IBM – NALAIYA THIRAN PROJECT

CAR RESALE VALUE PREDICTION

INDUSTRY MENTOR : Prof. SWETHA

FACULTY MENTOR : VENKATESAN R

TEAM ID : PNT2022TMID02053

TEAM LEAD : SUGANYA N TEAM

MEMBER : SANTHIYA R TEAM

MEMBER : SELVADHARANI K

TEAM MEMBER : PRITHIKA N

In partial fulfillment for the

award of degree of

Bachelor of Technology(B.Tech)

In

Information Technology

Stay Ahead

BANNARI AMMAN

INSTITUTE OF TECHNOLOGY

An Autonomous Institution Affiliated to Anna University, Approved by AICTE, Accredited by NAAC with 'A' Grade

Acknowledgement

We would like to express our special thanks of gratitude to our Faculty Mentor and Industry Mentor for their support and guidance in completing our project on the car resale value prediction.We would like to extend our gratitude to the IBM for Nalaiya Thiran project for providing us with all the facility that was required.It was a great learning experience. We would like to take this opportunity to express our gratitude.

TEAM MEMBERS

Suganya N (7376192IT242)

Santhiya R (7376192IT225)

Selvadharani K (7376192IT229)

Prithika N (7376192IT210)

DATE:

19/11/2022

ABSTRACT

Used car resale market in India was marked at 24.2 billion US dollars in 2019. Due to the huge requirement of used cars and lack of experts who can determine the correct valuation, there is an utmost need of bridging this gap between sellers and buyers. This project focuses on building a system that can accurately predict a resale value of the car based on minimal features like kms driven, year of purchase etc. without manual or human interference and hence it remains unbiased.

The production of cars has been steadily increasing in the past decade, with over 70 million passenger cars being produced in the year 2016. This has given rise to the used car market, which on its own has become a booming industry. The recent advent of online portals has facilitated the need for both the customer and the seller to be better informed about the trends and patterns that determine the value of a used car in the market. Using Machine Learning Algorithm like random regressor algorithm, we will try to develop a statistical model which will be able to predict the price of a used car, based on previous consumer data and a given set of features. We will also be comparing the prediction accuracy of these models to determine the optimal one.

1. INTRODUCTION

In this project, we developed automotive resale value prediction systems taking into account various car features using various algorithms and methodologies. In a word, car resale value prediction enables users to forecast the resale value of a car based on features such as miles driven, fuel type, etc.

1.1 NEED FOR THE SYSTEM

The sole goal of this general-purpose system for estimating resale value is to estimate the amount that the user can probably acquire. We attempt to anticipate the amount of resale with an accuracy of at least 70% so the user can receive an estimated value prior to reselling the vehicle and avoid closing a deal at a loss.

1.2 PROJECT PURPOSE

The main idea of making a car resale value prediction system is to get hands-on practice for python using Data Science.The system that forecasts the amount of resale value for cars is based on the user-provided parameters. The car's details are entered into the provided form by the user, and the value at which it will be sold is then forecasted.

1.3 PROJECT SCOPE

The system, which is written in the Python programming language, makes resale value predictions depending on the information provided. The machine learning program's training dataset, which calculates the exact value of the car, is what the system uses to operate. Only fields for the car's purchase price, miles driven, fuel, and purchase year are available for user input.

2. OBJECTIVE

The goal of a car resale value prediction system is to forecast the accurate worth of used cars, enabling customers to sell their vehicles remotely with unbiased valuation and without the need for human participation. In order to anticipate the car's resale value, the system only considers a small number of parameters due to the scarcity of data. The current system does not account for any physical damage to the car's body or engine when estimating its resale value because it is an online system.

Algorithms implemented	
Model Algorithm	**RMSE**
Linear Regression	1.6339649305195656
Decision Tree Regression	1.188564363381992
Random Forest Regression	0.6615636769718574

Our newly created system is divided into two components: data collection and prediction utilizing a machine learning-based algorithm. To train the model, we used publicly accessible Kaggle datasets.The second part is the web-based car resale value prediction.After preprocessing and cleaning the data from the previous stage, we trained an ML model that is based on the boosting technique. Prediction is made using the trained model. The front-end form requests information from users such as the city, miles traveled, year of purchase, and fuel type that are necessary for the ML model to produce a prediction.

Following form submission, data is transmitted via the Flask API to the ML model, which then responds with a forecasted automobile resale value based on user input. Utilizing a render template, this forecast is presented on the web page. As a result, a user may forecast the resale value of his car with little information, without human assistance, and without personal inspection.

3. PREDICTION APPROACH

For accurate prediction and better model training, a huge dataset of resale cars gathered.This dataset contains data of 5 main features i.e., fuel type, kms driven, city, car purchase year and resale value. Here resale value becomes our target column whereas other columns served as features for our model.

Data set consists of many null values and text values. which have to be removed as the model can only understand numbers. Moreover, fuel type was converted into numerical codes via one-hot encoding.A one hot encoding is a representation of categorical variables as binary vectors. This requires that the categorical values be mapped to integer values. After data pre-processing, all 5 files, each representing each city, have to be merged for model training.

Various different machine learning algorithms were implemented on the dataset along with hyperparameter tuning using **random_grid.** Reason behind this hyperparameter using is good performance, because of its mathematical working.

The reason why RandomSearchCV could outperform all other regression algorithms is the mathematics behind it.

1.Linear Regression:

A variable's value can be predicted using linear regression analysis based on the value of another variable. The dependent variable is the one you want to be able to forecast. The independent variable is the one you're using to make a prediction based on the value of the other variable.The Train Using AutoML tool employs the supervised machine learning technique of linear regression to identify the linear equation that most accurately captures the relationship between the explanatory variables and the dependent variable. This is accomplished by utilizing least squares to fit a line to the data.

More specifically, the nature and strength of the association between a dependent variable and a number of other independent variables are assessed using linear regression. It aids in the creation of models for making predictions, such as predicting the stock price of a company.

2.Decision Tree Regression:

A decision-making tool known as a "Decision Tree" uses a tree structure that resembles a flowchart or is a model of decisions and all of their potential outcomes, such as outcomes, input costs, and utility.The supervised learning algorithms group includes the decision-tree algorithm. It works with output variables that are categorized and continuous. The nodes have either: The branches/edges represent the node's outcome, and the nodes have

[Decision Nodes] Conditions

Result [End Nodes]

The example below illustrates a decision tree that assesses the smallest of three numbers: the branches/edges reflect the truth/falsity of the assertion, and a decision is made based on that.

Decision tree regression trains a model in the form of a tree to predict data in the future and generate useful continuous output by observing the properties of an item. Continuous output denotes the absence of discrete output, i.e., output that is not only represented by a discrete, well-known set of numbers or values.A weather prediction model that forecasts whether it will rain on a specific day is an example of a discrete output.

A profit prediction model that estimates the likely profit that may be made from the sale of a product is an example of a continuous output.Here, a decision tree regression model is used to predict continuous values.

3.Random Forest Regression:

A supervised learning technique called Random Forest Regression leverages the ensemble learning approach for regression. The ensemble learning method combines predictions from various machine learning algorithms to provide predictions that are more accurate than those from a single model.A separate sample of rows are used to build each tree, and a different sample of characteristics are chosen for splitting at each node.Each tree provides a unique prediction on its own.

A single outcome is then produced by averaging these predictions.A Random Forest's accuracy and overfitting are improved over a single Decision Tree thanks to the averaging. The average of the forecasts made by the forest's trees constitutes a prediction from the Random Forest Regressor.

4.TEST CASES

- **Missing values**

Four feature inputs are needed by the trained ML model to predict the output. If that doesn't work, the model returns an incorrect input error. The user must fill out every field because every one of the html form's necessary fields has been marked as such using CSS.

Output: If a field is left blank, the form will display a warning message and the user will need to fill it in.Therefore, there can be no model prediction mistakes.

- **Invalid Input**

For all 4 features, the trained ML model simply needs numerical input. Thus, the model may generate an error if the user inputs symbols like commas. Preprocessing script is deployed in the backend to eliminate all undesirable characters, such as commas and whitespace, in order to ensure that the model receives the input it needs.

Output: Because of the preprocessing script written in Python, the model will receive the correct input and produce accurate predictions.

- **Unseen year of purchase**

With data from vehicles acquired between 2011 and 2020, the model was trained. Since that data is fairly recent and unfamiliar to the model, it may become confused if the user enters information about a car they bought after that, in 2021.

Output: Because the model was trained using the boosting approach, it provides results that are fairly accurate, with an RMSE of only about 65,000 INR.

5. APPLICATION BUILDING

Build The Python Flask App

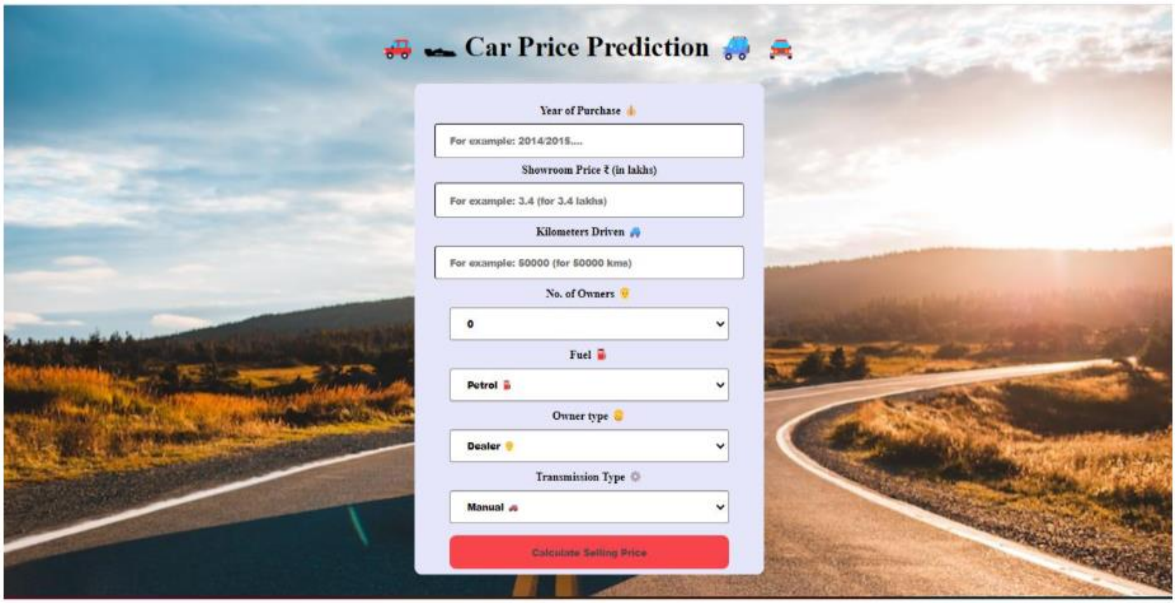
```
from flask import Flask, render_template,
request, jsonify

import requests

import pickle

import numpy as np

import sklearn
```

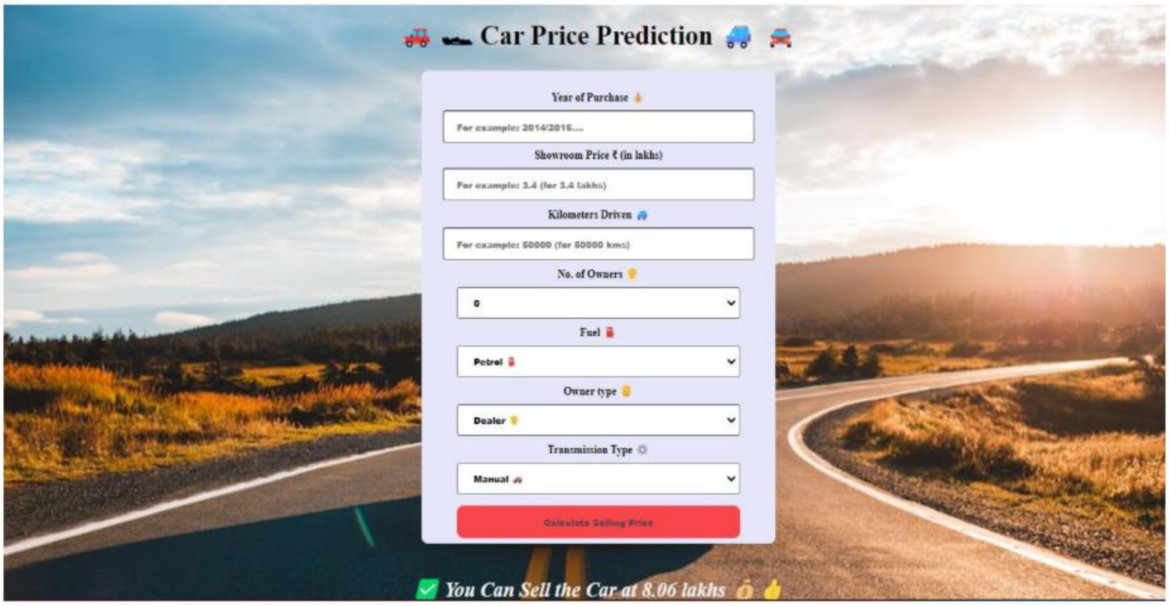
```
from sklearn.preprocessing import
StandardScaler

app = Flask("car_model")

model = pickle.load(open('model.pkl', 'rb'))

@app.route('/',methods=['GET'])

def Home():

    return render_template('index.html')

standard_to = StandardScaler()

@app.route("/predict", methods=['POST'])

def predict():

    Fuel_Type_Diesel=0

    if request.method == 'POST':

        Year = int(request.form['Year'])

        Year = 2020 - Year
Present_Price=float(request.form['Present_
Price'])
Kms_Driven=int(request.form['Kms_Drive
n'])

        Kms_Driven2=np.log(Kms_Driven)

        Owner=int(request.form['Owner'])
Fuel_Type_Petrol=request.form['Fuel_Type
_Petrol']

        if(Fuel_Type_Petrol=='Petrol'):

                Fuel_Type_Petrol=1
```

```
            Fuel_Type_Diesel=0
        elif(Fuel_Type_Petrol=='Diesel'):
           Fuel_Type_Petrol=0
           Fuel_Type_Diesel=1
        else:
           Fuel_Type_Petrol=0
           Fuel_Type_Diesel=0

Seller_Type_Individual=request.form['Selle
r_Type_Individual']
if(Seller_Type_Individual=='Individual'):
           Seller_Type_Individual=1
        else:
           Seller_Type_Individual=0

Transmission_Manual=request.form['Trans
mission_Manual']
       if(Transmission_Manual=='Mannual'):
           Transmission_Manual=1
        else:
           Transmission_Manual=0

prediction=model.predict([[Present_Price,K
ms_Driven2,Owner,Year,Fuel_Type_Diesel,
```

```
Fuel_Type_Petrol,Seller_Type_Individual,Transmission_Manual]])
        output=round(prediction[0],2)

        if output<0:
            return render_template('index.html',prediction_text="❌ Sorry you cannot sell this car. 🙁")

        else:
            return render_template('index.html',prediction_text="✅ You Can Sell the Car at {} lakhs 🤑 👍".format(output))
    else:
        return render_template('index.html')
if __name__=="__main__":
    app.run(debug=True)
```

6. EXECUTE AND TEST MODEL





7. CODING:

Form :

```
<!DOCTYPE html>
<html lang="en">
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <head>
        <title>🚗 🏎️ Car Price Prediction 🚙  🚘  </title>
    </head>
<style>
input[type=text], select {
  width: 80%;
  padding: 12px 20px;
  margin: 8px 0;
  display: inline-block;
  border-radius: 4px;
  font-weight: bolder;
}
```

```
input[type=submit] {
  width: 80%;
  background-color: red;
  opacity: 0.7;
  color:black;
  padding: 14px 20px;
  margin: 8px 0;
  border: none;
  border-radius: 10px;
  cursor: pointer;
  font-size:100;
  font-weight: bolder;
}
input[type=submit]:hover {
  background-color: red;
  opacity: 0.9;
}
form{
    border-radius: 10px;
    padding-top: 5%;
    font-weight: bolder;
}
div {

  border-radius: 5px;
  margin-left: 35%;
  width: 30%;

}
h1{
```

```
    color:color-blue;
    font-size: 40px;
    font-weight: bolder;
}
h3{
  color: white;
  font-size: 40px;
  border-radius: 10px;
  font-weight: bolder;
  padding: 14px 20px;
}


body {
            background-image: url("static/images/car.png");
            background-repeat: no-repeat;
            background-size: cover;
            text-align: center;
            padding: 0px;
        }
</style>
<body>
<h1>🚗 🏎 Car Price Prediction 🚙 🚘 </h1>
<div>
  <form action="{{ url_for('predict')}}" method="post"
style="background-color:#E6E6FA">
<label for="Year">Year of Purchase 💰 </label><br>
<input type="text" id="Year" name="Year" placeholder="For example:
2014/2015...." required="required"><br>
```

```html
<label for="Present_Price">Showroom Price ₹ (in lakhs)</label><br>
<input type="text" id="Present_Price" name="Present_Price"
placeholder="For example: 3.4 (for 3.4 lakhs)"
required="required"><br>
<label for="Kms_Driven">Kilometers Driven 🚙 </label><br>
<input type="text" id="Kms_Driven" name="Kms_Driven"
placeholder="For example: 50000 (for 50000 kms)"
required="required"><br>
<label for="Owner">No. of Owners 👱 </label><br>
<select id="owner" name="Owner">
      <option value="0">0</option>
      <option value="1">1</option>
      <option value="3">3</option>
   </select><br>
<label for="Fuel_Type_Petrol">Fuel ⛽ </label><br>
   <select id="Fuel_Type_Petrol" name="Fuel_Type_Petrol"
required="required">
      <option value="Petrol">Petrol ⛽ </option>
      <option value="Diesel">Diesel ⛽ </option>
      <option value="Cng">CNG ⛽ </option>
   </select><br>
<label for="Seller_Type_Individual">Owner type 👱 </label><br>
   <select id="Seller_Type_Individual" name="Seller_Type_Individual"
required="required">
      <option value="dealer">Dealer 👱 </option>
      <option value="individual">Individual 👱 </option>
      </select><br>
```

```
<label for="Transmission_Manual ">Transmission Type
⚙️ </label><br>
   <select id="Transmission_Manual" name="Transmission_Manual"
required="required">
     <option value="manual car">Manual 🚗 </option>
     <option value="automatic car ">Automatic 🚙 </option>
    </select> <br>

<input type="submit" value="Calculate Selling Price">
</form>
</div>
 <h3>{{ prediction_text }}</h3>
</body>
</html>
```

8. FUTURE ENHANCEMENT

The system is currently not very accurate. Additionally, only limited data has been gathered. To increase accuracy and usability, this might be expanded to include several automobile models and cities.Once sufficient data has been gathered, efficient deep learning techniques like LSTM (Long Short-Term Memory) or RNN (Recurrent Neural Networks) can be used. This can significantly increase accuracy while lowering RMSE.

Only a few features are currently used to forecast a car's resale value. Additional features could be added to this. CNN may also be used to assess the physical state of an automobile from photographs, such as seeing dents and scratches, and hence forecast more accurate resale value.

9. CONCLUSION

The increased prices of new cars and the financial incapability of the customers to buy them, Used Car sales are on a global increase. Therefore, there is an urgent need for a Used Car Price Prediction system which effectively determines the worthiness of the car using a variety of features. The proposed system will help to determine the accurate price of used car price prediction.As a result of pre-processing and transformation, Random Forest Regressor came out on top with 97% accuracy.

10.REFERENCES

1) Pudaruth, S., 2014. "Predicting the Price of Used Cars using Machine Learning Techniques." Vol 4, Number 7 (2014), pp. 753-76.

2) ijictv4n7spl_17.pdf (ripublication.com)

3) Gokce, E. (2020, January 10). "Predicting used car prices with machine learning techniques. "

4) Predicting Used Car Prices with Machine Learning Techniques | by Enes Gokce | Towards Data Science.

11.GITHUB AND VIDEO DEMO LINK

1.GITHUB LINK:

https://github.com/IBM-EPBL/IBM-Project-8934-1658938772

2.DEMO VIDEO LINK:

https://drive.google.com/file/d/17vvg7pxIMcI5oNywkRZ5_E5885Xz_uha/view?usp=sharing