**Data Analysis Report on**

**PRCP- 1017-AutoPricePred**

**BY:** Sameera Tanveer,

Arunmozhi Muthamizhan

Date: 11-11-2024

Table of Contents:

1. Introduction
2. Data Preprocessing & Feature Engineering
3. Modelling
4. Conclusion
5. Introduction

Purpose of the report:

The purpose of the report is to give you the complete data analysis report on the given data.

Overview of the dataset:

* The dataset is provided on the concept of Car Price Prediction which is a Regression problem. The dataset contains all the information that are required to predict the car price at the end given the information of the characteristics of the Car as mentioned in the dataset.
* The domain of the dataset is “Car Industry”
* The dataset contains 201 records and 26 columns (attributes/characteristics).
* The target column is a “price” column which a regression or Quantitative Continuous type of data.
* This makes this problem to be solved as a Regression problem narrowing down the possible algorithms or ML models applicable to solve this problem.

Objectives:

The main objective task is to get the best model for the production which predicts the ‘Car’ price given all the information or the values of Attributes.

2. Data Preprocessing & Feature Engineering.

1. **Adding Column Names to the Data Frame:** The dataset did not have Column Names. The column names were provided in the text file along with the metadata of the attributes. The first step was to set the column names of the dataframe.

2. **Handling Missing Values:** This dataset was a little bit complex in terms of the Missing Values, as the Missing values in the dataset are under hood disguised as ‘?’. It was a little hustle to observed and find this pattern.

To avoid the same confusion, we had converted all the ‘?’ to np.nan.

As this dataset was enriched with missing values, we have opted for Iterative Imputer technique to impute all the missing values of the dataset, making it robust and less biased dataset.

3. **Encoding all the Categorical Columns:** This dataset was as well enriched with Categorical data, which made it good with CatBoost ML Algorithm, but it we want to train the data with other algorithms, we need to encode all these columns.

We have used replace() method to encode all the categorical columns.

4. **Correct the Data Type of the columns:** Due to ‘?’ disguised as Missing Values, we have noticed that some of the numerical columns are data typed as Categorical which we have find them and converted their data type for ensuring the structure and correct data for modelling.

5. **Dealing with the Outliers:** We have used the Visualization technique of Box Plots to detect the possible candidates of Outliers.

But we have not found the extensive amounts of outliers that can be a hurdle within Model training. We have manually verified the Outlier candidates. With the domain knowledge, we had decided that these are not heavy weighted outliers and are less prone to faults.

6. **Correlation with Target Column(‘price’):** Using the corr() function we have particularly find the correlation of each independent features (input columns) to the dependent feature (Output column). And We have decided that all the columns are pretty much contributing to the prediction of the Price.

For further understanding of the correlation, we have used the heatmap() to understand it further and to find the multicollinearity as well.

7. **Feature Scaling:** Before feeding the data to the model, we have applied MinMax Scaler on the dataset, because we have observed that the data is very much skewed and is Non-normal distributed.

Hence, we decided to apply the scaling on the dataset using MinMax Scaler.

Feature scaling has been done after the splitting of the data, to avoid any data leakage which can later on lead to overfitting and wrong high-performance model that perform well on the test data, but not very well on the actual unseen data from the real world.

Modelling

For this specific dataset, by considering some facts about the data, we have decided to use all the possible algorithms that are suited to deal with regression problems.

1. **Linear Regression:** We have first tested the performance of the training model using Linear regression

After the training and prediction, we have calculated the mean-squared-error and r2-score of the model.

We have achieved the r2-score = 91.5%

1. **Lasso Regression:** After seeing the performance of the Linear Regression, we decided to try out using the Regularization techniques. We have used Lasso Regression for training the model.

After the training and prediction, we have calculated the mean-squared-error and r2-score of the model.

We have achieved the r2-score = 91.4%

1. **Ridge Regression:** After seeing the performance of the Linear Regression, we decided to try out using the Regularization techniques. We have used Ridge Regression for training the model.

After the training and prediction, we have calculated the mean-squared-error and r2-score of the model.

We have achieved the r2-score = 89.2%

1. **Elastic Net Regression**: We decided to try out the reliable regularization technique.

After the training and prediction, we have calculated the mean-squared-error and r2-score of the model. We have achieved the r2-score = 31.9%

1. **Random Forest Regressor**: After seeing the performance of the normal traditional ML algorithms, we have decided to try on with some Ensemble Algorithms. We started with Random Forest Regressor.

After the training and prediction, we have calculated the mean-squared-error and r2-score of the model.

We have achieved the r2-score = 93.4%

1. **XG Boost Regressor**: After seeing the performance of RFR, we have decided to try on with XG Boost Regressor.

After the training and prediction, we have calculated the mean-squared-error and r2-score of the model.

We have achieved the r2-score = 95.6%

1. **CatBoost Regressor:** As the dataset originally had a lot of Categorical columns, we tried using Cat Boost regression to test, if cat boost can capture all the patterns of the dataset.

After the training and prediction, we have calculated the mean-squared-error and r2-score of the model.

We have achieved the r2-score = 93.1%

1. **LGBM Regressor**: We then decided to try out with LGBM as well.

After the training and prediction, we have calculated the mean-squared-error and r2-score of the model.

We have achieved the r2-score = 91.9%

1. **AdaBoost Regressor:** At last, we decided to try out the last possible perfect model candidate for training, we have tried AdaBoost Regressor on the training dataset.

After the training and prediction, we have calculated the mean-squared-error and r2-score of the model.

We have achieved the r2-score = 92.7%

After trying out all the Possible ML candidate algorithms, we have found that “XG Boost Algorithm” yields the best results and this make it the best model to consider for this specific problem.

**Hyper-Parameter Tuning**:

After the normal parameter training of XG Boost, we have decided to improve the model performance by performing Hyper-parameter tuning on the XG Boost.

We have used, Grid Search, as the dataset to be consider was pretty small.

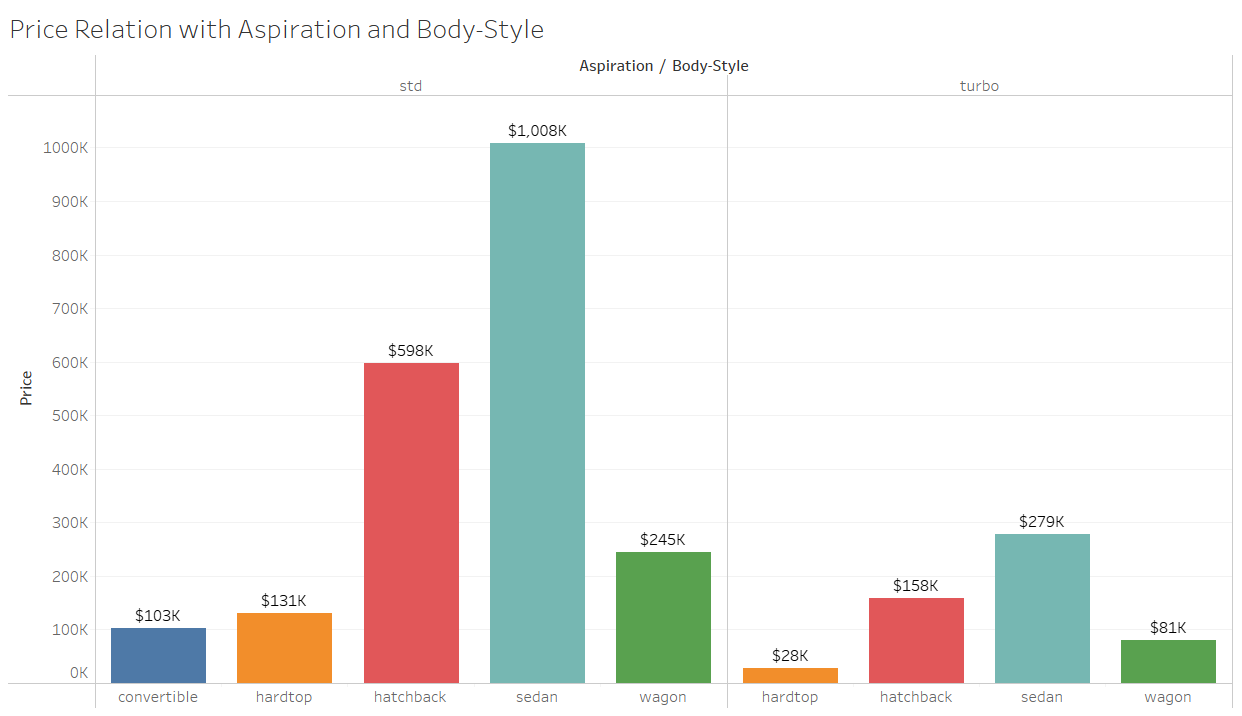
We have applied the Grid Search.

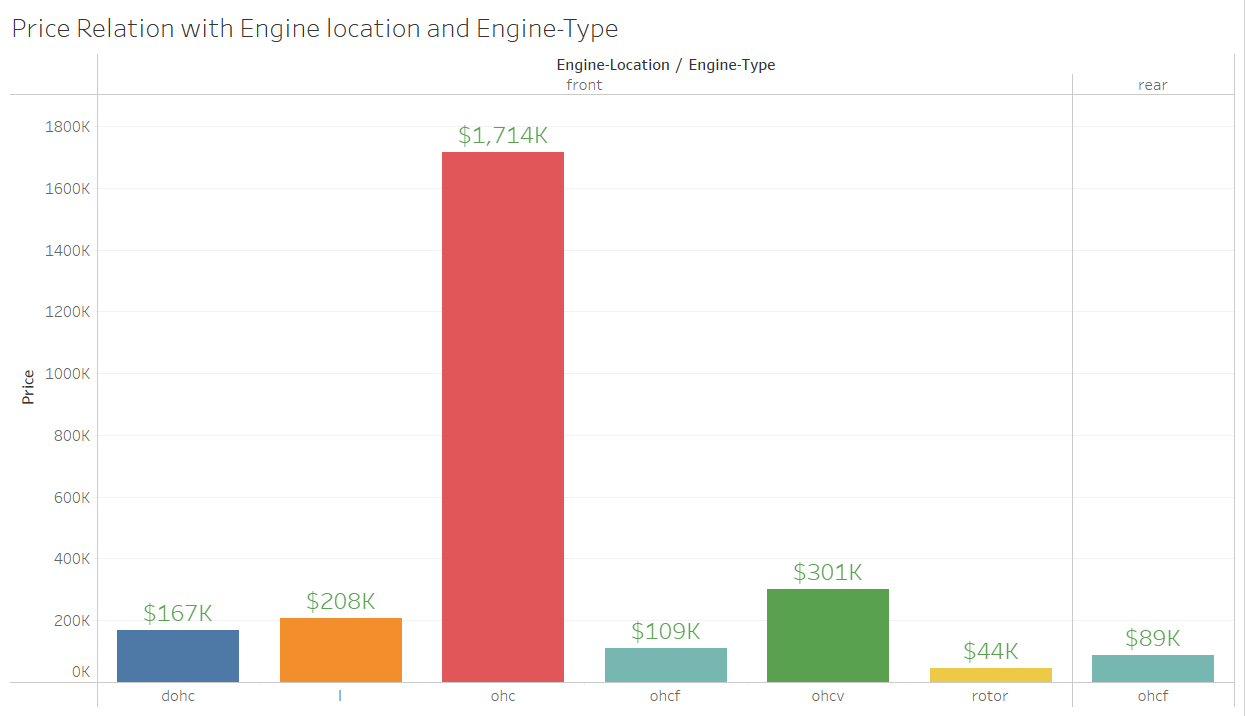
But unfortunately, we have achieved the same prediction or r2-score after the tuning as well.

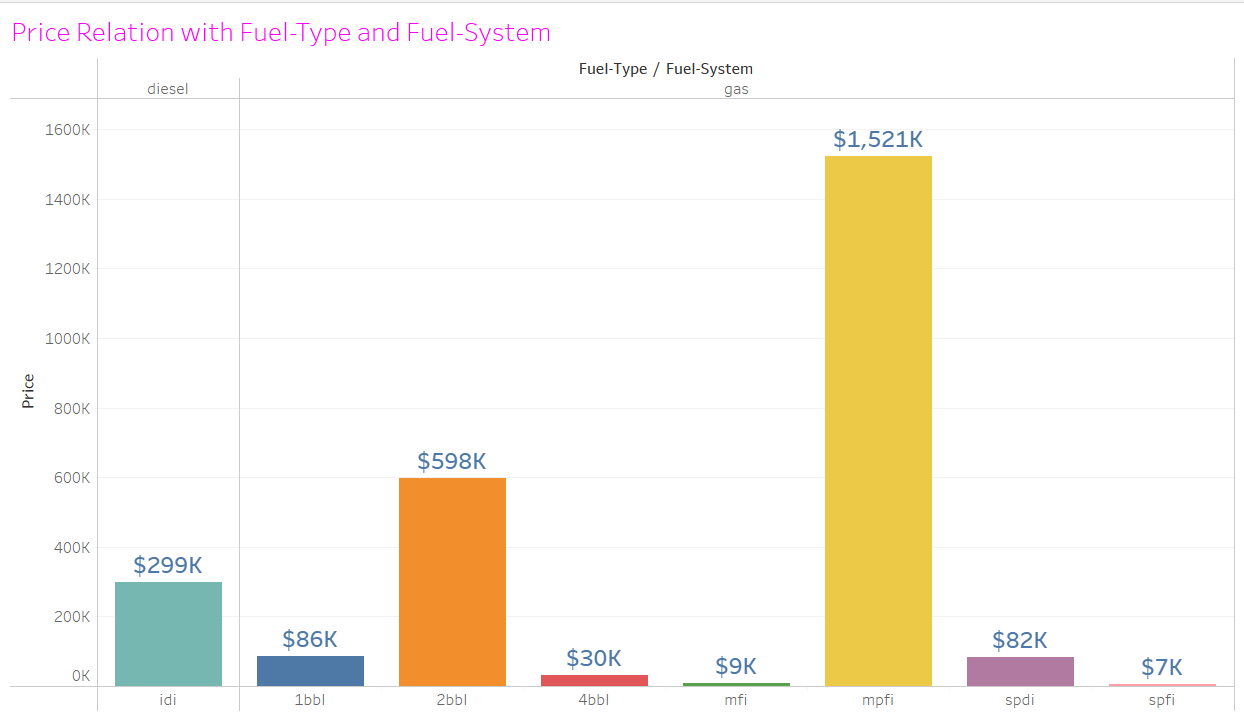
**Saving the Model and Variables:**

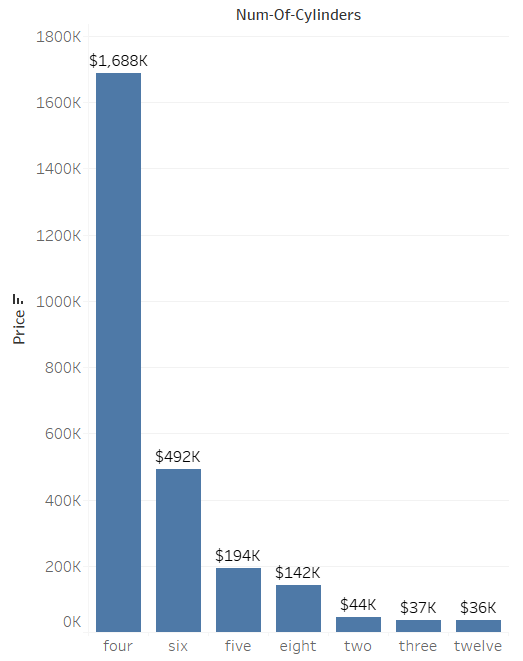
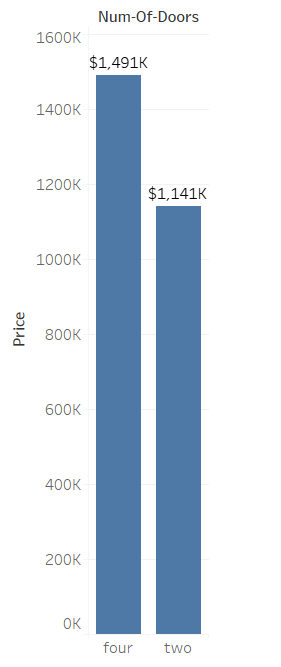
1. After all of these, we have saved our best model and some important variables using Joblib which reduces our task of rerunning all the cells again.
2. We have opted this approach as a solution for our issue of losing our work on the daily basis.

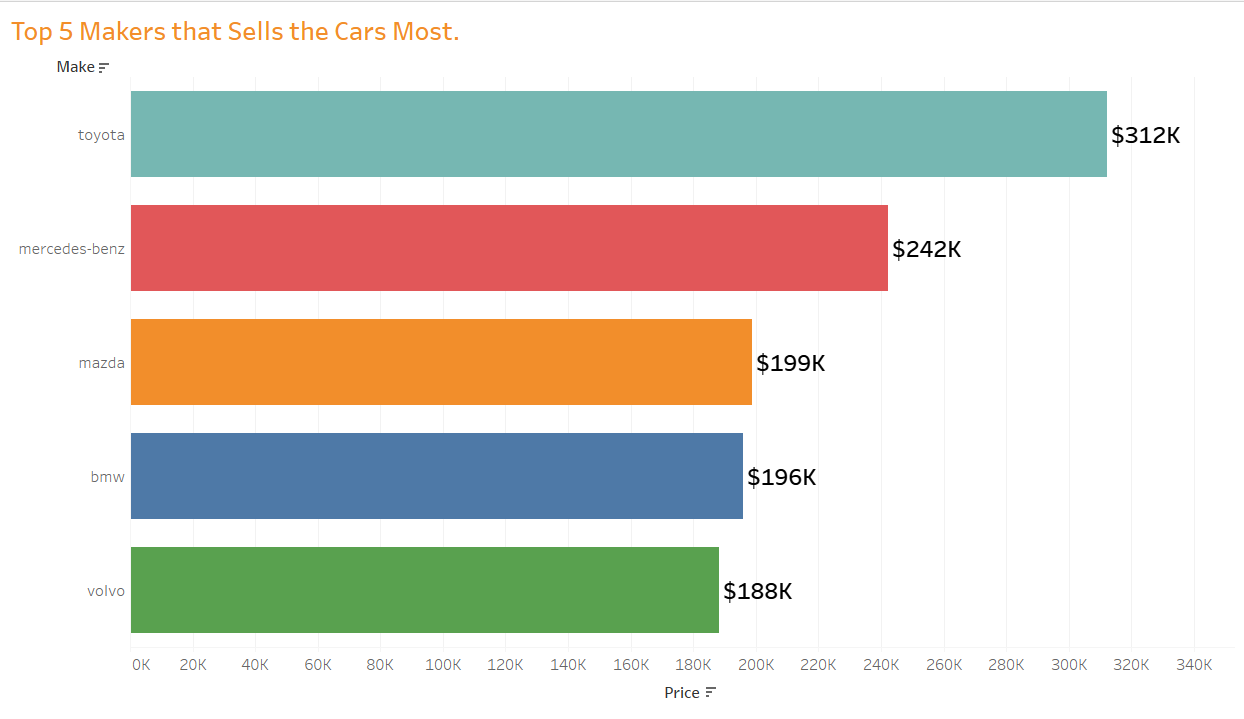
**Visualizations and Insights:**

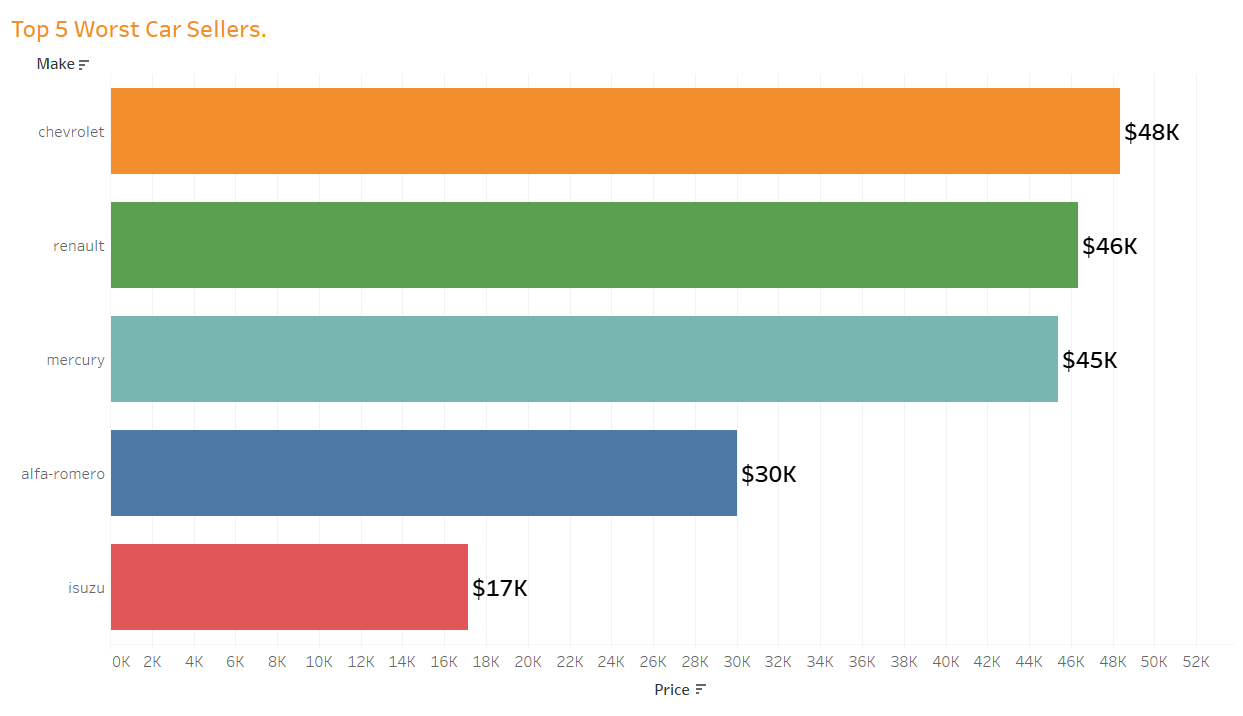
****

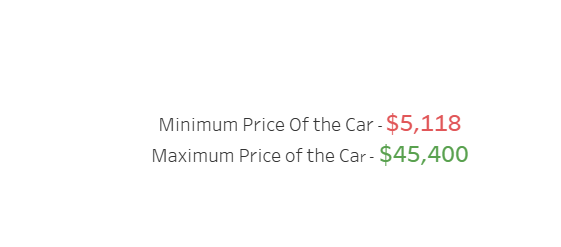










**Report on Challenges faced:**

1. Columns Names was in the other file.
2. Missing Values were disguised as ‘?’.
3. Too many Categorical Columns to encode.
4. Very small dataset to feed the model for training.

**Conclusion:**

We have successfully, solved the problem by creating the best predictive model for predicting Car Price.