelos de regressão na previsão de carros usados. Para selecionar apenas as variáveis mais relevantes usaram um método de eliminação recursiva (RFE), sendo as selecionadas, por ordem de importância, o ano de fabrico, quilometragem, marca, tipo de combustível, taxa fiscal (equivalente ao IUC em Portugal), e o modelo do carro. Comparando cinco regressores diferentes, nomeadamente, a regressão linear múltipla, o KNN, Random Forest, Gradient Boosting e redes neuronais, o regressor do Gradient Boosting apresentou os melhores resultados, com um valor mais elevado do R^2, 0.8, e o menor valor do erro médio. Para melhorar a performance, os autores sugerem normalizar os dados e a adição de mais variáveis descritivas.

Considerando a revisão feita, conclui-se que este tema, apesar de analisado várias vezes, ainda não foi posto claro quais as variáveis mais importantes para a previsão e o modelo com maior desempenho, embora os métodos de ensemble tenham mostrado, na sua maioria, os melhores resultados. Dependendo da fonte de dados, a previsão do preço para um carro usado é muito variante, dado os diferentes parâmetros para fazer esta avaliação, as variáveis e a quantidade de dados disponíveis.

No contexto deste problema, em que o objetivo é prever o preço de revenda num contexto B2B, esta incerteza torna-se ainda maior dado que vários comerciantes podem estabelecer preços diferentes para carros considerados no mesmo nível, e se torne assim difícil o modelo conseguir entender o padrão geral, sem ser afetado por *outliers*. É por isto importante, selecionar um modelo robusto a *outliers* e que consiga captar diferentes relações entre as variáveis.

Outro ponto que deve ser ressaltado é a esperada correlação existente entre as variáveis explicativas, na maioria das análises feitas incluem atributos tais como a marca, o modelo e o tipo de veículo, que são vistos como dependentes entre si. Sendo assim possível identificar que, uma regressão linear não deve ser aplicada a esta análise, dado o seu pressuposto de não multicolinearidade.

3. METODOLOGIA

Durante o estágio na Deloitte, com duração de 6 meses, houve a possibilidade de ter contacto com várias áreas e ferramentas. Dentro do mesmo projeto, existiu a oportunidade de pertencer a duas equipas diferentes, numa primeira fase à equipa de *Data Engineering* e na segunda fase à equipa de *Data Science*.

Para a gestão de ambas as fases, foi usada a metodologia Agile (Abrahamsson et al., 2017). Esta metodologia é reconhecida pela sua flexibilidade e capacidade de se adaptar às mudanças no decorrer de um projeto. Esta metodologia foi utilizada através da *framework* de gestão – Scrum. Este nome, derivado do rugby, caracteriza uma formação em que toda a equipa desempenha um papel específico, mas em que todos trabalham para uma rápida adoção de estratégias. Neste contexto, existiram três roles diferentes (Malsam, 2023), o *Scrum Master* – uma pessoa da Deloitte que ajudou a equipa a adotar um ambiente Agile e, sobretudo, na definição das tarefas, liderando todas as reuniões e fazendo a ponte com o *Scrum Product Owner* – uma pessoa do lado do cliente que participa ativamente nas reuniões, dando feedback sobre as tarefas feitas e priorizando as próximas. Por último, a *Scrum Development Team*, composta por todos os membros que contribuíram para o desenvolvimento do produto, neste caso apenas membros da Deloitte, incluindo a aluna.

Um dos princípios fundamentais do desenvolvimento ágil é o uso de ciclos de trabalho curtos, conhecidos como *sprints*, que normalmente têm uma duração fixa. Neste caso, foram usados ciclos de três semanas. No final de cada *sprint*, foi feita uma revisão de todas as tarefas definidas e foram planeados novos passos com o devido feedback e aprovação do *Scrum Product Owner*.

Para facilitar a gestão das tarefas e a colaboração da equipa, usou-se o software Jira, desenvolvido pela empresa Australiana Atlassian. O Jira permitiu que a equipa fizesse o planeamento de tarefas de forma eficiente e com visibilidade para todos, promovendo transparência e uma comunicação eficaz entre os membros da equipa. Uma *overview* do planeamento de tarefas pode ser consultada na figura 3.2.

3.1. ARQUITETURA GERAL DO PROJETO

Na figura 3.1, é possível identificar as várias componentes do projeto.

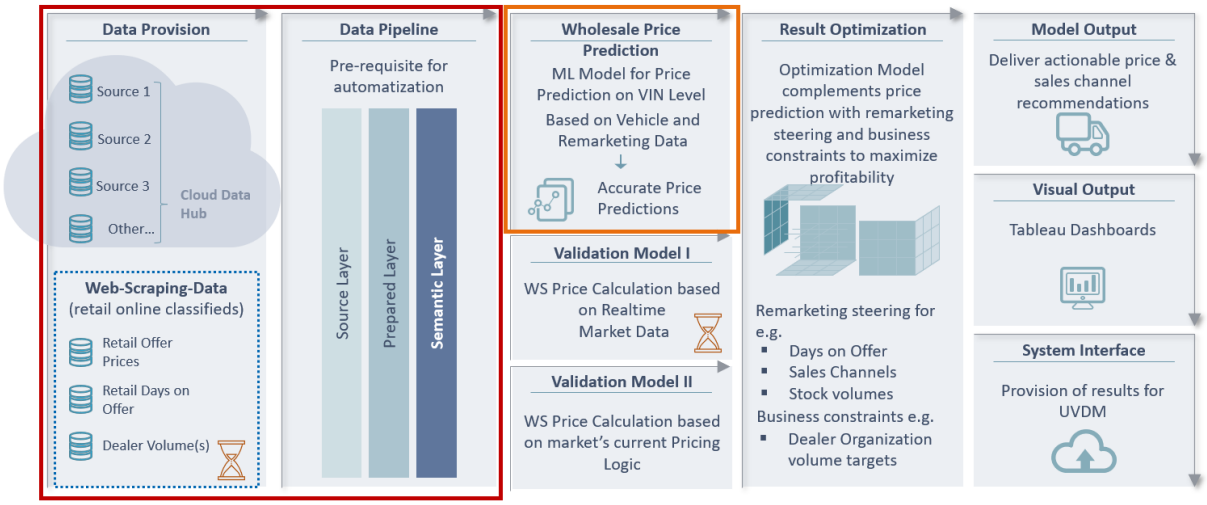


Figura 3.1 – Arquitetura do projeto

Delineado a vermelho, a primeira fase do projeto, caracterizada pela ingestão de dados de várias fontes e o tratamento de dados integrado numa data pipeline dividida em três *layers*.

Delineado a laranja, a segunda fase do projeto, a análise preditiva do preço com base nos dados descritivos de um veículo e o seu processo de revenda. Complementarmente ao modelo apresentado, foram ainda definidos dois modelos de validação, o modelo II definido pela lógica aplicada normalmente pelos comerciantes, caracterizado por um conjunto de regras, e o modelo I, com vista a adicionar inputs sobre a tendência do mercado e assim atribuindo preços mais competitivos.

Como auxiliar à estratégia de vendas, adicionalmente à previsão do preço, é aplicado um sistema de otimização que gere o volume e a escolha ideal de carros para cada comerciante definindo o canal de venda mais indicado para cada carro e tendo em conta várias restrições de negócio.

E finalmente, para a implantação da solução e apresentação de resultados foram definidos vários *dashboards* que irão apresentar não só a previsão do preço por canal de venda, como o canal de venda escolhido pelo modelo de otimização e outras características do veículo para que os comerciantes possam ter flexibilidade na sua escolha final.

Para ilustrar a divisão de tarefas ao longo do estágio, com o objetivo de atingir as componentes apresentadas, fez-se um esquema com as tarefas da aluna em *highlight*.

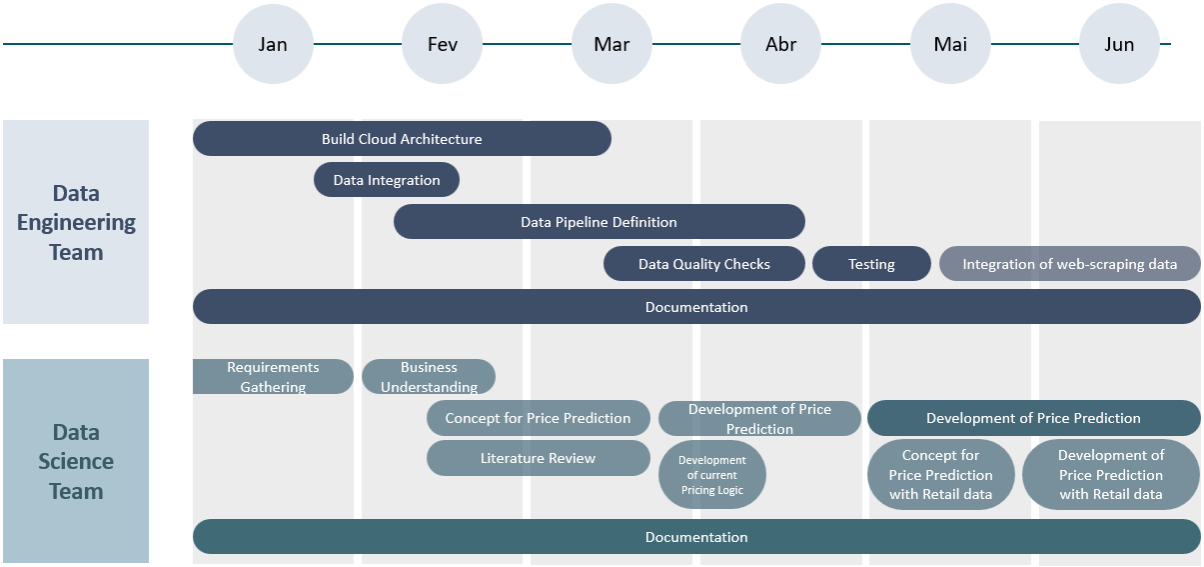


Figura 3.2 – *Roadmap* do projeto

Nos primeiros dois meses, essencialmente, a equipa fez o *setup* da *cloud,* neste caso usando os serviços da AWS (*Amazon Web Services*), através de *Infrastructure as Code (IaC)*, permitindo criar um processo mais automatizado e facilmente escalável, sem ser necessário interagir com a interface da *cloud* para usar os seus serviços. Durante este processo, ainda iterativo, pela definição dos serviços necessários ao longo do seu desenvolvimento, fez-se a integração dos dados do cliente e começou-se também a definir a pipeline dos dados, esta última, estendendo-se até quase ao final do quarto mês, dado a integração de vários mercados diferentes, com fontes e processamentos diferentes. Estes dois passos serão explicados em mais detalhe na secção 3.3.

Para complementar a pipeline dos dados, foram criados vários testes de validação da qualidade dos dados, incluindo regras de negócio e regras de qualidade, tais como o *VIN*, indicador único do veículo, não ser *null* e tabelas de dimensão não terem duplicados, respetivamente.

Já na transferência de equipas, a equipa de *Data Engineering* estava em fase de *testing*, para validação dos pressupostos do cliente e *deployment* para produção.

Nos últimos dois meses, o foco foi o desenvolvimento do modelo para prever o preço de revenda com os dados estáticos de um veículo e do processo de revenda. Tal como ilustrado, já teria havido trabalho prévio feito neste contexto, nomeadamente levantamento de requisitos, compreensão do processo de revenda e a definição do conceito. Com a implementação da lógica corrente da atribuição do preço, como modelo de validação da primeira fase do desenvolvimento do modelo foram apontados vários pontos a melhorar. Designadamente, uma melhoria no tratamento dos dados e a integração de um modelo de validação adicional com dados externos do mercado de carros usados.

A documentação foi algo essencial e comum às duas fases do projeto, para uma boa gestão da metodologia e apresentação dos resultados aos *stakeholders* do projeto.

Dado que o foco desta tese está na segunda fase do projeto, e mais concretamente, no primeiro módulo da previsão do preço de revenda, é feita uma introdução à sua metodologia na secção a seguir, e detalhada cada etapa do processo nas seguintes secções.

3.2. INTRODUÇÃO AO MÓDULO DA PREVISÃO DO PREÇO

O desenvolvimento deste módulo foi feito em linguagem Python, sendo o seu código e pipeline integrados no serviço de AWS Glue (*AWS Glue Documentation*, 2023). Este serviço permite automatizar o processo de tratamento dos dados e assim, tornar a fase de modelação mais dinâmica. Permite orquestrar vários scripts, os Glue Jobs, através de *workflows* que podem ser agendados para serem executados com uma dada frequência temporal ou até pela ação de outro evento, tal como, a ingestão de novos dados na *cloud*. Neste caso foi criado um Glue Job para o tratamento dos dados e outro com todas as etapas da modelação.

Dado o feedback da primeira análise dos modelos, a segunda iteração do seu desenvolvimento focou-se no tratamento dos dados e foram apenas testados modelos baseados no método de *ensemble*, usando as bibliotecas *sklearn* (Pedregosa et al., 2011) e *xgboost* (*XGBoost Documentation*, 2022). Estes modelos apresentam várias vantagens para o tipo de dados em questão, conseguindo captar comportamentos mais complexos e não lineares, serem robustos a *outliers* e produzirem resultados menos enviesados. Estes modelos são também bastante úteis para perceber a importância de cada variável na previsão do preço, e assim, auxiliar na identificação e seleção das mesmas.

Dado a diversidade de dados e discrepância entre os países em que a empresa tem atividade o desafio foi criar uma fase de modelação generalizada para receber diferentes variáveis e comportamentos de dados. No entanto, apesar de passarem pelo mesmo processo, foi estipulado dividir os dados entre mercados para a sua modelação individual, podendo serem escolhidos diferentes modelos e diferentes variáveis para cada mercado.

Para além do mercado, foi estipulado que para cada canal de venda, descritos na secção 1.1., deveriam também ser treinados modelos diferentes, dado que a lógica para prever o preço de cada carro varia em diferentes canais e diferentes tipos de carros são estrategicamente inseridos em diferentes canais para poder otimizar, assim, a sua margem de lucro.

No entanto, apesar da análise ser distinta para cada mercado, um carro deverá passar pelos três modelos de cada canal. Será o modelo de otimização a fazer a escolha ótima, tendo em conta outras variáveis, tais como custos associados e a exclusividade do veículo, sendo esta última uma variável bastante importante para a valorização do veículo quando estes são vendidos em leilões fechados.

Para além destas duas distinções, foi definido que os dias que o carro demora a ser vendido pode ser também uma variável determinística do preço, esta com mais peso quando o canal de venda escolhido é a venda direta. Ao ser uma variável de input do modelo e desconhecida na amostra a estimar definiu-se que a previsão iria ser feita para um dia, dado que será o momento em que o comerciante vendedor quer pôr o carro à venda.

Estando a estrutura e o conceito deste módulo definida, foi seguida uma metodologia baseada no CRISP-DM (*Cross Industry Standard Process for Data Mining*). O CRISP-DM (Chapman, 2000) consiste num conjunto de boas práticas para a execução de um projeto de *Data Science*, mas também já muito utilizado e adaptado para outras áreas. É composto por seis etapas, fazendo a associação para o português, o conhecimento do negócio, o conhecimento dos dados, a preparação dos dados, a modelação, a avaliação e a implantação da solução.

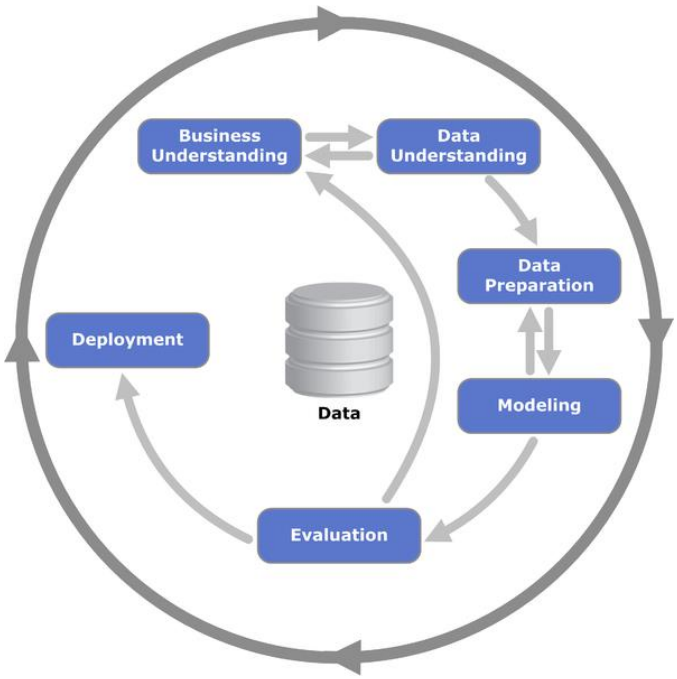


Figura 3.3 – Metodologia CRISP-DM

Estas etapas são bastante dependentes entre si e não são para ser executadas linearmente, esta metodologia caracteriza-se pela sua ciclicidade.

A primeira etapa caracteriza-se pela definição do problema e objetivos a serem atingidos, descritos no capítulo da introdução do documento. Nos próximos capítulos vamos ver a caracterização das quatro etapas a seguir, sendo que ainda não foi feita a implantação da solução apresentada. Considera-se que esta solução poderá ser melhorada, e por isso retornar à primeira etapa com as conclusões retiradas da sua avaliação.

De notar que para o estudo da solução e análise de resultados, nomeadamente as etapas do conhecimento dos dados, a preparação dos dados, e a avaliação foram feitas, em primeiro lugar, considerando o mercado da Alemanha de vendas diretas. Sendo assim apresentada a avaliação da etapa de modelação apenas nessa amostra, no entanto, o conceito foi desenhado considerando a possibilidade de input de outros mercados e também canais de venda.

3.3. Levantamento de Dados

Como referido anteriormente, na primeira fase do projeto foi feito o levantamento de dados. Para a sua descrição, é feita uma introdução à infraestrutura criada na *cloud* e posteriormente são detalhadas a estrutura e a preparação dos dados utilizados para o módulo da previsão do preço.

O processo para a integração dos dados e o seu tratamento na *cloud* pode ser descrito pelo seguinte esquema.

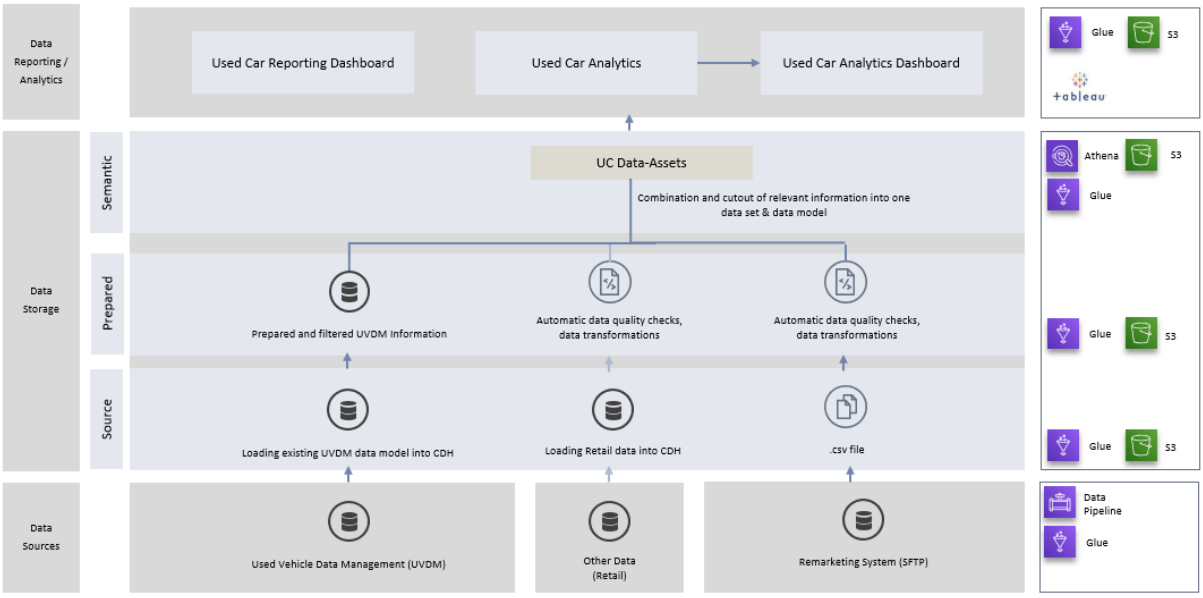


Figura 3.4 – *Flow* dos dados na *cloud*

Lendo o esquema de baixo para cima, é possível caracterizar que foram precisos vários sistemas diferentes para a ingestão de várias fontes, sendo estas uniformizadas na última *layer* da pipeline dos dados.

Para o processo de ingestão e transformações dos dados até à *Prepared Layer*, foi definido um *workflow* de Glue Jobs para cada fonte, permitindo a sua independência e execução em paralelo. Também, neste sentido, cada Glue Job, foi associado ao processamento de uma tabela, sendo que se ocorrer um erro no processamento de uma tabela, o processamento das restantes não é afetado.

Na *Source Layer*, os dados são guardados em bruto sem nenhuma transformação aplicada, as transformações são aplicadas a partir da *Prepared Layer*. Nesta *layer* a estrutura das tabelas são mantidas e são apenas feitas transformações aos tipos e aos nomes das colunas, para que depois na *Semantic Layer* seja feita a sua associação e várias bases de dados possam ser criadas tendo em conta o seu uso final. Para o processamento feito na *Semantic Layer*, um *workflow* é definido para cada uso dos dados. Neste caso, foram elaborados dois *workflows*, um para a elaboração de um *Reporting Dashboard*, implementado usando o software Tableau, com a construção de um modelo de dados incluindo as tabelas e o cálculo de KPIs úteis para a análise da performance das vendas de carros usados, e outro para a componente de *Analytics* com as tabelas necessárias para a execução da análise preditiva e a aplicação do modelo de otimização usando o serviço AWS Glue.

De forma complementar, em cada *layer* os dados foram guardados usando o serviço de armazenamento de objetos da AWS – o AWS S3, *Amazon Simple Storage Service* (*Amazon Simple Storage Service Documentation*, 2023). Este serviço oferece vários tipos de armazenamento, conforme a frequência do uso dos dados e faz o gerenciamento de acesso e segurança aos S3 Buckets e aos seus objetos. Os S3 Buckets, podem ser associados a pastas, e os objetos interpretados como as suas subpastas ou ficheiros. Neste caso, foi criado um S3 Bucket para cada *layer* e um objeto para cada tabela. Podendo assim, gerir o acesso dos dados ao nível das *layers* e/ou tabelas conforme a role atribuída a cada utilizador da plataforma, usando o serviço AWS IAM, *Identity and Access Management* (*AWS Identity and Access Management Documentation*, 2023). Este serviço permite gerenciar, de

maneira centralizada, o acesso a diversos serviços de AWS, tal como o AWS S3, podendo especificar a permissão de acesso a certos S3 Buckets e objetos.

Na última *layer*, é disponibilizado o serviço AWS Athena (*Amazon Athena Documentation*, 2023) para que os utilizadores finais possam explorar os dados, fazendo *queries* em SQL aos dados guardados em AWS S3. Este serviço oferece uma plataforma de análise de dados sem ser necessário um servidor, sendo este serviço caracterizado por *serverless*, pago consoante as consultas feitas.

3.3.1. Estrutura dos dados

Focando-nos no objetivo da segunda fase do projeto, para a análise aqui apresentada são usadas duas tabelas do mercado da Alemanha, uma com os dados estáticos do veículo e a outra com os dados do processo de revenda.

Após a uniformização e transformação dos tipos das colunas, a estrutura dos dados do veículo na *Prepared Layer* é descrita na tabela abaixo. A coluna "Exemplo", estará vazia para algumas variáveis dado os termos de confidencialidade da empresa.

Tabela 3.1 – Esquema da tabela do veículo

Variável	Descrição	Tipo	Exemplo
vin_17	Vehicle Identification Number (17 dígitos)	Texto	-
van	Código Hash identificar do vin_17	Texto	-
brand	Marca	Texto	-
chassis_code	Identificação do chassi do carro	Texto	-
model	Modelo	Texto	-
model_code	Código do modelo	Texto	-
date_production	Data de produção	Data	"2023-11-30"
body_type	Carroçaria	Texto	"Coupe", "Convertible"
transmission_type	Tipo de transmissão	Texto	"AUTOMATIC"
drive_type	Tipo de tração	Texto	"AWD", "RWD"
fuel_type	Tipo de combustível	Texto	"diesel"
power	Cavalos, hp	Número	220
color_exterior_code	Código da cor exterior do carro (3 dígitos)	Texto	-
color_exterior_description	Descrição da cor exterior do carro	Texto	-
upholstery_code	Código do estofamento	Texto	-
upholstery_description	Descrição do estofamento	Texto	-
upholstery_color_code	Cor do estofamento	Texto	-
option_codes_description	Códigos adicionados opcionalmente	Lista	-
doors	Número de portas	Número	4
co2_emission	Emissões de co2, g/km	Número	168
consumption_fuel	Consumo do carro, L/km	Número	7.4
electric_capacity	Bateria do carro, kWh	Número	40
facelift	Modificação do estilo do carro	Texto	-
line	Estilo de construção	Texto	-

Os dados do processo de revenda incluem vários campos dinâmicos característicos do veículo, tal como a quilometragem, e também os campos característicos do processo em si. A estrutura é descrita na tabela abaixo.

Tabela 3.2 – Esquema da tabela do processo de revenda

Variável	Descrição	Tipo	Exemplo
vin_17	Vehicle Identification Number (17 dígitos)	Texto	-
remarketing_process_step	Estado do processo de revenda	Texto	"SOLD"
arrival_date_oru	Data de chegada ao armazém	Data	"2023-05-30"
mileage_km	Quilometragem	Número	8000
damage	Dano em euros, líquido	Número	4000
first_offer_date	Data da primeira oferta	Data	"2023-11-30"
first_offer_price_net	Primeira oferta monetária, líquido	Número	30000
last_offer_price_net	Última oferta monetária, líquido	Número	30000
invoice_date	Data de revenda	Data	"2023-11-30"
vehicle_price_net	Preço da revenda, líquido	Número	30000
sales_channel	Canal de venda	Texto	"Direct"
manual_discount	Desconto aplicado	Número	300
date_first_registration	Data do primeiro registo do carro	Data	"2022-11-30"
previous_use	Tipo de uso anterior, tais como "Lease", "Rent a car"	Número	6
msrp_excl_options_net	Preço inicial do carro, excluindo opções, líquido	Número	34000
msrp_incl_options_net	Preço inicial do carro, incluindo opções, líquido	Número	41000

3.3.2. Preparação dos dados

Na *Semantic Layer* é feita, em primeiro lugar, a união das tabelas pelo seu campo comum, o *vin_17*. E em seguida, são executados dois passos principais, a filtragem de dados, remoção de observações classificadas como incorretas pelo cliente e a criação de variáveis-base para a análise exploratória dos dados, e claro, com vista a serem úteis para a análise preditiva.

1. Filtragem de dados
    a. Carros com o valor da coluna *van* em falta.
    b. Carros com o valor da coluna *model* em falta.
    c. Revendas que não respeitam a ordem temporal do processo de revenda. A ordem correta deverá ser *date_production – date_first_registration – arrival_date_oru – first_offer_date – invoice_date,* tal como ilustrado na figura 1.1.
    d. Carros com preços iguais ou abaixo de 0. Aplicando este filtro às colunas *vehicle_price_net*, *msrp_excl_options_net* e *msrp_incl_options_net*.
    e. Carros com marcas diferentes das requisitadas para a análise preditiva.

2. Criação de variáveis
    a. *days_of_use*, número de dias que o carro foi utilizado – subtração das variáveis *arrival_date_oru* e *date_first_registration.*
    b. *days_on_offer*, número de dias que o carro demorou até ser vendido desde a primeira oferta – subtração das variáveis *invoice_date* e *first_offer_date*.
    c. *days_to_sell*, número de dias que o carro demorou até ser vendido desde a chegada ao armazém – subtração das variáveis *invoice_date* e *arrival_date_oru*.
    d. *color_main_kr*, mapeamento da variável *color_exterior_code* para um grupo com menor granularidade. Para este mapeamento é usada a tabela dimensão que caracteriza as possíveis cores de um veículo.
    e. *reduction_factor*, percentagem de perda monetária referente ao preço inicial do veículo – subtração entre *msrp_incl_options_net* e *vehicle_price_net*, para a obtenção da perda (ou ganho) e divisão pelo *msrp_incl_options_net* para a obtenção da percentagem.
    f. *vehicle_price_net_without_discount*, valor da revenda sem incluir o desconto aplicado – subtração das variáveis *vehicle_price_net* e *manual_discount.*
    g. options_count, número de códigos opcionais do veículo – tamanho da lista *option_codes_description*.
    h. *engine_main*, categorização principal do motor – transformação baseada na variável *model*.
    i. *YUC,* variável binária que identifica se um veículo é definido como "Youth Used Car", caracterizado por ter uma idade abaixo de 1 ano e uma quilometragem menor que 35000 km.
    j. Variáveis temporais derivadas da coluna *first_offer_date*, para análise temporal dos dados, permitindo identificar possíveis tendências. Assume-se que esta data é a mais próxima da necessidade da previsão. São criadas cinco variáveis diferentes: *first_offer_date_year, first_offer_date_month, first_offer_date_day, first_offer_date_week* e *first_offer_date_dayofweek.*

3.4. ANÁLISE EXPLORATÓRIA

A seguinte análise exploratória tem como objetivo auxiliar na compreensão inicial da estrutura dos dados, ajudar a obter mais conhecimento sobre o comportamento das variáveis e, por consequente, proporcionar uma sólida fundamentação para as etapas seguintes. Nesse sentido, serão apenas descritas as análises com os insights mais úteis para o resto da análise.

3.4.1. Análise temporal

A amostra para esta análise foi extraída a 4 de maio de 2022, contando com 412 930 carros em processo de revenda, desde a chegada ao armazém depois do retorno, de 4 de maio de 2009 até 2 de maio de 2022.

Na figura 3.5, é possível tirar uma perceção da distribuição da venda de carros ao longo deste intervalo.

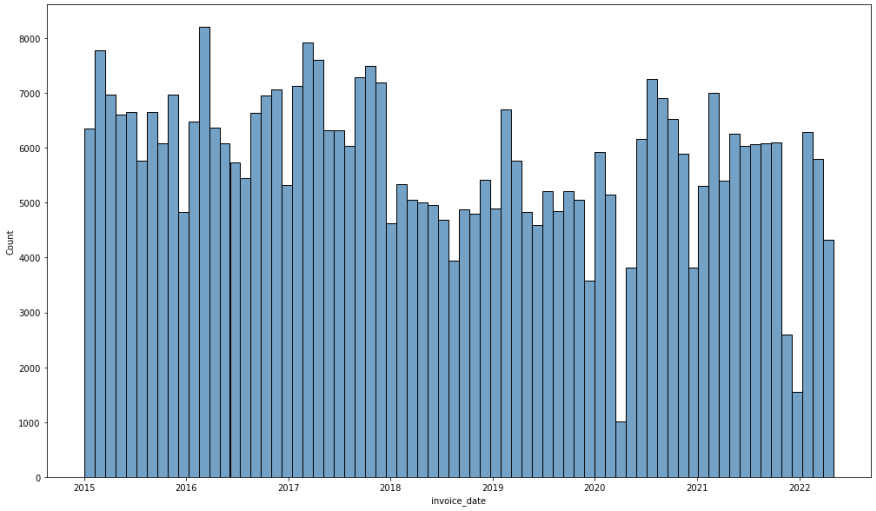


Figure 3.5 – Distribuição da venda de carros pelo tempo

Numa primeira análise, verifica-se uma descida das vendas em 2018, possivelmente relacionada com a queda na procura de automóveis com motor de combustão, identificada em (De Santis et al., 2022) e podendo ser esta extrapolada para o mercado de carros usados. Os autores descrevem esta queda como proveniente dos testes de emissões mais rigorosos implementados na União Europeia, UE, em 2018 e o acordo feito sobre as metas de emissões de CO_2 alcançado em dezembro de 2018, que geraram um incentivo a favor dos automóveis híbridos e elétricos em detrimento dos automóveis com motores de combustão.

É também de destacar, os acentuados decréscimos no início de 2020 e no final de 2021. O primeiro devido ao começo da pandemia, descendo não só o número de revendas como também a produção de novos carros, e o segundo decréscimo, também identificado pelos mesmo autores, assume-se que pode estar relacionado com a escassez de várias componentes para a produção de novos carros em meados de 2021 e o aumento do custo da energia em agosto de 2021, refletindo-se no mercado de carros usados no final do ano.

Em 2022, é de notar a tendência decrescente das vendas, podendo esta estar relacionada com a incerteza associada à guerra da Ucrânia e a consequente diminuição da procura de bens mais valiosos, como um carro.

É ainda possível aferir que a diferença entre a data mínima de chegada ao armazém e a data mínima de revenda, 1 de janeiro de 2015, é de quase 6 anos, um comportamento que não é normal para a nossa amostra. Sendo a diferença entre estas datas, representada pela variável *days_to_sell*, até 67 dias em 75% da amostra.

Ao analisar a figura 3.6, nota-se que a quantidade de carros, pós chegada ao armazém, começa apenas a ser mais evidente no ano de 2015, indo ao encontro da conclusão retirada anteriormente.

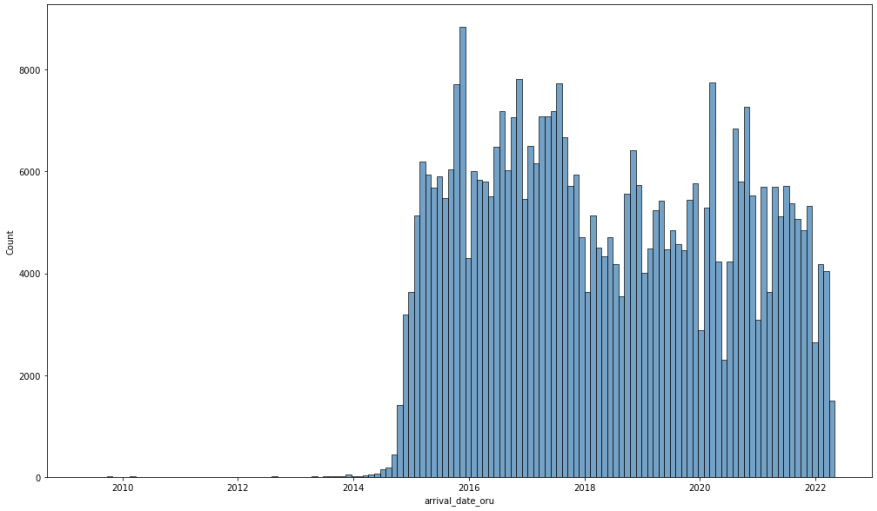


Figura 3.6 – Distribuição da chegada de carros usados ao armazém pelo tempo

3.4.2. Qualidade dos dados

Nesta secção irão ser analisados três aspetos diferentes dos dados. A avaliação de colunas com valores em falta, a análise de variáveis numéricas, mais detalhadamente à identificação de distribuições assimétricas, e por último uma análise às variáveis categóricas, nomeadamente às suas cardinalidades e a algumas distribuições interessantes para a etapa do tratamento dos dados.

3.4.2.1. Valores em falta

Serão apenas apontados os valores em falta das colunas da fonte, dado que as colunas derivadas, detalhadas na secção 3.3.2, irão ter os mesmos valores em falta.

Tabela 3.3 – Valores em falta

Variável	Valores em falta	Percentagem de valores em falta (%)
body_type	37	0.09
transmission_type	282 404	68.4
upholstery_code	100 80	2.44
co2_emission	297 494	72.04
consumption_fuel	297 494	72.04
electric_capacity	412 930	100
line	79 333	19.21
arrival_date_oru	3 834	0.93
mileage_km	3 834	0.93
first_offer_date	4 680	1.13
manual_discount	14 755	3.57
date_first_registration	714	0.17

Com o intuito de preencher os valores em falta na etapa do tratamento de dados, foram estudadas várias relações entre as variáveis.

1. Foi possível concluir que, para cada *model_code* só existe um *body_type* correspondente, no entanto os valores em falta são referentes apenas a um único *model_code*, que não tem nenhum veículo com o *body_type* preenchido. Por outro lado, por cada modelo pode haver até 5 valores diferentes da variável *body_type* com distribuições muito variáveis.
2. Para a variável *transmission_type*, todos os valores de *model_code* podem ter os 3 possíveis valores de transmissão (automático, manual ou outro).
3. Para a variável *upholstery_code*, um mesmo valor de *model_code* pode ter até 10 diferentes tipos de estofamento.
4. Para as variáveis *co2_emission* e *consumption_fuel*, com valores em falta em comum, pensou-se que fossem relacionadas com o veículo ser elétrico e daí poder aferir um valor 0, no entanto apenas 474 carros dos que têm estes valores em falta são elétricos, os restantes 297 020 oscilam em todas as outras categorias da variável *fuel_type.* Para além disto, existem 10 296 e 11 539 carros, respetivamente, com as variáveis *co2_emission* e *consumption_fuel* com um valor 0, o que era de esperar serem um veículo elétrico, no entanto existem apenas 2 860 carros com o *fuel_type* igual a "electrical". Tornando a credibilidade das duas variáveis muito baixa.
5. Para a variável *line*, um mesmo valor de *model_code* pode ter até 8 valores diferentes de estilo de construção.
6. Existem 152 923 com um valor de *manual_discount* igual a 0.

3.4.2.2. Assimetria dos dados e Outliers

Na análise das variáveis numéricas, calculando a métrica da assimetria (*skewness*), uma medida estatística que descreve a inclinação de uma distribuição de dados em relação à média, e fazendo gráficos de distribuição e caixa de bigodes, identificou-se que a maioria das variáveis têm uma assimetria positiva bastante acentuada. Exceto as variáveis *reduction_factor, options_count* e as variáveis temporais derivadas da *first_offer_date*.

De seguida analisa-se mais detalhadamente as variáveis que podem suscitar mais curiosidade ao leitor.

Inicialmente, analisa-se a distribuição da variável *reduction_factor*, uma variável bastante importante para o revendedor dado que representa a sua percentagem de perda em relação ao preço inicial do carro como novo.

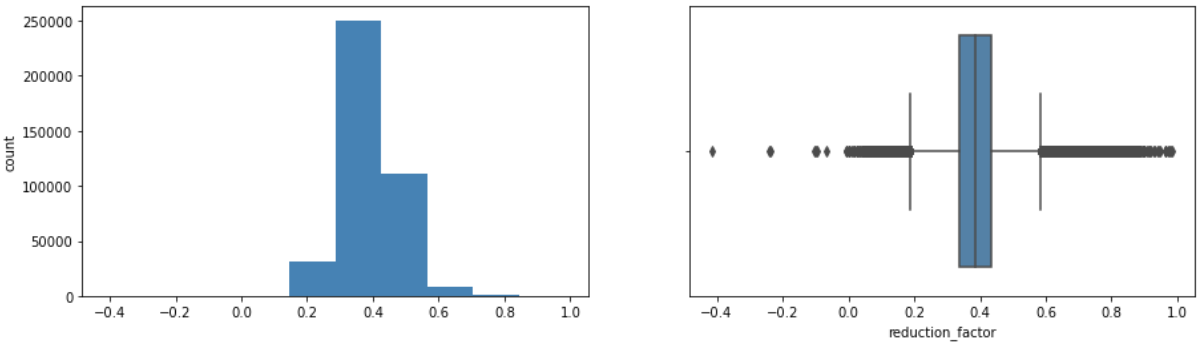


Figura 3.7 – Distribuição da percentagem de perda

É curioso notar que, na figura 3.7, embora não seja comum, existem 9 registos de carros com uma percentagem negativa, querendo isto dizer que o carro depois de usado foi vendido a um valor mais

elevado do que o preço inicial sugerido. É também de notar que, 50% da amostra varia apenas entre 3.3% e 4.3% de percentagem de perda.

A variável *manual_discount* tem uma *skewness* de 3.64, em que 85.2% dos carros tem um valor igual a 0 e o máximo de desconto aplicado foi de 5 114€.

As variáveis, que caracterizam o preço, nomeadamente *vehicle_price_net, msrp_excl_options_net e msrp_incl_options_net* apresentam uma s*kewness* de 1.41, 1.91 e 1.47, respetivamente. A distribuição da variável *vehicle_price_net* é apresentada com mais detalhe na figura 3.8. Esta variável, valor a prever, tem um valor mínimo de 0.01€, no entanto, tem um intervalo interquartil entre 20 840 € e 35 500 € com uma mediana de 27 043 € e uma média de 29 506 €.

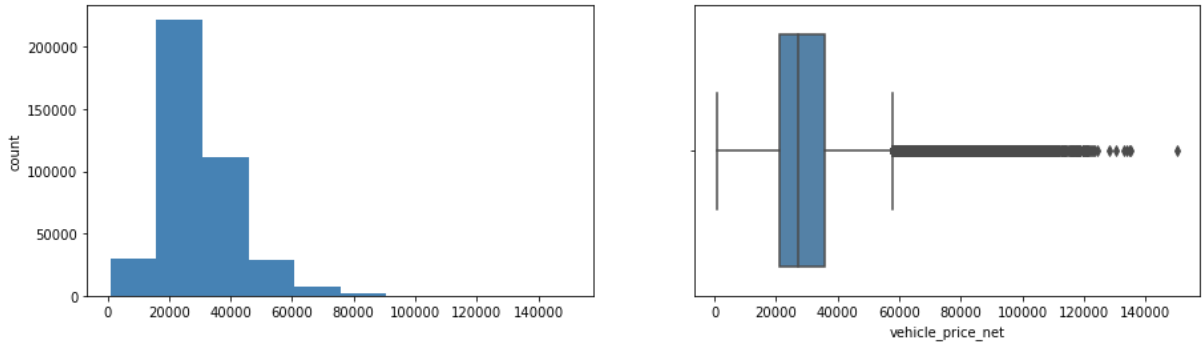


Figura 3.8 – Distribuição do preço de revenda

A variável que define a idade do carro em dias, tem também uma assimetria bastante positiva como se pode interpretar na figura 3.9. É ainda de notar que 75% da amostra é composta por carros com menos de 369 dias, aproximadamente 1 ano. Embora não seja bem visível no gráfico da caixa de bigodes da figura 3.9, vale ressaltar que o mínimo da idade do carro, presente na amostra, são 7 dias.

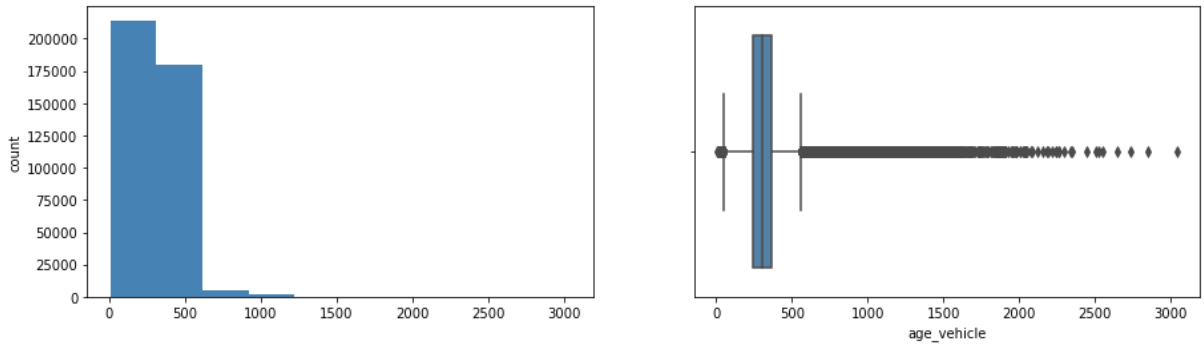


Figura 3.9 – Distribuição da idade do carro

De forma semelhante, na figura 3.10, é possível notar que a variável *days_on_offer* tem também uma assimetria positiva, ainda mais acentuada. Neste caso, 50% da amostra é composta por carros que são vendidos em menos de 19 dias depois da primeira oferta e 75% em menos de 43 dias. É ainda de destacar que o registo máximo desta variável é de 1405 dias, aproximadamente de 3 anos e 10 meses.

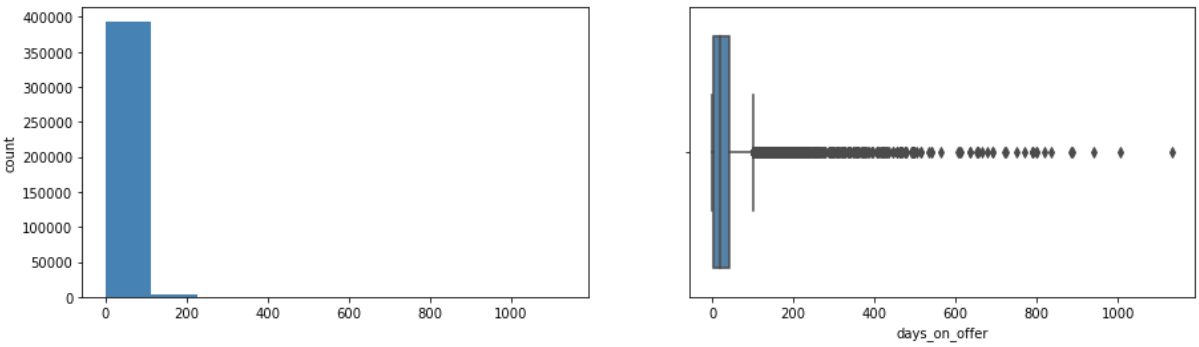


Figura 3.10 – Distribuição dos dias do carro em oferta

3.4.2.3. Distribuição das variáveis categóricas

Na análise das variáveis categóricas, é importante identificar a sua cardinalidade e distribuição para a determinação do seu e*ncoding*, transformação para variáveis numéricas, de forma perspicaz. A cardinalidade de cada variável é representada na tabela 3.4.

Tabela 3.4 – Cardinalidade das variáveis categóricas

Variável	Cardinalidade
brand	3
chassis_code	91
model	291
model_code	1111
body_type	9
previous_use	4
transmission_type	3
drive_type	3
fuel_type	5
color_exterior_code	194
color_main_kr	10
upholstery_code	98
upholstery_color_code	200
lci	3
line	52
engine_main	15
doors	4
YUC	2

Uma das variáveis com cardinalidade elevada é o código da cor exterior do carro, com 194 cores diferentes, no entanto com o mapeamento da variável *color_main_kr*, que agrupa esta variável em 10 categorias diferentes é possível manter alguma da sua caracterização com uma cardinalidade mais baixa. Na figura 3.11 é possível ver a distribuição das diferentes cores pelas categorias *main*. Ainda não tendo a sua descrição em texto, é fácil deduzir que as categorias com um maior número de cores são o preto e o cinzento.

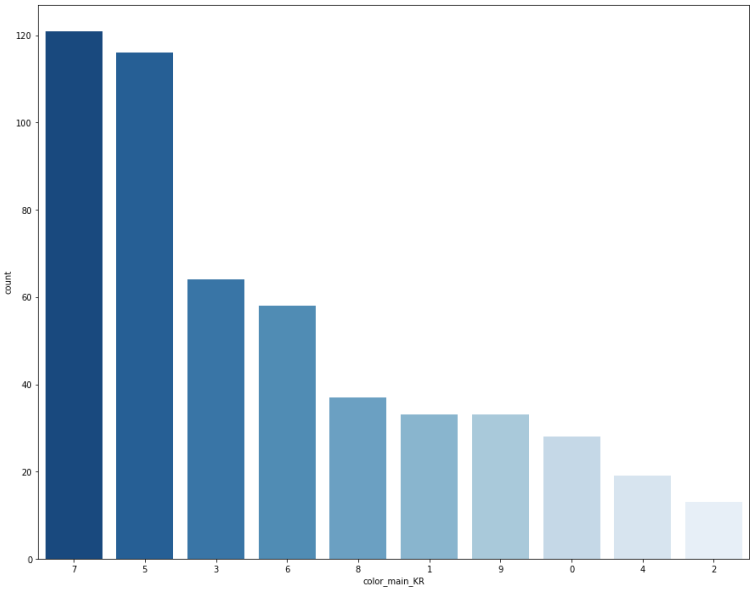


Figura 3.11 – Distribuição das cores pela categoria *main*

Para evidenciar as assimetrias positivas das variáveis numéricas *age_vehicle* e *mileage_km*, é relevante notar que a proporção de carros classificados como "Youth Used Cars" é bastante grande, cerca de 62.7%. A ilustração da sua distribuição é visível na figura 3.12.

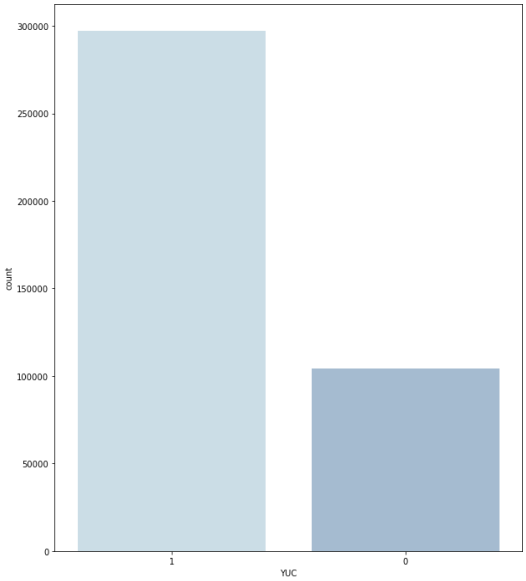


Figura 3.12 – Distribuição da variável *YUC*

Da mesma forma, a variável *brand* tem também uma distribuição bastante desproporcional pelas suas 3 categorias, como é ilustrado na figura 3.13. A marca "A" representa quase 85% da amostra.

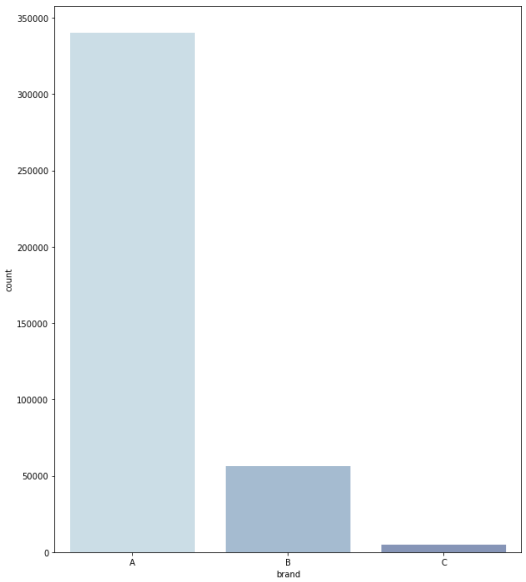


Figura 3.13 – Distribuição das marcas dos carros

Outra análise que se julgou importante mencionar foi a distribuição da variável de *reduction_factor* sobre cada categoria da variável *previous_use*. Ilustrada na figura 3.14.

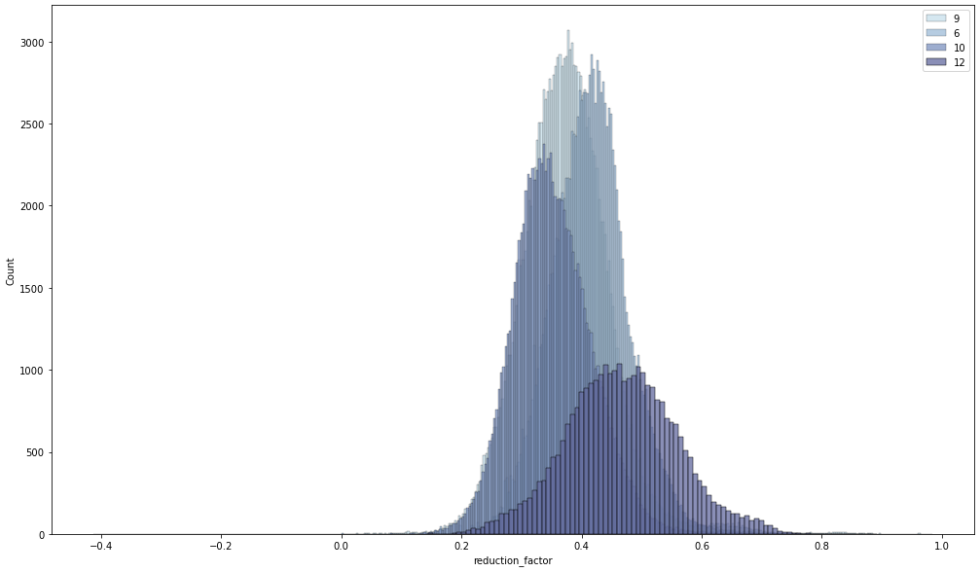


Figura 3.14 – Distribuição da percentagem de perda sobre a variável *previous_use*

Ao observar a figura 3.14, é de salientar as diferentes distribuições por cada categoria, e identificar as suas médias. Não sendo bem refletido no gráfico, por ordem ascendente, as médias das categorias 12, 6, 10 e 9 são respetivamente 0.35, 0.37, 0.41 e 0.47. Lamentavelmente, não foi possível obter a descrição dos códigos usados para este mercado, não podendo fazer a sua associação com os diferentes tipos de uso. Ainda assim, estima-se que o uso com menor percentagem de perda e também menor frequência será o uso para demostrações de veículos.

Com base nesta análise, foi possível perceber a distribuição temporal dos dados, identificar a presença de vários valores em falta, a acentuada assimetria dos dados e a consequente presença de *outliers*, e

a presença de variáveis categóricas com uma cardinalidade bastante elevada. Todos estes pontos irão contribuir para a escolha das transformações do capítulo a seguir.

3.5. TRATAMENTO DE DADOS

Nesta secção é apresentado o processo de transformação de dados com o objetivo de melhorar a sua qualidade e prepará-los para a sua utilização nos modelos descritos na próxima secção.

Sendo este um processo que se quer automatizado, os passos para o tratamento dos dados foram, não só estabelecidos com fundamentação na análise exploratória apresentada, como também no conhecimento de negócio. As regras estabelecidas foram derivadas de um processo iterativo, com a revisão do seu efeito na amostra estudada.

Esta etapa, é definida em 5 fases: a filtragem de dados, incluindo a remoção de o*utliers*; o preenchimento de dados em falta; a criação de novas variáveis, que tem como objetivo entregar ao modelo variáveis mais informativas e principalmente resolver o problema de cardinalidade das variáveis categóricas; a remoção das variáveis não utilizadas para a modelação; e, por último a transformação de variáveis, nomeadamente o *encoding* de variáveis categóricas e o escalonamento, mais conhecido como *scaling*, das variáveis numéricas.

1. Filtragem de dados
    a. Carros com a variável *remarketing_process_step* diferente de "SOLD", sendo incluídos na análise apenas carros que têm o processo de revenda concluído.
    b. Carros com a data do primeiro registo antes de 1 de janeiro de 2018, dado a instabilidade dos dados nos anos anteriores.
    c. Carros com um valor de *damage* superior a 20 000€.
    d. Carros com valor em falta na variável *vehicle_price_net*.
    e. Carros com preços fora dos limites definidos. Sendo que é esperado que os valores de *vehicle_price_net*, *msrp_excl_options_net* e *msrp_incl_options_net* não sejam menores que 5000, 8000 e 10000 euros respetivamente.
    f. Carros com preços iniciais (*msrp_incl_options_net* e *msrp_excl_options_net*) muito desviados da média, em particular com um Z-score maior que 2, estando este situado no percentil de 4.56% mais desviado da média.
2. Preenchimento de dados em falta
    a. Usar a mediana de dias entre *date_production* e *date_first_registration* por modelo para preencher os valores da variável ***date_first_registration***. Ao obter a mediana por modelo somar os dias à coluna *date_production* por cada carro, caso o modelo não tenha dados faz-se a mediana total e caso a variável tenha todos os dias em falta somam-se 10 dias.
    b. Usar a mediana de dias entre *arrival_date_oru* e *first_offer_date* por modelo para preencher os valores da variável ***first_offer_date***. Ao obter a mediana por modelo somar os dias à coluna *arrival_date_oru* por cada carro, caso o modelo não tenha dados faz-se a mediana total e caso a variável tenha todos os dias em falta soma-se 1 dia.

        Ao fazer esta transformação foi identificado que estariam a ser criados registos em que a *first_offer_date* seria maior que a *invoice_date*, por esta razão adicionou-se um

passo adicional para fazer esta verificação. Caso se verificasse este comportamento era subtraído 1 dia à variável *invoice_date*.

c. Usar o *model_code* para o preenchimento das variáveis ***model, doors, power, fuel_type, body_type*** *e* ***drive_type***, usando o seu valor mais comum, a moda.

d. Os restantes valores são preenchidos com "UNKNOWN" nas colunas categóricas e com 0 nas colunas numéricas.

3. Criação de novas variáveis

a. *options_price*, preço total das opções – subtração das variáveis *msrp_incl_options_net* e *msrp_excl_options_net*, na tentativa de esta ser mais relevante que a variável *options_count*, diferenciando casos em que um carro tenha a mesma quantidade de opções escolhidas que outro, mas estas serem mais valiosas.

b. *percent_damage*, percentagem de dano sobre o preço inicial do carro – divisão da variável *damage* pela variável *msrp_incl_options_net.*

c. *depreciation_rate*, rácio de depreciação do carro pelo número de dias em uso – divisão da variável *damage* pela variável *days_of_use*.

d. *main_model*, categorização da variável *model* em 10 categorias maiores – transformação da variável *model* com base na presença de certos dígitos.

e. *interior_mapping*, distinção do estofamento – *upholstery_description* – em duas categorias, "Leather" e "Cloth", baseado na inclusão de várias palavras-chave, como por exemplo, "LED" para "Leather" e "Stoff" para "Cloth".

f. *model_code_min_msrp_excl*, *model_min_msrp_excl* e *chassis_code_min_msrp_excl*, *encoding* das variáveis *model_code*, *model* e *chassis_code*, respetivamente, pelo mínimo do seu valor base inicial, excluindo as opções – mínimo da variável *msrp_excl_options_net* por categoria. Dado a assimetria positiva da variável foi decidido optar pelo seu valor mínimo para caracterizar cada categoria, no entanto, pode perder-se certas configurações de carros com valores mais elevados dentro da mesma categoria, presume-se que estes casos possam ser categorizados por outras variáveis, como por exemplo, o *power* e o *fuel_type*.

   Para a criação desta variável foi feita uma análise à oscilação do preço dos carros por cada grupo, o leitor pode consultar o apêndice A para ver a distribuição da diferença do valor máximo e mínimo da variável *msrp_excl_options_net* por cada grupo, representando o intervalo de preços.

g. *model_code_relative_msrp_excl_to_min*, *model_relative_msrp_excl_to_min* e *chassis_code_relative_msrp_excl_to_min,* relatividade do preço individual de cada carro sobre o mínimo de cada grupo *model_code*, *model* e *chassis_code*, respetivamente – divisão da variável *msrp_excl_options_net* pela variável do respetivo grupo mencionada na alínea f. Estas variáveis para além de representarem uma métrica de comparação relativa de cada carro de acordo com o grupo pertencente, são também uma normalização da variável *msrp_excl_options_net* dentro de cada grupo. À similaridade das variáveis da alínea f., o leitor pode consultar o apêndice B, para uma consulta mais detalhada da distribuição destas variáveis.

4. Remoção de variáveis

a. *vin_17* e *van*, variáveis com valor único por veículo e por sua vez não interessantes para a previsão do preço.

b. *options_codes_description*, não importante para a análise já sendo criadas variáveis com a sua origem.
c. *color_exterior_description* e *upholstery_description*, sendo que são usados os seus códigos como associação direta.
d. *co2_emission* e *consumption_fuel*, tendo a sua credibilidade sido questionado e descrita na secção 3.4.2.1.
e. *electric_capacity*, por ter todos os valores em falta.
f. *upholstery_code*, pela sua categorização que se julga mais relevante na variável *interior_mapping*.
g. *line*, dado a sua percentagem de valores em falta, descrita na tabela 3.3, ser bastante elevada. Não foi associada com a variável *model_code*, como outras características do veículo, descritas no ponto 2.c., pois tal como descrito na análise exploratória, cada *model_code* pode ter até 8 valores diferentes de *line*. Acredita-se, que esta variável possa ser explicada em conjunto das variáveis *model_code* e *options_price*.
h. *msrp_incl_options_net*, variável explicada pela soma entre as variáveis *msrp_incl_options_net* e *options_price.*
i. *reduction_factor* e *vehicle_price_net_without_discount,* sendo que foram apenas usadas para uma primeira análise exploratória e não deverão ser usadas na previsão do preço, tendo em conta que ambas incluem esta variável no seu cálculo, e por isso, definidas como *data leakage*.

5. Transformação de variáveis
   a. *Robust Scaling* das variáveis numéricas, usando a função *RobustScaler* do *sklearn*. É um escalonamento robusto a *outliers*, apropriado ao tipo de dados presentes nesta análise, dado a sua acentuada assimetria. Em vez de usar a média e o desvio padrão como o mais conhecido método de escalonamento – *Standard Scaling* –, usa a mediana e o IQR (intervalo interquartil). A sua fórmula é definida da seguinte maneira,
   $$X_{scaled} = \frac{X - X_{median}}{IQR}.$$
   b. *Label Encoding* das variáveis categóricas, usando a função *LabelEncoder* do *sklearn*. Este método substituí as categorias entre 0 e o número de categorias menos 1 de forma arbitrária. Para a variável *door* é feito um tratamento especial, dado que tem uma ordem a querer ser respeitada, e assim, é só feita a sua transformação de texto para número.
   c. *Target Encoding* das variáveis categóricas, usando a biblioteca *category_encoders*, especificamente a função *TargetEncoder*, este método de *encoding* é baseado na substituição de cada categoria pela média da variável target, no entanto tem um parâmetro de ponderação entre a média da variável target e a média total da amostra de treino (Micci-Barreca, 2001). Este parâmetro de ponderação é representado por uma função de formato em "s", dependente do número de observações de cada categoria i, n_i. Representada pela seguinte fórmula,
   $$s(n_i) = \frac{1}{1 + \exp\left(-\frac{n_i - k}{f}\right)}$$

O parâmetro f, controla a inclinação da função e o parâmetro k define a metade do valor mínimo que queremos considerar para n_i para podermos confiar na média da categoria i, um valor de k= n_i daria um valor de 0.5 a $s(n_i)$.
Assim, o valor escalar, t_i, a ser substituído na variável X, quando esta toma um valor igual a X_i, é traduzido na seguinte fórmula.

$$T_i = E[Y] * \big(1 - s(n_i)\big) + s(n_i) * E[Y|X = X_i]$$

Desta forma, o valor T_i irá inclinar-se para $E[Y|X = X_i]$ quando o número de observações de X_i é grande e para a média total da amostra de treino quando o número de observações é pequeno. Ainda assim, é um método que suscita dúvidas em relação ao *overfitting* do modelo, dado que o valor do target é utilizado no seu cálculo. Para não ter resultados enviesados é também importante garantir que a amostra de treino apresenta uma boa diversidade de comportamentos.

É de notar que os pontos da fase 1 são feitos apenas para a avaliação e aprendizagem do modelo, não deverão ser aplicados em dados com o preço desconhecido, em que o objetivo é fazer a sua previsão. E ainda que os pontos 2.a., 2.b., 2.c., 3.f., 3.g. e todos os pontos da fase 5, deverão ser aplicados aos dados apenas com o conhecimento da amostra de treino, isto é, para a amostra de teste, e no futuro em amostras desconhecidas, todas as métricas usadas para o tratamento das variáveis, tais como a mediana, deverão ser calculadas com base na amostra conhecida, de treino do modelo. Este ponto, previne, numa primeira fase, o vazamento de informações (*data leakage*) da amostra de teste para a amostra de treino e pode evitar o enviesamento (*bias*) das métricas necessárias, ao ser calculadas apenas entre a amostra de teste, dado que esta nem sempre será representativa da amostra geral, podendo conter carros com características únicas.

Como vamos ver no capítulo seguinte, os pontos descritos na fase 5, e o ponto 3.f., opções de *encoding* e *scaling* das variáveis, serão alvos de uma pesquisa em grelha, considerando várias combinações destas transformações, para se poder avaliar a sua performance nos modelos a aplicar. É ainda de ressaltar que os pontos 3.f., 5.b. e 5.c. são mutuamente exclusivos para a aplicação da mesma variável, no sentido em que se se optar pela aplicação de um ponto o outro não é aplicado.

Estas opções foram escolhidas com fundamento na análise exploratória, o *robust scaling* devido à presença de *outliers*, e os diferentes tipos de *encoding* como alternativa ao *one hot encoding*, que, embora não obrigue os valores das variáveis a ter uma ordem, como o *label encoding*, evitando o enviesamento do modelo, com a grande cardinalidade de algumas variáveis categóricas, iria aumentar bastante a dimensionalidade dos dados, podendo assim criar uma matriz esparsa e diminuindo a velocidade do treino dos modelos. Ainda assim, poderia optar-se por diminuir a cardinalidade das variáveis categóricas, limitando as categorias ao top n mais frequentes e agregando as restantes numa só categoria. Esta opção foi descartada numa primeira iteração pelo feedback do cliente de não querer juntar algumas categorias e assim não perder informação que possa vir a ser útil para a previsão. No entanto, após a primeira avaliação de resultados estas transformações devem ser revistas e experimentadas outras alternativas.

3.6. MODELAÇÃO

Esta secção visa apresentar uma solução de modelação para a previsão de carros usados robusta e adaptável a diferentes variáveis, mercados e canais de venda.

O processo desta solução pode ser definido em três tópicos diferentes, a seleção das variáveis mais importantes para a previsão do preço, o treino dos modelos que inclui a afinação de hiperparâmetros, e por último, as métricas usadas para a sua avaliação.

3.6.1. Seleção de variáveis

Tendo um conjunto de variáveis para cada mercado e cada canal de venda, foi aplicado o método RFE, para a seleção de variáveis mais relevantes para a previsão. Este algoritmo remove as variáveis menos importantes do conjunto de dados, recursivamente, até atingir o número desejado de variáveis ou otimizar o desempenho do modelo. Neste caso, sendo o número ideal de variáveis desconhecido, optou-se por escolher a melhor combinação de variáveis dado o coeficiente de determinação, R^2, do modelo. Este método foi aplicado com a função *RFECV* do *sklearn*, permitindo fazer uma validação cruzada do modelo para avaliar a sua performance em dados não observados, e geralmente evitar previsões enviesadas. Sendo que os dados têm cadência temporal foi usada a função *TimeSeriesSplit*, também do *sklearn*, para estabelecer diferentes subconjuntos da amostra de treino, assegurando que as observações do treino ocorrem sempre antes das observações do teste.

Adicionalmente, o modelo utilizado para a aplicação do RFE foi o XGBoost, tendo sido este o melhor modelo avaliado na primeira iteração desta análise. Com o objetivo de prevenir *overfitting*, o número de estimadores, *base learners*, foi definido como 100, a profundidade das árvores como 3 e o parâmetro *alpha* como 2, que representa o termo de regularização L1 dos pesos de cada variável, com o intuito de inclinar os pesos menores para 0, e assim ser possível ter uma seleção de variáveis mais crítica.

Tendo em consideração os diferentes tipos de transformações descritos na secção do tratamento de dados, como input do RFE são dadas 8 combinações diferentes de dados por mercado e canal de venda. O conjunto de variáveis selecionado é o conjunto que proporciona um melhor desempenho do modelo. Um esquema deste processo é visível na figura 3.15, que descreve os diferentes tipos de transformações usadas e os parâmetros do método RFE.

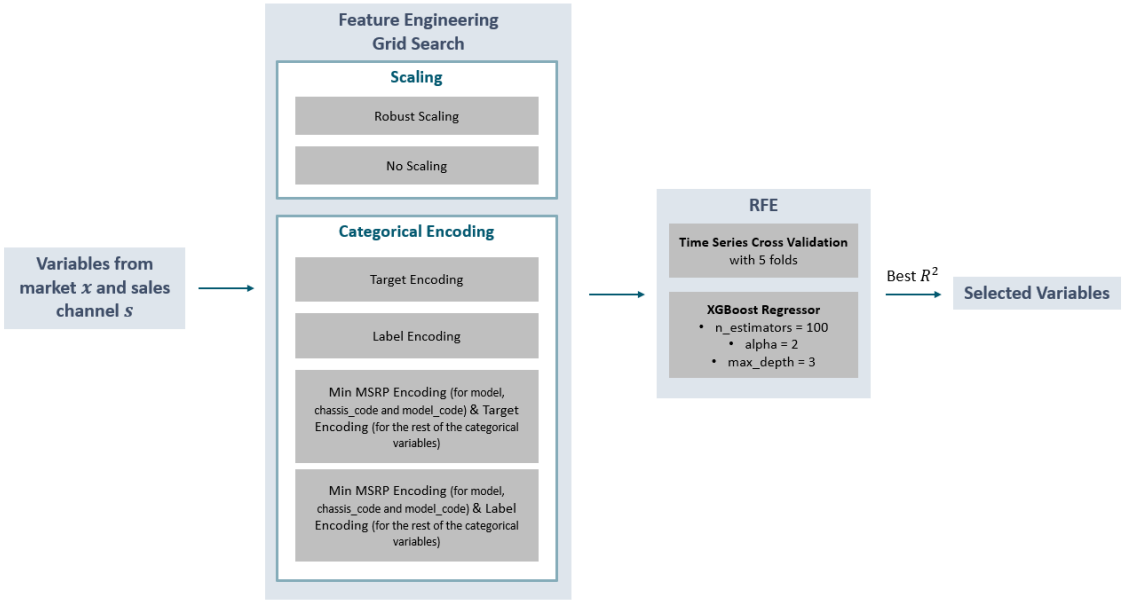


Figura 3.15 – Processo de seleção de variáveis

3.6.2. Treino dos modelos

A fase de treino dos modelos, igualmente à anterior, inclui uma validação cruzada de séries temporais dividida em 5 subconjuntos, no entanto, inclui também a afinação dos hiperparâmetros de cada modelo sobre uma pesquisa em grelha. O modelo é treinado usando todas as combinações dos valores de hiperpârametros por cada subconjunto de treino e avaliado em cada subconjunto de validação.

Na tabela 3.5, pode-se consultar os modelos aplicados, os hiperparâmetros testados, e a sua lista de opções. Estes hiperparâmetros foram definidos, sobretudo, com vista a prevenir o *overfitting* do modelo. O leitor pode consultar o seu significado e associação com o algoritmo acedendo à documentação da biblioteca *sklearn.ensemble* (Pedregosa et al., 2011) e à documentação do xgboost em (*XGBoost Parameters*, 2022).

Tabela 3.5 – Opções de hiperparâmetros por modelo

Modelo	Hiperparâmetros
Random Forest	Max depth: {3, 4, 5} N estimators: {50, 100, 200} Min samples split: {100, 200} Max features: {sqrt, log2} Bootstrap: {True}
AdaBoost (base estimator: Decision Tree)	Learning rate: {0.01, 0.1, 0.2} Base estimator max depth: {3, 4, 5} N estimators: {50, 100, 200, 500}
Gradient Boosting	Learning rate: {0.01, 0.1, 0.2} Max depth: {3, 4, 5} Min samples split: {100, 200} N estimators: {100, 200, 500}
XGBoost	Learning rate: {0.01, 0.1, 0.2} Max depth: {3, 4, 5}

	N estimators: {10, 100, 200, 500}
	Lambda: {0.1, 0.5, 2}

Em resumo, foram estudados valores baixos para a profundidade das árvores, sendo estas limitadas por 3 a 5 divisões; a opção com maior número de estimadores, 500, foi apenas testada nos modelos que seguem o conceito de *boosting*, por serem mais robustos a *overfitting*; de modo complementar à limitação da profundidade das árvores, foi testado limitar o número de observações presentes num "nó" após uma divisão, sendo este definido como 100 ou 200 observações para os modelos Random Forest e Gradient Boosting; para o Random Forest foram ainda definidos um limite máximo para o número de variáveis usadas para cada divisão, sendo testadas as duas opções dinâmicas que a função oferece, a raiz quadrada e o logaritmo binário do número de variáveis, e a obrigação da aplicação do *bootstrap*, pois caso contrário não é feito o *resampling* e toda a amostra é usada para a construção de todas as árvores; em complementaridade com o número de estimadores, para os modelos com o conceito de *boosting,* são testados três valores diferentes para o parâmetro da taxa de aprendizagem que controla a redução da contribuição de cada árvore/estimador, sendo que valores menores requerem mais iterações, mas podem melhorar a generalização; e finalmente, para o modelo XGBoost são testados vários valores para o parâmetro *lambda*, que representa o termo de penalização *Ridge* (McDonald, 2009), maiores valores irão tornar o modelo mais conservativo e assim prevenir o seu *overfitting*.

Obtendo o conjunto de hiperpâramentos que proporciona o melhor desempenho para cada modelo, os modelos são replicados com o melhor conjunto de hipêrparametros usando a totalidade da amostra de treino. Definidos os modelos, todos são usados para prever o preço das observações presentes na amostra de teste.

3.6.3. Avaliação dos modelos

Para avaliar e escolher o melhor modelo, a principal métrica usada foi o coeficiente de determinação, R^2, sendo esta representativa da proporção da variabilidade do preço que é explicada pelas variáveis independentes, que varia entre 0 e 1, em que 1 significa um ajustamento perfeito (Ozer, 1985).

Ainda assim, o erro médio absoluto, mais conhecido como, MAE, do inglês *Mean Absolute Error*, foi também importante para a análise de resultados, sendo esta métrica bastante fácil de interpretar e menos sensível a *outliers* do que outras métricas, como o erro quadrático médio, RMSE. Apesar de ambos serem uma medida de dispersão, o RMSE ao elevar o erro ao quadrado irá penalizar erros maiores. Assim, dado o considerável número de *outliers* dos dados, o uso do MAE irá dar uma melhor perceção do ajuste do modelo aos carros com valores mais comuns e comportamentos mais normais (Willmott & Matsuura, 2005).

Adicionalmente, o erro percentual absoluto médio, MAPE, do inglês *Mean Absolute Percentage Error*, foi usado para se poder ter uma maior sensibilidade do valor do erro proporcionalmente ao preço do carro.

Para a análise de resultados deste trabalho em específico, a amostra de dados foi dividida na amostra de treino, contendo dados de 1 ano, com chegada ao armazém entre 03-05-2020 e 02-05-2021, e na amostra de teste com dados de aproximadamente 2 meses, com chegada ao armazém entre 03-05-2021 e 30-06-2021.

É de notar que os dados da amostra inicial teriam carros com chegada ao armazém desde 04-05-2009 até 02-05-2022, no entanto, fazendo uma primeira divisão dos dados com a amostra de treino até 02-05-2021 e de teste até 02-05-2022 foram estudados comportamentos completamente diferentes, sendo estas amostras não representativas de um todo e bastante prejudiciais para a aprendizagem do modelo. Por isso, decidiu-se limitar a amostra de treino com dados apenas desde 03-05-2020 e a previsão do preço apenas com 2 meses de avanço, dado que é esperado que o modelo possa ser readaptado, com um novo processamento de treino mensalmente.

4. DISCUSSÃO E RESULTADOS

A divisão da amostra de dados, descrita na secção anterior, após ser aplicado o tratamento dos dados, resultou numa divisão de 46 180 registos na amostra de treino e 9 036 registos na amostra de teste, sendo o teste aproximadamente 20% da amostra total estudada.

O primeiro resultado a analisar é então o output do módulo da seleção de variáveis, que segue o processo da figura 3.15. Das 34 variáveis iniciais para cada combinação de *feature engineering*, a combinação escolhida foi usando o método de *target encoding* para as variáveis categóricas sem aplicar o método de *robust scaling*, foram selecionadas 30 variáveis. Sendo estas enumeradas na tabela 4.1.

Tabela 4.1 – Variáveis de input vs. selecionadas

Variáveis de input	Número	Variáveis selecionadas	Número
['brand', 'e_code', 'model', 'model_code', 'body_type', 'transmission_type', 'drive_type', 'fuel_type', 'power', 'color_exterior_code', 'upholstery_color_code', 'color_main_KR', 'interior_mapping', 'doors', 'lci', 'options_price', 'YUC', 'engine_main', 'main_model', 'mileage_km', 'msrp_excl_options_net', 'manual_discount','days_of_use', 'previous_use', 'days_on_offer', 'percent_damage', 'first_offer_date_year', 'first_offer_date_month', 'first_offer_date_day', 'first_offer_date_week', 'first_offer_date_dayofweek', 'e_code_relative_msrp_excl_to_min', 'model_relative_msrp_excl_to_min', 'model_code_relative_msrp_excl_to_min']	34	['brand', 'e_code', 'model', 'model_code', 'body_type', 'transmission_type', 'drive_type', 'fuel_type', 'power', ~~'color_exterior_code'~~, 'upholstery_color_code', ~~'color_main_KR'~~, ~~'interior_mapping',~~ 'doors', 'lci', 'options_price', 'YUC', 'engine_main', 'main_model', 'mileage_km', 'msrp_excl_options_net', 'manual_discount','days_of_use', 'previous_use', 'days_on_offer', 'percent_damage', 'first_offer_date_year', 'first_offer_date_month', 'first_offer_date_day', 'first_offer_date_week', ~~'first_offer_date_dayofweek'~~, 'e_code_relative_msrp_excl_to_min', 'model_relative_msrp_excl_to_min', 'model_code_relative_msrp_excl_to_min']	30

Interpretando os seus resultados, conseguimos perceber que as duas variáveis que representam a cor, o tipo de estofamento e o dia da semana foram eliminados, e por isso não considerados importantes para a variação do preço. O modelo de XGBoost usado para esta seleção teve um R^2 de 0.983, e como este seria um valor demasiado elevado, que apontaria para um efeito de *overfitting* foram feitos vários testes usando menos variáveis. Ainda assim, considerou-se o passo de *feature engineering* como ótimo, assumindo que o *encoding* com melhor performance é o *target encoding*, e que o escalonamento das variáveis não é importante para o desempenho do modelo. Considerando algum conhecimento de negócio, e os outputs obtidos, as variáveis selecionadas são descritas na tabela 4.2.

Tabela 4.2 – Variáveis de input final

Variáveis de input	Número
['e_code', 'model', 'model_code', 'main_model' 'transmission_type', 'power', 'interior_mapping', 'lci', 'YUC', 'options_price', 'mileage_km', 'msrp_excl_options_net', 'manual_discount','days_of_use', 'previous_use', 'days_on_offer', 'percent_damage', 'first_offer_date_day', 'first_offer_date_month']	19

Com este conjunto, o modelo obteve um R^2 de 0.982, assumindo assim, que esta seria uma seleção de variáveis mais apropriada para a previsão, dado o seu menor risco de *overfitting* e o seu elevado desempenho, pouco mais baixo que o anterior, que incluía 30 variáveis, mais 11 que este.

Prosseguiu-se então para a fase de modelação dos dados, onde aplicando os modelos e hiperparâmetros definidos na tabela 3.5, foram obtidos os seguintes resultados.

Tabela 4.3 - Resultados da modelação

Modelo	Hiperparâmetros	R^2	MAE (€)	MAPE (%)
Random Forest	Max depth: 5 N estimators: 50 Min samples split: 100 Max features: sqrt Bootstrap: True	Treino: 0.947 Teste: 0.882	2866	7.6
AdaBoost (base estimator: Decision Tree)	Learning rate: 0.1 Base estimator max depth: 5 N estimators: 50	Treino: 0.950 Teste: 0.890	2945	8.0
Gradient Boosting	Learning rate: 0.2 Max depth: 5 Min samples split: 100 N estimators: 200	Treino: 0.989 Teste: 0.949	1395	3.8
XGBoost	Learning rate: 0.1 Max depth: 3 N estimators: 500 Lambda: 2	Treino: 0.980 Teste: 0.965	1298	3.6

Pela leitura da tabela 4.3, é possível verificar que o modelo com o melhor desempenho na amostra de teste foi o XGBoost, e por isso o modelo final escolhido. No entanto, é possível também verificar que o modelo a ter o melhor desempenho na amostra de treino foi o Gradient Boosting, mesmo que com uma *learning rate* mais elevada e um número de estimadores, *base learners,* mais baixo, porém, cada um com uma profundidade mais elevada.

O modelo de XGBoost foi capaz de gerar um MAE de 1298€, representando uma percentagem média do erro das previsões de 3.6% sobre o preço.

Ainda que estes resultados pareçam satisfatórios, devem ser analisados com mais cuidado, identificando o tipo de carros que obtiveram uma previsão pior, sendo que o preço de carros usados pode assumir valores de 5000€, e assim, um erro de 1298€ não parecer tão satisfatório quando comparado com carros de maior valor. E o modelo deve ser aplicado a outras amostras de dados para perceber se não se trata de *overfitting*.

5. CONCLUSÕES E TRABALHOS FUTUROS

Este relatório apresenta o trabalho desenvolvido durante um estágio de 6 meses na Deloitte, onde a aluna teve a oportunidade de trabalhar especificamente para um cliente no ramo da indústria automóvel em que o seu objetivo seria transferir todos os seus dados para a *cloud*, para usufruir de uma maior centralidade dos dados, e assim, sendo possível criar vários *use cases* à volta dos mesmos. Neste caso, os dados seriam sobre o processo de revenda de carros usados e o objetivo final seria criar uma plataforma de *analytics*, para que fosse possível auxiliar os comerciantes na definição do preço ótimo para a revenda de carros usados.

O primeiro desafio da aluna foi criar uma arquitetura em AWS para criar um sistema de ETL dinâmico, pronto a receber diferentes dados de diversas fontes. Nesta fase, foi importante o conhecimento da aplicabilidade dos serviços de AWS e a sua definição usando um processo de *Infrastructure as Code* (IaC).

O segundo desafio, foi a gestão de vários mercados, contendo tabelas e variáveis diferentes, que tiveram de ser normalizadas para poderem ser orquestradas no mesmo sistema. Nesta fase, foram bastante importantes o conhecimento de negócio e a implementação de *data quality checks* durante todo o processo da pipeline. Permitindo assim, uma boa gestão das transformações aplicadas para que a integridade dos dados fosse assegurada.

O terceiro desafio, foi finalmente, o foco deste documento, a análise preditiva do preço dos carros usados. Esta etapa, sendo a mais interessante para a aluna teve um sabor "agridoce", dado que pôde finalmente ter contacto com a área de *Data Science*, no entanto, não conseguindo produzir as análises desejadas dado as exigências do cliente no tempo limitado do projeto. A maior limitação e diferença do seu percurso académico, foi a exigência da generalização da solução a outros tipos de dados, não podendo, assim, aplicar transformações muito específicas aos dados analisados.

O trabalho apresentado, apesar de sugerir várias técnicas de tratamento de dados e *feature engineering*, torna-se mais fraco na componente da modelação, dado que esta fase não teve o devido foco e foi implementada em pouco tempo do projeto, sendo assim apresentado um trabalho ainda em progresso.

Para a análise da metodologia e conceito definidos, os resultados foram analisados com dados do mercado da Alemanha de vendas diretas, contendo carros de 03-05-2020 a 30-06-2021. Estes resultados destacaram a eficácia dos modelos de *ensemble* nos dados sobre o processo de revenda de carros usados, sendo o XGBoost o modelo com maior desempenho, obtendo um R^2 de 0.965 na amostra de teste. Esta performance sugere que o modelo é capaz de identificar os comportamentos complexos nos dados, porém precisa de ser analisado com mais cuidado e aplicado a outras amostras para perceber a sua generalidade.

Como trabalho futuro, deverá fazer-se uma análise mais profunda à fase da seleção de variáveis para observar a evolução do desempenho do modelo e perceber em que momento este estabiliza, podendo assim identificar o número e quais as variáveis mais significativas para a explicação da variação do preço. Complementarmente poderá ser feita uma análise à importância das variáveis, com a ajuda de várias métricas, como o *gain, weight* e *cover*, no entanto é preciso ter cuidado na sua análise, podendo estas produzir valores inconsistentes como identificado por (S. Lundberg, 2018), o autor sugere o uso

de uma nova métrica baseada em teoria de jogos, o valor de SHAP, SHapley Additive exPlanation (S. M. Lundberg et al., 2017).

Outra análise interessante de perceber também, seria tentar validar se o modelo consegue aprender tendências temporais dos preços, caso contrário, deverá ter-se muito cuidado com a amostra que é usada para o treino, para que esta não represente quaisquer comportamentos que não possam ser explicados pelas variáveis de input. Neste sentido, poderá haver a possibilidade de explorar outros modelos que consigam lidar com tendências temporais, tais como Large Language Models, LLMs (Jin et al., 2023).

Por último, no sentido de complementar a avaliação feita ao modelo, sugere-se fazer uma análise detalhada dos erros por vários grupos de carros, com o intuito de perceber se o modelo está a ter uma boa generalização ou se aprendeu apenas a previsão do preço com mais precisão em certos grupos.

5.1. AVALIAÇÃO DO ESTÁGIO

Durante o período deste estágio na Deloitte, a aluna teve a oportunidade de participar num projeto bastante desafiante com um cliente do setor automóvel. Teve contacto com várias ferramentas novas, tal como a *cloud*, percebendo a sua importância para a gestão de dados das empresas. Pôde também aplicar os conhecimentos técnicos que obteve no mestrado durante a sua participação no módulo da análise preditiva e a sua aplicação num ambiente profissional. A equipa que integrou, proporcionou-lhe um ambiente de trabalho colaborativo e estimulante, onde pode contribuir ativamente para as discussões e processos de decisão.

A experiência adquirida durante o estágio reforçou o seu interesse pela área de *Data Science* e despertou curiosidade pela área de *Cloud Computing*. A aluna, após o estágio, continuou a trabalhar na Deloitte permitindo aprofundar o seu conhecimento noutros campos na área de *Data Science*, como em sistemas de recomendação e problemas de otimização para a gestão de stocks, também para um cliente da indústria automóvel.

REFERÊNCIAS BIBLIOGRÁFICAS

Abrahamsson, P., Salo, O., Ronkainen, J., & Warsta, J. (2017). *Agile Software Development Methods: Review and Analysis*.

Amazon Athena Documentation. (2023). Amazon Web Services. https://docs.aws.amazon.com/en_us/athena/

Amazon Simple Storage Service Documentation. (2023). Amazon Web Services. https://docs.aws.amazon.com/s3/

AWS Glue Documentation. (2023). Amazon Web Services. https://docs.aws.amazon.com/glue/?icmpid=docs_homepage_analytics

AWS Identity and Access Management Documentation. (2023). Amazon Web Services. https://docs.aws.amazon.com/en_us/iam/

B. Nithya, & Dr. V. Ilango. (2017). Predictive Analytics in Health Care Using Machine Learning Tools and Techniques. *International Conference on Intelligent Computing and Control Systems*.

Breiman, L. (2001). Random Forests. Em H. Blockeel (Ed.), *Machine Learning* (Vol. 45, pp. 5–32). Springer.

Chapman, P. (2000). *CRISP-DM 1.0: Step-by-step data mining guide.* (SPSS, Ed.).

Chen, T., & Guestrin, C. (2016). *XGBoost: A Scalable Tree Boosting System*.

De Santis, R., Di Sano, M., Gunnella, V., & Neves, P. (2022). Motor vehicle sector: explaining the drop in output and the rise in prices. *ECB Economic Bulletin*, *7*. https://www.ecb.europa.eu/pub/economic-bulletin/focus/2022/html/ecb.ebbox202207_02~5bde8eeff0.en.html

Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. *The Annals of Statistics*, *29*(5). https://doi.org/10.1214/aos/1013203451

GeeksforGeeks. (2023). *Supervised and Unsupervised learning*. https://www.geeksforgeeks.org/supervised-unsupervised-learning/.

Gegic, E., Isakovic, B., Keco, D., Masetic, Z., & Kevric, J. (2019). Car Price Prediction using Machine Learning Techniques. *TEM Journal*, *8*(1), 113–118. https://doi.org/10.18421/TEM81-16

Hankar, M., Birjali, M., & Beni-Hssane, A. (2022). Used Car Price Prediction using Machine Learning: A Case Study. *11th International Symposium on Signal, Image, Video and Communications, ISIVC 2022*. https://doi.org/10.1109/ISIVC54825.2022.9800719

Jin, M., Wen, Q., Liang, Y., Zhang, C., Xue, S., Wang, X., Zhang, J., Wang, Y., Chen, H., Li, X., Pan, S., Tseng, V. S., Zheng, Y., Chen, L., & Xiong, H. (2023). *Large Models for Time Series and Spatio-Temporal Data: A Survey and Outlook*.

John D. Kelleher, Brian Mac Namee, & Aoife D'Arcy. (2020). *Fundamentals of Machine Learning for Predictive Data Analytics, Second Edition Algorithms, Worked Examples, and Case Studies*. The MIT Press.

Kingsford, C., & Salzberg, S. L. (2008). What are decision trees? *Nature Biotechnology*, *26*(9), 1011–1013. https://doi.org/10.1038/nbt0908-1011

Kumar, R. A., & Samruddhi, K. (2020). Used Car Price Prediction using K-Nearest Neighbor Based Model. *International Journal of Innovative Research in Applied Sciences and Engineering (IJIRASE)*, *4*(3). https://doi.org/10.29027/IJIRASE.v4.i3.2020.686-689

Kumar, S. (2021, Maio 31). *3 Techniques to Avoid Overfitting of Decision Trees*. Towards Data Science. https://towardsdatascience.com/3-techniques-to-avoid-overfitting-of-decision-trees-1e7d3d985a09

Lundberg, S. (2018, Abril 17). *Interpretable Machine Learning with XGBoost*. Towards Data Science. https://towardsdatascience.com/interpretable-machine-learning-with-xgboost-9ec80d148d27

Lundberg, S. M., Allen, P. G., & Lee, S.-I. (2017). A Unified Approach to Interpreting Model Predictions. *31st Conference on Neural Information Processing Systems (NIPS 2017)*. https://github.com/slundberg/shap

Malsam, W. (2023, Julho 11). *Scrum Methodology: An Introduction to the Scrum Process*. Project Manager. https://www.projectmanager.com/blog/scrum-methodology

Mane, P. (2021, Setembro 30). *Multicollinearity in Tree Based Models*. https://medium.com/@manepriyanka48/multicollinearity-in-tree-based-models-b971292db140.

McDonald, G. C. (2009). Ridge regression. *WIREs Computational Statistics*, *1*(1), 93–100. https://doi.org/10.1002/wics.14

Micci-Barreca, D. (2001). A preprocessing scheme for high-cardinality categorical attributes in classification and prediction problems. *ACM SIGKDD Explorations Newsletter*, *3*(1), 27–32. https://doi.org/10.1145/507533.507538

Monburinon, N., Chertchom, P., Kaewkiriya, T., Rungpheung, S., Buya, S., & Boonpou, P. (2018). Prediction of prices for used car by using regression models. *Proceedings of 2018 5th International Conference on Business and Industrial Research: Smart Technology for Next Generation of Information, Engineering, Business and Social Science, ICBIR 2018*, 115–119. https://doi.org/10.1109/ICBIR.2018.8391177

Morde, V. (2019, Abril 8). *XGBoost Algorithm: Long May She Reign!* Towards Data Science. https://towardsdatascience.com/https-medium-com-vishalmorde-xgboost-algorithm-long-she-may-rein-edd9f99be63d

Ozer, D. J. (1985). Correlation and the coefficient of determination. *Psychological Bulletin*, *97*(2), 307–315. https://doi.org/10.1037/0033-2909.97.2.307

Pal, N., Arora, P., Sundararaman, D., Kohli, P., & Palakurthy, S. S. (2018). How much is my car worth? A methodology for predicting used cars prices using Random Forest. *Future of Information and Communications Conference (FICC)*. www.pakwheels.com

Pedregosa et al. (2011). Scikit-learn: Machine Learning in Python. *JMLR 12*, 2825–2830.

Pudaruth, S. (2014). Predicting the Price of Used Cars using Machine Learning Techniques. *International Journal of Information & Computation Technology*, *4*(7), 753–764. http://www.irphouse.com

Schapire, R. E. (2013). Explaining AdaBoost. Em B. Schölkopf, Z. Luo, & V. Vovk (Eds.), *Empirical Inference: Festschrift in Honor of Vladimir N. Vapnik* (pp. 1–287). Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-41136-6

Su, X., Yan, X., & Tsai, C. L. (2012). Linear regression. *Wiley Interdisciplinary Reviews: Computational Statistics*, *4*(3), 275–294. https://doi.org/10.1002/wics.1198

Tibshirani, R. (1996). Regression Shrinkage and Selection Via the Lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, *58*(1), 267–288. https://doi.org/10.1111/J.2517-6161.1996.TB02080.X

Used Car Market Size, Share & Trend Analysis Report By Vehicle Type (Hybrid, Conventional, Electric), By Vendor Type, By Fuel Type, By Size, By Sales Channel, By Region, And Segment Forecasts, 2023 - 2030. (2022). Em *Grand View Research*. https://www.grandviewresearch.com/industry-analysis/used-car-market

Walczak, S., & Cerpa, N. (2003). Artificial Neural Networks. Em Academic Press (Ed.), *Encyclopedia of Physical Science and Technology* (3.ª ed., pp. 631–645). Elsevier. https://doi.org/10.1016/B0-12-227410-5/00837-1

Wang, G., Hao, J., Ma, J., & Jiang, H. (2011). A comparative assessment of ensemble learning for credit scoring. *Expert Systems with Applications*, *38*(1), 223–230. https://doi.org/10.1016/j.eswa.2010.06.048

Willmott, C., & Matsuura, K. (2005). Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance. *Climate Research*, *30*, 79–82. https://doi.org/10.3354/cr030079

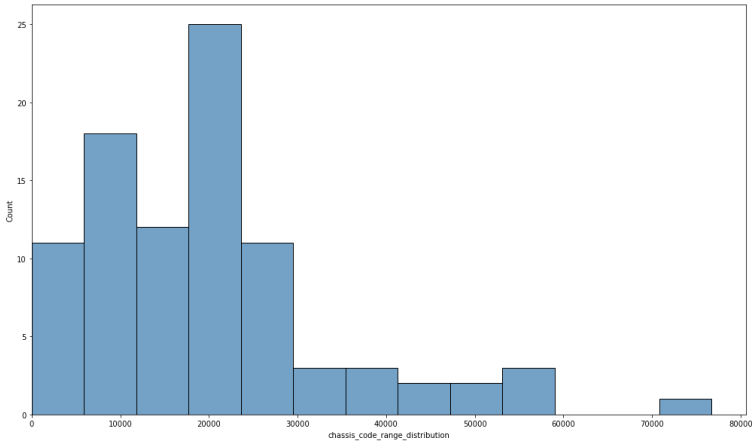
XGBoost Documentation. (2022, Dezembro). https://xgboost.readthedocs.io/en/latest/index.html

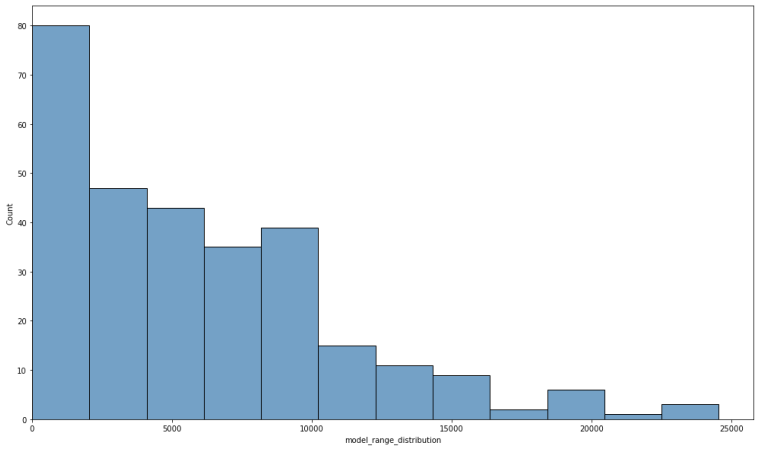
XGBoost Parameters. (2022, Dezembro). XGBoost Documentation. https://xgboost.readthedocs.io/en/latest/parameter.html

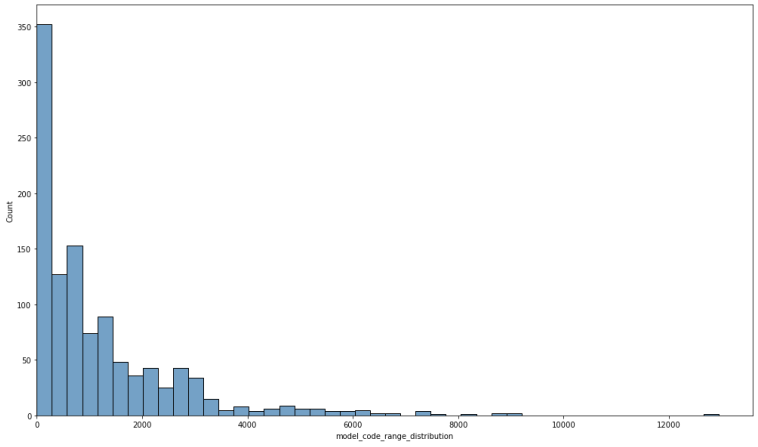
Zhang, Y., & Haghani, A. (2015). A gradient boosting method to improve travel time prediction. *Transportation Research Part C: Emerging Technologies*, *58*, 308–324. https://doi.org/10.1016/J.TRC.2015.02.019

APÊNDICE A

Distribuição do valor máximo e mínimo da variável de cada categoria das variáveis: *chassis_code, model* e *model_code*. É de notar que os eixos do x, representados nas figuras abaixo são diferentes para cada grupo. Tal como era esperado, dado a sua menor granularidade, o grupo *chassis_code* é o que tem o maior intervalo de preços.







APÊNDICE B

Distribuição das variáveis que representam a relatividade do preço do carro ao preço mínimo do seu grupo, por ordem ascendente de granularidade, os grupos do *chassis_code, model* e *model_code*.