INDICIUM

UBIRATAN DA SILVA TAVARES

DESAFIO CIENTISTA DE DADOS

Rio de Janeiro

2023

1. Objetivos

- Analisar os dados e responder perguntas de negócios do cliente cujo *core business* é compra e venda de veículos usados.
- Criar modelo preditivo que precifique os carros do cliente.

2. Base de Dados

Para alcançar os objetivos definidos anteriormente foram utilizados dois *datasets:*

- Um dataset para treinamento chamado **cars_training** composto por 29584 linhas, 28 colunas de informação (features) e a variável a ser prevista (“preco”).
- Um segundo dataset para teste chamado de **cars_test** composto por 9862 linhas e 28 colunas, sendo que este dataset não possui a coluna “preco”.

A Tabela 1 apresenta a identificação e a descrição de cada atributo presente nos datasets de treinamento e teste.

Tabela 1 – atributos da base de dados de treinamento e teste

Atributo	Descrição
id	Identificador único dos veículos cadastrados na base de dados
num_fotos	Quantidade de fotos que o anúncio do veículo contém
marca	Marca do veículo anunciado
modelo	Modelo do veículo anunciado
versao	Versão do veículo anunciando. Sua cilindrada, quantidade de válvulas, se é flex ou não, etc.
ano_de_fabricacao	Ano de fabricação do veículo anunciado
ano_modelo	Modelo do ano de fabricação do veículo anunciado
hodometro	Valor registrado no hodômetro do veículo anunciado
cambio	Tipo de câmbio do veículo anunciado
num_portas	Quantidade de portas do veículo anunciado
tipo	Tipo do veículo anunciado. Se ele é sedã, hatch, esportivo, etc.
blindado	Informação se o veículo anunciado é blindado ou não
cor	Cor do veículo anunciado
tipo_vendedor	Tipo do vendedor do veículo anunciado
cidade_vendedor	Cidade em que vendedor do veículo anunciado reside
estado_vendedor	Estado em que vendedor do veículo anunciado reside
anunciante	Tipo de anunciante do vendedor do veículo anunciado
entrega_delivery	Vendedor faz ou não delivery do veículo anunciado
troca	Veículo anunciado já foi trocado anteriormente
elegivel_revisao	Veículo anunciado precisa ou não de revisão
dono_aceita_troca	Vendedor aceita ou não realizar uma troca com o veículo anunciado
veiculo_único_dono	Veículo anunciado é de um único dono
revisoes_concessionaria	Veículo anunciado teve suas revisões feitas em concessionárias
ipva_pago	Veículo anunciado está com o IPVA pago ou não
veiculo_licenciado	Veículo anunciado está com o licenciamento pago ou não
garantia_de_fábrica	Veículo anunciado possui garantia de fábrica ou não
revisoes_dentro_agenda	Avalia se as revisões feitas do veículo anunciado foram realizadas dentro da agenda prevista
veiculo_alienado	Veículo anunciado está alienado ou não
preco (target)	Preço do veículo anunciado

3. Análise das Principais Estatísticas da Base de Dados de Treinamento

Para iniciar a compreensão dos dados foi utilizado a biblioteca pandas para carregar a base de dados **cars_training** para o ambiente de desenvolvimento no Google Colab.

Foi criado a variável **df** (objeto do DataFrame do pandas) no momento do carregamento da base de dados **cars_training.**

3.1 Análise Preliminar

Na análise preliminar foram levantadas as seguintes informações com base na variável **df:**

- A base de dados **cars_training** carregada possui **29.584** registros (linhas) e **29** atributos (colunas);
- Quanto ao tipo de atributos contidos na base de dados **cars_training**:
  - object: **id**, **marca**, **modelo**, **versao**, **cambio**, **tipo**, **blindado**, **cor**, **tipo_vendedor**, **cidade_vended**or, **estado_vendedo**r, **anunciante**, **dono_aceita_troca**, **veiculo_unico_dono**, **revisoes_concessionaria**, **ipva_pago**, **veiculo_licenciado**, **garantia_de_fabrica** e **revisoes_dentro_agenda**.
  - float: **num_fotos**, **ano_modelo**, **hodometro**, **veiculo_alienado** e **preco** são do tipo float;
  - int: **ano_de_fabricacao** e **num_portas**.
  - bool: **entrega_delivery**, **troca** e **elegivel_revisao.**
- Quanto aos atributos contidos na base de dados **cars_training** com valores faltantes: **num_fotos** (177); **dono_aceita_troca** (7.662); **veiculo_unico_dono** (19.161); **revisoes_concessionaria** (20.412); **ipva_pago** (9.925); **veiculo_licenciado** (13.678); **garantia_de_fabrica** (25.219); **revisoes_dentro_agenda** (23.764); **veiculo_alienado** (29.584).
- Quanto aos atributos contidos na base de dados **cars_training** contendo um valor único: **elegivel_revisao, dono_aceita_troca, veiculo_unico_dono, revisoes_concessionaria, ipva_pago, veiculo_licenciado, garantia_de_fabrica, revisoes_dentro_agenda.**

3.2 Estatísticas Resumidas e Visualização de Dados

A entender os dados significa ter uma melhor compreensão das distribuições dos atributos e das relações entre os atributos da base de dados **cars_training.**

Dois grandes ramos de métodos estatísticos são usados para auxiliar na compreensão dos dados; eles são: estatísticas resumidas e visualização de dados.

3.2.1 Estatísticas Resumidas

Entende-se por estatísticas resumidas, os métodos usados para resumir a distribuição e relacionamentos entre variáveis usando quantidades estatísticas.

A Tabela 2 apresenta um resumo das principais estatísticas descritivas dos, tais como quantidade da amostra, média, desvio-padrão, valor mínimo e máximo e quartis, dos atributos numéricos da base de dados **cars_training.**

Tabela 2 – Resumo estatístico dos atributos numéricos

	num_fotos	ano_de_fabricacao	ano_modelo	hodometro	num_portas	veiculo_alienado	preco
count	29407	29584	29584	29584	29584	0	29584
mean	10.32383446	2016.758552	2017.808985	58430.59208	3.940677393		133023.8799
std	3.487334496	4.062422107	2.673930199	32561.76931	0.3383602707		81662.87225
min	8	1985	1997	100	2		9869.950645
25%	8	2015	2016	31214	4		76571.76846
50%	8	2018	2018	57434	4		114355.797
75%	14	2019	2020	81953.5	4		163679.6174
max	21	2022	2023	390065	4		1359812.892

É importante destacar da Tabela 2 que o desvio-padrão dos atributos **hodometro** e **preco** são bastante elevados, significando que há uma alta dispersão nas amostras dos dados, o que pode configurar à presença de *outliers, a* ser confirmado posteriormente na etapa de visualização de dados.

Observa-se que o atributo **veiculo_alienad**o não contém nenhuma informação, isto é, todas 29.584 amostras com valores faltantes, revelando a insignificância deste atributo para a análise dos dados.

Nota-se que o atributo **num_fotos** contém valores faltantes, por conter uma quantidade menor de amostra comparado ao total de amostras da base de dados, confirmando a diferença das 177 amostras com valores faltantes.

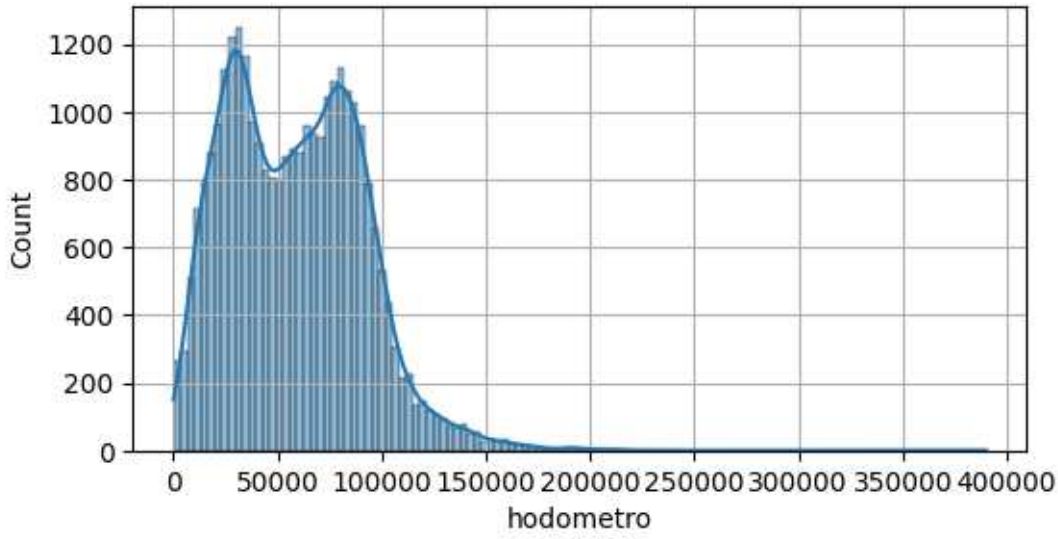
3.2.2 Visualização de Dados

Entende-se por visualização de dados, os métodos usados para resumir a distribuição e as relações entre os atributos usando visualizações gráficas.

Foram construídos histogramas e *boxplots* para visualizar a distribuição das amostras dos dados dos atributos numéricos contínuos, como os atributos **hodometro** e **preco**.

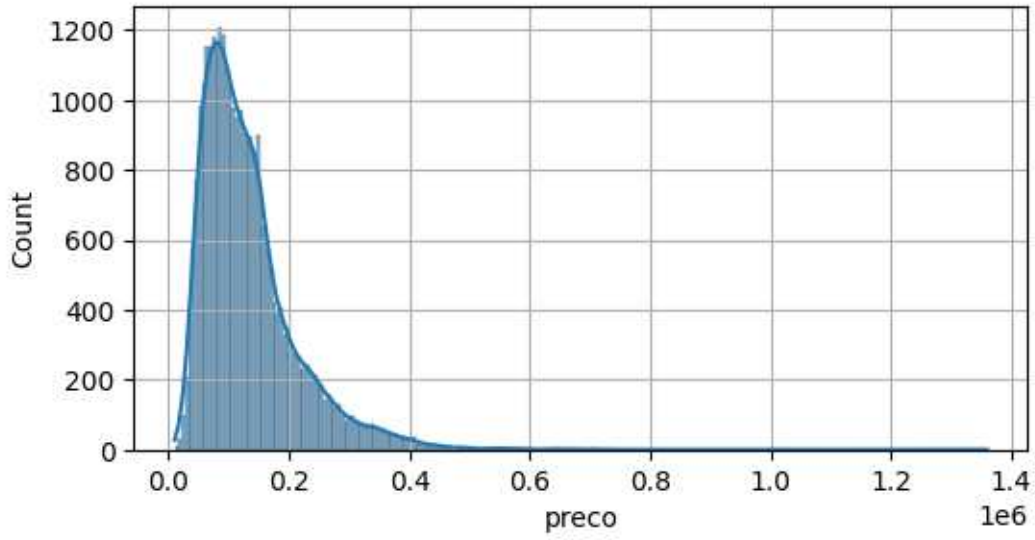
Na Figura 1 é apresentado a distribuição das amostras do atributo **hodometro**. Observa-se que a distribuição é bimodal, isto é, existem dois picos ou modas distintas, ou seja, dois valores que são mais frequentes do que os demais na amostra. Por fim, nota-se que a cauda direita é mais longa e os valores concentram-se mais à esquerda, o que pode indicar a presença de *outliers*.

Figura 1 – Histograma do atributo hodometro



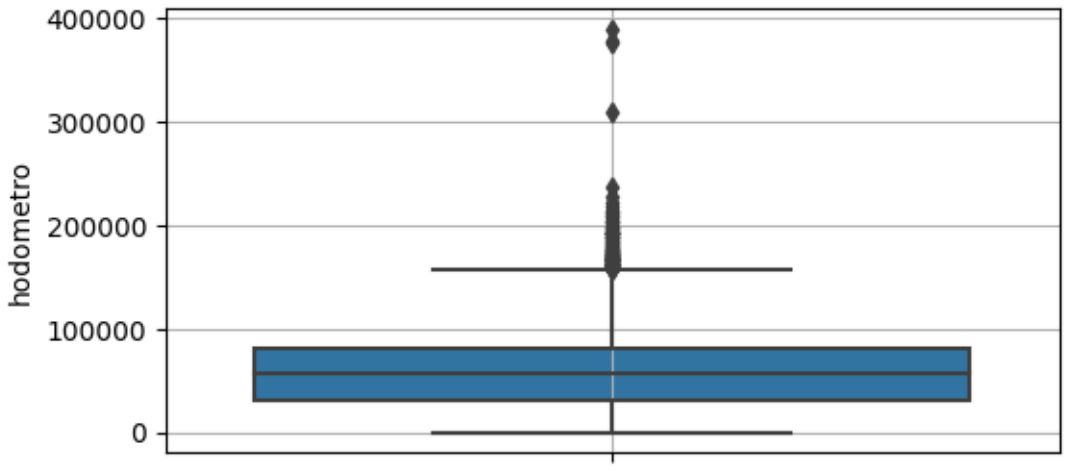
Na Figura 2 é apresentado a distribuição das amostras do atributo **preco**. Observa-se que as amostras dos dados não estão simetricamente distribuídos em torno da média. A cauda direita é mais longa e os valores concentram-se mais à esquerda da medida central, o que pode indicar a presença de *outliers.* A distribuição abaixo é dita assimétrica positiva (à direita).

Figura 2 – Histograma do atributo preco



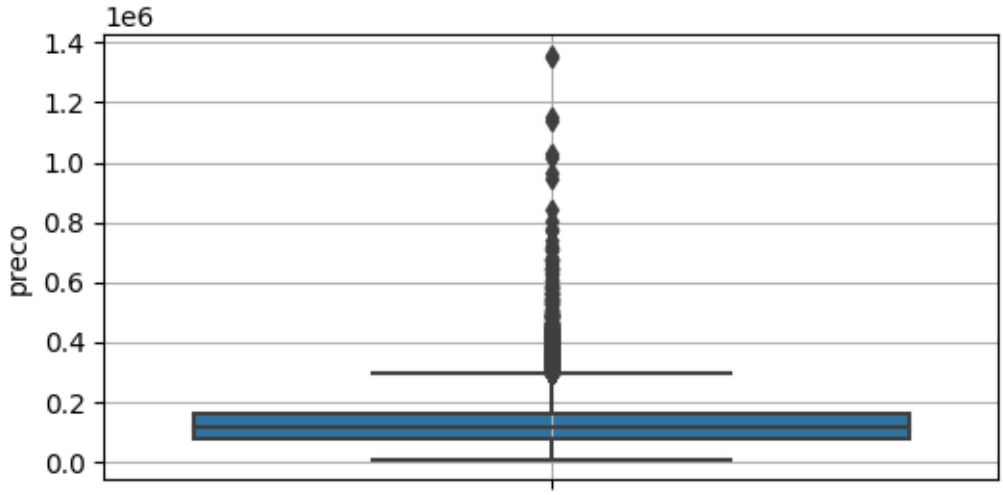
Na Figura 3 é apresentado a distribuição das amostras dos dados do atributo **hodometro** por meio do gráfico *boxplot*. Observa-se a presença de valores *outliers*, o que confirma a alta dispersão das amostras dos dados.

Figura 3 – Boxplot do atributo hodometro



Na Figura 4 é apresentado a distribuição das amostras dos dados do atributo **preco** por meio do gráfico *boxplot*. Observa-se a presença de valores *outliers*, o que confirma a alta dispersão das amostras dos dados.

Figura 4 – Boxplot do atributo preco



Os *outliers*, também conhecidos como “valores atípicos”, são pontos de dados que estão muito distantes dos demais valores, podendo indicar anomalias ou erros de leitura ou medição.

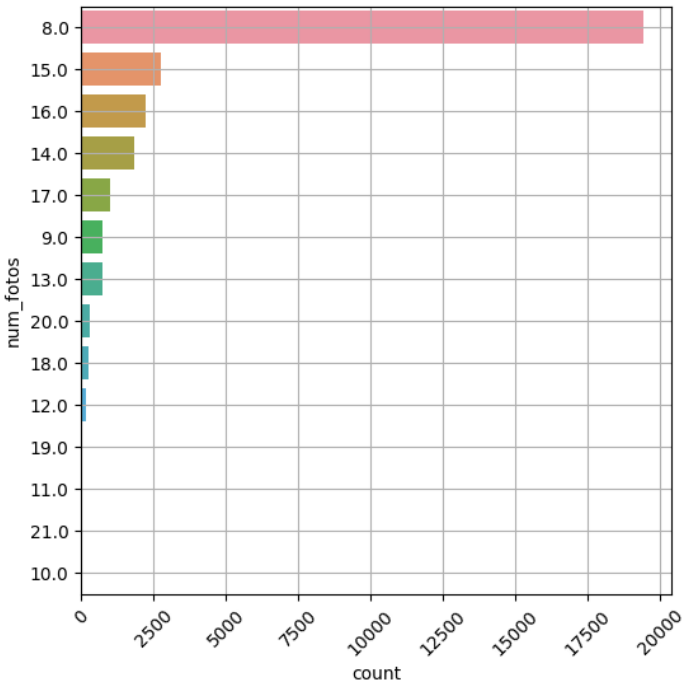
Estes “valores atípicos”, visualizados na Figura 3 e Figura 4, em alguns casos podem ser de grande importância e relevância para a análise dos dados, enquanto em outros casos, podem ser resultado de erros ou eventos raros e devem ser tratados com cautela.

A presença de *outliers* pode afetar a interpretação de estatísticas descritivas e análises. Por isso, é importante investigar e entender a natureza desses “valores atípicos” antes de tomar decisões sobre como tratá-los em uma análise estatística ou modelagem de dados.

Para os atributos numéricos discretos e categóricos foram construídos gráficos de barras para visualizar a frequência dos valores discretos e categóricos.

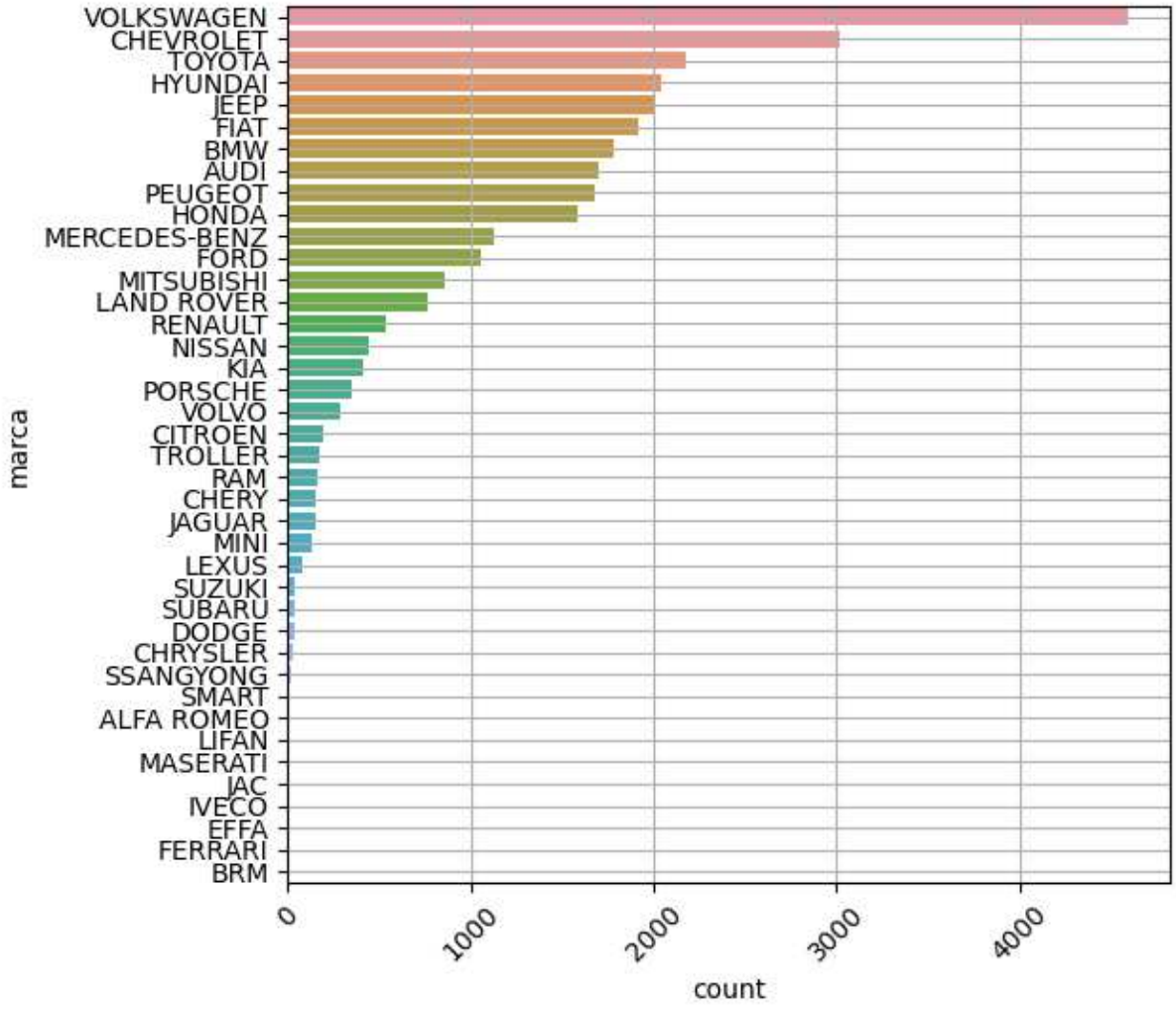
Na Figura 5 é apresentado o gráfico de barra das amostras dos dados do atributo **num_fotos.** Observa-se a predominância do anúncio de veículos contendo 8 fotos e a existência de um grande desbalanceamento entre a quantidade de amostras de veículos anunciados contendo 8 fotos e as demais quantidades de fotos apresentadas no eixo y.

Figura 5 – Gráfico de Barra do atributo num_fotos



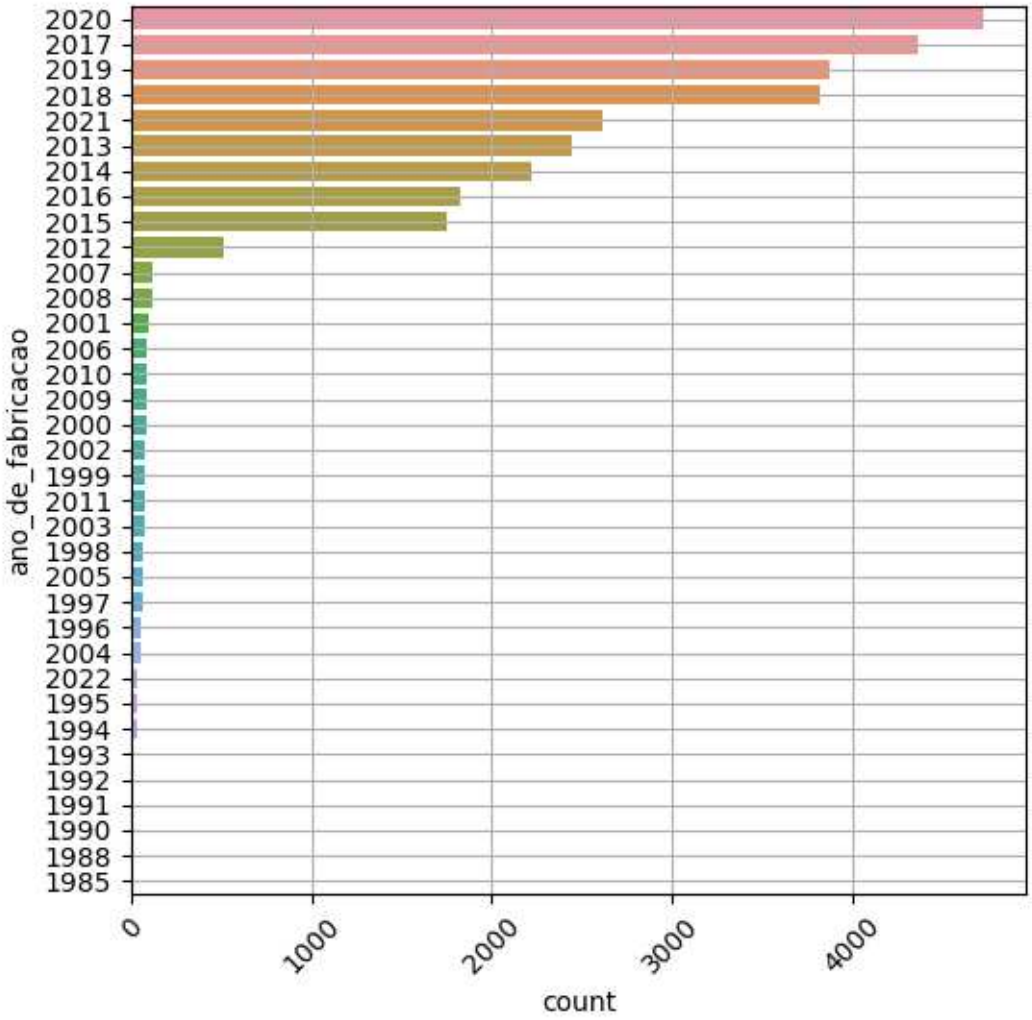
Na Figura 6 é apresentado o gráfico de barra das amostras dos dados do atributo **marca**. Observa-se a predominância de veículos anunciados da marca VOLKSWAGEM e a existência de um grande desbalanceamento das marcas de veículos com maiores amostras e as marcas de veículos com menores amostras, conforme visualizadas no eixo y.

Figura 6 – Gráfico de Barra do atributo marca



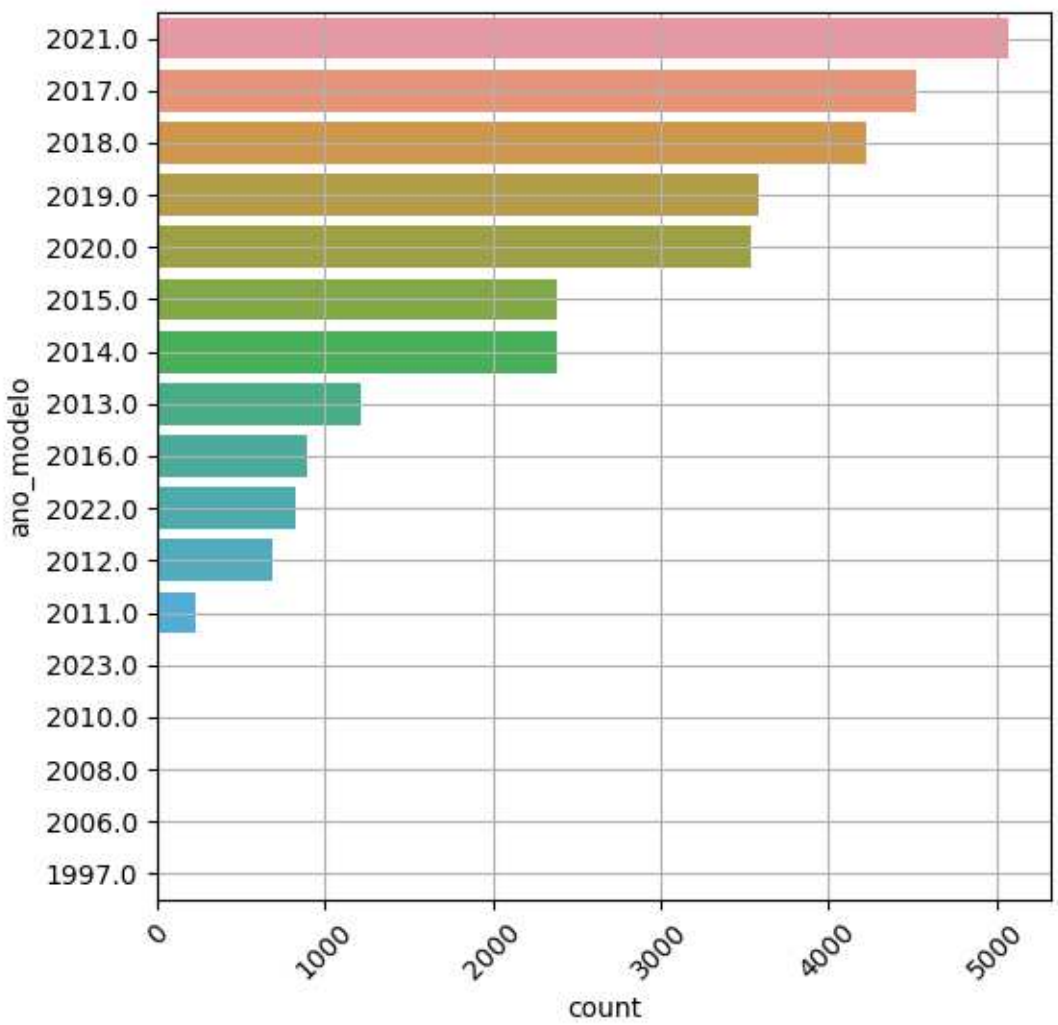
Na Figura 7 é apresentado o gráfico de barra das amostras dos dados do atributo **ano_de_fabricacao**. Observa-se a predominância de veículos fabricados no ano de 2020 e a existência de um grande desbalanceamento dos veículos mais novos (ano_de_fabricacao >= 2015) e os veículos mais antigos (ano_de_fabricacao < 2015), conforme visualizadas no eixo y.

Figura 7 – Gráfico de Barra do atributo ano_de_fabricacao



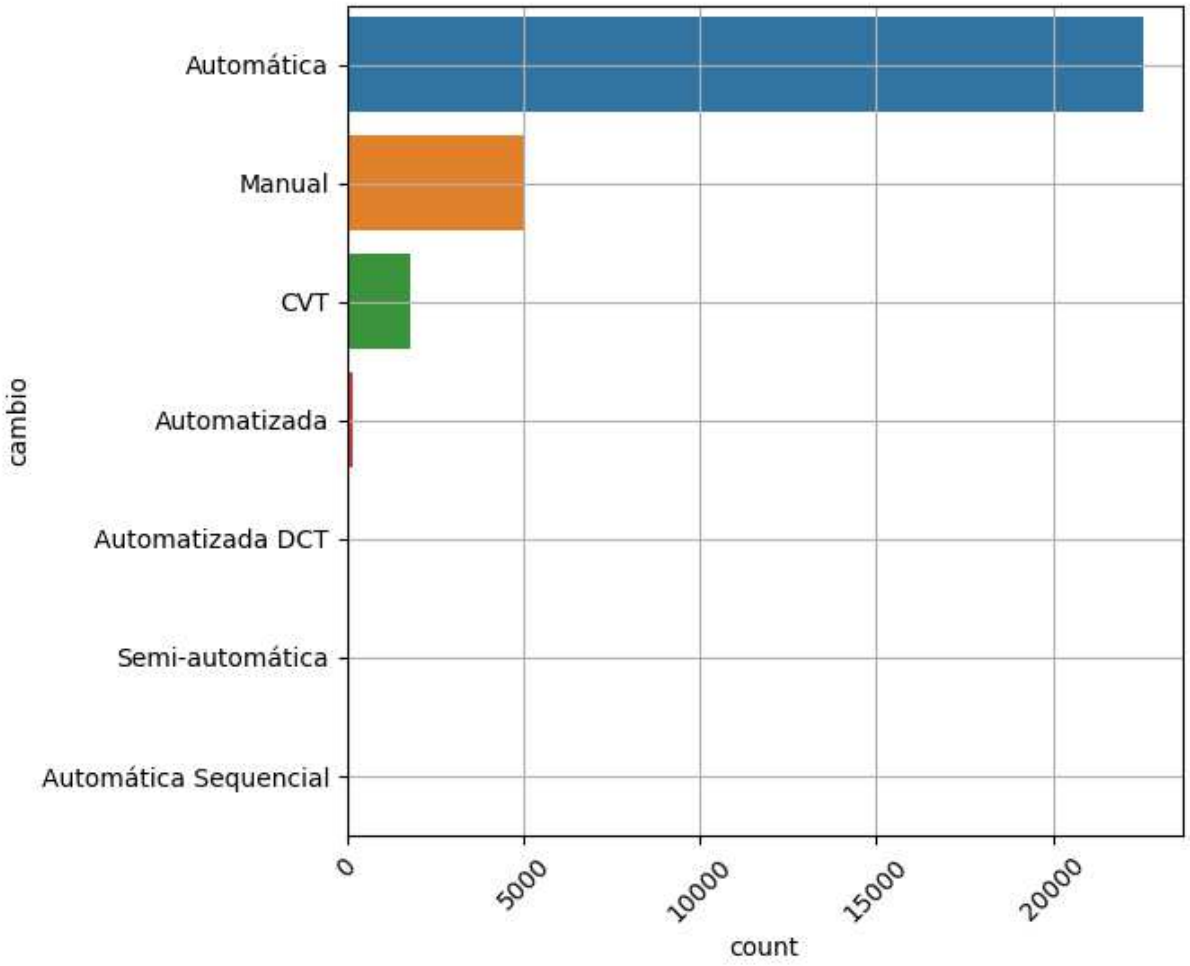
Na Figura 8 é apresentado o gráfico de barra das amostras dos dados do atributo **ano_de_modelo**. Observa-se a predominância de veículos com modelo do ano de fabricação de 2021 e a existência de um grande desbalanceamento entre as 17 modelos do ano de fabricação.

Figura 8 – Gráfico de Barra do atributo ano_modelo



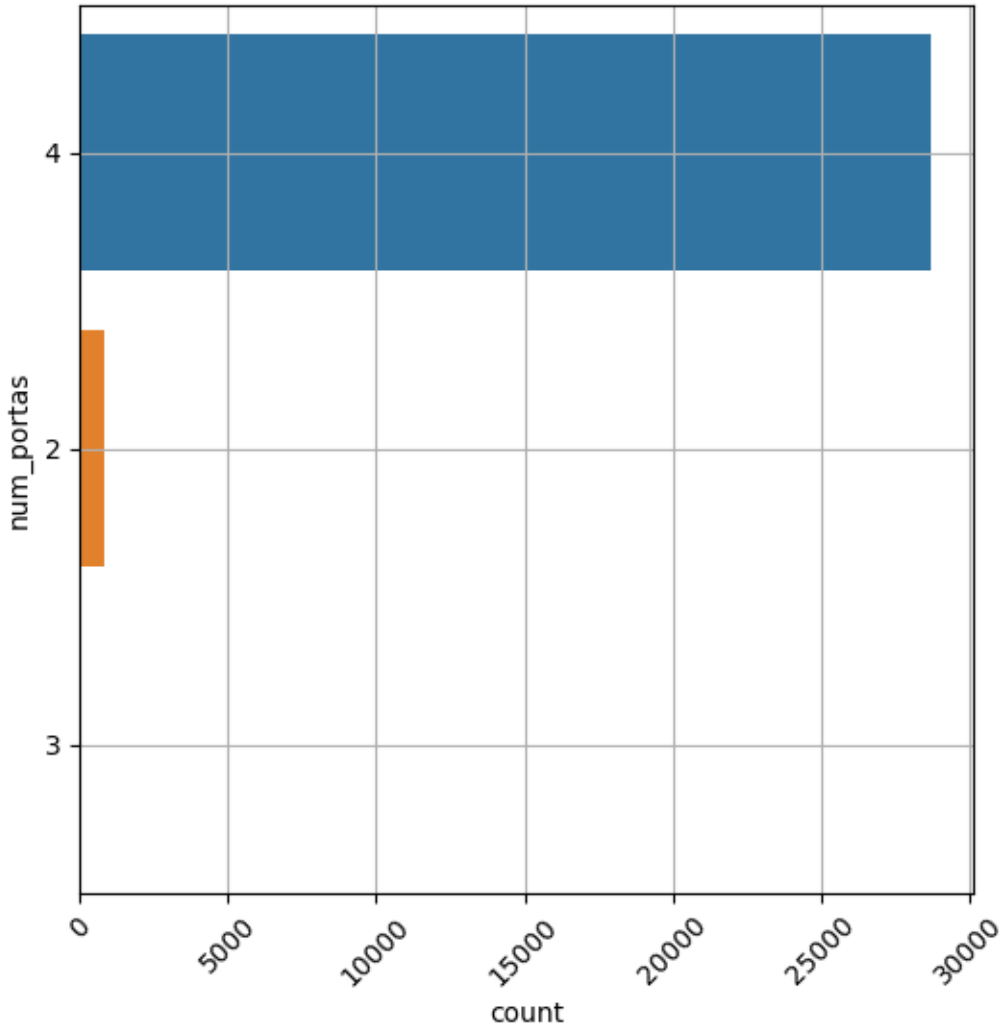
Na Figura 9 é apresentado o gráfico de barra das amostras dos dados do atributo **cambio.** Observa-se a predominância de veículos com cambio do tipo automático e a existência de um grande desbalanceamento entre os 7 tipos de cambio.

Figura 9 – Gráfico de Barra do atributo cambio



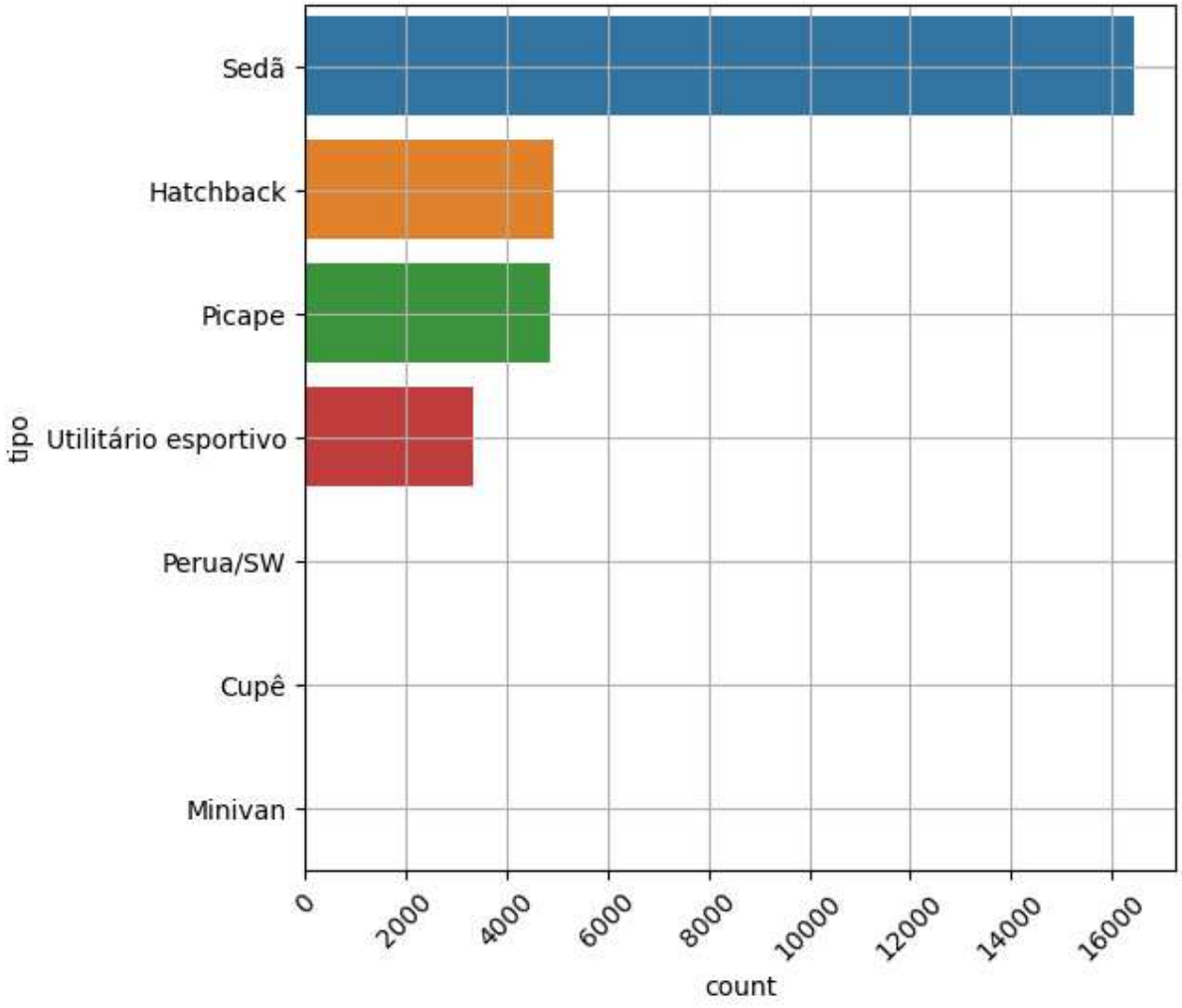
Na Figura 10 é apresentado o gráfico de barra das amostras dos dados do atributo **num_portas.** Observa-se a predominância de veículos anunciados contendo 4 portas e o grande desequilíbrio entre as amostras dos veículos de acordo com o seu número de portas.

Figura 10 – Gráfico de Barra do atributo num_portas



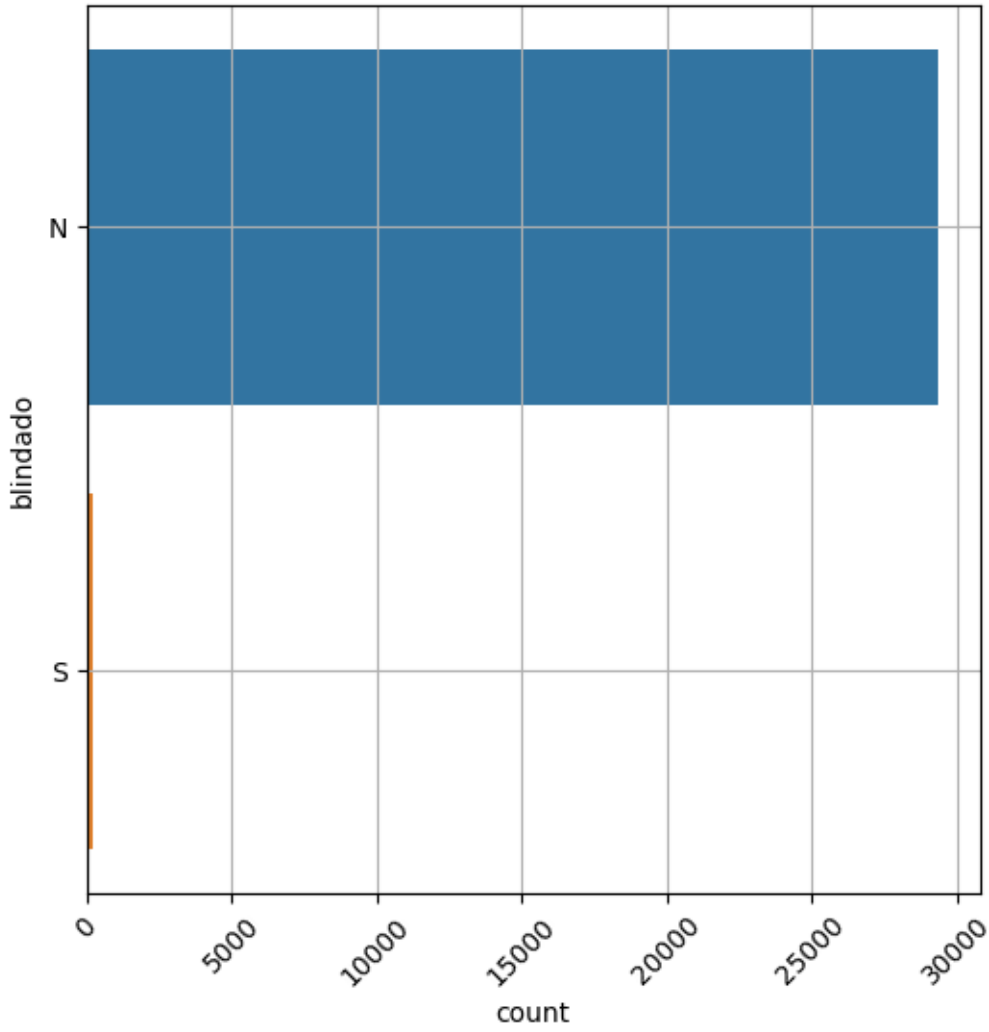
Na Figura 11 é apresentado o gráfico de barra das amostras dos dados do atributo **tipo.** Observa-se a predominância de veículos anunciados do tipo Sedã e existência de um grande desbalanceamento entre os sete tipos de veículos

Figura 11 – Gráfico de Barra do atributo tipo



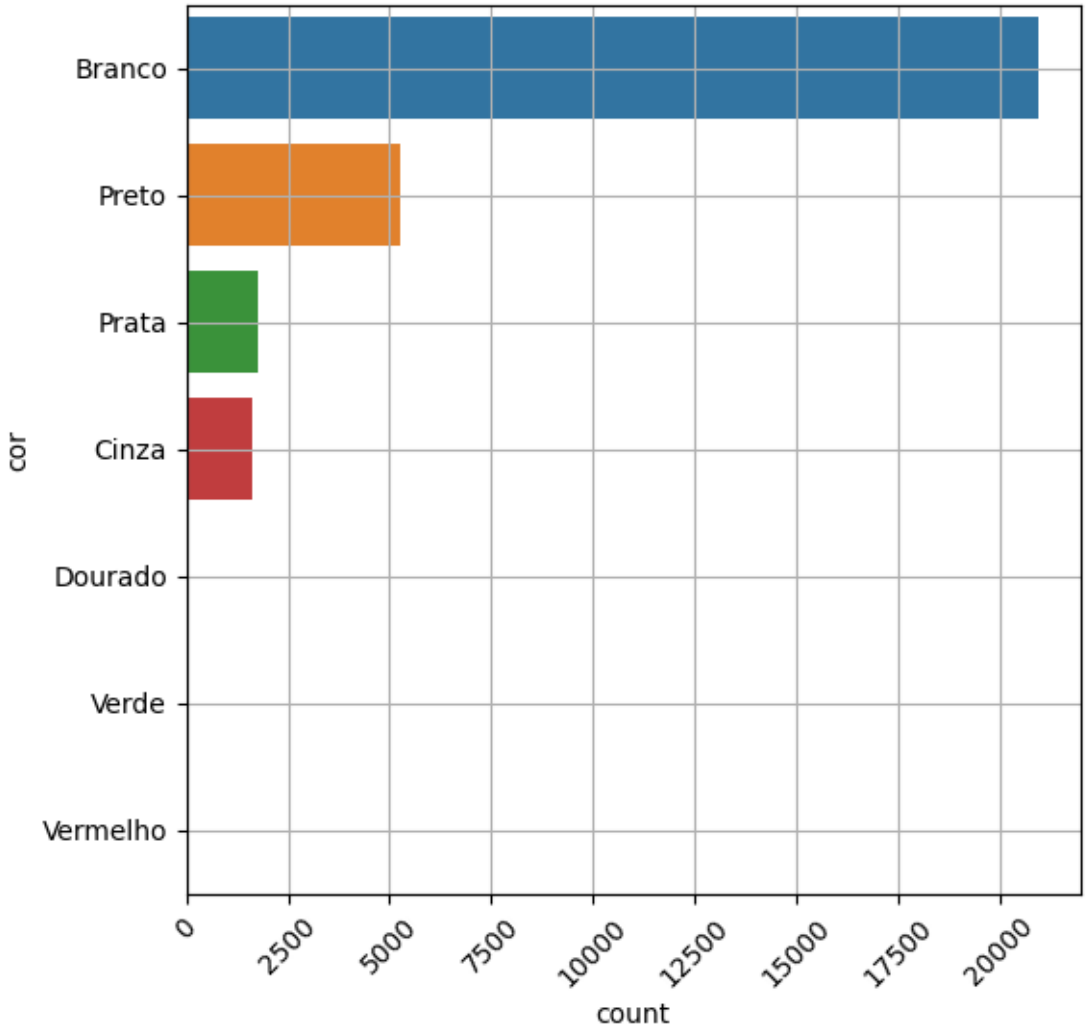
Na Figura 12 é apresentado o gráfico de barra das amostras dos dados do atributo **blindado.** Observa-se a predominância de veículos anunciados que não possuem blindagem e a existência de um desequilíbrio acentuado entre as duas classes deste atributo.

Figura 12 – Gráfico de Barra do atributo blindado



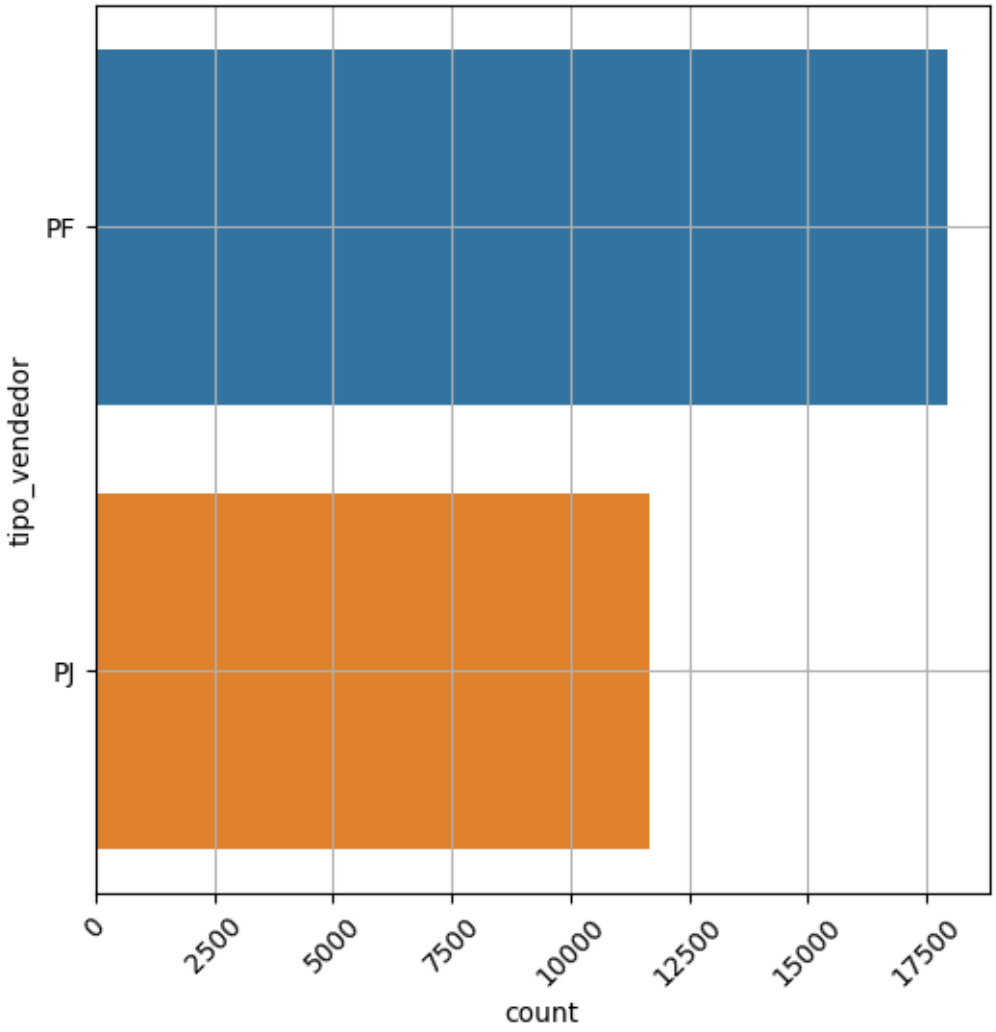
Na Figura 13 é apresentado o gráfico de barra das amostras dos dados do atributo **cor.** Observa-se a predominância de veículos anunciados com a cor branca e um desequilíbrio acentuado entre as sete classes deste atributo.

Figura 13 – Gráfico de Barra do atributo cor



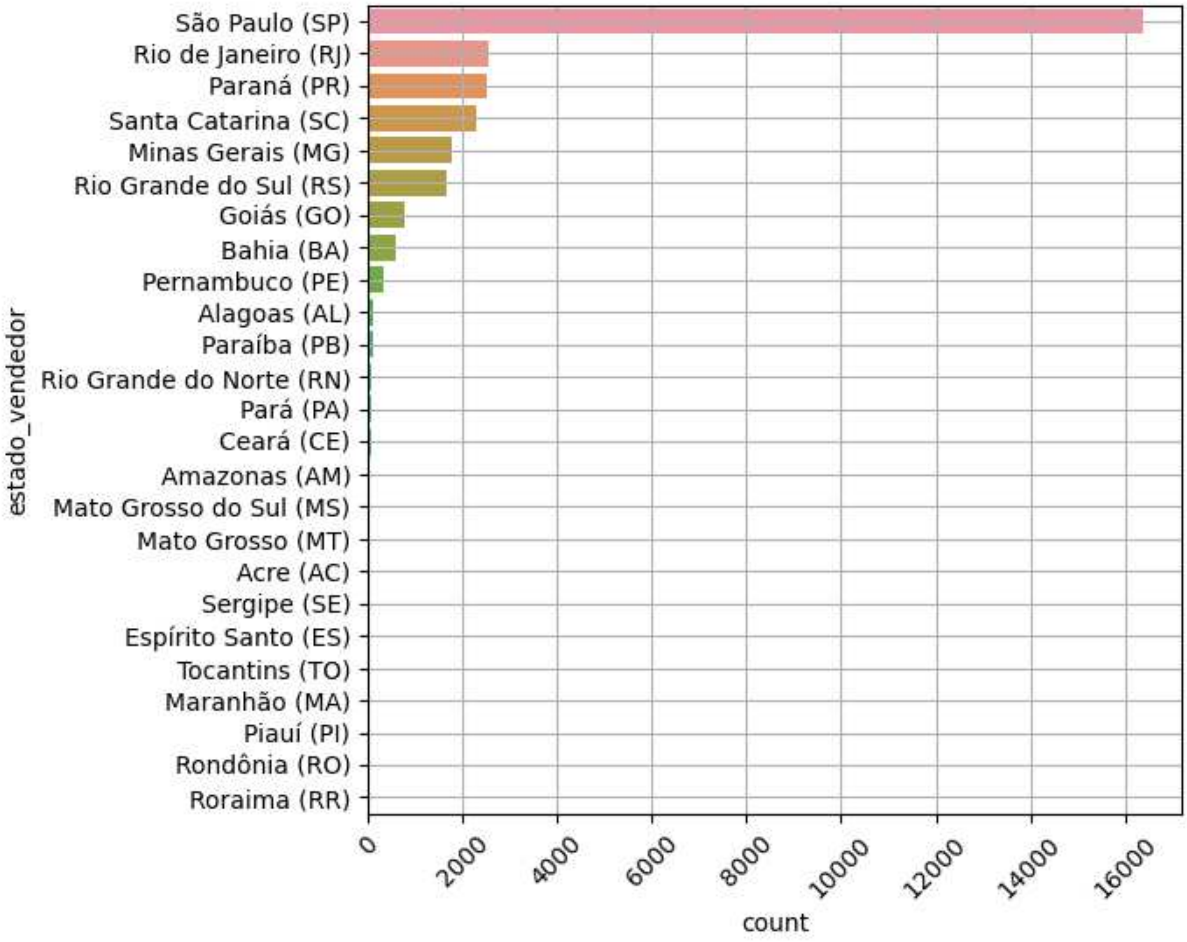
Na Figura 14 é apresentado o gráfico de barra das amostras dos dados do atributo **tipo_vendedor.** Observa-se a predominância da “PESSOA FÍSICA” como tipo de vendedor do veiculo anunciado e um pequeno desbalanceamento entre as duas classes deste atributo.

Figura 14 – Gráfico de Barra do atributo tipo_vendedor



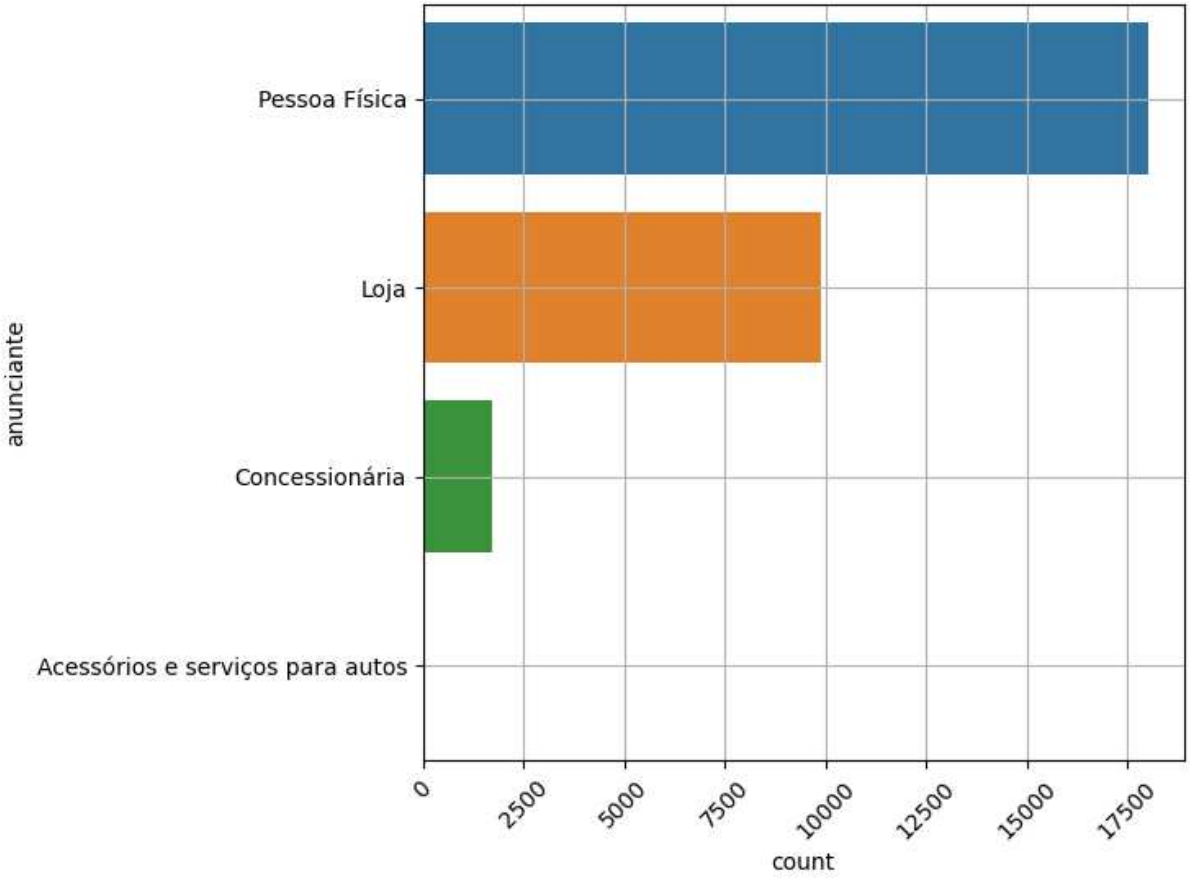
Na Figura 15 é apresentado o gráfico de barra das amostras dos dados do atributo **estado_vendedor.** Observa-se a predominância veículos anunciados com residência no estado de São Paulo e um grande desbalanceamento entre as 25 classes deste atributo.

Figura 15 – Gráfico de Barra do atributo estado_vendedor



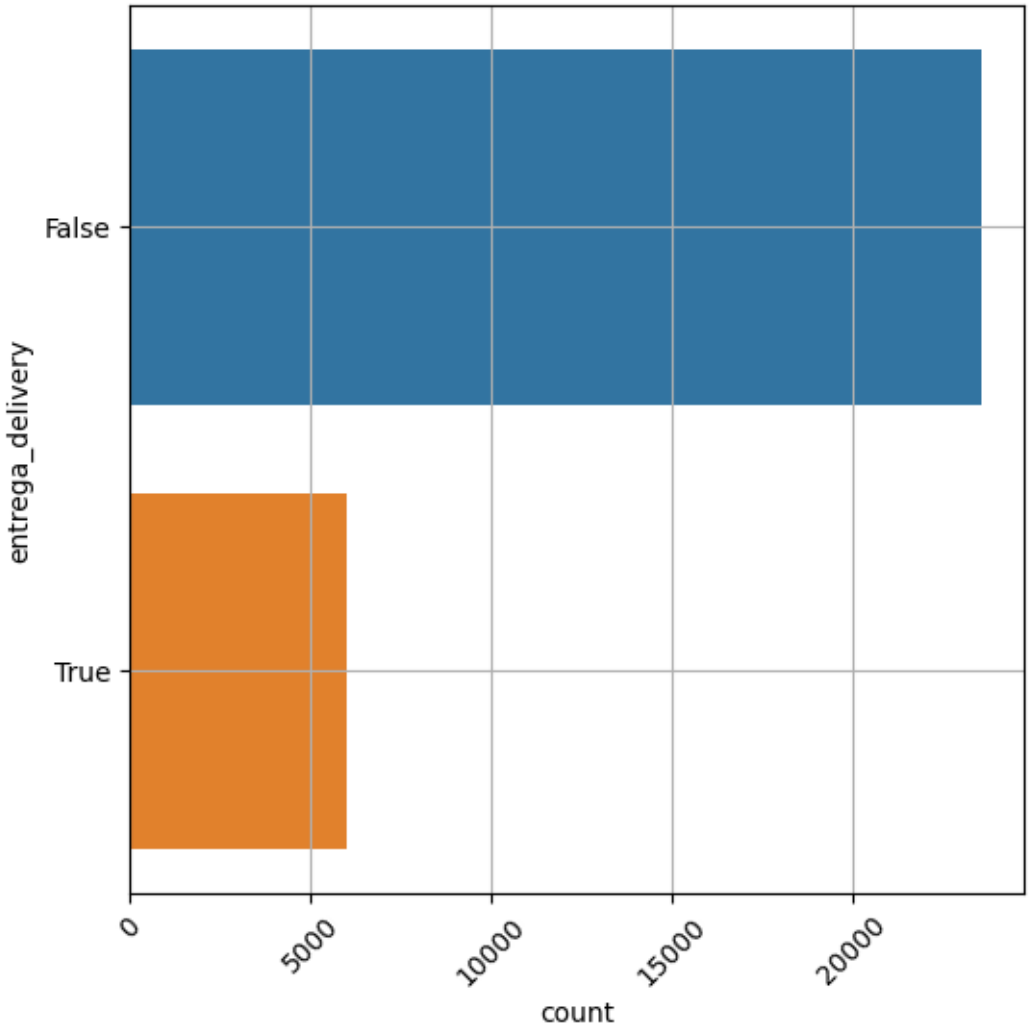
Na Figura 16 é apresentado o gráfico de barra das amostras dos dados do atributo **anunciante.** Observa-se a predominância do anunciante do veículo ser pessoa física. Observa-se também a existência de um grande desbalanceamento entre as 4 classes deste atributo.

Figura 16 – Gráfico de Barra do atributo anunciante



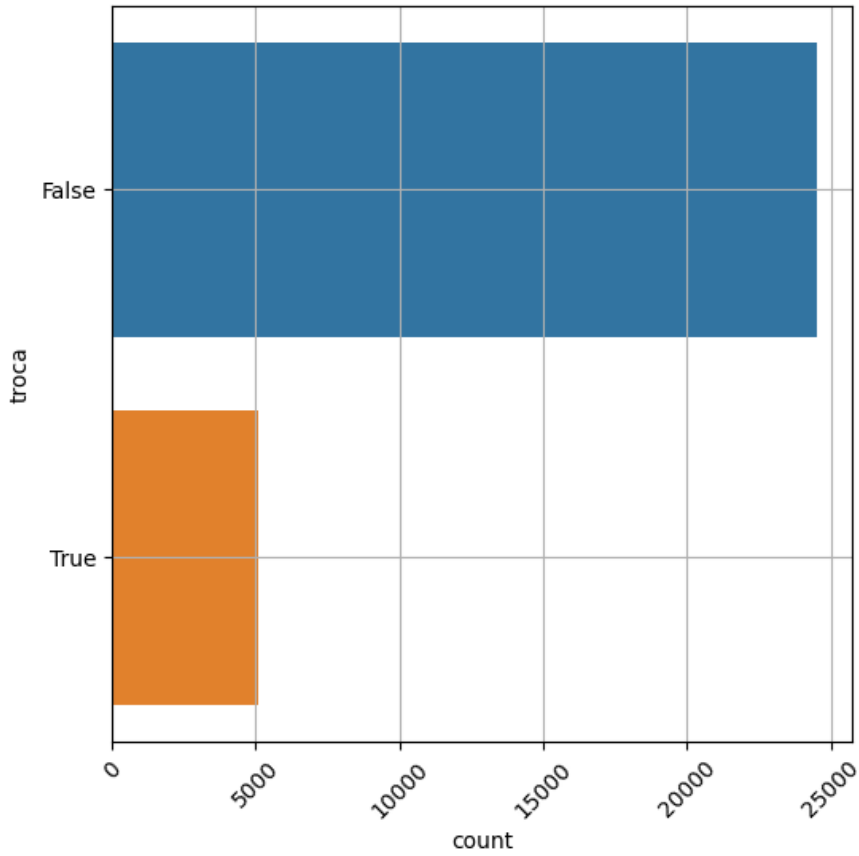
Na Figura 17 é apresentado o gráfico de barra das amostras dos dados do atributo **entrega_delivery.** Observa-se a predominância de que o veículo anunciado não ser utilizado para entrega delivery. Observa-se a existência de um grande desbalanceamento entre as duas classes deste atributo.

Figura 17 – Gráfico de Barra do atributo entrega_delivery



Na Figura 18 é apresentado o gráfico de barra das amostras dos dados do atributo **troca.** Observa-se a predominância de que o veículo anunciado não foi trocado anteriormente. Observa-se a existência de um grande desbalanceamento entre as duas classes deste atributo.

Figura 18 – Gráfico de Barra do atributo troca



NOTA: Não foram construídos os gráficos de barra das amostras dos dados dos atributos **modelo, versão e cidade_vendedor,** por conter, respectivamente, 457, 1916, 575 classes distintas. Os demais atributos não foram construídos gráficos por conter apenas um valor único nas amostras dos dados.

Através de todos os gráficos de barras construídos foi possível identificar a predominância de uma classe específica e o desbalanceamento entre as classes de cada atributo analisado.

4. Análise Exploratória de Dados (EDA) e Hipóteses de Negócio

Nesta seção apresenta-se a utilização da análise exploratória de dados para responder inicialmente três hipóteses de negócio criadas por este autor, tais como:

- Qual a cidade com o menor preço médio de venda de um veículo do tipo sedã com transmissão automática?
  - Resposta: A cidade de Registro/SP é a melhor cidade para aqueles que desejam comprar um veículo do tipo sedã com transmissão automática, em virtude de uma maior economia.

- Qual a cidade com o maior preço médio de venda entre os veículos anunciados?
  - Resposta: A cidade de Cocalzinho de Goiás/GO é a melhor cidade para aqueles que desejam vender um carro, em virtude de uma maior lucratividade.

- Qual a cidade com o menor preço médio de venda entre os veículos anunciados da marca FORD, modelo FIESTA?
  - Resposta: A cidade de Mogi das Cruzes/SP é a melhor cidade para aqueles que desejam comprar este tipo de marca/modelo, em virtude de uma maior economia.

As respostas às perguntas de negócio apresentadas no desafio:

- Qual o melhor estado cadastrado na base de dados para se vender um carro de marca popular e por quê?
  - Resposta: O estado do Mato Grosso é o melhor estado para se vender um carro da marca popular, considerando nesta análise somente o veículo da marca VOLKSWAGEM, por ter o maior preço médio de venda comparado aos demais estados.

- Qual o melhor estado para se comprar uma picape com transmissão automática e por quê?
    - Resposta: O estado da Paraíba é o melhor estado para se comprar uma picape com transmissão automática, por ter o menor preço médio de venda comparado aos demais estados.

- Qual o melhor estado para se comprar carros que ainda estejam dentro da garantia de fábrica e por quê?
    - Resposta: O estado da Paraíba é o melhor estado para se comprar um carro com garantia de fábrica, por ter o menor preço médio de venda comparado aos demais estados.

5. Explicação da Previsão do Preço a partir da Base de Dados

Para criar um modelo preditivo visando a precificação do preço de um carro com base nos atributos previsores disponíveis na base de dados de treinamento foi desenvolvido as seguintes etapas:

- Preparação dos dados;
- Seleção das *features*;
- Divisão dos dados;
- Seleção e treinamento do modelo;
- Avaliação do modelo treinado;
- Precificando com o modelo treinado

5.1 Preparação dos dados

Nesta etapa foi realizado o processo de pré-processamento dos dados, tais como a utilização de um atributo para ser configurado como índice; eliminação de atributos irrelevantes; tratamento de valores ausentes; conversões de dados dos tipos float e bool para int; eliminação de atributos categóricos com quantidade excessiva de classes, a fim de se evitar maldição da dimensionalidade, entre outros.

A partir do levantamento das informações obtidas na análise preliminar, apresentada na Seção 3.1, foram realizados as seguintes manipulações na variável **df** visando o processo de pré-processamento dos dados da base de treinamento:

- O atributo **id** foi configurado como índice da variável **df**;
- O atributo **veiculo_alienado** foi excluído da base de dados de treinamento por não conter nenhum valor presente;
- Os atributos **elegivel_revisao, dono_aceita_troca, veiculo_único_dono, revisoes_concessionaria, ipva_pago, veiculo_licenciado, garantia_de_fabrica, revisoes_dentro_agenda** também foram excluídos da base de dados por possuir valor único. Estes atributos não fornecem nenhuma informação útil para o treinamento de um modelo preditivo, por não conter nenhuma variação entre os registros;
- Os 177 valores faltantes do atributo **num_fotos** foram preenchidos com o valor mediano do atributo;
- Os atributos **num_fotos** e **ano_modelo** foram convertidos de float para valores int;
- O atributo **blindado** foi convertido do tipo object para o tipo int com valores setados em 0 e 1;
- Os atributos **entrega_delivery** e **troca** foram convertidos do tipo bool para o tipo int com valores setados em 0 e 1;

Dando continuidade ao processo de pré-processamento dos dados de treinamento foi decidido por este autor a exclusão dos seguintes atributos **modelo**, **versao** e **cidade_vendedo**r, consideradas descartadas por serem atributos categóricos uma quantidade excessiva de classes, o que contribuiria para a

maldição da dimensionalidade. O atributo numérico contínuo, **hodometro**, também foi excluído neste momento da construção da primeira versão do modelo preditivo, podendo num ciclo posterior da aplicação da metodologia CRISP-DM ser reavaliado para utilização em uma versão posterior do modelo.

O passo seguinte do processo foi realizar a separação da variável **df** em duas partes: variável **X**, representadas pelos atributos **num_fotos, marca, ano_de_fabricacao, ano_modelo, cambio, num_portas, tipo, blindado, cor, tipo_vendedor, estado_vendedor, anunciante, entrega_delivery** e **troca**; e a variável **y**, representada pelo atributo **preco**.

Para finalizar o processo de pré-processamento foi aplicado na variável **X** a codificação one-hot por meio da classe OneHotEncoder da biblioteca scikit-learn, resultando na criação da variável **X_onehot** contendo 29.584 linhas e 167 colunas.

5.2 Seleção das *features*

Nesta etapa teve como objetivo identificar as *features* mais relevantes para serem utilizadas como atributos previsores para entrada o processo de treinamento do modelo preditivo de precificação dos carros.

Considerando a elevada dimensionalidade da variável **X_onehot** para ser utilizada como entrada para o treinamento do modelo preditivo foi aplicado o método *Recursive Feature Elimination* (RFE) disponível por meio da classe RFE do scikit-learn para inicialmente identificar o número de recursos (features) a serem selecionados para o treinamento do modelo preditivo.

Mas antes disto foi aplicado a técnica *Local Outlier Factor* (LOF) para identificar e eliminar de forma automática **outliers** da base de dados de treinamento, resultando na redução da quantidade de linhas das variáveis **X** e **y**, de modo que a variável **X_onehot** passou a conter 24.260 linhas e 167 colunas e a variável **y** passou a conter 24.260 linhas e 1 coluna. Além disto, foi aplicado a normalização dos dados da variável y por meio da classe **MinMaxScaler** da biblioteca scikit-learn gerando como resultado a variável **y_norm.** A normalização da variável pode ajudar o modelo a convergir mais rapidamente e melhorar o desempenho do algoritmo.

A identificação do número de recursos foi obtido com a criação das funções get_models(num_features) e evaluate_model(model, X, y), responsáveis pela criação de um dicionário de modelos RFE e pela avaliação de cada modelo RFE por meio da validação cruzada, resultando na definição de 15 recursos entre os 167 atributos da variável **X_onehot,** que otimizam a métrica *Root Mean Squared Error* (RMSE) de forma a se obter o menor valor possível, no caso um RMSE equivalente a 0,04565.

Para finalizar esta etapa foi aplicado o método RFE para identificar entre as 167 colunas da variável **X_onehot** as 15 colunas definidas anteriormente, resultando na seleção das seguintes colunas como recursos mais relevantes para a entrada para o treinamento do modelo preditivo, a saber: [16, 25, 34, 38, 43, 44, 86, 94, 95, 96, 97, 111, 120, 122, 125]. Com a seleção destas colunas foi possível definir a variável **X_onehot_features** para ser utilizada como entrada para o treinamento do modelo.

5.3 Divisão dos Dados

Nesta etapa foi utilizado a função train_test_split da biblioteca sckit-learn para realizar a divisão das variáveis X_one_features e y_norm, resultando as variáveis **X_train, X_test, y_train_norm, y_test_norm**, como sendo o conjunto de dados de treinamento e teste do modelo. Foi adotado 80% dos dados para treinamento e 20% para teste das variáveis **X_one_features** e **y_norm.**

5.4 Seleção e Treinamento do Modelo

Esta etapa teve como objetivo definir o modelo a ser utilizado no processo de treinamento utilizando-se a features selecionadas definidas na variável **X_onehot_features** e separada nas variáveis **X_train** e **X_test.**

Nesta etapa foi utilizado o AutoKeras, que é uma implementação do AutoML[1] para modelos de aprendizado profundo que usa pesquisa de arquitetura neural. A

1 O termo AutoML refere-se a técnicas para descobrir automaticamente o modelo de melhor desempenho para um determinado conjunto de dados.

pesquisa é realizada usando os modelos Keras por meio da API TensorFlow tf.keras.

O AutoKeras é uma biblioteca que usa redes neurais para resolver tarefas de aprendizado de máquina automatizado. Ela oferece suporte para uma ampla gama de tarefas de modelagem preditiva, como problemas de regressão e classificação, onde as arquiteturas da rede são otimizadas automaticamente.

Esta biblioteca fornece uma abordagem simples e eficaz para encontrar automaticamente modelos de alto desempenho.

O *AutoKeras* foi aplicado para a tarefa de regressão usando a classe *StructuredDataRegressor* e configurado para o número de modelos a serem testados. Para o treinamento foi setado um total de 10 modelos.

Posteriormente foi realizado o treinamento fornecer as entradas das variáveis **X_train** e **y_train_norm.**

No final do processo de treinamento são gerados os modelos e os resultados são salvos na pasta denominada **model_auto** no diretório de trabalho atual.

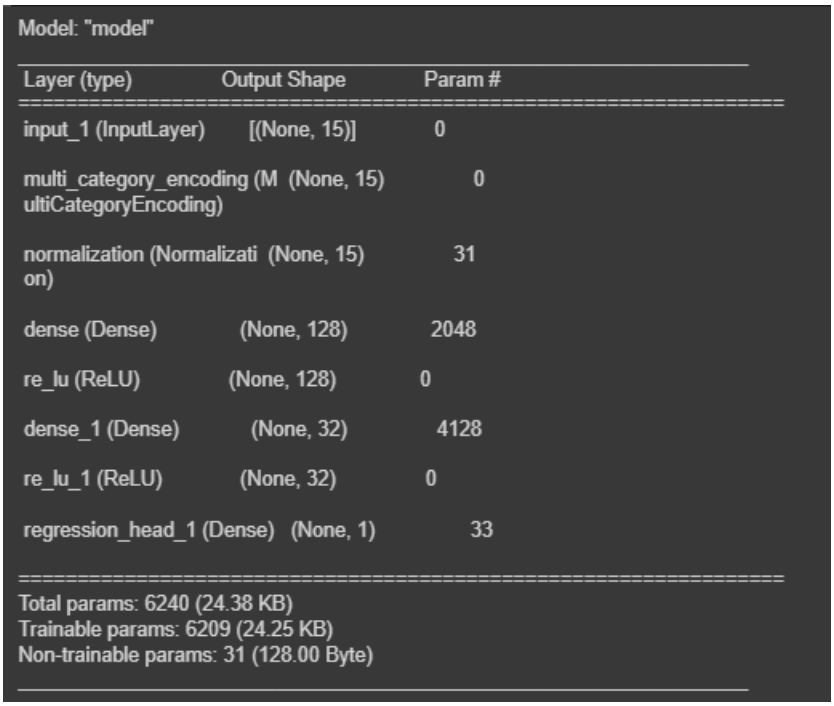
5.5 Avaliação do Modelo Treinado

Para avaliar o desempenho do modelo treinado foi definido a métrica RMSE. Esta é uma métrica normalmente utilizada para avaliação de um modelo de regressão, tal como o modelo de precificação para a venda de veículos. Esta métrica ajudará a verificar o quão bem o modelo treinado está prevendo os preços dos veículos. Nesta etapa foi utilizado o método evaluate da classe *StructuredDataRegressor* ao fornecer como entradas as variáveis **X_test** e **y_test_norm,** que retorna como resultado o valor da métrica *Mean Squared Error* (MSE), que aplicando a raiz quadrada no valor de MSE obtém-se a métrica RMSE, cujo valor obtido foi 0.044925.

5.6 Precificando com o Modelo Treinado

Usando o método export_model() da classe *StructuredDataRegressor* obtém-se o modelo de melhor desempenho obtido resultando na variável **model**.

O resumo deste modelo pode ser obtido com a função summary da variável **model**, gerando o seguinte resultado:



O modelo pode ser salvo em arquivo através da função save da variável **model** resultando no arquivo batizado por este autor com o nome **model_auto.h5**.

Este foi o modelo utilizado para realizar as previsões dos preços de vendas dos veículos fornecidos no arquivo **cars_test.csv.**

6. Link

O desafio cientista de dados foi desenvolvido utilizando a linguagem python no ambiente de desenvolvimento computacional Google Colab. O repositório público utilizado para registrar e documentar os artefatos produzidos encontra-se disponível no link https://github.com/ubiratantavares/desafio_cientista_de_dados.