CAR PROJECT

Submitted by:

Leena Chatterjee

ACKNOWLEDGMENT

I wish to express my sincere gratitude to Data trained Academy and FlipRobo Technologies who gave me the opportunity to do this Project. It helped me to do a lot of research and I have grasped many new things. The data source is internet, we fetched data through scrapping. The training I have taken from Data trained, Kaggle and GitHub helped me to complete this project

INTRODUCTION

Business Problem Framing:

CARS24 is an online used car marketplace that is headquartered in Gurugram, Haryana, India.[2] The company sells more than 1,50,000 cars annually.[3] The company is considered among the four major organised players in the used car segment in India.

Cars24 was founded in 2015 by Vikram Chopra, Mehul Agrawal, Gajendra Jangid and Ruchit Agarwal as a platform to buy and sell used cars. In 2021, the company expanded operations internationally in several countries, including the United Arab Emirates and Australia.

The Cars24 platform facilitates the transaction and has an offline presence. Apart from selling used cars, the company's services include paperwork such as transferring the car to the name of the new owner which enables end-to-end transactions and offers an online auction platform to businesses looking to sell their pre-owned cars. In 2019 the company started offering verified used cars where the company offered a buyback guarantee on the vehicles verified by inspection.

The company operates 202 branches across 73 cities in India as of 2019. Apart from its own branches, the company has a tie up with more than 10,000 channel partners across 230 cities in India.

In May 2020 the company launched Cars24 Moto. Cars24 Moto is a service which allows customers to sell used two wheelers such as motorbikes, mopeds and scooters on its platform. It also launched a service offering vehicles inspection services at the customers location in place of their branch

Here we have fetched used car data for buying a car and trying to predicting the price based on car history(accidental or non-accidental), monthly Emi ,fueltype

Conceptual Background of the Domain Problem:

Data science comes as a very important tool to solve problems in the domain to help the companies increase their overall revenue, profits, improving their marketing strategies and focusing on changing trends in house sales and purchases.

Goals of the Study:

The main objectives of this study are as follows:

- To apply data pre-processing and preparation techniques in order to obtain clean data
- To build machine learning models able to car price based on features
- To analyse and compare models performance in order to choose the best model

Review of Literature

Trends in car prices indicate the current economic situation and also are a concern to the buyers. There are many factors that have an impact on car prices, such as Emi and fuel type. A house with nonaccidental, high milage would have a greater price as compared to a used with no such accessibility. Predicting car prices manually are a difficult task and generally not very accurate, hence there are many systems developed for price prediction. The variety of approaches either consider the entire data for modelling, or split the entire data and model each partition independently for the choice of prediction model. There were may not sufficient training samples per partition for the latter approach. So, such modelling ignores the relatedness between partitions, and for all prediction scenarios.

Motivation for the Problem Undertaken:

Learning the theoretical background for data science or machine learning are often a frightening experience, because it involves multiple fields of mathematics and an extended list of online resources. By proper practical research and practice I can become better in this field. These suggestions are derived from my mentors/SME's and my own experience in the beginner projects.

Analytical Problem Framing

Mathematical/ Analytical Modelling of the Problem

- In the project, we need to predict the price of the car using the provided features in the dataset.

- In the project we have used regression as price is continuous, while feting data it came along with ~rupee sign so python assumed it object column but it is continuous we have performed here regression not categorial
-
- There are outliers in the dataset. But all the column being object so we did not remove skew or outlier

- Describe method is used to display all the summary statistics of the dataset. In Model building we split the dataset into train and test data and is scaled. The scaled data is passed into the model for training and results are predicted. The performance of the model is found by error metrics like r2 score, mae, mse.

Data Sources and their formats:

The Data source refers to Housing Project. The dataset is in a csv(comma separated values) format. The whole dataset contains 500 rows and 10 columns. The following are the features with their description in detail:

df.columns:

=➔Index(['accidental_history', 'car_model', 'ownership', 'Milage',
'Manufacture_year', 'fuel_type', 'Emi', 'Price', 'car_name', 'Model'],
dtype='object')

1.'accidental_history'-For buyers, one of the most significant benefits is the lower retail price. Used cars that have been in an accident are, on average, 60% of the price of undamaged cars, even if the repairs are flawless . ... Because an accident history impacts vehicle value, **a used car in bad condition is no longer considered an asset**.

2.'car_model'- Some people so passionate about cars, so model play a vital role here

3.Ownership- Does multiple owners mean the car is in bad shape? **No**. ... No different than when people trade their car right before its about to run out of warranty because they do not want to be the ones hit with the big expense.

Milage -A number of miles travelled or covered. Mileage indicates the distance that a vehicle can travel with a specific amount of fuel. The car that can travel a good distance with just a small amount of fuel is stated as a vehicle with good mileage or high fuel efficiency, which also means that the owner of that car need not spend much money on expensive

'Manufacture_year'--On the driver's side door, **there is usually a sticker placed by the manufacturer in between the door and the doorjamb**, which contains a lot important information about your vehicle. Usually on this sticker, it will list the date of manufacture, or production date. User always prefer recently means not too old manufacture year

fuel_type'-**Diesel cars** last longer and their value belittles relatively slower than the petrol variants, which makes it a preferable choice for the commuters needing to cover long distances. Diesel cars pull off better mileage when compared to petrol cars.

Emi- Equated Monthly Instalment (or EMI) consists of the principal portion of the loan amount and the interest. Therefore, **EMI = principal amount + interest paid on the Car Loan**.

Price-Here it is the Target variable

Car_name- Some people so passionate about cars, so model play a vital role here

Data Pre-processing Done:

Data Pre-processing is defined as the process of preparing the data and making it suitable for a machine learning model. It's the primary and crucial step while creating a machine learning model. When creating a machine learning project, it's not always a case that we encounter the clean and formatted data. And while doing any operation with data, it's mandatory to wash it and put during a formatted way. So for this, we use data pre-processing task.

In simple terms, Data pre-processing is a data mining technique which is used to transform the raw data in a useful and efficient format. In the present project, data pre-processing steps are as follows: \

- **Collection of Data**: The data collected is in the form of a csv file.

- **Importing the required Libraries**: Initially we import the libraries like NumPy, pandas and matplotlib. These libraries are well explained in the below section.

- **Importing the Dataset**: This is also known as loading/reading the data which can be done by using Pandas. Dataset is read by giving the right path where it is located.

- **Identifying the missing values and Handling them**: In the given dataset, we do not have missing values in most of the columns.

- **Encoding the categorical data**: This section is of converting Categorical columns to Numerical columns since, model takes only numerical values. There are varieties of encoding techniques. Here, we use Label Encoding to convert the categorical columns. Before using Label Encoding, we import label encoder from

sklearn.preprocessing library. In the given dataset, we have nearly most of the features as categorical (object type) columns.

-

```
for i in df1.columns:
    df[i]=le.fit_transform(df[i])
```

Data Inputs- Logic- Output Relationships:

In the given dataset, label is the output feature and all the remaining features can be taken as input. The relationship between the input and output variables are found by correlation. In general, the correlation can be found using heatmap in which all the relationship between the feature variables can be observed clearly. We can also use different plots by giving the target variable against the remaining columns.

Hardware and Software Requirements and Tools Used:

Hardware:HP Laptop, Intel Core i5 Processor, 8th Generation

Software: The complete project is done using Jupyter Notebook.

Libraries and Packages used:

Initially we load the basic libraries such as pandas, seaborn and matplotlib **NumPy:** NumPy is the library used for scientific computing in python. It is also termed as numerical python.

NumPy is imported as "import numpy".

Pandas: Pandas is mainly used for data analysis. It allows importing data from various file formats.

The Pandas library is imported as "import pandas"

Seaborn: Seaborn is a Python data visualization library based on matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphics.

Matplotlib: Matplotlib is a visualization library in Python for 2D plots of arrays.

sklearn.preprocessing:The sklearn.preprocessing package provides several common utility functions and transformer classes to change raw feature vectors into a representation that is more suitable for the downstream estimators. In general, learning algorithms benefit from standardization of the data set.

LabelEncoder: LabelEncoder library is used to convert categorical columns to numerical columns. It is imported from sklearn.preprocessing package.LabelEncoder is imported as

from sklearn.preprocessing import LabelEncoder

StandardScaler: StandardScaler is used to standardize the data to a single scale and it is imported as

from sklearn.preprocessing import StandardScaler

SciPy: SciPylibrary includes modules for linear algebra, integration, optimization, and statistics. Skew is used to treat the skewness in the dataset. It is imported from scipy.stats library in the following way:

from scipy.stats import skew

StandardScaler: It is used to scale all the values of each feature such that its distribution will have a mean value 0 and standard deviation of 1. It is imported from sklearn.preprocessing library.

from sklearn.preprocessing import StandardScaler train_test_split: Used in breaking our input and target variable into train and test data. In the project we have split train and test into 80:20 respectively. It is imported from sklearn.model_selectionlibrary. **from sklearn.model_selection import train_test_split**

All the used Algorithms are imported as shown in figure below.

from sklearn import linear_model from **sklearn.linear_model import LinearRegression,Lasso,Ridge**
from sklearn.tree import DecisionTreeRegressor
from sklearn.ensemble import RandomForestRegressor, GradientBoostingRegressor

Performance metrics:

All the Regression model's performance is found by mae, rmse, r2_score.

from sklearn.metrics import mean_absolute_error,mean_squared_error,r2_score

As some of these libraries are frequently used, we write these in short form as following:

import numpy as np import
pandas as pd import matplotlib.pyplot
as plt import seaborn as sns

All the machine learning algorithms libraries/packages are explained detail in further sections.

Model/s Development and Evaluation

Identification of possible problem-solving approaches (methods) The whole problem-solving approach includes the following steps:

- **Problem Framing:** It includes understanding the problem that is whether the problem is of regression or classification. The present project is of regression type. As mentioned Earlier price cannot be categorial

- **Data Understanding:** Data understanding means having an intimate grasp of both the distributions of variables and the relationships between variables. It also includes summary statistics and data visualization.

- **Data Cleaning:** The process of identifying and repairing issues with the data is termed as data cleaning. Statistical methods are used for data cleaning. Some of them are outlier detection and imputation. There are many outliers in the present dataset, but as being categorial column we did not remove those

- **Data Selection:** The process of reducing the scope of data to those elements that are most useful for making predictions is called data selection.

- **Data Preparation:** It includes the data to identify the features to be selected and removed. We converted obj data to numeric .

- **Model Evaluation:** Model evaluation consists of identifying input and output variables and splitting the dataset into train and test datasets. The output variable is “Price ” and the remaining features are input variables. In this project, we have split into 80:20 train and test respectively.
- **Model Configuration:** Hyperparameter tuning the models will get the best fit parameters of each and every model. In this project we use GridSearchCV for knowing the best fit parameters.
- **Model Selection:** The process of selecting one method as the solution is called model selection. It includes the regression model performance metrics.

Testing of Identified Approaches (Algorithms):

The Machine Learning Algorithms used in this project for training and testing to predict the prices of the houses are namely:

- Linear Regression
- Lasso Regression
- Ridge Regression
- Decision Tree Regressor

In addition to the above algorithms, ensembling techniques are also use. They are:

- Random Forest Regressor
- Gradient Boosting Regressor

Linear Regression:

Linear regression is a linear model, e.g. a model that assumes a linear relationship between the input variables (x) and the single output variable (y). More specifically, that y can be calculated from a linear combination of the input variables (x). When there is a single input variable (x), the method is referred to as simple linear regression. When there are multiple input variables, it refers to the method as multiple linear regression.

Lasso Regression:

Lasso regression is a type linear regression that uses shrinkage. Shrinkage is where data values are shrunk towards a central point, like the mean. The lasso procedure encourages simple, sparse models .This particular type of regression is well-suited for models showing high levels of multicollinearity.

Ridge Regression:

Ridge regression is a model tuning method that is used to analyse any data that suffers from multicollinearity. This method performs L2 regularization. When the issue of multicollinearity occurs, least-squares are unbiased, and variances are large, this results in predicted values to be far away from the actual values.

Decision Tree Regressor:

Decision tree builds regression or classification models in the form of a tree structure. It breaks down a dataset into smaller and smaller subsets while at the same time an associated decision tree is incrementally developed. The final result is a tree with decision nodes and leaf nodes. **Random Forest Regressor:**

A random forest is a meta estimator that fits a number of classifying decision trees on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting.

Gradient Boosting Regressor:

Gradient boosting is a machine learning technique for regression and classification problems, which produces a prediction model in the form of an ensemble of weak prediction models, typically decision trees.

Run and Evaluate selected models:

The above-mentioned algorithms have been run in the jupyter notebook and the performance metrics are found .

To check whether the model is overfitting/underfitting GridSearchCV is used

We have chosen the best R2 score by running the random state of 42 to 100.

Random forest Regressor gave the best R2 score with random state of 72.

Key Metrics for success in solving problem under consideration:

Evaluating a model is a major part of building an effective machine learning model. The performance metrics of a Regression model are R2 score, Mean absolute error, mean squared error, Root Mean Squared Error. Among these R2 score should be in between 0 & 1.

Observations from modelling:

- Target variable is selected as “Price”
- Split the dataset to 80:20 for train and test respectively.
- Used standard scaler to scale the values
- All the data obtained from above steps given to model and made the predictions.
- Observed the best fit parameters from GridSearchCV
- Gradient Boosting Regressor has the better performance metrics.

CONCLUSION:

Key Findings and Conclusions of the Study:

The goal is to achieve the system which will reduce the human effort to find a car price. The proposed system car Price Prediction model approximately tries to achieve the same one. We have managed out how to prepare a model that gives users for a best approach with future lodging value predictions. Proposed system focused on predict the price according to car history and Emi , model .

Limitations of this work and Scope for Future Work:

- **Limitations:** Datas are limited

We found after all analysis RandomForestRegressor() is the best model and 'criterion': ' mae', 'max_depth': 18, 'n_estimators': 2 we got the final model which accuracy is 99. 9540419237838

```
Finalmodel=RandomForestRegressor(criterion= 'mae', max_depth= 18, n_estimators = 2)

Finalmodel.fit(x_train, y_train)

pred=Finalmodel.predict(x_test)
R2score=r2_score(y_test,pred)

print(R2score*100)
```

```
99.9540419237838
```

```
import joblib
joblib.dump(Finalmodel,"Car_data_output.pkl")
```

```
['Car_data_output.pkl']
```