TECHNOLOGICAL INSTITUTE OF THE PHILIPPINES

938 Aurora Blvd., Cubao, Quezon City

COLLEGE OF ENGINEERING AND ARCHITECTURE

ELECTRONICS ENGINEERING DEPARTMENT

2ND SEMESTER SY 2023 - 2024

FINAL PERIOD

Computational Thinking with Python

COE 003 - ECE32-COE1

Finals - EDA Document

Final Project

Submitted to:

Engr. Joesmart Apan

Submitted on:

May 19, 2024

Submitted by:

Enriquez, Franz Ivan
Montecillo, Sean Andre
Panopio, Armin Rucos

Technical Specifications and Prices for Leading Automotive Companies

Exploratory Data Analysis

Enriquez, Montecillo, Panopio
Electronics Engineering Department
Technological Institute of the Philippines
Quezon City, Philippines

I. Introduction

In the intense and competitive automotive industry, understanding the fine details of vehicle specifications and pricing is important for consumers, manufacturers, and industry analysts. This project applies exploratory data analysis (EDA) to technical specifications and prices across leading automotive companies. By using EDA techniques, the group aims to uncover patterns, trends, and insights that can affect the decision-making processes, identify market positioning, and highlight areas of improvement within the industry. This project aims to provide a well-defined overview of the current landscape of the automotive industry, offering valuable perspectives on how technical features and pricing strategies impact market performance and consumer preferences.

II. Source Code

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import requests
from io import StringIO
from wordcloud import WordCloud

# Load the dataset from the provided link
url = "https://raw.githubusercontent.com/rushabh-mehta/EDA-on-Automobile-Dataset/master/Automobile_data.csv"
response = requests.get(url)
csv_data = StringIO(response.text)
data = pd.read_csv(csv_data)

# Summary statistics
summary_stats = data.describe()
print("Summary Statistics:")
print(summary_stats)

#Histogram of Makes
plt.figure(figsize=(22, 6))
sns.histplot(data=data, x='make', kde=True)
plt.title('Distribution of Makers')
plt.xlabel('Makers')
plt.ylabel('Frequency')
plt.show()

#Histogram of Fuel system
plt.figure(figsize=(8, 6))
sns.histplot(data=data, x='fuel-system', kde=True)
plt.title('Distribution of Fuel system')
plt.xlabel('Type of Fuel System')
plt.ylabel('Frequency')
plt.show()

#Histogram of Fuel type
plt.figure(figsize=(2, 6))
sns.histplot(data=data, x='fuel-type', kde=True)
plt.title('Distribution of Fuel Type')
plt.xlabel('Type of Fuel')
plt.ylabel('Frequency')
plt.show()

#Top 20 makes
top_20_titles = data['make'].value_counts().head(20)

# Create a bar plot for the top 20 makes
plt.figure(figsize=(12, 6))
plt.bar(top_20_titles.index, top_20_titles.values)
plt.xlabel('makes')
plt.ylabel('Count')
plt.title('Top 20  Makes')
plt.xticks(rotation=45, ha='right')
plt.tight_layout()
plt.show()

top_20_titles = data['make'].value_counts().head(20)

# Create a dictionary of job titles and their counts
title_counts = dict(top_20_titles)

#Word Cloud Object
wordcloud = WordCloud(width=800, height=400, background_color='white').generate_from_frequencies(title_counts)

plt.figure(figsize=(10, 6))
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis('off')
plt.title('Top 20 - Make Word Cloud')
plt.show()
#line plot
body_style_counts = data['body-style'].value_counts().reset_index()
body_style_counts.columns = ['body-style', 'count']

# Sort the data by the 'fuel-system' column if necessary
body_style_counts = body_style_counts.sort_values('body-style')

# Line Plot
plt.figure(figsize=(8, 6))
sns.lineplot(data=body_style_counts, x='body-style', y='count', marker='o')
plt.title('Distribution of Body System')
plt.xlabel('Type of Body')
plt.ylabel('Frequency')
plt.show()

#Scatter Plot
top_10_data = data.head(20)

# Scatter Plot
plt.figure(figsize=(10, 6))
sns.scatterplot(data=top_10_data, x='make', y='price')
plt.title('Prices by Makes(TOP 20)')
plt.xlabel('Makes')
plt.ylabel('Prices')
plt.show()

#pie chart
cylinder_counts = data['num-of-cylinders'].value_counts().reset_index()
cylinder_counts.columns = ['num-of-cylinders', 'count']

# Pie Chart
plt.figure(figsize=(10, 6))
plt.pie(cylinder_counts['count'], labels=cylinder_counts['num-of-cylinders'], autopct='%1.1f%%', startangle=140)
plt.title('Distribution of Number of Cylinders')
plt.axis('equal')  # Equal aspect ratio ensures that pie is drawn as a circle.
plt.show()
```

III. Theory of Operation / Explanation of Code used

In this section, we will tackle the theory of operation behind this project. By breaking down our code into parts, to achieve our desired output.

Libraries

We utilized Jupyter Notebook to achieve exploratory data analysis (EDA). In our chosen data set, the automobile data set, we used various libraries, such as Numpy, Pandas, Matplotlib, and Seaborn, to represent them using different charts to show the relationships between variables.

Loading Dataset

Pandas and Request are the libraries used to load our dataset from the net. Request Library is concerned with getting the data set from the net, and Pandas Library for reading and using the data set in a structured format using the command (.describe()) will show you a quick overview of the data and show the count, mean, standard deviation, minimum, 25th percentile, median, 75th percentile , and maximum of the columns [1]. of the data.

Visualization of Data

Matplotlib and Seaborn work hand in hand to provide a visual representation of our data files. They are both popular libraries used in data visualization within the Python programming language [2]. The charts we used are histograms, bar charts, pie charts, scatter plots, line graphs, and word clouds. We used this to represent and correlate data from the dataset in a visually appealing and informative manner.

IV. Data & Results

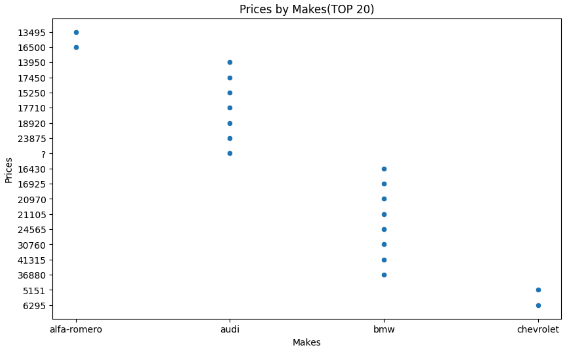

Figure 1: Prices by Makes (Top 20)

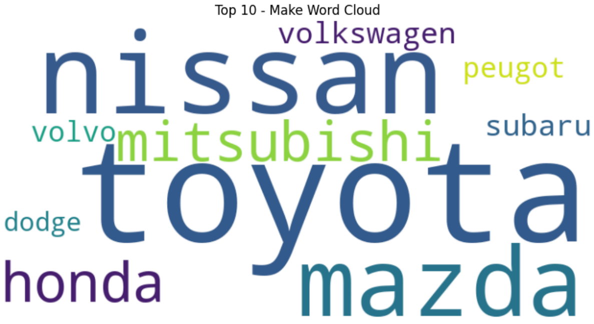

Figure 2: Word Cloud (Top 10 Overall Companies)

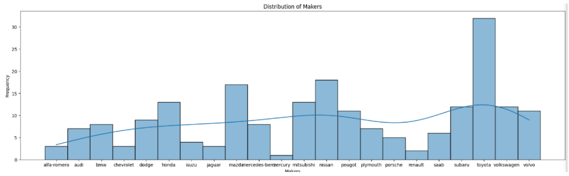

Figure 3: Distribution of Makers' Frequency

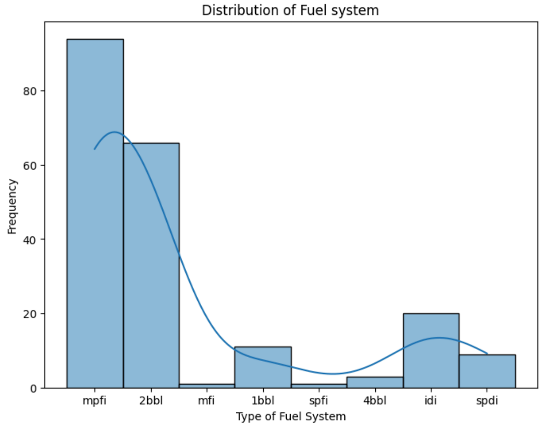

Figure 4: Fuel System Performance Distribution

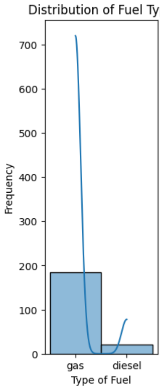

Figure 5: Fuel Type Distribution (Gasoline or Diesel)

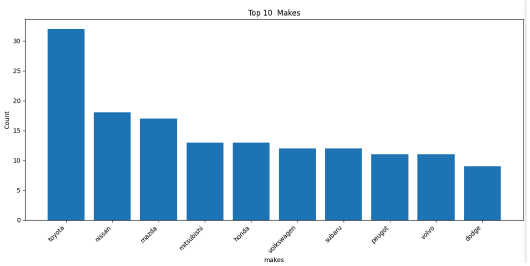

Figure 6: Top 10 Companies for Makes Bar Graph

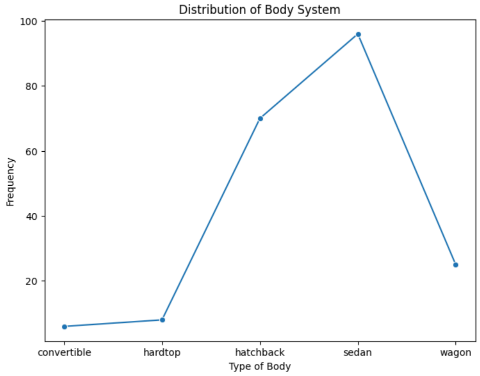

Figure 7: Body System Distribution Line Graph

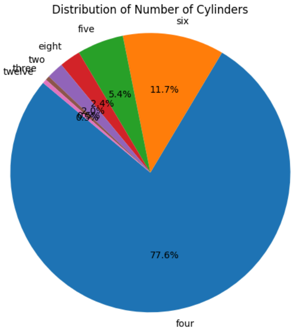

Figure 8: Cylinder Count Pie Chart

V. Interpretation of Data

Figure 1 represents a scatter plot. A scatter plot, also called a scatter chart or scatter graph, is a visual aid for observing relationships between two different numerical variables. It uses dots to represent the values for the variables [3]. The data we used to show its relationship are prices and make.

Figure 2 shows a word cloud of the top 10 makes in the data set. The basic principle behind word clouds, also called text clouds or tag clouds, is that a word appears to be bigger in the word cloud the more times it appears in the data set [4].

Figures 3, 4, and 5 are examples of histograms. A histogram is a correlation between the variable and its frequency distribution in the data set. Figure 3 shows the frequency of all variables under makes in the data set. Figure 4 shows the distribution of the variables under the fuel system and Figure 5 shows variables from fuel types.

Figure 6 shows a bar plot of the top 10 makes in the data set. We used bar plots to correlate variables and their frequencies. We used the following commands to limit the selection to 10 in order to make a top 10 make bar plot. The command (.value_counts()) is to determine the number of appearances in the data set for each variable, and (.head(n)) is used to select the first n rows in the (.value_countss()).

Figures 7 and 8 show a line graph and a pie chart. Both can be used to represent the frequency of each variable in a specific section of the data set. The line graph displays trends over time but can also be used to represent frequency, while the pie chart shows the distribution of categories as a whole.

VI. References

[1] “Python: Display All Columns of a Pandas DataFrame in ‘.describe()’ | Saturn Cloud Blog,” Nov. 02, 2023. https://saturncloud.io/blog/python-spyder-display-all-columns-of-a-pandas-dataframe-in-describe/#:~:text=describe()%E2%80%9D%20Method-,The%20.,and%20maximum%20of%20the%20columns.

[2] S. Pierre, “Python Data Visualization with Seaborn and Matplotlib,” Built In, Feb. 16, 2023. https://builtin.com/data-science/data-visualization-tutorial

[3] Atlassian, “Mastering Scatter Plots: Visualize data correlations,” Atlassian. https://www.atlassian.com/data/charts/what-is-a-scatter-plot

[4] “What are Word Clouds? The Value of Simple Visualizations...,” Boost Labs - Digital Product Agency. https://boostlabs.com/what-are-word-clouds-value-simple-visualizations/