



Project Report on

“Used Car Price Prediction”



Name : Ravisankar P K



Post : Intern

ACKNOWLEDGMENT

I would like to express my sincere thanks of gratitude to my SME as well as “Flip Robo Technologies” team for letting me work on “Used Car Price Prediction” project also huge thanks to my academic team “DataTrained”. Their suggestions and directions have helped me in the completion of this project successfully. This project also helped me in doing lots of research wherein I came to know about so many new things.

All the required information and dataset are provided by Flip Robo Technologies (Bangalore) that helped me to complete the project.

1. INTRODUCTION

1.1 Business Problem Framing

With the covid 19 impact in the market, we have seen lot of changes in the car market. Now some cars are in demand hence making them costly and some are not in demand hence cheaper. One of our clients works with small traders, who sell used cars. With the change in market due to covid 19 impact, our client is facing problems with their previous car price valuation machine learning models. So, they are looking for new machine learning models from new data. We have to make car price valuation model.

This project contains two phases:

Data Collection Phase

In this section we need to scrape the data of used cars from websites. We need web scraping for this. we have to fetch data for different locations. The number of columns for data doesn't have limit. Generally, these columns are Brand, model, variant, manufacturing year, driven kilometers, fuel, number of owners, location and at last target variable Price of the car. This data is to give you a hint about important variables in used car model.

We have collected the dataset from the website www.cardekho.com which is a

web platform where seller can sell their used car. The data is scrapped using Web scraping technique and the framework used is Selenium. We scrapped nearly 5960 of the data and fetched the data for different locations and collected the information of different features of the car and saved the collected data in excel format. The dimension of the dataset is 5960 rows and 20 columns including target variable "Car_Price". The particular dataset contains both categorical and numerical data type. The data description is as follows:

Data Collection Phase

Car_Name	Name of the cars with manufacturing year
Fuel_type	Type of fuel used for car engine
Kilometers	Car running in kms till the date
Engine_disp	Engine displacement/engine CC
Gear_transmission	Type of gear transmission used in car
Milage_in_km/ltr	Overall milage of car in Km/ltr
Seating_cap	Number of seats available in the car
color	Color of the car

Max_power	Maximum power of engine used in car in bhp
front_brake_type	Type of brake system used for front-side wheels
rear_brake_type	Type of brake system used for back-side wheels
cargo_volume	Total cubic feet of space in a car's cargo area
height	Total height of car in mm
width	Width of car in mm
length	Total length of the car in mm
Weight	Gross weight of the car in kg
Insp_score	Inspection rating out of 10
top_speed	Maximum speed limit of the car in km per hours
City_url	Url of the page of cars from a particular city/location
Car_price	Price of the car

Model Building Phase

After collecting the data, you need to build a machine learning model.

Before model building do all data pre-processing steps. Try different models with different hyper parameters and select the best model.

- Data Cleaning
 - Exploratory Data Analysis
- Data Pre-processing
- Model Building
- Model Evaluation
- Hyper parameter Turning
- Selecting the best model

Data Pre-processing Done

Data pre-processing is the process of converting raw data into a well-readable format to be used by Machine Learning model. Data pre-processing is an integral step in Machine Learning as the quality of data and the useful information that can be derived from it directly affects the ability of our model to learn; therefore, it is extremely important that we pre-process our data before feeding it into our model.

We used following pre-processing steps:

- Importing necessary libraries and loading collected dataset as a data

frame.

- Checked some statistical information like shape, number of unique values present, info, unique() data types etc.
- From the dataset I found some numerical features having “-” sign and string value like “null” so I replaced them with NAN values and dropped
- The columns having more than 50% of “-” sign as they were of no use for prediction.
- Done feature engineering on some features as they had some irrelevant values like kms, kmpl and replaced them by empty space.
- Extracted the features Brand, Model and Manufacturing_year from the column Car_Name and created Car_age by subtracting the Manufacturing year of car from the year 2021. Replaced string values and “-” sign by empty space in some of the columns to get the numerical data.
- The target variable "Car_price" should be continuous data but due to some string values it was showing as object data type. So, I replaced those entries with appropriate values. Then split the column into two as price_a and price_b and stored numerical values in price_a column and string values in price_b column and after that multiplied those two columns to get exact car price in numerical format.
- The column City_url contained the urls of the cities, so I created a

new column as Location by replacing the urls by specific city name.

➤ The columns "front_brake_type" and "rear_brake_type" were had some duplicate entries that is the entries belongs to same categories so, I replaced/grouped the same categories by appropriate values.

➤ Converted all the numerical continuous columns from object data type into float data type after cleaning the data and saved the cleaned data in excel file format.

➤ Checked for null values and treated them using imputation techniques like mean, median and mode methods. Checked the statistical summary of the dataset using describe () method. Separated both numerical and categorical columns for further process.

➤ Performed univariate, bivariate and multivariate analysis to visualize the data. Visualized each feature using seaborn and matplotlib libraries by plotting several categorical and numerical plots like pie plot, count plot, bar plot, reg plot, strip plot, line plot, violin plot, distribution plot, box plots and pair plot.

➤ Identified outliers using box plots and removed outliers in continuous numerical columns using Zscore and stored the data frame after removing outliers as "new_df".

➤ Checked for skewness and removed skewness in numerical columns using power transformation method (yeo-johnson).

➤ Encoded the columns having object data type using Label Encoder

method. Used Pearson's correlation coefficient to check the correlation between label and features. With the help of heatmap and correlation bar graph was able to understand the Feature vs Label relativity and insights on multicollinearity among the feature columns.

- Separated feature and label data and feature scaling is performed using Standard Scaler method to avoid any kind of data biasness.
- As we noticed multicollinearity we used to find Variance Inflation Factor (VIF) values and dropped the column "Max_power" as it was containing VIF above 10 and got rid of multicollinearity issue.

Data Inputs- Logic- Output Relationships

The dataset consists of label and features. The features are independent and label is dependent as the values of our independent variables changes as our label varies.

- Since we had both numerical and categorical columns, I checked the distribution of skewness using dist plots for numerical features and checked the counts using count plots & pie plots for categorical features as a part of univariate analysis.
- To analyse the relation between features and label We used many plotting techniques where I found numerical continuous variables having

strong relation with label Car_Price with the help of reg plot.

- We checked the correlation between the label and features using heat map and bar plot. Where I got both positive and negative correlation between the label and features.

Hardware & Software Requirements & Tools Used

To build the machine learning projects it is important to have the following hardware and software requirements and tools.

Hardware required:

- Processor: core i5 or above
- RAM: 8 GB or above
- ROM/SSD: 250 GB or above

Software required:

- Distribution: Anaconda Navigator
- Programming language: Python
- Browser based language shell: Jupyter Notebook
- Selenium
- Chrome: To scrape the data

MODEL/S DEVELOPMENT AND EVALUATION

Identification of possible Problem-solving approaches

(Methods):

We have used both statistical and analytical approaches to solve the problem which mainly includes the pre-processing of the data also used EDA techniques and heat map to check the correlation of independent and dependent features. Treated null values using imputation methods. Removed outliers and skewness using Zscore and yeo-johnson methods respectively. Encoded data using Label Encoder. Also, before building the model, I made sure that the input data is cleaned and scaled before it was fed into the machine learning models. Checked for the best random state to be used on our Regression Machine Learning model pertaining to the feature importance details. Finally created multiple regression models along with evaluation metrics.

For this particular project we need to predict price of used cars. In this dataset, Car_Price is the target variable, which means our target column is continuous in nature so this is a regression problem. We used many regression algorithms and predicted the car price. By doing various evaluations we have selected GradientBoostingRegressor as our best suitable algorithm to create our final model as it is giving least difference in R2 score and cross validation score among all the algorithms used. In order to get good performance and to check whether my model getting

over-fitting and under-fitting We made use of the K-Fold cross validation by setting cv=5 and then hyper parameter tuning on best model. Then I saved my final model and loaded the same for predictions.

3.2 Testing of Identified Approaches (Algorithms)

Since “Car_Price” is my target variable which is continuous in nature, from this we can conclude that it is a regression type problem hence have used following regression algorithms. After the pre-processing and data cleaning we left with 18 columns including target and with the help of feature importance bar graph we used these independent features for model building and prediction. We have used 7 regression algorithms after choosing random state among 1-200 number. We have used Random Forest Regressor to find best random state The algorithms used on training the data are as follows:

1. Decision Tree Regressor
2. Random Forest Regressor
3. Extra Trees Regressor
4. Gradient Boosting Regressor
5. Extreme Gradient Boosting Regressor (XGB)
6. Bagging Regressor
7. KNeighbors Regressor (KNN)

Model Selection:

Model	R2 Score	Cross_Validation_ Score	Difference
Decision Tree Regressor	88.10	81.09	7.01
Random Forest Regressor	96.5	90.88	5.62
Extra Trees Regressor	95.4	91.33	4.07
Gradient Boosting Regressor	92.98	89.39	3.59
XGB Regressor	96.09	91.06	5.03
Bagging Regressor	95.5	88.5	7
K Neighbors Regressor	88.4	83.75	4.65

From the difference between R2 score and Cross Validation score, it can be seen that the Gradient Boosting Regressor has least difference and low evaluation metrics compared to other models. So, we can conclude that Gradient Boosting Regressor as our best fitting model. Let's try to increase our model score by tuning the best model using different types of hyper parameters.

Key Evaluation Tools

The essential step in any machine learning model is to evaluate the accuracy and determine the metrics error of the model. We have used Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE) and R2 Score metrics for my model evaluation:

❖ **Mean Absolute Error (MAE):** MAE is a popular error metric for regression problems which gives magnitude of absolute difference between actual and predicted values. The MAE can be calculated as follows:

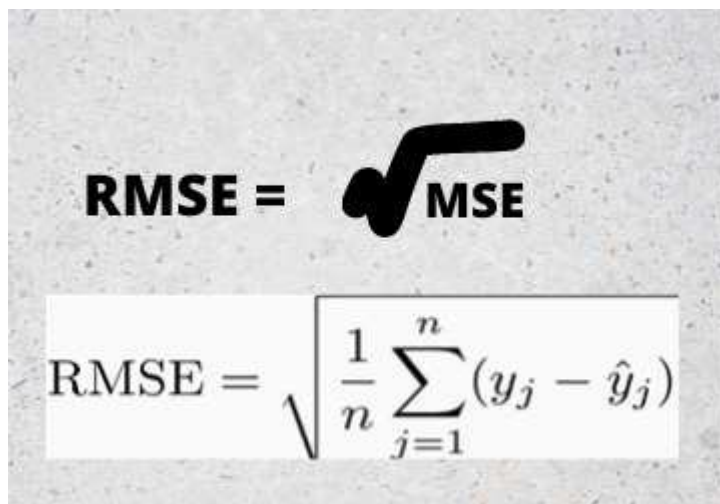
The diagram illustrates the formula for Mean Absolute Error (MAE). The formula is $MAE = \frac{1}{N} \sum |Y - \hat{Y}|$. Annotations include: 'Divide by total Number of Data Points' pointing to $\frac{1}{N}$; 'Actual Output' pointing to Y ; 'Predicted Output' pointing to \hat{Y} ; 'Sum Of' pointing to the summation symbol \sum ; and 'Absolute Value of residual' pointing to the absolute value bars $|Y - \hat{Y}|$. A yellow bracket underlines the term $|Y - \hat{Y}|$.

❖ **Mean Squared Error (MSE):** MSE is a most used and very simple metric with a little bit of change in mean absolute error. Mean squared error states that finding the squared difference between actual and predicted value. We perform squared to avoid the cancellation of

negative terms and it is the benefit of MSE.

$$MSE = \frac{1}{n} \sum \left(\underbrace{y - \hat{y}}_{\substack{\text{The square of the difference} \\ \text{between actual and} \\ \text{predicted}}} \right)^2$$

❖ **Root Mean Squared Error (RMSE):** RMSE is an extension of the mean squared error. The square root of the error is calculated, which means that the units of the RMSE are the same as the original units of the target value that is being predicted.



The image shows a hand-drawn formula for RMSE on a textured background. At the top, it says 'RMSE = √ MSE' with a large, bold square root symbol. Below this, a white rectangular box contains the mathematical formula:
$$RMSE = \sqrt{\frac{1}{n} \sum_{j=1}^n (y_j - \hat{y}_j)^2}$$

❖ **R2 Score:** We used R2 score which gives the accurate value for the models used. On the basis of R2 score We created final model

Visualizations

We used pandas profiling to get the over viewed visualization on the pre-processed data. Pandas is an open-source Python module with which we can do an exploratory data analysis to get detailed description of the features and it helps in visualizing and understanding the distribution of each variable. We have analysed the data using univariate, bivariate and multivariate analysis. In univariate analysis We have used distribution plot, pie plot and count plot and in bivariate analysis We used bar plots and strip plots to get the relation between categorical variable and target column Car price and used line plot, reg plots to understand the relation between continuous numerical variables and target variable. Apart from these plots We used violin plot, pair plot (multivariate analysis) and box plots to get the insight from the features.

Univariate Analysis: Univariate analysis is the simplest way to analyse data. “Uni” means one and this means that the data has only one kind of variable. The major reason for univariate analysis is to use the data to describe. The analysis will take data, summarise it, and then find some pattern in the data. Mainly we will get the counts of the values present in the features.

Bivariate Analysis: Bivariate analysis is one of the statistical analysis where two variables are observed, Bi means two variables. One variable here is dependent while the other is independent. These variables are usually denoted by X and Y. We can also analyse the data using both independent variables. Bivariate analysis is finding some kind of empirical relationship between two variables. In this study we will be showing the relation between dependent (target) and independent variables (features) using different plotting techniques.

CONCLUSION

Key Findings and Conclusions of the Study

The case study aims to give an idea of applying Machine Learning algorithms to predict the sale price of the used cars. After the completion of this project, we got an insight of how to collect data, pre-processing the data, analysing the data, cleaning the data and building a model.

In this study, we have used multiple machine learning models to predict the sale price of the used cars. We have gone through the data analysis by performing feature engineering, finding the relation between features and label through visualizations. And got the important feature and we used these features to predict the car price by building ML models. After training the model we checked CV score to overcome with the overfitting

issue. Performed hyper parameter tuning on the best model and the best model's R2 score increased and was giving R2 score as 96.90%. We have also got good prediction results of car price.

Findings: From the whole study we got to know that the continuous numerical variables having some strong positive linear relation with the label "Car_Price". By comparing car price and categorical variables we got to know that the cars having automatic gear transmission, cars from the city Bangalore, cars using petrol and deisel as fuels, cars having the brands Benz and BMW and cars with 5-7 seating capacity have high sale price. While comparing continuous numerical variables and Car_Price we found that cars which are having good milage, engine displacement, less running in kms have good linear relation with the price that is the cars with this kind of qualities have high selling prices. We found outliers and removed them and further removed skewness. Looking at the heat map, I could see there were some features which were correlated with each other, so I used VIF method to remove the feature causing multicollinearity and scaled the data to overcome with the data biasness.

Learning Outcomes of the Study in respect of Data Science

While working on this project I learned many things about the features of cars and about the car selling web platforms and got the idea that how the machine learning models have helped to predict the price of

used cars which provides greater understanding into the many causes and benefits of selling old cars. I found that the project was quite interesting as the dataset contains several types of data. I used several types of plotting to visualize the relation between target and features. This graphical representation helped me to understand which features are important and how these features describe selling price of the old cars. Data cleaning was one of the important and crucial things in this project where I dealt with features having string values, features extraction and selection. Finally got XGBoosting Regressor as best model. The challenges I faced while working on this project was when I was scrapping the real time data from cardekho website, it took so much time to gather data. The data was quite difficult to handle and cleaning part was challenging for me but fixed it well and it was unable to remove skewness in Seating_cap column so moved further keeping it as it is. Finally, our aim was achieved by predicting the sale price of used cars and built car price evaluation model that could help the clients to understand the future price of used cars.

Limitations of this work and scope for future work

Limitations: The main limitation of this study is the low number of records that have been used. In the dataset our data is not properly distributed in some of the columns many of the values in the columns are “-” and some values which are not realistic. Because I have seen the

column Running in kms showing 0 kms and some of the cars having age as 0 years which are not possible in case of used cars. So, because of that data our models may not make the right patterns and the performance of the model also reduces. So that issues need to be taken care.

Future work: As future work, we intend to collect more data and to use more advanced techniques like artificial neural networks and genetic algorithms to predict car prices. In future this machine learning model may bind with various website which can provide real time data for price prediction. Also, we may add large historical data of car price which can help to improve accuracy of the machine learning model.