on-modeling-for-used-cars-in-india

November 26, 2024

[ ]:

```

```

1 Project Title: Price prediction modeling for used cars in India

1.0.1 Overview

This dataset contains information about used cars in the Indian market, comprising 9,582 entries with 11 detailed attributes. The data appears to be collected up to November 2024, providing a comprehensive view of the second-hand car market in India.

1.0.2 Dataset Features

- Brand: Car manufacturer (e.g., Volkswagen, Maruti Suzuki, Honda, Tata)
- Model: Specific car model (e.g., Taigun, Baleno, Polo, WRV)
- Year: Manufacturing year of the vehicle (ranging from older models to 2024)
- Age: Age of the vehicle in years
- kmDriven: Total kilometers driven by the vehicle
- Transmission: Type of transmission (Manual or Automatic)
- Owner: Ownership status (first or second owner)
- FuelType: Type of fuel (Petrol, Diesel, Hybrid/CNG)
- PostedDate: When the car listing was posted
- AdditionalInfo: Extra details about the vehicle
- AskPrice: Listed price in Indian Rupees ( )

[ ]:

```

```

[ ]:

```

```

[6]:

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

[149]:

```python
# loading dataset

data = pd.read_csv("used_car_dataset.csv")
```

[150]:
```
data.head()
```

[150]:
```
           Brand      model  Year  Age     kmDriven Transmission   Owner  \
0          Honda       City  2001   23    98,000 km       Manual  second
1         Toyota     Innova  2009   15  190000.0 km       Manual  second
2     Volkswagen  VentoTest  2010   14    77,246 km       Manual   first
3  Maruti Suzuki      Swift  2017    7    83,500 km       Manual  second
4  Maruti Suzuki     Baleno  2019    5    45,000 km    Automatic   first

  FuelType PostedDate                                        AdditionInfo  \
0   Petrol     Nov-24  Honda City v teck in mint condition, valid gen…
1   Diesel     Jul-24  Toyota Innova 2.5 G (Diesel) 7 Seater, 2009, D…
2   Diesel     Nov-24  Volkswagen Vento 2010-2013 Diesel Breeze, 2010…
3   Diesel     Nov-24     Maruti Suzuki Swift 2017 Diesel Good Condition
4   Petrol     Nov-24       Maruti Suzuki Baleno Alpha CVT, 2019, Petrol

   AskPrice
0  1,95,000
1  3,75,000
2  1,84,999
3  5,65,000
4  6,85,000
```

[151]:
```
data.tail()
```

[151]:
```
              Brand     model  Year  Age    kmDriven Transmission   Owner  \
9577          Skoda   Octavia  2014   10  105,904 km    Automatic  second
9578  Maruti Suzuki  Alto-800  2020    4   55,000 km       Manual   first
9579  Maruti Suzuki      Ritz  2013   11   92,000 km       Manual   first
9580        Hyundai     Verna  2019    5   72,000 km    Automatic   first
9581        Hyundai   New i20  2021    3   83,228 km       Manual  second

       FuelType PostedDate  \
9577     Diesel     Oct-24
9578  Hybrid/CNG     Nov-24
9579     Diesel     Nov-24
9580     Petrol     Oct-24
9581     Petrol     Nov-24

                                           AdditionInfo     AskPrice
9577       Skoda Octavia 1.9 Elegance TDI, 2014, Diesel   10,40,000
9578  Maruti Suzuki Alto 800 CNG LXI Optional, 2020,…     3,75,000
9579                   Maruti Suzuki Ritz VDi, 2013, Diesel     4,15,000
9580  Hyundai Verna VTVT 1.6 AT SX Option, 2019, Petrol     8,55,000
9581         Hyundai New i20 1.2 Asta IVT, 2021, Petrol     6,99,000
```

[152]:
```
data.shape
```

[152]:
```
(9582, 11)
```

[153]:
```python
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9582 entries, 0 to 9581
Data columns (total 11 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   Brand         9582 non-null   object
 1   model         9582 non-null   object
 2   Year          9582 non-null   int64
 3   Age           9582 non-null   int64
 4   kmDriven      9535 non-null   object
 5   Transmission  9582 non-null   object
 6   Owner         9582 non-null   object
 7   FuelType      9582 non-null   object
 8   PostedDate    9582 non-null   object
 9   AdditionInfo  9582 non-null   object
 10  AskPrice      9582 non-null   object
dtypes: int64(2), object(9)
memory usage: 823.6+ KB
```

[154]:
```python
# find the null values

data.isna().sum()
```

[154]:
```
Brand            0
model            0
Year             0
Age              0
kmDriven        47
Transmission     0
Owner            0
FuelType         0
PostedDate       0
AdditionInfo     0
AskPrice         0
dtype: int64
```

[155]:
```python
data.columns
```

[155]:
```
Index(['Brand', 'model', 'Year', 'Age', 'kmDriven', 'Transmission', 'Owner',
       'FuelType', 'PostedDate', 'AdditionInfo', 'AskPrice'],
      dtype='object')
```

[156]:
```
# drop the column

data.drop(columns = ['AdditionInfo'], axis=1, inplace=True)
```

[157]:
```
data.head()
```

[157]:
```
            Brand      model  Year  Age     kmDriven Transmission   Owner  \
0           Honda       City  2001   23    98,000 km       Manual  second
1          Toyota     Innova  2009   15  190000.0 km       Manual  second
2      Volkswagen  VentoTest  2010   14    77,246 km       Manual   first
3   Maruti Suzuki      Swift  2017    7    83,500 km       Manual  second
4   Maruti Suzuki     Baleno  2019    5    45,000 km    Automatic   first

  FuelType PostedDate    AskPrice
0   Petrol     Nov-24    1,95,000
1   Diesel     Jul-24    3,75,000
2   Diesel     Nov-24    1,84,999
3   Diesel     Nov-24    5,65,000
4   Petrol     Nov-24    6,85,000
```

[158]:
```
data.shape
```

[158]:
```
(9582, 10)
```

[159]:
```
# Clean the AskPrice column
data['AskPrice'] = (
    data['AskPrice']
    .str.replace(' ', '', regex=False)  # Remove rupee symbol
    .str.replace(',', '', regex=False)  # Remove commas
    .str.strip()  # Remove any leading/trailing spaces
    .astype(int)  # Convert to integer
)
```

[160]:
```
data.head()
```

[160]:
```
            Brand      model  Year  Age     kmDriven Transmission   Owner  \
0           Honda       City  2001   23    98,000 km       Manual  second
1          Toyota     Innova  2009   15  190000.0 km       Manual  second
2      Volkswagen  VentoTest  2010   14    77,246 km       Manual   first
3   Maruti Suzuki      Swift  2017    7    83,500 km       Manual  second
4   Maruti Suzuki     Baleno  2019    5    45,000 km    Automatic   first

  FuelType PostedDate  AskPrice
0   Petrol     Nov-24    195000
1   Diesel     Jul-24    375000
2   Diesel     Nov-24    184999
3   Diesel     Nov-24    565000
```

```
4   Petrol     Nov-24    685000
```

[161]:

```
# Clean the kmDriven column
data['kmDriven'] = (
    data['kmDriven']
    .str.replace('km', '', regex=False)  # Remove rupee symbol
    .str.replace(',', '', regex=False)  # Remove commas
    .str.strip()  # Remove any leading/trailing spaces
    .astype(float)  # Convert to integer
)
```

[162]:

```
data.head()
```

[162]:

```
           Brand      model  Year  Age  kmDriven Transmission   Owner  \
0          Honda       City  2001   23   98000.0       Manual  second
1         Toyota     Innova  2009   15  190000.0       Manual  second
2     Volkswagen  VentoTest  2010   14   77246.0       Manual   first
3  Maruti Suzuki      Swift  2017    7   83500.0       Manual  second
4  Maruti Suzuki     Baleno  2019    5   45000.0    Automatic   first

  FuelType PostedDate  AskPrice
0   Petrol     Nov-24    195000
1   Diesel     Jul-24    375000
2   Diesel     Nov-24    184999
3   Diesel     Nov-24    565000
4   Petrol     Nov-24    685000
```

[173]:

```
data['kmDriven'] = data['kmDriven'].fillna(data['kmDriven'].mean())
```

[174]:

```
data.isna().sum()
```

[174]:

```
Brand           0
model           0
Year            0
Age             0
kmDriven        0
Transmission    0
Owner           0
FuelType        0
PostedDate      0
AskPrice        0
dtype: int64
```

[175]:

```
data.drop(columns = ['Age'],axis=1,inplace=True)
```

[176]:

```
data.head()
```

[176]:

```
           Brand      model  Year  kmDriven Transmission   Owner FuelType  \
0          Honda       City  2001   98000.0       Manual  second   Petrol
1         Toyota     Innova  2009  190000.0       Manual  second   Diesel
2     Volkswagen  VentoTest  2010   77246.0       Manual   first   Diesel
3  Maruti Suzuki      Swift  2017   83500.0       Manual  second   Diesel
4  Maruti Suzuki     Baleno  2019   45000.0    Automatic   first   Petrol

  PostedDate  AskPrice
0     Nov-24    195000
1     Jul-24    375000
2     Nov-24    184999
3     Nov-24    565000
4     Nov-24    685000
```

[177]:

```
data['PostedDate'] = pd.to_datetime(data['PostedDate'] + '-01',␣
 ↪format='%b-%y-%d')
```

[178]:

```
# Extract Year and Month from the PostedDate into separate columns
data['Year_Posted'] = data['PostedDate'].dt.year
data['Month_Posted'] = data['PostedDate'].dt.month
```

[179]:

```
data.head()
```

[179]:

```
           Brand      model  Year  kmDriven Transmission   Owner FuelType  \
0          Honda       City  2001   98000.0       Manual  second   Petrol
1         Toyota     Innova  2009  190000.0       Manual  second   Diesel
2     Volkswagen  VentoTest  2010   77246.0       Manual   first   Diesel
3  Maruti Suzuki      Swift  2017   83500.0       Manual  second   Diesel
4  Maruti Suzuki     Baleno  2019   45000.0    Automatic   first   Petrol

  PostedDate  AskPrice  Year_Posted  Month_Posted
0 2024-11-01    195000         2024            11
1 2024-07-01    375000         2024             7
2 2024-11-01    184999         2024            11
3 2024-11-01    565000         2024            11
4 2024-11-01    685000         2024            11
```

[180]:

```
data.drop(columns = 'PostedDate',axis=1, inplace=True)
```

[181]:

```
data.head()
```

[181]:

```
           Brand      model  Year  kmDriven Transmission   Owner FuelType  \
0          Honda       City  2001   98000.0       Manual  second   Petrol
1         Toyota     Innova  2009  190000.0       Manual  second   Diesel
2     Volkswagen  VentoTest  2010   77246.0       Manual   first   Diesel
3  Maruti Suzuki      Swift  2017   83500.0       Manual  second   Diesel
4  Maruti Suzuki     Baleno  2019   45000.0    Automatic   first   Petrol
```

```
   AskPrice  Year_Posted  Month_Posted
0    195000         2024            11
1    375000         2024             7
2    184999         2024            11
3    565000         2024            11
4    685000         2024            11
```

[182]:
```
data['Month_Posted'].value_counts()
```

[182]:
```
11    8693
10     616
9      145
8       63
7       29
6       18
4        6
5        5
12       3
2        2
1        1
3        1
Name: Month_Posted, dtype: int64
```

[183]:
```
data['Year_Posted'].value_counts()
```

[183]:
```
2024    9579
2023       3
Name: Year_Posted, dtype: int64
```

[184]:
```
# Replace 'Manual' with 0 and 'Automatic' with 1 in the 'Transmission' column
data['Transmission'] = data['Transmission'].replace({'Manual': 0, 'Automatic':␣
  ↪1})
```

[185]:
```
data.head()
```

[185]:
```
           Brand      model  Year  kmDriven  Transmission   Owner FuelType  \
0          Honda       City  2001   98000.0             0  second   Petrol
1         Toyota     Innova  2009  190000.0             0  second   Diesel
2     Volkswagen  VentoTest  2010   77246.0             0   first   Diesel
3  Maruti Suzuki      Swift  2017   83500.0             0  second   Diesel
4  Maruti Suzuki     Baleno  2019   45000.0             1   first   Petrol

   AskPrice  Year_Posted  Month_Posted
0    195000         2024            11
1    375000         2024             7
2    184999         2024            11
```

```
3    565000         2024             11
4    685000         2024             11
```

[186]:
```
data['FuelType'] = data['FuelType'].replace({'Diesel': 0, 'Petrol': 1, 'Hybrid/
 ↪CNG':2})
```

[187]:
```
data.head()
```

[187]:
```
           Brand      model  Year  kmDriven  Transmission   Owner  FuelType  \
0          Honda       City  2001   98000.0             0  second         1
1         Toyota     Innova  2009  190000.0             0  second         0
2     Volkswagen  VentoTest  2010   77246.0             0   first         0
3  Maruti Suzuki      Swift  2017   83500.0             0  second         0
4  Maruti Suzuki     Baleno  2019   45000.0             1   first         1

   AskPrice  Year_Posted  Month_Posted
0    195000         2024            11
1    375000         2024             7
2    184999         2024            11
3    565000         2024            11
4    685000         2024            11
```

[188]:
```
data['Owner'].value_counts()
```

[188]:
```
first     4800
second    4782
Name: Owner, dtype: int64
```

[189]:
```
data['Owner'] = data['Owner'].replace({'first': 0, 'second': 1})
```

[190]:
```
data.head()
```

[190]:
```
           Brand      model  Year  kmDriven  Transmission  Owner  FuelType  \
0          Honda       City  2001   98000.0             0      1         1
1         Toyota     Innova  2009  190000.0             0      1         0
2     Volkswagen  VentoTest  2010   77246.0             0      0         0
3  Maruti Suzuki      Swift  2017   83500.0             0      1         0
4  Maruti Suzuki     Baleno  2019   45000.0             1      0         1

   AskPrice  Year_Posted  Month_Posted
0    195000         2024            11
1    375000         2024             7
2    184999         2024            11
3    565000         2024            11
4    685000         2024            11
```

[191]:
```
data['model'].value_counts()
```

```
[191]: City                   330
       Wagon-R                311
       Swift                  283
       Creta                  260
       Ertiga                 249
                             …
       H5x                      1
       Punch                    1
       Tiguan All Space         1
       Cedia                    1
       Gran Turismo             1
       Name: model, Length: 400, dtype: int64
```

```
[192]: from sklearn.preprocessing import LabelEncoder
       # Initialize the LabelEncoder
       le = LabelEncoder()
```

```
[193]: # Convert the 'Model' column to numeric values
       data['model'] = le.fit_transform(data['model'])
```

```
[194]: data.head()
```

```
[194]:            Brand  model  Year  kmDriven  Transmission  Owner  FuelType  \
       0           Honda     84  2001   98000.0             0      1         1
       1          Toyota    187  2009  190000.0             0      1         0
       2      Volkswagen    347  2010   77246.0             0      0         0
       3   Maruti Suzuki    317  2017   83500.0             0      1         0
       4   Maruti Suzuki     52  2019   45000.0             1      0         1

          AskPrice  Year_Posted  Month_Posted
       0    195000         2024            11
       1    375000         2024             7
       2    184999         2024            11
       3    565000         2024            11
       4    685000         2024            11
```

```
[195]: data['Brand'] = le.fit_transform(data['Brand'])
```

```
[196]: data.head()
```

```
[196]:    Brand  model  Year  kmDriven  Transmission  Owner  FuelType  AskPrice  \
       0     12     84  2001   98000.0             0      1         1    195000
       1     36    187  2009  190000.0             0      1         0    375000
       2     37    347  2010   77246.0             0      0         0    184999
       3     23    317  2017   83500.0             0      1         0    565000
       4     23     52  2019   45000.0             1      0         1    685000
```

```
   Year_Posted  Month_Posted
0         2024            11
1         2024             7
2         2024            11
3         2024            11
4         2024            11
```

[197]:
```
# split the data into X and Y
x= data.drop(columns = 'AskPrice')
y = data['AskPrice']
```

[198]:
```
x.head()
```

[198]:
```
   Brand  model  Year  kmDriven  Transmission  Owner  FuelType  Year_Posted  \
0     12     84  2001   98000.0             0      1         1         2024
1     36    187  2009  190000.0             0      1         0         2024
2     37    347  2010   77246.0             0      0         0         2024
3     23    317  2017   83500.0             0      1         0         2024
4     23     52  2019   45000.0             1      0         1         2024

   Month_Posted
0            11
1             7
2            11
3            11
4            11
```

[199]:
```
y.head()
```

[199]:
```
0    195000
1    375000
2    184999
3    565000
4    685000
Name: AskPrice, dtype: int32
```

[200]:
```
# normalization

from sklearn.preprocessing import MinMaxScaler
# Initialize MinMaxScaler
scaler = MinMaxScaler()
```

[201]:
```
data_normalized = pd.DataFrame(scaler.fit_transform(data), columns=data.columns)
```

[202]:
```
data_normalized.head()
```

[202]:

```
      Brand     model      Year  kmDriven  Transmission  Owner  FuelType  \
0  0.315789  0.210526  0.394737  0.100000           0.0    1.0       0.5
1  0.947368  0.468672  0.605263  0.193877           0.0    1.0       0.0
2  0.973684  0.869674  0.631579  0.078822           0.0    0.0       0.0
3  0.605263  0.794486  0.815789  0.085204           0.0    1.0       0.0
4  0.605263  0.130326  0.868421  0.045918           1.0    0.0       0.5

   AskPrice  Year_Posted  Month_Posted
0  0.004237          1.0      0.909091
1  0.008474          1.0      0.545455
2  0.004001          1.0      0.909091
3  0.012946          1.0      0.909091
4  0.015770          1.0      0.909091
```

[203]:

```
from sklearn.model_selection import train_test_split
```

[204]:

```
X_train, X_test, y_train, y_test = train_test_split(x, y, test_size=0.2,␣
 ↪random_state=42)
```

[205]:

```
# Display the shapes of the resulting datasets
print("X_train shape:", X_train.shape)
print("X_test shape:", X_test.shape)
print("y_train shape:", y_train.shape)
print("y_test shape:", y_test.shape)
```

```
X_train shape: (7665, 9)
X_test shape: (1917, 9)
y_train shape: (7665,)
y_test shape: (1917,)
```

[206]:

```
#building the XgBoost model

!pip install xgboost
```

```
Requirement already satisfied: xgboost in c:\users\user\anaconda3\lib\site-
packages (2.1.2)
Requirement already satisfied: numpy in c:\users\user\anaconda3\lib\site-
packages (from xgboost) (1.24.4)
Requirement already satisfied: scipy in c:\users\user\anaconda3\lib\site-
packages (from xgboost) (1.10.1)
```

[207]:

```
import xgboost as xgb
from sklearn.metrics import mean_squared_error, r2_score
```

[208]:

```
# Initialize the XGBoost regressor model
xg_reg = xgb.XGBRegressor(objective='reg:squarederror', eval_metric='rmse',␣
 ↪random_state=42)
```

```

# Train the model
xg_reg.fit(X_train, y_train)

# Predict on the test set
y_pred = xg_reg.predict(X_test)
```

[209]:
```
y_pred
```

[209]:
```
array([ 346213.9 ,  302984.  , 1098718.4 , …,  661576.56,  108354.48,
        685818.9 ], dtype=float32)
```

[210]:
```
# Evaluate the model
rmse = mean_squared_error(y_test, y_pred, squared=False)
r2 = r2_score(y_test, y_pred)

# Output the results
print(f'Root Mean Squared Error (RMSE): {rmse}')
print(f'R2 Score: {r2}')
```

```
Root Mean Squared Error (RMSE): 769862.3129815074
R2 Score: 0.7788931419558567
```

[ ]:
```

```

[211]:
```
from sklearn.ensemble import AdaBoostRegressor
from sklearn.tree import DecisionTreeRegressor
```

[212]:
```
# Initialize the AdaBoost Regressor with a weak learner (Decision Tree)
adaboost_regressor =␣
 ↪AdaBoostRegressor(base_estimator=DecisionTreeRegressor(max_depth=2),␣
 ↪n_estimators=50, random_state=42)

# Train the model
adaboost_regressor.fit(X_train, y_train)
```

```
C:\Users\USER\anaconda3\Lib\site-packages\sklearn\ensemble\_base.py:166:
FutureWarning: `base_estimator` was renamed to `estimator` in version 1.2 and
will be removed in 1.4.
  warnings.warn(
```

[212]:
```
AdaBoostRegressor(base_estimator=DecisionTreeRegressor(max_depth=2),
                  random_state=42)
```

[213]:
```
# Predict on the test set
y_pred1 = adaboost_regressor.predict(X_test)
```

[214]:

```
y_pred1
```

[214]:

```
array([1062533.33455545, 1062533.33455545, 1386868.25885958, …,
       1386868.25885958, 1062533.33455545, 1062533.33455545])
```

[215]:

```
# Evaluate the model
rmse = mean_squared_error(y_test, y_pred, squared=False)  # Root Mean Squared␣
 ↪Error
r2 = r2_score(y_test, y_pred)  # R2 Score

# Print results
print(f'Root Mean Squared Error (RMSE): {rmse}')
print(f'R2 Score: {r2}')
```

```
Root Mean Squared Error (RMSE): 769862.3129815074
R2 Score: 0.7788931419558567
```

[ ]:

```

```

[ ]:

```

```