

**Project Report**

Bachelor Of Computer Applications

2nd Semester

Exploratory Data Analysis Project

**Global Product Inventory Dataset 2025 Analysis**

**By**

**Name :** Amaraiah.C

**Reg No. :** 2411021240004

**Github Link :** https://github.com/Amaraiah11/IDS-Project-SEM-2/upload

Department Of Computer Application

Alliance University

Chandrapur a- Anekal Main Road,Anekal

Bengaluru

**Introduction:**

In today’s rapidly evolving automotive market, the **used two-wheeler segment** has shown significant growth, fueled by rising consumer demand for cost-effective mobility. Buyers and sellers alike need accurate insights into the fair market value of used bikes.

This project focuses on predicting the **resale price of used motorcycles** using a supervised machine learning approach.

**Dataset Source:**

The dataset used in this project was obtained from Kaggle and is titled **“Used Bikes Dataset”**. It consists of various features that influence the price of second-hand motorcycles, including:

* Brand of the bike (e.g., Bajaj, Honda, Royal Enfield)
* City where the bike is listed
* Bike age
* Kilometers driven
* Engine power (cc)
* Ownership history (e.g., First Owner, Second Owner)
* Selling price

**Problem Statement:**

"Given the specifications and condition of a used motorcycle, can we accurately predict its selling price?"

This is framed as a regression problem since the target variable (price) is continuous in nature.

**Project Objective:**

To build a robust machine learning model that can:

* Understand the key factors influencing used bike prices.
* Predict the resale price of a bike based on available features.
* Help buyers/sellers make data-driven decisions.

**Proposed Solution:**

1. Clean and preprocess the raw dataset to handle missing values, duplicates, and outliers.
2. Perform Exploratory Data Analysis (EDA) to uncover trends and relationships.
3. Use feature engineering and one-hot encoding fo
4. r categorical variables.
5. Build and train a Linear Regression model to predict bike prices.
6. Evaluate the model using performance metrics like RMSE and R² score.
7. Visualize actual vs predicted values to understand model effectiveness.

**Conclusion & Insights**

In this project, we successfully built a Linear Regression model to predict the resale price of used motorcycles using a dataset sourced from Kaggle. Through thorough data cleaning, preprocessing, and exploratory analysis, we uncovered important trends and patterns that impact pricing in the second-hand two-wheeler market.

**Key Insights:**

* Power (cc) and age of the bike are strong predictors of resale value. As expected, newer and more powerful bikes fetch higher prices.
* Kilometres driven negatively impacts price — bikes with higher mileage tend to be cheaper.
* Brand and city also play an important role; bikes from premium brands and metro cities are usually priced higher.
* Ownership history affects value — first-owner bikes typically command better resale prices.

**Model Performance**:

* The Linear Regression model achieved a good fit, with an R² score of approximately 0.87, indicating that around 87% of the variance in bike prices is explained by the selected features.
* The Root Mean Squared Error (RMSE) was within a reasonable range, suggesting accurate predictions for most test cases.

**Coding**

import pandas as pd

df=pd.read\_csv(r"C:\Users\amara\Downloads\Used\_Bikes.csv.zip") df

bike\_name price city

kms\_driven \

1. TVS Star City Plus Dual Tone 110cc 35000.0 Ahmedabad

17654.0

1. Royal Enfield Classic 350cc 119900.0 Delhi

11000.0

1. Triumph Daytona 675R 600000.0 Delhi 110.0
2. TVS Apache RTR 180cc 65000.0 Bangalore

16329.0

1. Yamaha FZ S V 2.0 150cc-Ltd. Edition 80000.0 Bangalore

10000.0

... ... ... ...

...

1. Hero Passion Pro 100cc 39000.0 Delhi

22000.0

1. TVS Apache RTR 180cc 30000.0 Karnal

6639.0

1. Bajaj Avenger Street 220 60000.0 Delhi

20373.0

1. Hero Super Splendor 125cc 15600.0 Jaipur

84186.0

1. Bajaj Pulsar 150cc 22000.0 Pune

60857.0

owner age power brand 0 First Owner 3.0 110.0 TVS

1. First Owner 4.0 350.0 Royal Enfield
2. First Owner 8.0 675.0 Triumph
3. First Owner 4.0 180.0 TVS
4. First Owner 3.0 150.0 Yamaha ... ... ... ... ...
5. First Owner 4.0 100.0 Hero
6. First Owner 9.0 180.0 TVS
7. First Owner 6.0 220.0 Bajaj
8. First Owner 16.0 125.0 Hero
9. First Owner 13.0 150.0 Bajaj

[32648 rows x 8 columns] df.describe()

price kms\_driven age power count 3.264800e+04 32648.000000 32648.000000 32648.000000 mean 6.829542e+04 26344.625184 8.048211 213.511302

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| |  | | --- | | std 9.071860e+04 22208.527695 4.031700 134.428868 min 4.400000e+03 1.000000 1.000000 100.000000 25% 2.500000e+04 12000.000000 5.000000 150.000000  50% 4.300000e+04 20373.000000 7.000000 150.000000 |   75% 8.000000e+04 35000.000000 10.000000 220.000000  max 1.900000e+06 750000.000000 63.000000 1800.000000  df.info()  <class 'pandas.core.frame.DataFrame'>  RangeIndex: 32648 entries, 0 to 32647 Data columns (total 8 columns):  # Column Non-Null Count Dtype  --- ------ -------------- -----   1. bike\_name 32648 non-null object 2. price 32648 non-null float64 3. city 32648 non-null object 4. kms\_driven 32648 non-null float64 5. owner 32648 non-null object 6. age 32648 non-null float64 7. power 32648 non-null float64 7 brand 32648 non-null object   dtypes: float64(4), object(4) memory usage: 2.0+ MB df.isna()   |  |  | | --- | --- | | bike\_name price city kms\_driven owner age power brand 0 False False False False False False False False   1. False False False False False False False False 2. False False False False False False False False 3. False False False False False False False False 4. False False False False False False False False... ... ... ... ... ... ... ... ... 5. False False False False False False False False 6. False False False False False False False False 7. False False False False False False False False 8. False False False False False False False False 9. False False False False False False False False | | | [32648 rows x 8 columns] |   df.isnull().sum()   |  | | --- | | bike\_name 0 price 0 city 0 kms\_driven 0 owner 0 age 0 power 0 | |

brand 0 dtype: int64 df.head(10)

bike\_name price city

kms\_driven \

1. TVS Star City Plus Dual Tone 110cc 35000.0 Ahmedabad

17654.0

1. Royal Enfield Classic 350cc 119900.0 Delhi

11000.0

1. Triumph Daytona 675R 600000.0 Delhi 110.0
2. TVS Apache RTR 180cc 65000.0 Bangalore

16329.0

1. Yamaha FZ S V 2.0 150cc-Ltd. Edition 80000.0 Bangalore

10000.0

1. Yamaha FZs 150cc 53499.0 Delhi

25000.0

1. Honda CB Hornet 160R ABS DLX 85000.0 Delhi

8200.0

1. Hero Splendor Plus Self Alloy 100cc 45000.0 Delhi

12645.0

1. Royal Enfield Thunderbird X 350cc 145000.0 Bangalore

9190.0

1. Royal Enfield Classic Desert Storm 500cc 88000.0 Delhi

19000.0

owner age power brand 0 First Owner 3.0 110.0 TVS

1. First Owner 4.0 350.0 Royal Enfield
2. First Owner 8.0 675.0 Triumph
3. First Owner 4.0 180.0 TVS
4. First Owner 3.0 150.0 Yamaha
5. First Owner 6.0 150.0 Yamaha
6. First Owner 3.0 160.0 Honda
7. First Owner 3.0 100.0 Hero
8. First Owner 3.0 350.0 Royal Enfield 9 Second Owner 7.0 500.0 Royal Enfield df.tail(10)

bike\_name price city kms\_driven

\

1. Yamaha Fazer 25 250cc 123000.0 Kadapa 14500.0
2. Royal Enfield Classic 350cc 95500.0 Delhi 18000.0
3. Hero Passion Pro 100cc 32000.0 Delhi 12000.0
4. Bajaj Avenger 220cc 41000.0 Delhi 20245.0

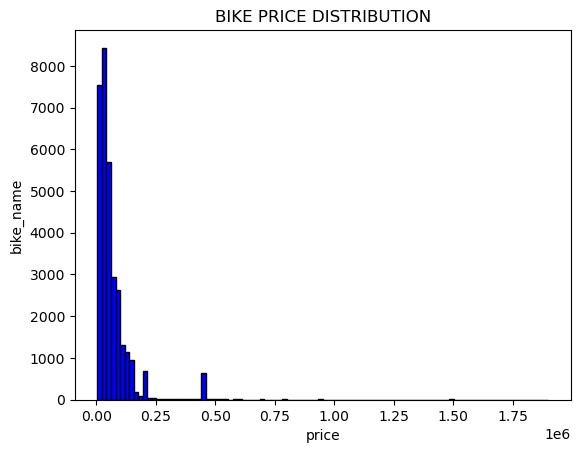
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1. Hero Passion 100cc 15000.0 Perumbavoor 35000.0 2. Hero Passion Pro 100cc 39000.0 Delhi 22000.0 3. TVS Apache RTR 180cc 30000.0 Karnal 6639.0 4. Bajaj Avenger Street 220 60000.0 Delhi 20373.0 5. Hero Super Splendor 125cc 15600.0 Jaipur 84186.0 32647 Bajaj Pulsar 150cc 22000.0 Pune 60857.0  |  | | --- | | owner age power brand 32638 First Owner 4.0 250.0 Yamaha   1. First Owner 8.0 350.0 Royal Enfield 2. First Owner 6.0 100.0 Hero 3. Second Owner 11.0 220.0 Bajaj 4. Second Owner 19.0 100.0 Hero 5. First Owner 4.0 100.0 Hero 6. First Owner 9.0 180.0 TVS 7. First Owner 6.0 220.0 Bajaj 8. First Owner 16.0 125.0 Hero 9. First Owner 13.0 150.0 Bajaj |   df.duplicated()   |  |  |  | | --- | --- | --- | | 1. False 2. False 3. False 4. False 5. False | |  | | .. | . | | 1. True 2. True 3. True 4. True 5. True | | | Length: 32648, dtype: bool | | |   df.drop\_duplicates(inplace=True)  df*#lots of rows are dropped because many bikes are same modle*   |  |  |  | | --- | --- | --- | | bike\_name price city | | | | kms\_driven \ | |  | | 0 TVS Star City Plus Dual Tone 110cc 35000.0 Ahmedabad | | | | 17654.0 |  | | | 1 Royal Enfield Classic 350cc 119900.0 Delhi | | | | 11000.0 |  | | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1. Triumph Daytona 675R 600000.0 Delhi   110.0   1. TVS Apache RTR 180cc 65000.0 Bangalore   16329.0   1. Yamaha FZ S V 2.0 150cc-Ltd. Edition 80000.0 Bangalore   10000.0  ... ... ... ...  ...  9362 Hero Hunk Rear Disc 150cc 25000.0 Delhi  48587.0   1. Bajaj Avenger 220cc 35000.0 Bangalore   60000.0   1. Harley-Davidson Street 750 ABS 450000.0 Jodhpur   3430.0   1. Bajaj Dominar 400 ABS 139000.0 Hyderabad   21300.0   1. Bajaj Avenger Street 220 80000.0 Hyderabad   7127.0   |  |  | | --- | --- | | owner age power brand 0 First Owner 3.0 110.0 TVS   1. First Owner 4.0 350.0 Royal Enfield 2. First Owner 8.0 675.0 Triumph 3. First Owner 4.0 180.0 TVS 4. First Owner 3.0 150.0 Yamaha ... ... ... ... ...   9362 First Owner 8.0 150.0 Hero   1. First Owner 9.0 220.0 Bajaj 2. First Owner 4.0 750.0 Harley-Davidson 3. First Owner 4.0 400.0 Bajaj 4. First Owner 5.0 220.0 Bajaj | | | [7324 rows x 8 columns] |   df.columns=df.columns.str.strip() df   |  |  |  |  | | --- | --- | --- | --- | | bike\_name price city | | | | | kms\_driven \ | | |  | | 0 TVS Star City Plus Dual Tone 110cc 35000.0 Ahmedabad | | | | | 17654.0 | |  | | | 1 Royal Enfield Classic 350cc 119900.0 Delhi | | | | | 11000.0 | |  | | | 2 Triumph Daytona 675R 600000.0 Delhi | | | | | 110.0 |  | | | | 3 TVS Apache RTR 180cc 65000.0 Bangalore | | | | | 16329.0 | |  | | | 4 Yamaha FZ S V 2.0 150cc-Ltd. Edition 80000.0 Bangalore | | | | | 10000.0 | |  | | |

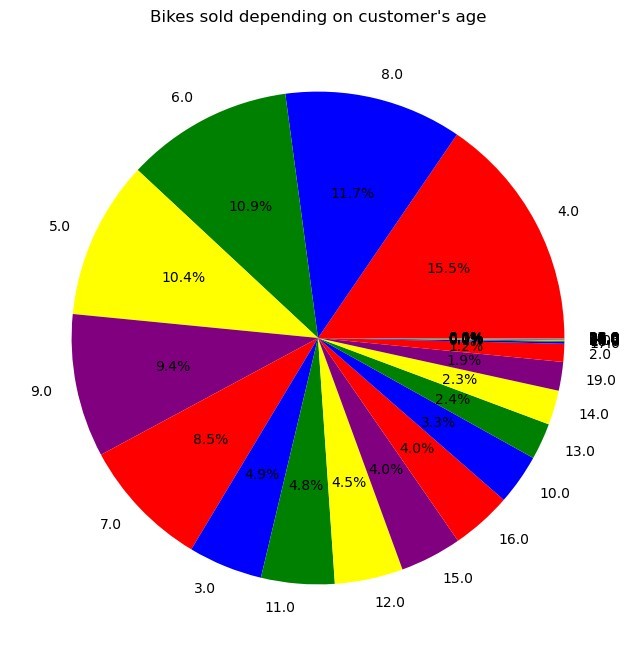
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ... ... ... ...  ...  9362 Hero Hunk Rear Disc 150cc 25000.0 Delhi  48587.0   1. Bajaj Avenger 220cc 35000.0 Bangalore   60000.0   1. Harley-Davidson Street 750 ABS 450000.0 Jodhpur   3430.0   1. Bajaj Dominar 400 ABS 139000.0 Hyderabad   21300.0   1. Bajaj Avenger Street 220 80000.0 Hyderabad   7127.0   |  |  | | --- | --- | | owner age power brand 0 First Owner 3.0 110.0 TVS   1. First Owner 4.0 350.0 Royal Enfield 2. First Owner 8.0 675.0 Triumph 3. First Owner 4.0 180.0 TVS 4. First Owner 3.0 150.0 Yamaha ... ... ... ... ...   9362 First Owner 8.0 150.0 Hero   1. First Owner 9.0 220.0 Bajaj 2. First Owner 4.0 750.0 Harley-Davidson 3. First Owner 4.0 400.0 Bajaj 4. First Owner 5.0 220.0 Bajaj | | | [7324 rows x 8 columns] |   df=pd.read\_csv(r"C:\Users\amara\Downloads\Used\_Bikes.csv.zip")  df.columns =df.columns.str.strip() print(df.columns)   |  |  |  |  | | --- | --- | --- | --- | | Index(['bike\_name', 'price', 'city', 'kms\_driven', 'owner', 'age', | | | | | 'power', |  | | | | 'brand'], | |  | | | dtype='object') | | |  |   df.columns =df.columns.str.strip() df["price"]= df["price"].astype(int) df   |  |  |  |  | | --- | --- | --- | --- | | bike\_name price city | | | | | kms\_driven \ | | |  | | 0 TVS Star City Plus Dual Tone 110cc 35000 Ahmedabad | | | | | 17654.0 | |  | | | 1 Royal Enfield Classic 350cc 119900 Delhi | | | | | 11000.0 | |  | | | 2 Triumph Daytona 675R 600000 Delhi | | | | | 110.0 |  | | | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1. TVS Apache RTR 180cc 65000 Bangalore   16329.0   1. Yamaha FZ S V 2.0 150cc-Ltd. Edition 80000 Bangalore   10000.0  ... ... ... ...  ...   1. Hero Passion Pro 100cc 39000 Delhi   22000.0   1. TVS Apache RTR 180cc 30000 Karnal   6639.0   1. Bajaj Avenger Street 220 60000 Delhi   20373.0   1. Hero Super Splendor 125cc 15600 Jaipur   84186.0   1. Bajaj Pulsar 150cc 22000 Pune   60857.0   |  |  | | --- | --- | | owner age power brand 0 First Owner 3.0 110.0 TVS   1. First Owner 4.0 350.0 Royal Enfield 2. First Owner 8.0 675.0 Triumph 3. First Owner 4.0 180.0 TVS 4. First Owner 3.0 150.0 Yamaha ... ... ... ... ... 5. First Owner 4.0 100.0 Hero 6. First Owner 9.0 180.0 TVS 7. First Owner 6.0 220.0 Bajaj 8. First Owner 16.0 125.0 Hero 9. First Owner 13.0 150.0 Bajaj | | | [32648 rows x 8 columns] |   df.columns =df.columns.str.strip() df["price"]= df["price"].astype(float) df   |  |  |  |  | | --- | --- | --- | --- | | bike\_name price city | | | | | kms\_driven \ | | |  | | 0 TVS Star City Plus Dual Tone 110cc 35000.0 Ahmedabad | | | | | 17654.0 | |  | | | 1 Royal Enfield Classic 350cc 119900.0 Delhi | | | | | 11000.0 | |  | | | 2 Triumph Daytona 675R 600000.0 Delhi | | | | | 110.0 |  | | | | 3 TVS Apache RTR 180cc 65000.0 Bangalore | | | | | 16329.0 | |  | | | 4 Yamaha FZ S V 2.0 150cc-Ltd. Edition 80000.0 Bangalore | | | | | 10000.0 | |  | | | ... ... ... ... | | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| ...   1. Hero Passion Pro 100cc 39000.0 Delhi   22000.0   1. TVS Apache RTR 180cc 30000.0 Karnal   6639.0   1. Bajaj Avenger Street 220 60000.0 Delhi   20373.0   1. Hero Super Splendor 125cc 15600.0 Jaipur   84186.0   1. Bajaj Pulsar 150cc 22000.0 Pune   60857.0   |  |  | | --- | --- | | owner age power brand 0 First Owner 3.0 110.0 TVS   1. First Owner 4.0 350.0 Royal Enfield 2. First Owner 8.0 675.0 Triumph 3. First Owner 4.0 180.0 TVS 4. First Owner 3.0 150.0 Yamaha ... ... ... ... ... 5. First Owner 4.0 100.0 Hero 6. First Owner 9.0 180.0 TVS 7. First Owner 6.0 220.0 Bajaj 8. First Owner 16.0 125.0 Hero 9. First Owner 13.0 150.0 Bajaj | | | [32648 rows x 8 columns] |   import matplotlib.pyplot as plt  df=pd.read\_csv(r"C:\Users\amara\Downloads\Used\_Bikes.csv.zip") plt.hist(df['price'],bins=100,color='Blue',edgecolor="Black" ) plt.xlabel("price") plt.ylabel("bike\_name")  plt.title("BIKE PRICE DISTRIBUTION") plt.show() |



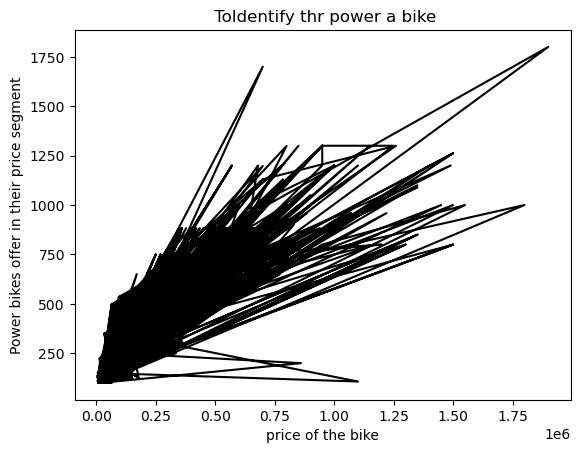
|  |
| --- |
| df=pd.read\_csv(r"C:\Users\amara\Downloads\Used\_Bikes.csv.zip")  ratings\_count = df['age'].value\_counts() plt.figure(figsize=(8, 8))  plt.pie(ratings\_count, labels=ratings\_count.index, autopct='%1.1f%%', colors=['red', 'blue', 'green', 'yellow', 'purple']) plt.title("Bikes sold depending on customer's age") plt.show() |



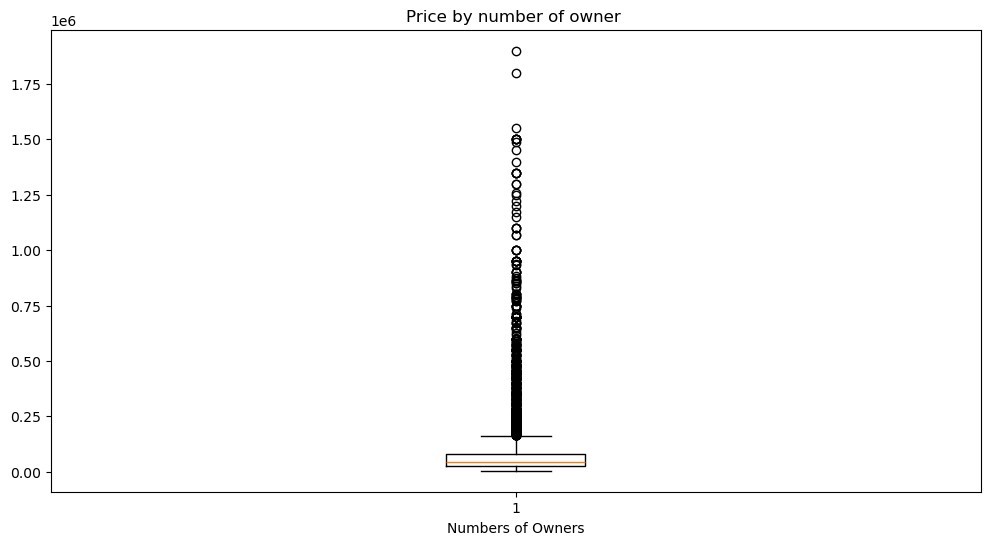
plt.plot(df['price'],df['power'],color='Black')

plt.xlabel("price of the bike")

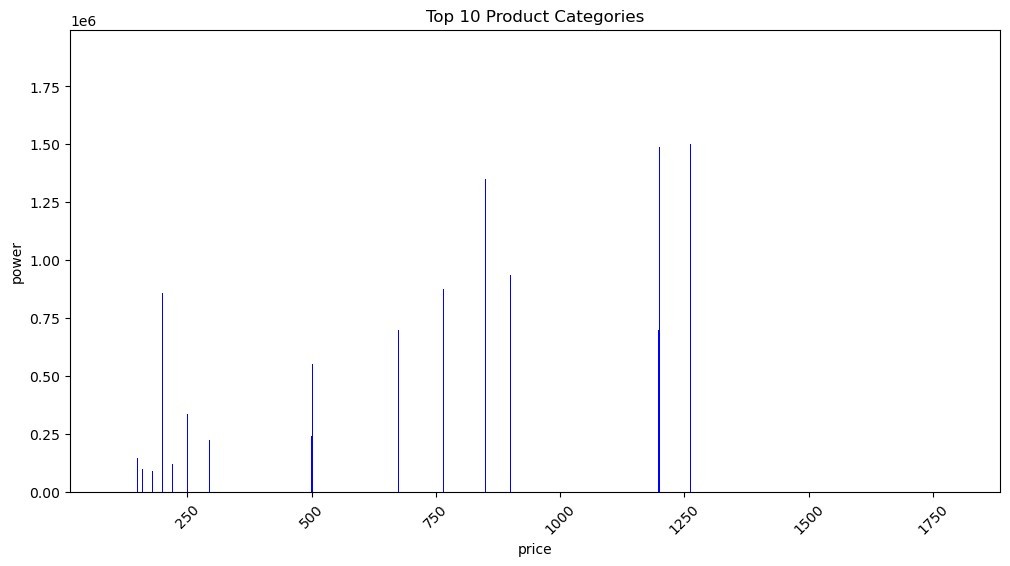
plt.ylabel("Power bikes offer in their price segment") plt.title(" ToIdentify thr power a bike") plt.show()



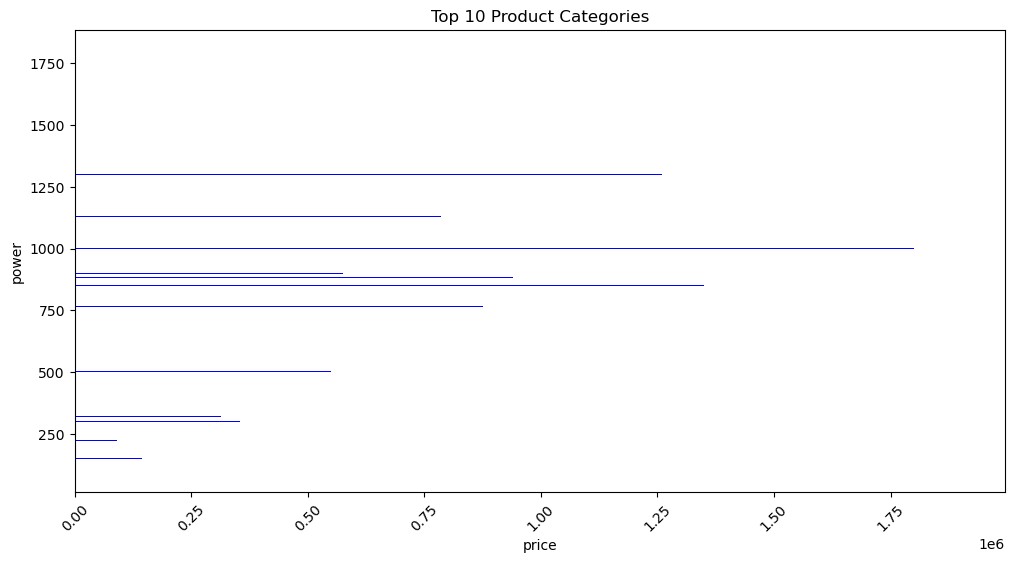
plt.figure(figsize=(12,6)) plt.boxplot(df['price']) plt.xlabel("Numbers of Owners") plt.title("Price by number of owner ") plt.show()



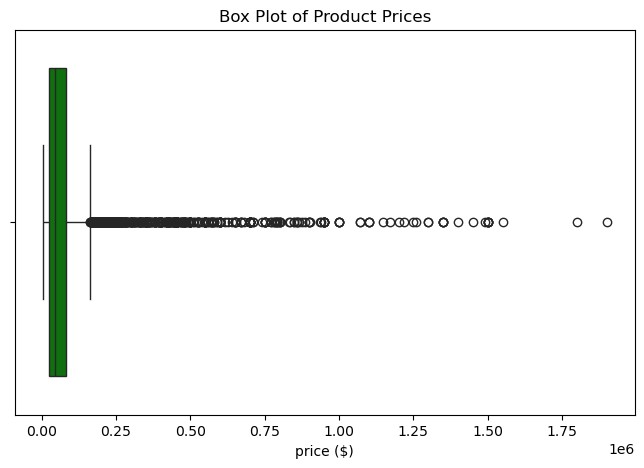
|  |  |
| --- | --- |
| print(df.shape)   |  | | --- | | (32648, 8) |   import matplotlib.pyplot as plt  plt.figure(figsize=(12, 6))  plt.bar(df["power"],df["price"], color='blue') plt.xticks(rotation=45) plt.xlabel("price") plt.ylabel("power")  plt.title("Top 10 Product Categories") plt.show() |



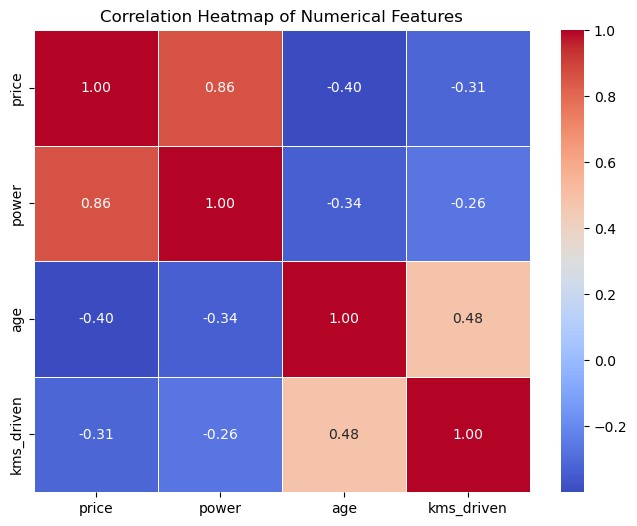
|  |
| --- |
| import matplotlib.pyplot as plt  plt.figure(figsize=(12, 6))  plt.barh(df["power"],df["price"], color='blue')  plt.xticks(rotation=45) plt.xlabel("price") plt.ylabel("power")  plt.title("Top 10 Product Categories") plt.show() |



|  |
| --- |
| import matplotlib.pyplot as plt import seaborn as sns  plt.figure(figsize=(8, 5))  sns.boxplot(x=df["price"], color="green") plt.title("Box Plot of Product Prices") plt.xlabel("price ($)") plt.show() |

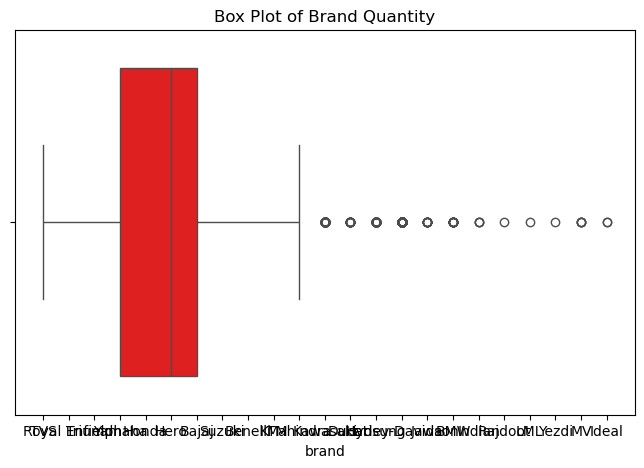


|  |
| --- |
| import matplotlib.pyplot as plt import seaborn as sns  *# Compute correlation matrix*  correlation\_matrix = df[["price", "power", "age", "kms\_driven"]].corr()  *# Plot heatmap*  plt.figure(figsize=(8, 6))  sns.heatmap(correlation\_matrix, annot=True, cmap="coolwarm", fmt=".2f", linewidths=0.5)  plt.title("Correlation Heatmap of Numerical Features") plt.show() |



|  |  |
| --- | --- |
| df.describe().T  from sklearn.model\_selection import train\_test\_split from sklearn.linear\_model import LinearRegression  *# Split data*  X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)  *# Train model*  model = LinearRegression() model.fit(X\_train, y\_train)   |  | | --- | | LinearRegression() |   from sklearn.metrics import mean\_squared\_error, r2\_score import numpy as np  *# Predictions*  y\_pred = model.predict(X\_test) |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *# Evaluation metrics*  rmse = np.sqrt(mean\_squared\_error(y\_test, y\_pred)) r2 = r2\_score(y\_test, y\_pred)  print(f"RMSE: ₹{rmse:.2f}") print(f"R² Score: {r2:.4f}")   |  |  | | --- | --- | | RMSE: ₹1799702077.85 |  | | R² Score: -371886386.3628 | |   plt.figure(figsize=(8, 5))  sns.boxplot(x=df["brand"], color="red") plt.title("Box Plot of Brand Quantity") plt.xlabel("brand") plt.show() |



|  |
| --- |
| import pandas as pd import numpy as np  import matplotlib.pyplot as plt import seaborn as sns  from sklearn.model\_selection import train\_test\_split |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| from sklearn.linear\_model import LinearRegression from sklearn.metrics import mean\_squared\_error, r2\_score  *# ------------------ Step 1: Preprocessing ------------------*  *# One-hot encode categorical variables*  df\_encoded = pd.get\_dummies(df, columns=['city', 'owner', 'brand'], drop\_first=True)  *# Define features and target*  X = df\_encoded.drop(columns='price') y = df\_encoded['price']  *# Split into train and test sets*  X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)  *# ------------------ Step 2: Model Training -----------------*model = LinearRegression() model.fit(X\_train, y\_train)  *# ------------------ Step 3: Evaluation -----------------*y\_pred = model.predict(X\_test)  rmse = np.sqrt(mean\_squared\_error(y\_test, y\_pred)) r2 = r2\_score(y\_test, y\_pred)  print(f"Root Mean Squared Error (RMSE): ₹{rmse:.2f}") print(f"R² Score: {r2:.4f}")  *# ------------------ Step 4: Actual vs Predicted Plot*  *-----------------*plt.figure(figsize=(8, 5))  plt.scatter(y\_test, y\_pred, alpha=0.6, color='teal')  plt.plot([y\_test.min(), y\_test.max()], [y\_test.min(), y\_test.max()],  'r--')  plt.xlabel("Actual Price") plt.ylabel("Predicted Price")  plt.title("Actual vs Predicted Bike Prices") plt.tight\_layout()  plt.show()   |  |  | | --- | --- | | Root Mean Squared Error (RMSE): ₹1799702077.85 | | | R² Score: -371886386.3628 |  | |

