Regenesys School of Technology

Post Graduate Diploma in Data Science

Academic Year 2024-25

Class/Intak:	**Mini Project – Post Graduate Diploma in Data Science**
Registration Number:	**Reg1516369**
Name	**Lesego Dlamini**

I. **Title:** *Exploratory Data Analysis of Car Price dataset in Python.*

II. **Introduction:** *The purpose of this mini project is to perform Exploratory Data Analysis (EDA) on the provided car_price_dataset (given as a csv file) in order to understand the correlation or relationship between selling price and different car features such as vehicle age, milage, km driven, fuel type, transmission type, engine size, max power , number of seats etc. EDA plays a crucial role in understanding the structure of the dataset, identifying patterns, detecting anomalies (such as outliers) and preparing the data for future modeling or analysis. In Data Science, EDA reveals insights in our data that guide us in feature selection, data cleaning and model building.*

This report documents the EDA process carried out using Python with interpretations and conclusions drawn at each stage of analysis.

III. **Problem Statement:** *Perform Exploratory Data Analysis (EDA) on the given dataset using Python libraries such as Numpy, Pandas, and Matplotlib/Seaborn. The goal is to uncover underlying patterns, relationships, and insights from the data. Additionally, document your process with Python comments explaining your code, and for each section provide detailed conclusions and observations.*

IV. **Requirements:**

Software requirements:

- *Python 3.12.10*
- *Jupyter Notebook(or any other Integrated Development Environment (IDE) such as VS Studio code)*

- *Python libraries used: Pandas, Numpy, Matplotlib, Seaborn, Scikit-Learn*

Dataset:

Car_price_dataset.csv provided as project scope

V. **Proposed Solution:**

Performing EDA using Python. This includes data inspection, preprocessing, visualization, correlation analysis, feature scaling, and feature scaling to identify most influential variables and uncover patterns, trends and draw conclusive insights.

VI. **Design*:***

The project workflow entails:

1. Data loading and inspection
2. Data type transformation
3. Descriptive statistics analysis
4. Feature scaling
5. Categorical and correlation analysis
6. Feature selection using SelectKBest

VII. **Dataset Description:**

The dataset consists of 15411 rows and 14 columns. It included categorical attributes such as car name, car brand, car model, seller type, fuel type, and transmission type. Numerical variables consisted of milage, engine, vehicle age, km driven, number of seats, maximum power, and selling price.

No missing values were found in the data set.

Selling Price is our target variable (dependent variable) and the rest are independent variables (i.e. Features).

There are three data types in our dataset: object (car_name, brand, model), integerts (vehicle_age, km_driven, engine, seats, selling_price), and categorical (seller_type, fuel_type,transmission_type).

The shape of our dataset is 15411 rows and 14 columns (13 features and 1 index column to be dropped) and each column consists of 15411 entries so our dataset consist of no null-values.

Overall, selling_price, engine, and max_power are right-skewed variables. While milage is relatively symmetric.

Seats column has an anomaly of minimum of 0 seats recorded which doesn't make logical sense. This column will have to be cleaned.

Duplicates: there are 167 duplicated rows in our dataframe. We used the .drop_duplicates() to drop the duplicated rows, which left us with a dataframe of 15244 rows and 13 columns.

Outliers: The outlier analysis revealed the presence of extreme values across multiple numerical variables, particularly in price-related features, indicating a skewed distribution and the need for careful preprocessing before modeling.

VIII. Implementation:

1. Understanding the Dataset

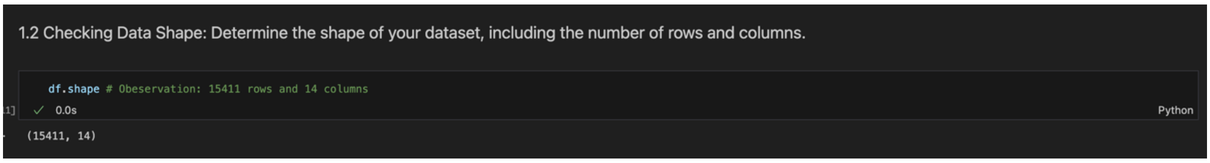
1.1 Data Loading

The dataset was imported into the python environment using Pandas library. This allowed for efficient data manipulation, inspection and analysis.
Further, the dataset was loaded without any errors, including correct formatting and compatibility with our analysis tools.

1.2 Dataset Shape

The shape of the dataset was examined using df.shape method to determine the number or instances (rows) and features (columns). This was determined to be a dataset of 15411 rows and 14 columns, of which 1 column is a column of indexes.

Understanding the shape of the dataset is crucial for understanding how our data is structured.



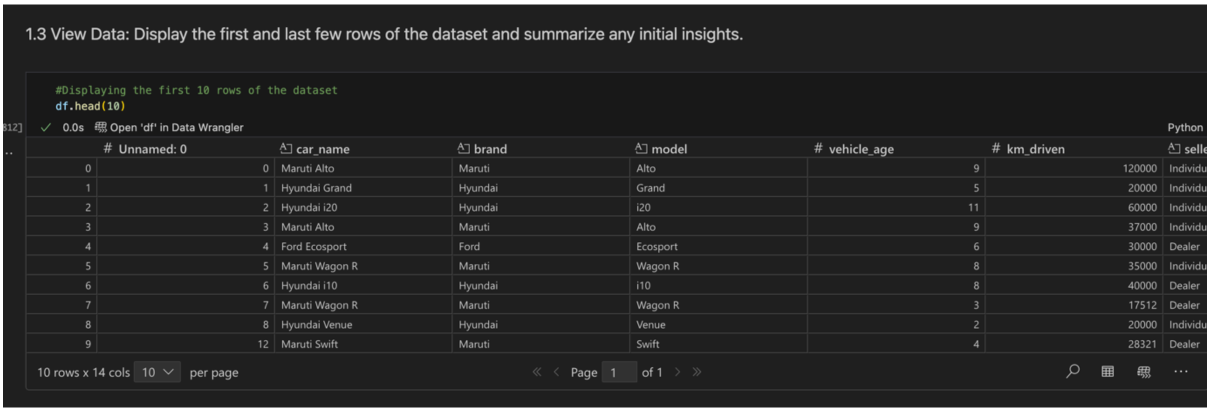
1.2 Checking Data Shape: Determine the shape of your dataset, including the number of rows and columns.

```
df.shape # Obeservation: 15411 rows and 14 columns
```

```
(15411, 14)
```

1.3 Viewing the Data

The first and last 10 rows were displayed to gain an initial understanding of feature names, value ranges, and data consistency. The dataset showed to contains a mix of numerical and categorical variables.



1.3 View Data: Display the first and last few rows of the dataset and summarize any initial insights.

```
#Displaying the first 10 rows of the dataset
df.head(10)
```

	# Unnamed: 0	car_name	brand	model	# vehicle_age	# km_driven	selle
0	0	Maruti Alto	Maruti	Alto	9	120000	Individu
1	1	Hyundai Grand	Hyundai	Grand	5	20000	Individu
2	2	Hyundai i20	Hyundai	i20	11	60000	Individu
3	3	Maruti Alto	Maruti	Alto	9	37000	Individu
4	4	Ford Ecosport	Ford	Ecosport	6	30000	Dealer
5	5	Maruti Wagon R	Maruti	Wagon R	8	35000	Individu
6	6	Hyundai i10	Hyundai	i10	8	40000	Dealer
7	7	Maruti Wagon R	Maruti	Wagon R	3	17512	Dealer
8	8	Hyundai Venue	Hyundai	Venue	2	20000	Individu
9	12	Maruti Swift	Maruti	Swift	4	28321	Dealer

2. *Initial Data Examination*

2.1 Dataset Information

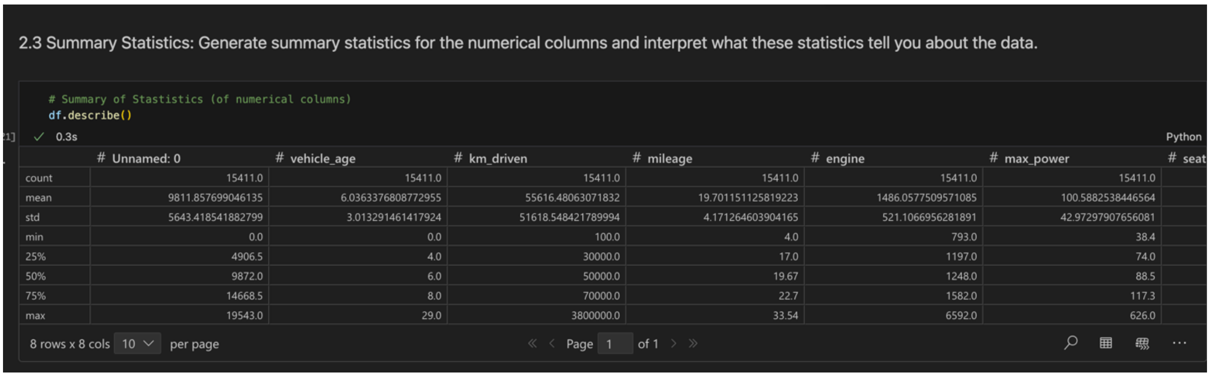
The .info() method was used to inspect column names, data types and non-null columns. It was found that the column names consist of: 'car_name', 'brand', 'model', 'vehicle_age', 'km_driven', 'seller_type' (Individual/Dealer/Trustmark Dealer), 'fuel_type' (Petrol/Diesel/CNG/LPG/Electric), 'transmission_type' (Automatic/Manual), 'milage', 'engine', 'max_power', 'seats' and 'selling_price'. Further, the dataset consisted of no null counts (each column has a recorded 15411 instances meaning there are no missing values in our dataset). Data types included integers, floats and object types.

2.2 Data Type inspection and Conversion

Each column’s data type was reviewed to ensure accuracy and easy of analysis. Categorical variables that were initially stored as object type variables were converted to categorical types (namely: fuel type, transmission type, and seller type) to ensure accurate statistical analysis, visualization and feature selection.

2.3 Summary of Statistics

Descriptive statistics were generated for numerical features, including: mean, median, standard deviation, minimum and maximum values, quartiles. Large differences between mean and median we found in some variables suggest skewness in the data. Sdditionally, extreme maximum and minimum values hinted at presence of potential outliers.

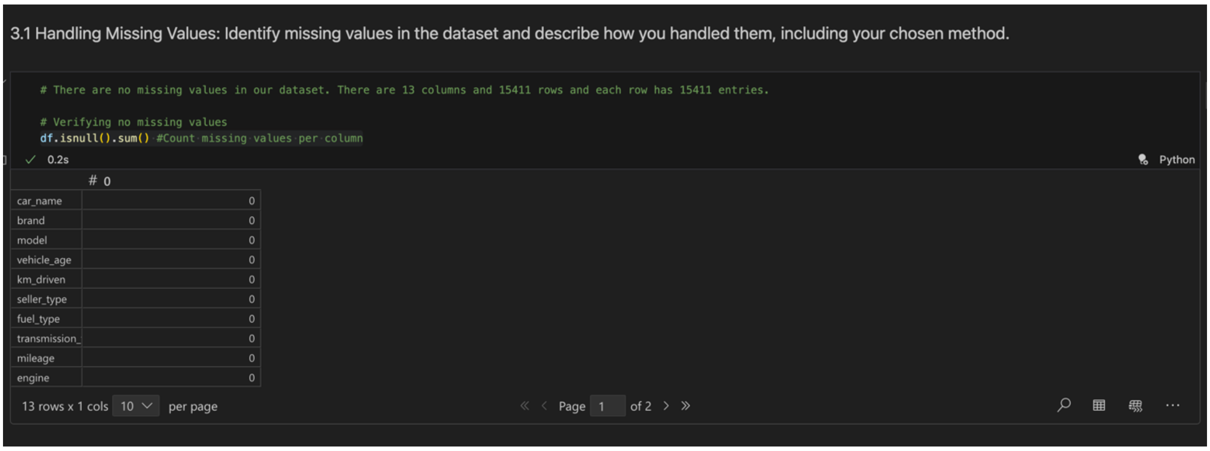


2.3 Summary Statistics: Generate summary statistics for the numerical columns and interpret what these statistics tell you about the data.

```python
# Summary of Stastistics (of numerical columns)
df.describe()
```

	# Unnamed: 0	# vehicle_age	# km_driven	# mileage	# engine	# max_power	# seat
count	15411.0	15411.0	15411.0	15411.0	15411.0	15411.0	
mean	9811.857699046135	6.0363376808772955	55616.48063071832	19.701151125819223	1486.0577509571085	100.5882538446564	
std	5643.418541882799	3.013291461417924	51618.548421789994	4.171264603904165	521.1066956281891	42.97297907656081	
min	0.0	0.0	100.0	4.0	793.0	38.4	
25%	4906.5	4.0	30000.0	17.0	1197.0	74.0	
50%	9872.0	6.0	50000.0	19.67	1248.0	88.5	
75%	14668.5	8.0	70000.0	22.7	1582.0	117.3	
max	19543.0	29.0	3800000.0	33.54	6592.0	626.0	

3. *Data Cleaning*
   3.1 Missing values calculations



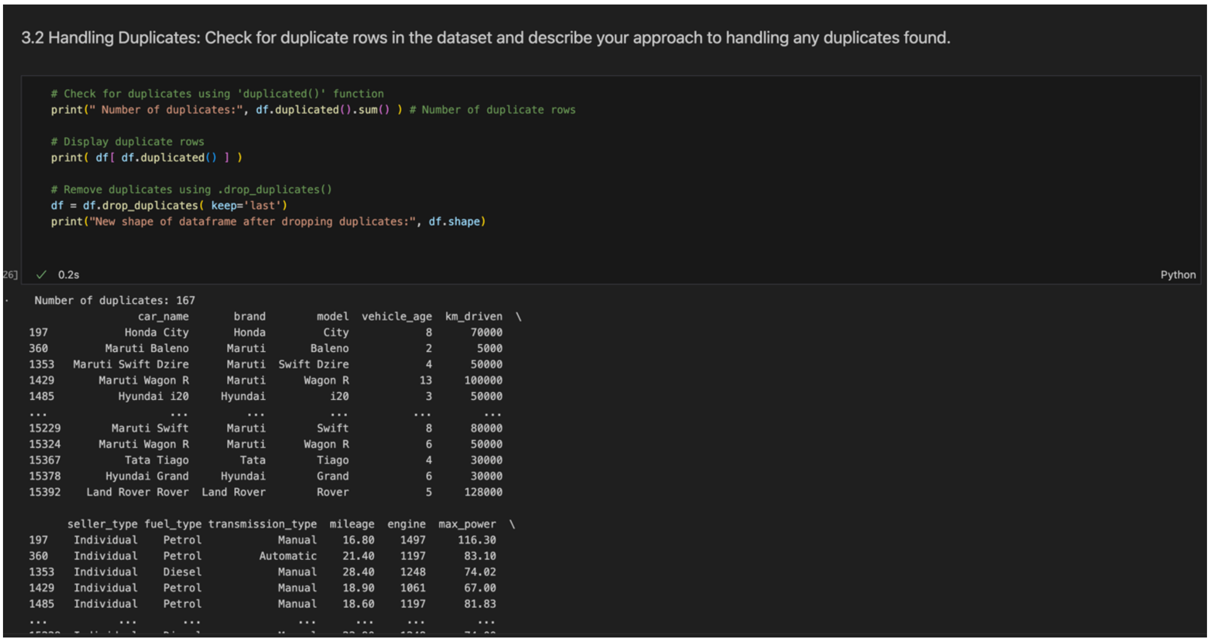
3.1 Handling Missing Values: Identify missing values in the dataset and describe how you handled them, including your chosen method.

```python
# There are no missing values in our dataset. There are 13 columns and 15411 rows and each row has 15411 entries.

# Verifying no missing values
df.isnull().sum() #Count missing values per column
```

	# 0
car_name	0
brand	0
model	0
vehicle_age	0
km_driven	0
seller_type	0
fuel_type	0
transmission_	0
mileage	0
engine	0

No missing values were found in the dataset so no row removals was necessary.

3.2 Duplicate values



3.2 Handling Duplicates: Check for duplicate rows in the dataset and describe your approach to handling any duplicates found.

```python
# Check for duplicates using 'duplicated()' function
print(" Number of duplicates:", df.duplicated().sum() ) # Number of duplicate rows

# Display duplicate rows
print( df[ df.duplicated() ] )

# Remove duplicates using .drop_duplicates()
df = df.drop_duplicates( keep='last')
print("New shape of dataframe after dropping duplicates:", df.shape)
```

26] ✓ 0.2s Python

```
Number of duplicates: 167
                car_name       brand        model  vehicle_age  km_driven  \
197           Honda City       Honda         City            8      70000
360        Maruti Baleno      Maruti       Baleno            2       5000
1353   Maruti Swift Dzire     Maruti  Swift Dzire            4      50000
1429       Maruti Wagon R     Maruti      Wagon R           13     100000
1485          Hyundai i20    Hyundai          i20            3      50000
...                  ...        ...          ...          ...        ...
15229        Maruti Swift     Maruti        Swift            8      80000
15324      Maruti Wagon R     Maruti      Wagon R            6      50000
15367          Tata Tiago       Tata        Tiago            4      30000
15378       Hyundai Grand    Hyundai        Grand            6      30000
15392    Land Rover Rover  Land Rover       Rover            5     128000

      seller_type fuel_type transmission_type  mileage  engine  max_power  \
197    Individual    Petrol            Manual    16.80    1497     116.30
360    Individual    Petrol         Automatic    21.40    1197      83.10
1353   Individual    Diesel            Manual    28.40    1248      74.02
1429   Individual    Petrol            Manual    18.90    1061      67.00
1485   Individual    Petrol            Manual    18.60    1197      81.83
...           ...       ...               ...      ...     ...        ...
```

Using the df.duplicated().sum() method we found 167 duplicated rows in our dataset. We then proceeded to remove said duplicates by using the df.drop_duplicated() function, making sure to reset the index after operation.

3.3 Outlier detection and Treatment

Outliers were detected for numerical columns using statistical techniques (such as the Interquartile Range method).

Process:

- *Lower and upper bounds were calculated for each numerical feature.*
- *Observations falling outside these bounds were flagged as outliers.*
- *Outlier rows were removed to create a refined dataset.*

Results:

- *The number of outliers varied across features.*
- *The dataset shape changed after removal, confirming successful filtering.*

Impact:
Removing outliers helped:

- *Reduce noise*
- *Improve data reliability*
- *Prevent skewed statistical interpretations*

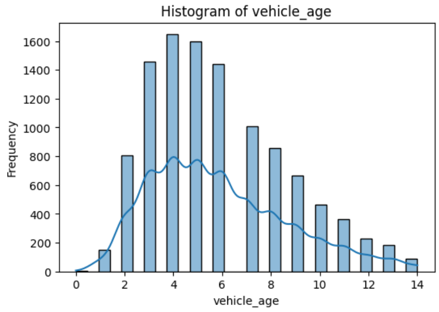
The outlier analysis revealed the presence of extreme values across multiple numerical variables, particularly in price-related features, indicating a skewed distribution and the need for careful preprocessing before modeling.

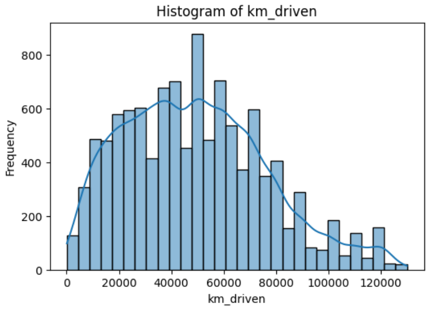
- *For vehicle_age : most cars fall within a reasonable age range, the outliers are very old or very new cars.*
- *For km_driven: there are a large amount of outliers present in this column. These are cars with very high milage, from this we see that some cars are driven much more than average (likely due to long-distance use or being used as fleet cars). High milage will affetct the selling price.*
- *Milage: has fewer outliers than km_driven. The outliers here likely represent cars with high fuel efficiency or with poor fuel effciency. With there being fewer outliers in this column we can deduce that most cars have a similar fuel efficiency.*
- *Engine: has many outliers, large engine sizes are most notable meaning most cars have small to mid-sized engine sizes making large engines out of the norm.*
- *Max_power: most cars exeplify as normal 'everyday' cars which makes high-performance cars uncommon and outliers.T*
- *Seats: Many values are flagged as outliers. This is because most cars have 5 seats so anything else stands out statistically as an outlier.*
- *Selling_price: this column has significant outliers. There are cars priced significantly above the typical price range which tells us that the price distribution is rightly-skewed.These high prices could be due to luxury/premium vehicles (which is rare as most present as everyday cars).*

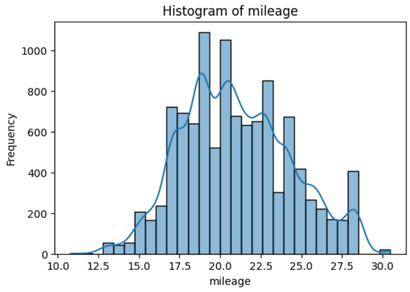
Overall most cars are normal cars. Outliers would represent cars with high selling price, high milage or large engine sizes.

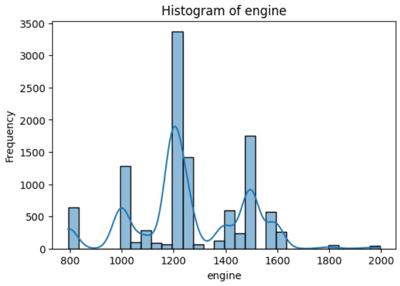
4. *Data Analysis*
5.

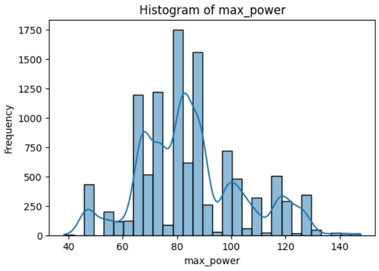
We generated histograms for numerical data columns:

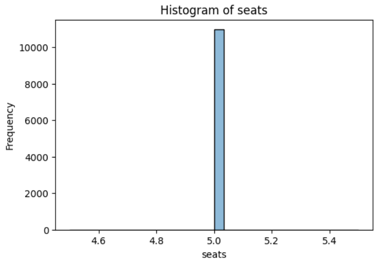


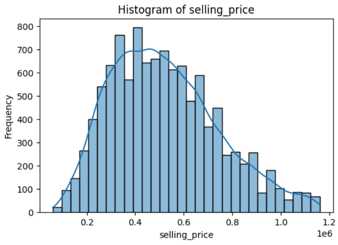












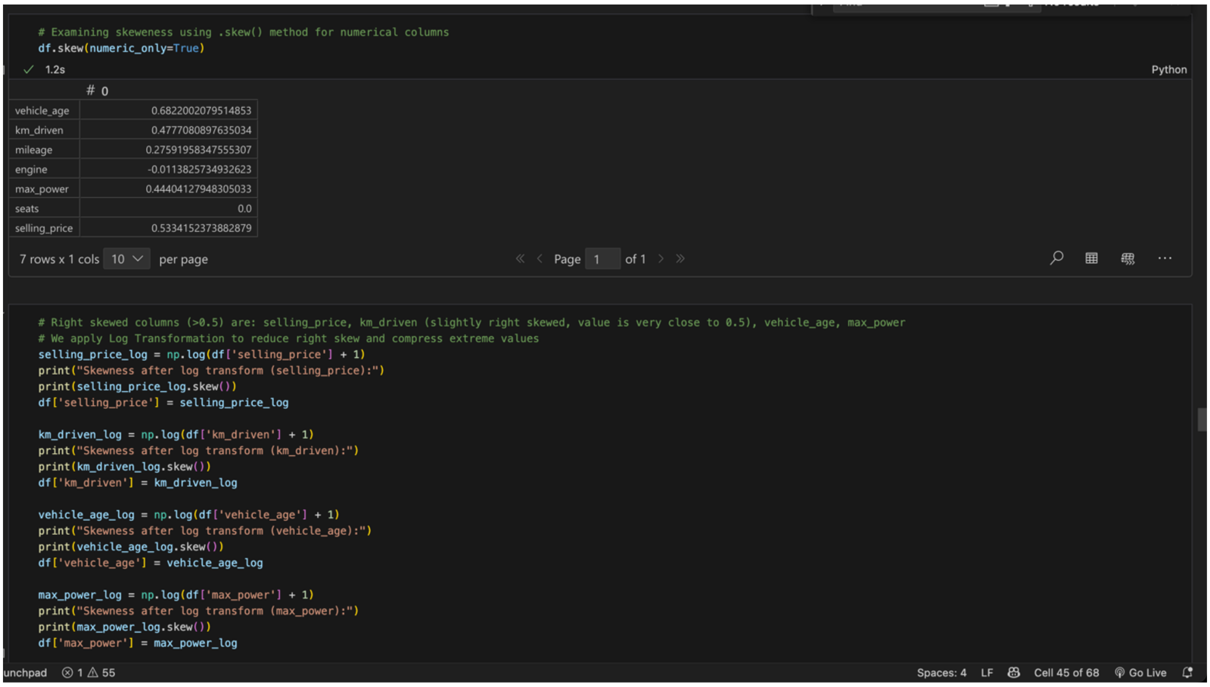
Insights:
Most numerical features show right-skewed distributions (>0.5), especially selling_price, km_driven, engine, and max_power.

This means most cars fall into lower or mid-range values, with a few extreme cases. Variables like mileage are more evenly distributed.

The seats variable shows little variation, with most vehicles having 5 seats.

The distributions visually support the presence of outliers identified using the IQR method.

We further went to statistically measure skewness and apply Log Transformations to certain features.



```
# Examining skeweness using .skew() method for numerical columns
df.skew(numeric_only=True)
```

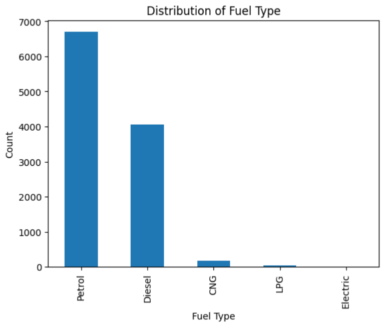
✓ 1.2s

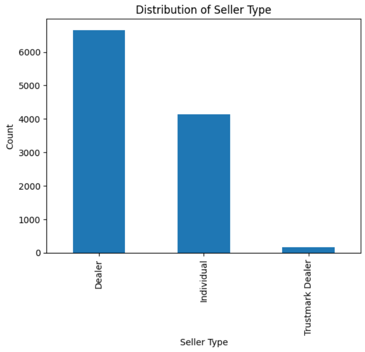
	# 0
vehicle_age	0.6822002079514853
km_driven	0.4777080897635034
mileage	0.275919583475553 07
engine	-0.0113825734932623
max_power	0.44404127948305033
seats	0.0
selling_price	0.5334152373882879

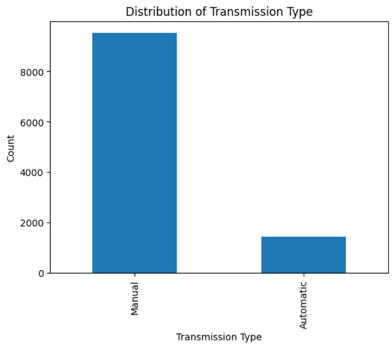
7 rows x 1 cols 10 per page Page 1 of 1

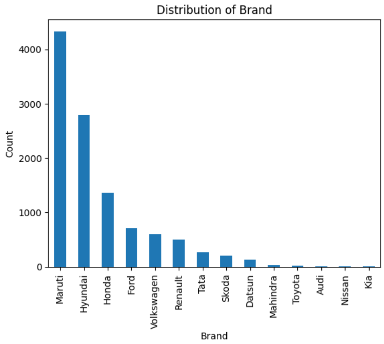
```
# Right skewed columns (>0.5) are: selling_price, km_driven (slightly right skewed, value is very close to 0.5), vehicle_age, max_power
# We apply Log Transformation to reduce right skew and compress extreme values
selling_price_log = np.log(df['selling_price'] + 1)
print("Skewness after log transform (selling_price):")
print(selling_price_log.skew())
df['selling_price'] = selling_price_log

km_driven_log = np.log(df['km_driven'] + 1)
print("Skewness after log transform (km_driven):")
print(km_driven_log.skew())
df['km_driven'] = km_driven_log

vehicle_age_log = np.log(df['vehicle_age'] + 1)
print("Skewness after log transform (vehicle_age):")
print(vehicle_age_log.skew())
df['vehicle_age'] = vehicle_age_log

max_power_log = np.log(df['max_power'] + 1)
print("Skewness after log transform (max_power):")
print(max_power_log.skew())
df['max_power'] = max_power_log
```

For categorical features, we plotted distribution plots and found the following observatiosn:









Here we see that Maruti brand accounts for the most common brand, followed by Hyundai and Honda. The rest of the brands have relatively low counts.

The data is imbalanced with respect to brands, there's only a few dominant brands and many with low frequencies.

Manual transmission type if by far the highest frequency, Automatic transmission cars are significantly fewer.

This indicates either preference popularity or availability.

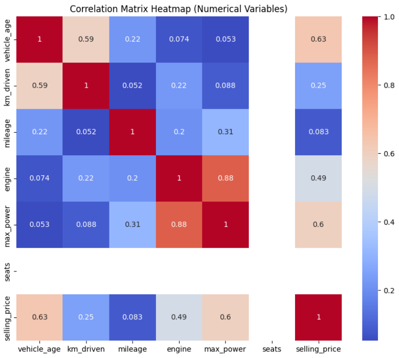
Petrol and Diesel fuel types are by far the most significant, more so than CNG, LPG and Electric types which have extremely low counts.

Dealers account for majority of listings, while Individuals form a smaller portion and very few Trustmark Dealers.

This indicates that it's a dealer dominated market.

6. *Feature selection*
7.

Feature selection techniques were applied to identify the most influential variables with respect to the target variable. The top-ranked features were identified based on their scores.



The Correlation analysis shows that selling price is strongly influenced by engine size and maximum power, while milage and vehicle age have a negative relationship with price (price decreases as milage and vehicle age).

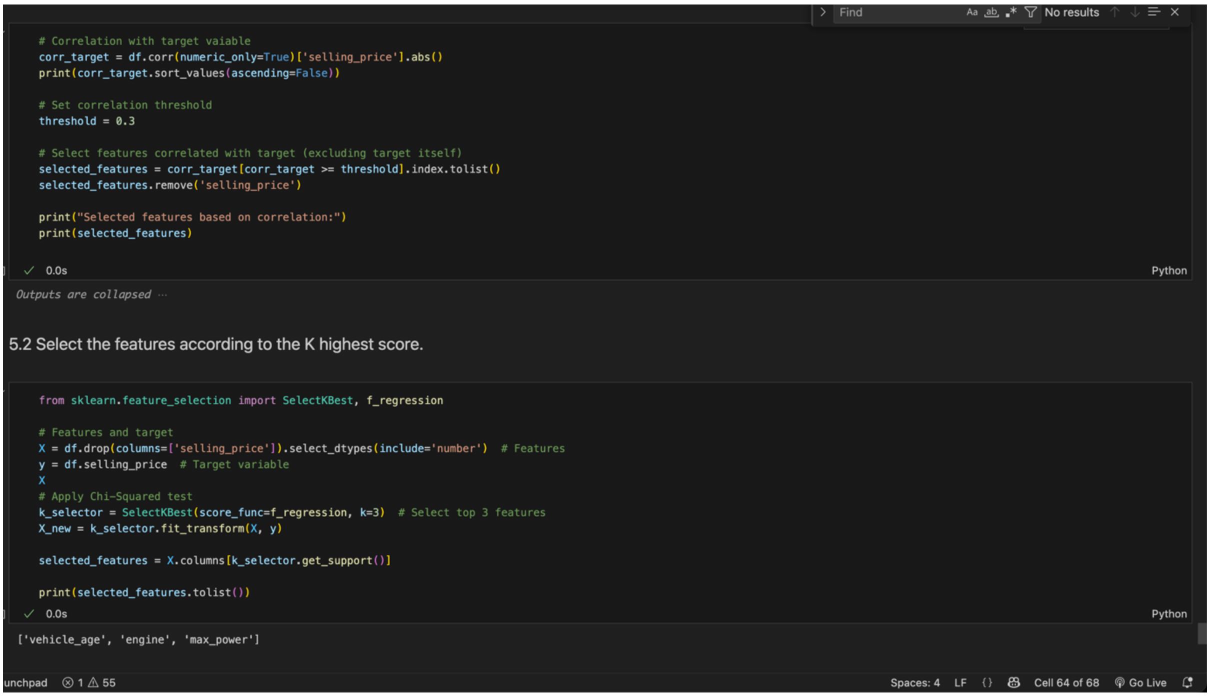
Engine size and power are highly correlated with each other, while fuel efficiency decreases as engine size and max power increase.

Seats has no variance and show a weak relationship with selling price.

Performance variables (engine, max_power) are the strongest predictors of price;

Efficiency variables (mileage) move in the opposite direction;

Age matters, but not as strongly as power.



```
# Correlation with target vaiable
corr_target = df.corr(numeric_only=True)['selling_price'].abs()
print(corr_target.sort_values(ascending=False))

# Set correlation threshold
threshold = 0.3

# Select features correlated with target (excluding target itself)
selected_features = corr_target[corr_target >= threshold].index.tolist()
selected_features.remove('selling_price')

print("Selected features based on correlation:")
print(selected_features)
```

✓ 0.0s

Outputs are collapsed ...

5.2 Select the features according to the K highest score.

```
from sklearn.feature_selection import SelectKBest, f_regression

# Features and target
X = df.drop(columns=['selling_price']).select_dtypes(include='number')  # Features
y = df.selling_price  # Target variable
X
# Apply Chi-Squared test
k_selector = SelectKBest(score_func=f_regression, k=3)  # Select top 3 features
X_new = k_selector.fit_transform(X, y)

selected_features = X.columns[k_selector.get_support()]

print(selected_features.tolist())
```

✓ 0.0s

```
['vehicle_age', 'engine', 'max_power']
```

Strong correlations with selling price: max_power,engine and vehicle_age.

Moderate correlation: milage

Weak correlation: km_driven, seats

K highest score also confirms that selected features are ['vehicle_age', 'engine', 'max_power']

IX. Results and Discussion:

The analysis revealed that vehicle age, engine size, and maximum power have strong correlations with selling price. Feature selection confirmed these variables as the top predictors. Categorical analysis highlighted class imbalances that should be considered in modeling.

X. Conclusion:

This project successfully applied exploratory data analysis techniques to understand the car price dataset. The findings demonstrate that vehicle specifications significantly influence selling price, and the prepared dataset is suitable for predictive modeling.

XI. References:

Han, J., Kamber, M., & Pei, J. (2012). *Data mining: Concepts and techniques* (3rd ed.). Morgan Kaufmann.

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2021). *An introduction to statistical learning: With applications in R* (2nd ed.). Springer. https://doi.org/10.1007/978-1-0716-1418-1

Kuhn, M., & Johnson, K. (2013). *Applied predictive modeling*. Springer. https://doi.org/10.1007/978-1-4614-6849-3

Géron, A. (2022). *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow* (3rd ed.). O'Reilly Media.

Tukey, J. W. (1977). *Exploratory data analysis*. Addison-Wesley.

Matplotlib Development Team. (n.d.). *Matplotlib documentation*. https://matplotlib.org/

Pandas Development Team. (n.d.). *Pandas documentation*. https://pandas.pydata.org/

Python Software Foundation. (n.d.). *Python documentation*. https://www.python.org/

Scikit-learn Developers. (n.d.). *Scikit-learn documentation*. https://scikit-learn.org/

Seaborn Developers. (n.d.). *Seaborn documentation*. https://seaborn.pydata.org/

Formatting Notes:

1. **Line spacing:** 1.5 spacing, except references where single spacing is required.
2. **Font type:** Arial throughout.
3. **Font size:** A size 12 font is required throughout, except Chapter headings that must have a size 16 font.

4. **Justification:** The main text must be fully justified. Text within tables must be left justified.