- Home
- About
- Contact

CARDEKHO

MACHINE LEARNING CASE STUDY PROJECT

Learn More

Play Video

Get Started »

SAYYED NIDA [COHORT 4]

Home About Contact

CARDEKHO.COM

CarDekho.com is a one-stop, tech-driven platform for car seekers in India—starting from in-depth research to final purchase—backed by a powerful ecosystem that includes vehicle listings, virtual tours, insurance, and finance. Founded in 2008, it's a unicorn-scale venture with robust multi-channel monetization.

Learn More

PROBLEM STATEMENT

The used car market in India is a dynamic and ever-changing landscape. Prices can fluctuate wildly based on a variety of factors including the make and model of the car, its mileage, its condition and the current market conditions. As a result, it can be difficult for sellers to accurately price their cars.

DATASET LINK

APPROACH

We propose to develop a machine learning model that can predict the price of a used car based on its features. The model will be trained on a dataset of used cars that have been sold on Cardekho.com in India. The model will then be able to be used to predict the price of any used car, given its features.

OBJECTIVE

To build suitable Machine Learning Model for Used Car Price Prediction.

STEPS TO BUILD A ML MODEL:

1. Understanding the problem statement
2. Data collection (Bringing all the data together)
3. Understanding the data (Description, checking null values)
4. Data cleaning
5. Exploratory data analysis (EDA)

IMPORTING ALL RELAVANT LIBRARIES

```
[ ]  # Importing all relavant libraries
     import pandas as pd
     import matplotlib.pyplot as plt
     import seaborn as sns
     import numpy as np
```

UNDERSTANDING THE DATA

```
[ ] data.shape
```

```
(15411, 14)
```

COLUMNS & ROWS

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 15411 entries, 0 to 15410
Data columns (total 14 columns):
 #   Column             Non-Null Count  Dtype  
---  ------             --------------  -----  
 0   Unnamed: 0         15411 non-null  int64  
 1   car_name           15411 non-null  object 
 2   brand              15411 non-null  object 
 3   model              15411 non-null  object 
 4   vehicle_age        15411 non-null  int64  
 5   km_driven          15411 non-null  int64  
 6   seller_type        15411 non-null  object 
 7   fuel_type          15411 non-null  object 
 8   transmission_type  15411 non-null  object 
 9   mileage            15411 non-null  float64
 10  engine             15411 non-null  int64  
 11  max_power          15411 non-null  float64
 12  seats              15411 non-null  int64  
 13  selling_price      15411 non-null  int64  
dtypes: float64(2), int64(6), object(6)
memory usage: 1.6+ MB
```

NO NULL VALUES

EXPLORATORY DATA ANALYSIS

Univariate analysis
When we analyze each column one by one

Feature engineering
the process of selecting, transforming, and creating new features from raw data to improve the performance of machine learning models

Home About Contact

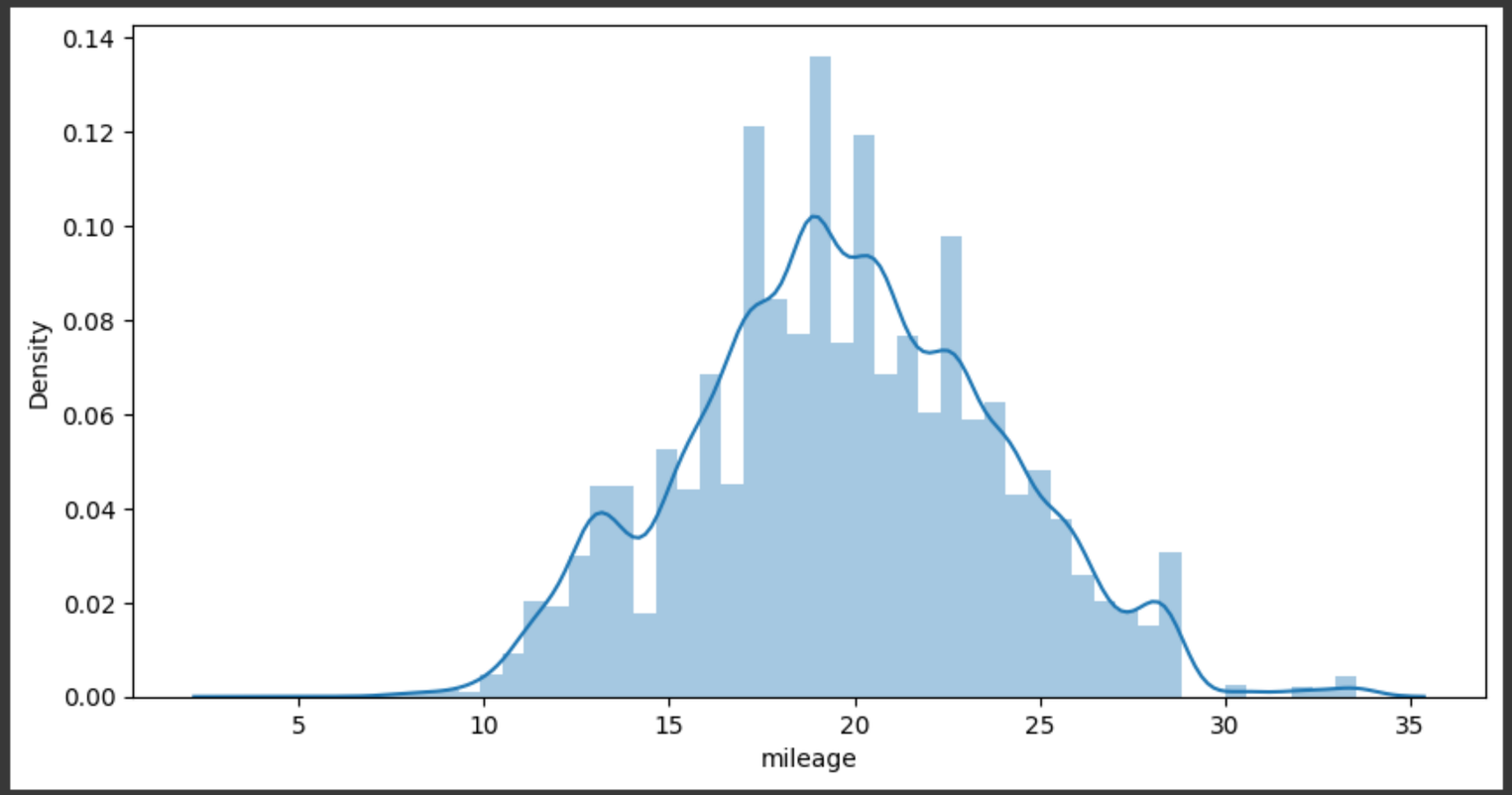
UNIVARIATE ANALYSIS

CREATE A HISTOGRAM (FREQUENCY DISTRIBUTION GRAPH) FOR NUMERICAL COLUMNS

```
#Create a Histogram (Frequency distribution graph) for Numerical columns

plt.figure(figsize = [10,5])
sns.distplot(data['mileage'])

plt.show()
```



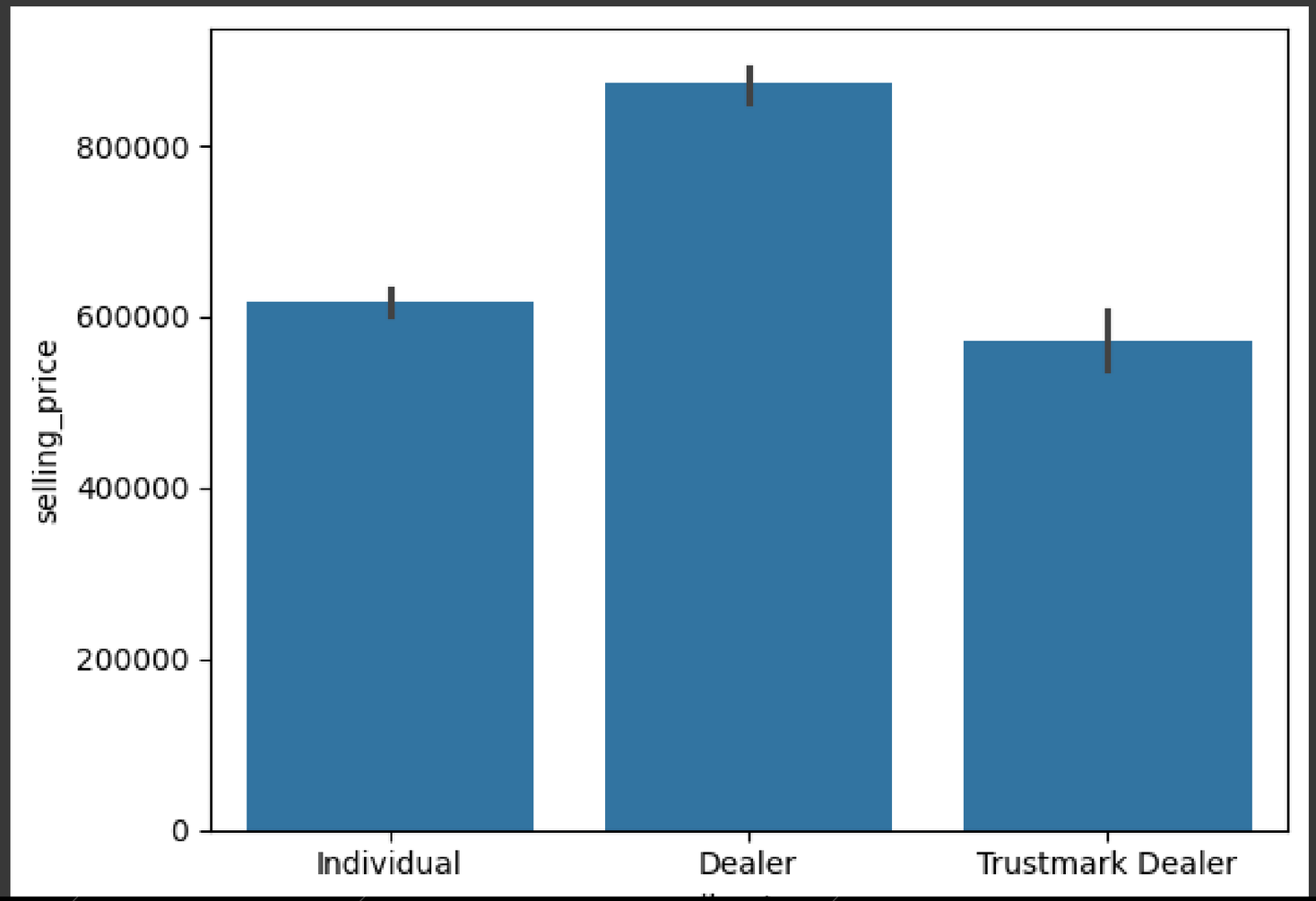
INSIGHTS

- The majority of cars in the dataset have a mileage between approximately 15 kmpl and 25 kmpl, with the distribution peaking around 20 kmpl. There are fewer cars with very low or very high mileage.

CREATE A BARPLOT FOR NUMERICAL COLUMNS

```
[ ]  sns.barplot(x = data['seller_type'],y = data['selling_price'])

     plt.show()
```



INSIGHTS

- Cars sold by 'Dealer' generally have a higher average selling price compared to those sold by 'Individual' sellers. The average selling price for 'Trustmark Dealer' appears to be somewhere between 'Dealer' and 'Individual'

Home About Contact

FEATURE ENGINEERING

```
#Feature engineering

data.head()
model_data = data.copy()

model_data.drop(labels = ['car_name','brand','model','Unnamed: 0'], axis = 1, inplace = True)

model_data
```

	vehicle_age	km_driven	seller_type	fuel_type	transmission_type	mileage	engine	max_power	seats	selling_price
0	9	120000	Individual	Petrol	Manual	19.70	796	46.30	5	120000
1	5	20000	Individual	Petrol	Manual	18.90	1197	82.00	5	550000
2	11	60000	Individual	Petrol	Manual	17.00	1197	80.00	5	215000
3	9	37000	Individual	Petrol	Manual	20.92	998	67.10	5	226000
4	6	30000	Dealer	Diesel	Manual	22.77	1498	98.59	5	570000
...	...	...	...	...	...	...	...	...	...	...
15406	9	10723	Dealer	Petrol	Manual	19.81	1086	68.05	5	250000
15407	2	18000	Dealer	Petrol	Manual	17.50	1373	91.10	7	925000
15408	6	67000	Dealer	Diesel	Manual	21.14	1498	103.52	5	425000
15409	5	3800000	Dealer	Diesel	Manual	16.00	2179	140.00	7	1225000
15410	2	13000	Dealer	Petrol	Automatic	18.00	1497	117.60	5	1200000

15411 rows × 10 columns

Car_dekho Linear Regression.ipynb ☆ Changes will not be saved

File Edit View Insert Runtime Tools Help

Share Gemini N

Commands + Code + Text Run all Copy to Drive Connect

```
[ ] one hot encoding. dummies


    Transmission type :

    Petrol 1
    Diesel 0
    Electric 2
```

```
[ ] model_data = pd.get_dummies(model_data,dtype = float)

    model_data
```

	vehicle_age	km_driven	mileage	engine	max_power	seats	selling_price	seller_type_Dealer	seller_type_Individual	seller_type_Trustmark Dealer	fuel_type_CNG	fuel_type_Diesel	fuel_type_Electric	fuel_type_LPG	fuel_type_Petrol	t
0	9	120000	19.70	796	46.30	5	120000	0.0	1.0	0.0	0.0	0.0	0.0	0.0	1.0	
1	5	20000	18.90	1197	82.00	5	550000	0.0	1.0	0.0	0.0	0.0	0.0	0.0	1.0	
2	11	60000	17.00	1197	80.00	5	215000	0.0	1.0	0.0	0.0	0.0	0.0	0.0	1.0	
3	9	37000	20.92	998	67.10	5	226000	0.0	1.0	0.0	0.0	0.0	0.0	0.0	1.0	
4	6	30000	22.77	1498	98.59	5	570000	1.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
15406	9	10723	19.81	1086	68.05	5	250000	1.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	
15407	2	18000	17.50	1373	91.10	7	925000	1.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	
15408	6	67000	21.14	1498	103.52	5	425000	1.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	
15409	5	3800000	16.00	2179	140.00	7	1225000	1.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	
15410	2	13000	18.00	1497	117.60	5	1200000	1.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	

15411 rows × 17 columns

Home About Contact

Lets create independent and dependent variables

```
x = model_data.drop('selling_price',axis = 1)

x
```

```
y = model_data['selling_price']

y
```

independent variables (features) are the inputs used to predict an outcome, while the dependent variable (target) is the outcome we are trying to predict

Home About Contact

'''Model building: Train-test split'''

```
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(x,y,test_size = 0.2)
#80% of data will be used in training and 20 % will be used in Testing
```

```python
#Import libraries for Model Building

from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_absolute_error,r2_score

Regressor = LinearRegression().fit(x_train,y_train)

pred = Regressor.predict(x_test)

# Evaluation of the model


r_square = r2_score(y_true = y_test,y_pred = pred)
```

```
[ ] r_square

0.530661637160916
```

```
pred

array([ 423825.77755564, 2935009.24577014,  838308.67127006, ...,
       1992224.98994278, 2093721.91521314, -190963.57873385])
```

Machine Learning Model

```
[ ] x_test['predicted price'] = pred
```

```
x_test
```

	vehicle_age	km_driven	mileage	engine	max_power	seats	seller_type_Dealer	seller_type_Individual	seller_type_Trustmark Dealer	fuel_type_CNG	fuel_type_Diesel	fuel_type_Electric	fuel_type_LPG	fuel_type_Petrol	transmission_typ
14442	8	54000	14.00	2523	75.00	7	0.0	1.0	0.0	0.0	1.0	0.0	0.0	0.0	
285	3	13000	12.19	1997	237.36	4	1.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	
13167	2	10000	23.84	1199	84.00	5	0.0	1.0	0.0	0.0	0.0	0.0	0.0	1.0	
8069	7	77000	12.55	2982	168.50	7	1.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	
1815	3	78000	26.08	1462	91.19	7	1.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
8523	8	104500	22.32	1582	126.30	5	0.0	1.0	0.0	0.0	1.0	0.0	0.0	0.0	
14684	3	78000	24.70	1498	98.63	5	1.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	
11718	7	89000	18.88	1995	184.00	5	1.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	
6657	8	50000	19.62	1998	192.00	5	1.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	
12899	8	52000	22.74	796	47.30	5	1.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	

3083 rows × 17 columns

INSIGHTS

Relationships between variables:

- The bar plot of seller_type vs selling_price suggests that cars sold by 'Dealer' and 'Trustmark Dealer' generally have higher selling prices than those sold by 'Individual' sellers.

Model Performance:

- The **R-squared value of approximately 0.53** indicates that the linear regression model explains about 53% of the variance in the selling price. This suggests that the model captures some of the relationships between the features and the selling price, but there is still a significant portion of the variance that is not explained by the model. This could be due to various factors, such as missing important features, non-linear relationships, or noise in the data.

THANK YOU

I welcome and appreciate your thoughts about this project.
Always open to suggestions, let’s connect!