Cure Deliverable 3

Due: October 30, 2023

Made By:

Group 103

Editors:

Opeyemi Sanyaolu (30156143)
Mendjemo Ngangom Gerard Ledoux (30192835)
Kosy Onyejemezi (30189156)
Chidera Awiaka (30165302)

Task 1

Member	Topic	# of Classes	# of Numeric Attributes	# of Instances
Opeyemi Sanyaolu	Classification of Vehicle Data	2	2	5
Kosy Onyejemezi	Classification of houses	2	3	12
Mendjemo Gerard	Classification of animals	2	2	10
Chidera Awiaka	Classification of Stationary Sales	2	3	14

We chose to expand on the classification of vehicle dataset. The dataset has been initialized with 100 instances instead of 5.

Task 2

a) Classification of Vehicle Data.
b) The Classification of vehicle data describes different attributes customers may want to consider when purchasing a new vehicle, allowing customers to easily compare the make, price, horsepower, and top speed.
   Description of data:
   - Number of instances: 100 (at least 30 per class)
   - Number of attributes: 3 numeric attributes (Price, horsepower, top speed)
   - Number of classes: 2 (Overseas, American)
   - Creator: Opeyemi Sanyaolu

c) Dataset:

Classification of Vehicle Data

```
import pandas as pd

df = pd.read_excel("Vehicle_Data.xlsx")

print(df)
```

[100] ✓ 0.0s

```
...          Car Make   Price (USD)  Horsepower (hp)  Top Speed (mph)  \
0      Nissan Micra         17000              109              115
1     Nissan Sentra         23000              149              120
2       Nissan 370Z         30000              332              160
3       Nissan GT-R        210740              600              205
4     Infiniti Q50         53500              400              155
..             ...           ...              ...              ...
96        Kia Niro         24690              139              105
97      Kia Sedona         42000              276              130
98       Acura TLX         50000              355              155
99    Jeep patriot         28000              172              115
100   Chrysler 300         50000              363              140

    Category (class)
0           Overseas
1           Overseas
2           Overseas
3           Overseas
4           Overseas
..               ...
96          Overseas
97          Overseas
98          Overseas
99          American
100         American

[101 rows x 5 columns]
```

	A	B	C	D	E
1	Car Make	Price (USD)	Horsepower (hp)	Top Speed (mph)	Category (class)
2	Nissan Micra	17,000	109	115	Overseas
3	Nissan Sentra	23,000	149	120	Overseas
4	Nissan 370Z	30,000	332	160	Overseas
5	Nissan GT-R	210,740	600	205	Overseas
6	Infiniti Q50	53,500	400	155	Overseas
7	Nissan Altima	35,100	182	130	Overseas
8	Nissan Maxima	36,990	300	140	Overseas
9	Nissan Versa	18,340	122	120	Overseas
10	Nissan Rogue	30,000	175	125	Overseas
11	Nissan Murano	44,000	260	125	Overseas
12	Nissan Pathfinder	45,000	284	125	Overseas
13	Nissan Armada	53,000	390	130	Overseas
14	Nissan Juke	30,000	215	135	Overseas
15	Nissan Kicks	21,000	122	120	Overseas
16	Nissan Xterra	30,000	265	115	Overseas
17	Nissan Frontier	28,000	220	120	Overseas

Sources for the table above can be found in the reference section at the very bottom.

Task 3

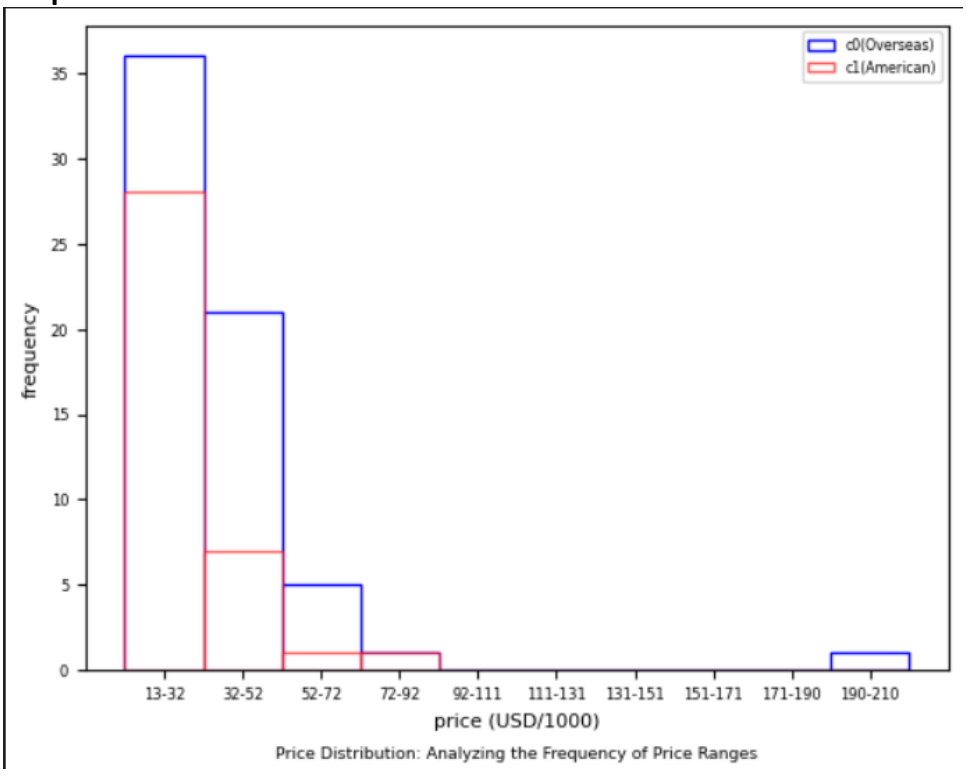
a) b) c)

Code

```
import math

import matplotlib.pyplot as plt
fig, ax = plt.subplots()
plt.rcParams.update({'font.size': 5.6}) #Change the font size of the table to 6 for better readability

import numpy as np
category = np.array(df.iloc[:, 4]) # Get an array of all values in Category column of dataframe

## Separate the instances into their different categories

pos = 0 # position of the price in the dataframe
overseas = np.array([]) # Store the values of prices that fall under the Overseas category
american = np.array([]) # Store the values of prices that fall under the American category

for row in category:
    # Filter categories
    if row == "Overseas":
        overseas = np.append(overseas, df.iloc[pos, 1])
    else:
        american = np.append(american, df.iloc[pos, 1])
    pos += 1

## Create class ranges for the price attribute

# Statistics of Price column
price = np.array(df.iloc[:, 1]) # Get an array of all values in Price column
class_ranges = [] # store class ranges in bins format
max_price = np.max(price) # Minimum price
min_price = np.min(price) # Maximum price

# Class boundary calculations
n_classes = math.floor(math.pow(len(price), 1/2)) # Number of classes
interval_len = (max_price - min_price) / n_classes # Interval length

# Class boundaries
lcb = min_price # Lower class boundary
```

```
ucb = lcb + interval_len # Upper class boundary

for i in range(n_classes):
    # Adding class boundaries
    class_ranges.append(lcb)
    class_ranges.append(ucb)

    # Update class boundry
    lcb = ucb
    ucb = lcb + interval_len

# Make x labels math the class boundaries
xticks_val = [] # Store values of xticks
xticks_label = [] # Store labels name xtick

# Class boundaries
lcb = min_price # Lower class boundary
ucb = lcb + interval_len # Upper class boundary

for i in range(n_classes):
    # Position and name  x labels
    xticks_val.append(lcb + ((ucb - lcb) / 2)) # Current value of xtick
    xticks_label.append(str(math.floor(lcb/1000)) + "-" + str(math.floor(ucb/1000))) # Class boundary matching xtick

    # Update class boundry
    lcb = ucb
    ucb = lcb + interval_len


## Draw Histogram
plt.hist(overseas, bins=class_ranges, color='w', edgecolor='b', alpha=1.0, label='c0(Overseas)')
plt.hist(american, bins=class_ranges, color = 'w', edgecolor='r', alpha=0.65, label='c1(American)')
plt.xticks(xticks_val, xticks_label)
plt.title("Price Distribution: Analyzing the Frequency of Price Ranges", y = -0.16)
plt.xlabel("price (USD/1000)")
plt.ylabel("frequency")
plt.legend()
plt.show()
```

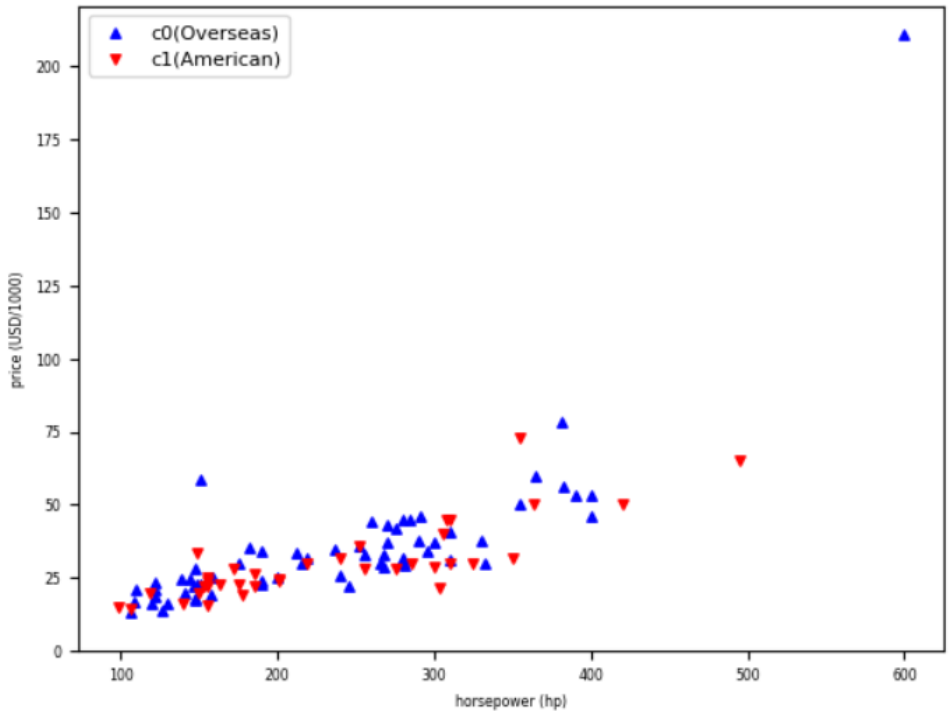
Graph



Code

```
    import math

    import matplotlib.pyplot as plt
    fig, ax = plt.subplots()
    plt.rcParams.update({'font.size': 8}) #Change the font size of the table to 6 for better readability

    import numpy as np
    category = np.array(df.iloc[:, 4]) # Get an array of all values in Category column of dataframe

    ## Separate the instances into their different categories

    pos = 0 # position of the price in the dataframe
    price_overseas = np.array([]) # Store the values of prices that fall under the Overseas category
    price_american = np.array([]) # Store the values of prices that fall under the American category
    hp_overseas = np.array([]) # Store the values of horsepowers that fall under the Overseas category
    hp_american = np.array([]) # Store the values of horsepowers that fall under the Overseas category

    for row in category:
        # Filter categories
        if row == "Overseas":
            price_overseas = np.append(price_overseas, df.iloc[pos, 1])
            hp_overseas = np.append(hp_overseas, df.iloc[pos, 2])
        else:
            price_american = np.append(price_american, df.iloc[pos, 1])
            hp_american = np.append(hp_american, df.iloc[pos, 2])
        pos += 1


    ## Adjust y labels
    # Statistics of Price column
    price = np.array(df.iloc[:, 1]) # Get an array of all values in Price column
    class_ranges = [] # store class ranges in bins format
    max_price = np.max(price) # Minimum price
    min_price = 0

    # Make intervals
    yticks_val = [] # Store values of xticks
    yticks_label = [] # Store labels name xtick

cur_price = 0

while cur_price <= max_price:
    yticks_label.append(str(math.floor(cur_price/1000)))
    yticks_val.append(cur_price)
    cur_price += 25000


## Draw scatter plot
ax.scatter(hp_overseas, price_overseas, c='b', marker='^', s=15, label="c0(Overseas)")
ax.scatter(hp_american, american, c='r', marker='v', s=15, label="c1(American)")

plt.yticks(yticks_val, yticks_label)
plt.title("Top Speed vs. Price: A Comparison of Car Performance and Cost", y = -0.16)
plt.xlabel("horsepower (hp)")
plt.ylabel("price (USD/1000)")
plt.legend()
plt.show()
```

Graph



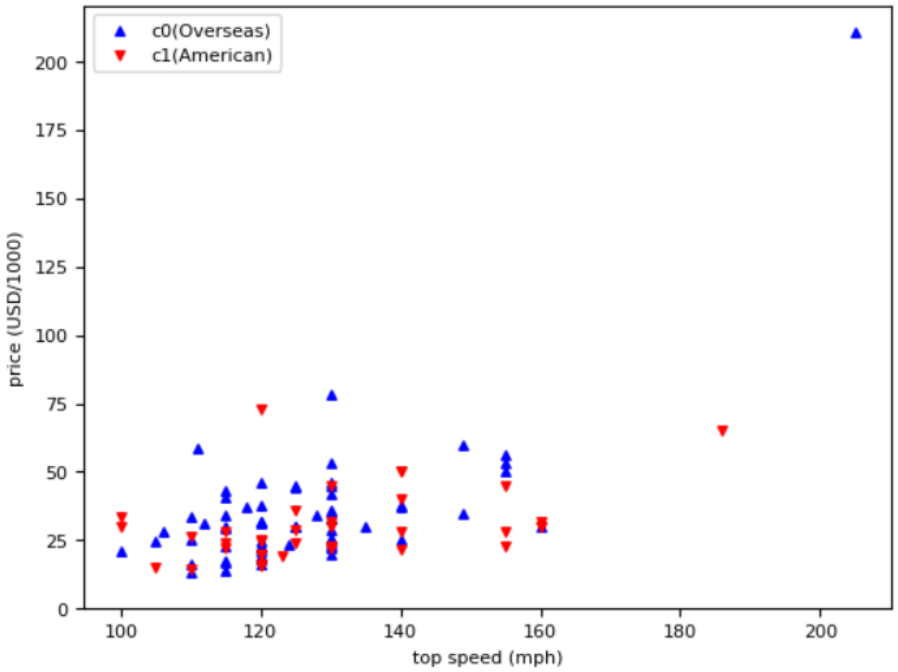
Horsepower vs. Price: A Comparison of Car Performance and Cost

Description: The graph above reveals the relationship between the horsepower and the price in the cars made in America (red), vs the ones made overseas (blue). The scatter plot shows that a car with a higher horsepower would usually be more expensive than one with a lower horsepower, with the cars made overseas being a bit more expensive. However, there are some exceptions as other factors could also affect the price of the car.

Code

```
    import math

    import matplotlib.pyplot as plt
    fig, ax = plt.subplots()
    plt.rcParams.update({'font.size': 8}) #Change the font size of the table to 6 for better readability

    import numpy as np
    category = np.array(df.iloc[:, 4]) # Get an array of all values in Category column of dataframe

    ## Separate the instances into their different categories

    pos = 0 # position of the price in the dataframe
    price_overseas = np.array([]) # Store the values of prices that fall under the Overseas category
    price_american = np.array([]) # Store the values of prices that fall under the American category
    ts_overseas = np.array([]) # Store the values of top speeds that fall under the Overseas category
    ts_american = np.array([]) # Store the values of top speeds that fall under the Overseas category

    for row in category:
        # Filter categories
        if row == "Overseas":
            price_overseas = np.append(price_overseas, df.iloc[pos, 1])
            ts_overseas = np.append(ts_overseas, df.iloc[pos, 3])
        else:
            price_american = np.append(price_american, df.iloc[pos, 1])
            ts_american = np.append(ts_american, df.iloc[pos, 3])
        pos += 1


    ## Adjust y labels
    # Statistics of Price column
    price = np.array(df.iloc[:, 1]) # Get an array of all values in Price column
    class_ranges = [] # store class ranges in bins format
    max_price = np.max(price) # Minimum price
    min_price = 0

    # Make intervals
    yticks_val = [] # Store values of yticks
    yticks_label = [] # Store labels name ytick

cur_price = 0

while cur_price <= max_price:
    yticks_label.append(str(math.floor(cur_price/1000)))
    yticks_val.append(cur_price)
    cur_price += 25000


## Draw scatter plot
ax.scatter(ts_overseas, price_overseas, c='b', marker='^', s=15, label="c0(Overseas)")
ax.scatter(ts_american, american, c='r', marker='v', s=15, label="c1(American)")

plt.yticks(yticks_val, yticks_label)
plt.title("Horsepower vs. Price: Exploring the Relationship between Car Performance and Cost", y = -0.16)
plt.xlabel("top speed (mph)")
plt.ylabel("price (USD/1000)")
plt.legend()
plt.show()
```

Graph



Top Speed vs. Price: Exploring the Relationship between Car Performance and Cost

Description: The graph above reveals the relationship between the top speed and the price in the cars made in America (red), vs the ones made overseas (blue). The scatter plot shows that the top speed of a car does not typically affect the price of a car the same way its horsepower does as there are many instances where cars with the same top speed have a wide range of prices. Furthermore, the cars made overseas were more expensive in comparison to the American made ones in terms of top speed.

d) We believe a machine learning model should easily be able to classify a new instance within our dataset. As it should be easy to tell whether the car was made overseas or in America from the make of the car, as well as find out where the top speed, horsepower, and price should fit in.

References

https://www.autoevolution.com/cars/
https://fastestlaps.com/makes
https://www.edmunds.com/sedan/#x-small-sedans
https://driving.ca/nissan/altima/