

AUTOMOBILE PRICE PREDICTION

Presented By
ALWIN M
AKASHKUMAR A H
ADHITHYAKUMAR A S

INTRODUCTION

Car price prediction using machine learning is an innovative approach that leverages data-driven models to estimate the value of new and used vehicles. Determining the fair market price of a car relied heavily on manual assessments and subjective evaluations. However, with the advent of machine learning, this process has become more accurate, consistent, and efficient. By analyzing historical car sales data, vehicle specifications, and prevailing market trends, machine learning models such as Linear Regression, Random Forest, and Neural Networks can predict car prices based on key features like make, model, year, mileage, condition, and location.

PROBLEM

STATEMENT

Develop a machine learning-based model that can accurately predict the price of a used car based on several input features such as car brand, model, year of manufacture, mileage, fuel type, transmission type, and ownership history. By utilizing historical data of car sales, the model will be trained to learn patterns and relationships between these features and the market value of cars. This will allow users to receive an estimated price for a car instantly, making the process more efficient and reliable.

EXISTING SYSTEM

An existing system for car price prediction using machine learning typically involves analyzing historical car sales data, vehicle specifications, and market trends to train models like Linear Regression, Random Forest, or Neural Networks. These models then predict the fair market value of cars based on various features such as make, model, year, mileage, condition, and location.

PROPOSED SYSTEM



The proposed system uses advanced machine learning techniques to forecast more accurate, scalable, and user-friendly predictions of new and used cars. To do this, it's going to build upon existing methodologies by making use of real-time data ingestion, feature rich in analysis and explainability, all together with modern practices of deployment for it to reach out to a wider audience like buyers, sellers, and dealerships.

HOW IT WORKS

```
[ ] model = LinearRegression()

[ ] model.fit(x_train, y_train)

[ ] predict = model.predict(x_test)

[ ] predict
array([[378844.93028124, 347161.92986535, 381845.91985837, ...,
        84333.96746337, -26166.2537024 , 332515.64289612]])

x_train.head(1)
  name  year  km_driven  fuel  seller_type  transmission  owner  mileage  engine  max_power  seats
3864   1  2015    30000    2             1             1         23.1    998.0      67.04     5.0

[ ] input_data_model = pd.DataFrame([[4,2020,5900,1,2,2,1,17.01,1582.0,126.2,5.0]],columns=['name', 'year', 'km_driven',
[ ] input_data_model

  name  year  km_driven  fuel  seller_type  transmission  owner  mileage  engine  max_power  seats
0     4  2017     5900    1             2             2         17.01  1582.0    126.2     5.0
```

```
app.py 3 x Training.ipynb Cardetails.csv
app.py 3
1 import pandas as pd
2 import numpy as np
3 import pickle as pk
4 import streamlit as st
5
6 model = pk.load(open('model.pkl','rb'))
7
8 st.header('Car Price Prediction ML Model')
9
10 cars_data = pd.read_csv('Cardetails.csv')
11
12 def get_brand_name(car_name):
13     car_name = car_name.split(' ')[0]
14     return car_name.strip()
15 cars_data['name'] = cars_data['name'].apply(get_brand_name)
16
17 name = st.selectbox('Select Car Brand', cars_data['name'].unique())
18 year = st.slider('Car Manufactured Year', 1994,2024)
19 km_driven = st.slider('No of kms Driven', 11,200000)
20 fuel = st.selectbox('Fuel type', cars_data['fuel'].unique())
21 seller_type = st.selectbox('Seller type', cars_data['seller_type'].unique())
22 transmission = st.selectbox('Transmission type', cars_data['transmission'].unique())
23 owner = st.selectbox('Seller type', cars_data['owner'].unique())
24 mileage = st.slider('Car Mileage', 10,40)
25 engine = st.slider('Engine CC', 700,5000)
26 max_power = st.slider('Max Power', 0,200)
27 seats = st.slider('No of Seats', 5,10)
28
```

```
PS D:\VScod>
* History restored

PS D:\VScod>
* History restored

PS D:\VScod>
* History restored

PS D:\VScod>
```

Automobile price prediction

HOW IT WORKS

```

.ipynb ★
File Edit Insert Runtime Tools Help Last edited on October 9

Connect

def clean_data(value):
    value = value.split(' ')[0]
    value = value.strip()
    if value == '':
        return 0
    return float(value)

cars_data['brand_name'] = cars_data['name'].apply(get_brand_name)

cars_data['name'].unique()

cars_data['name'].unique()
Maruti, 'Skoda', 'Honda', 'Hyundai', 'Toyota', 'Ford', 'Renault',
Mahindra, 'Tata', 'Chevrolet', 'Datsun', 'Jeep', 'Mercedes-Benz',
Mitsubishi, 'Audi', 'Volkswagen', 'BMW', 'Nissan', 'Lexus',
Jaguar, 'Land', 'MG', 'Volvo', 'Daewoo', 'Kia', 'Fiat', 'Force',
Ambassador, 'Ashok', 'Isuzu', 'Opel'], dtype=object)

cars_data['mileage'] = cars_data['mileage'].apply(clean_data)

cars_data['max_power'] = cars_data['max_power'].apply(clean_data)

cars_data['engine'] = cars_data['engine'].apply(clean_data)

for col in cars_data.columns:
    print('Unique values of ' + col)

```

```

VS Code
File Edit Selection View Go Run Terminal Help

EXPLORER
app.py 3 x Training.ipynb Cardetails.csv
CODE
app.py 3 x
25 engine = st.slider('Engine CC', 700,5000)
26 max_power = st.slider('Max Power', 0,200)
27 seats = st.slider('No of Seats', 5,10)
28
29
30 if st.button("Predict"):
31     input_data_model = pd.DataFrame(
32         [[name,year,km_driven,fuel,seller_type,transmission,owner,mileage,engine,max_power,seats]],
33         columns=['name','year','km_driven','fuel','seller_type','transmission','owner','mileage','engine','max_power','seats'])
34
35     input_data_model['owner'].replace(['First Owner', 'Second Owner', 'Third Owner',
36         'Fourth & Above Owner', 'Test Drive Car'],
37         [1,2,3,4,5], inplace=True)
38     input_data_model['fuel'].replace(['Diesel', 'Petrol', 'LPG', 'CNG'],[1,2,3,4], inplace=True)
39     input_data_model['seller_type'].replace(['Individual', 'Dealer', 'Trustmark Dealer'],[1,2,3], inplace=True)
40     input_data_model['transmission'].replace(['Manual', 'Automatic'],[1,2], inplace=True)
41     input_data_model['name'].replace(['Maruti', 'Skoda', 'Honda', 'Hyundai', 'Toyota', 'Ford', 'Renault',
42         'Mahindra', 'Tata', 'Chevrolet', 'Datsun', 'Jeep', 'Mercedes-Benz',
43         'Mitsubishi', 'Audi', 'Volkswagen', 'BMW', 'Nissan', 'Lexus',
44         'Jaguar', 'Land', 'MG', 'Volvo', 'Daewoo', 'Kia', 'Fiat', 'Force',
45         'Ambassador', 'Ashok', 'Isuzu', 'Opel'],
46         [1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31],
47         inplace=True)
48
49     car_price = model.predict(input_data_model)
50
51     st.markdown('Car Price is going to be ' + str(car_price[0]))

PROBLEMS 3 OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS D:\VScode>
* History restored
PS D:\VScode>
* History restored
PS D:\VScode>
* History restored
PS D:\VScode>

```

Automobile price prediction

Proposed Output

Car Price Prediction ML Model

Select Car Brand

Ford

Car Manufactured Year

1994 2024

2020

No of kms Driven

11 200000

10749

Fuel type

Petrol

Seller type

Individual

Transmission type

Manual

Seller type

First Owner

Car Mileage

10 40

20

Engine CC

700 5000

809

Max Power

0 200

20

No of Seats

5 10

5

Predict

Car Price is going to be 160347.2262430936

CONCLUSION

9

The new method was able to efficiently automate the complicated process of the estimation of car prices, which is one of the benefits both car sellers and buyers get in terms of decision-making, as well as dealers, through the immediate prediction of prices. As far as the integration of the model with Streamlit seems to open up an interface where the user will be required to input the details of the car that they are interested in acquiring. That is to say, the accuracy of predictions is highly dependent on the quality and breadth of the training data in addition to the capacity of this model to generalize to unseen data.



The background features three vertical stripes on the left: a wide pink stripe, a narrower blue stripe, and a medium-width beige stripe. The right side of the image is a light cream color, decorated with two rectangular areas of a pink dot pattern. One area is in the top right corner, and the other is in the bottom right corner. Both dot patterns are arranged in a grid that tapers off to the right.

THANK YOU