C.I.E - I

Section-1 (Theory) – 10 marks

1.a) How chatbot help businesses to improve customer services.

Chatbots can play a significant role in improving customer services for businesses in several ways:

1. 24/7 Availability: Chatbots operate round the clock, providing customers with assistance and information at any time. This ensures that customers can get help outside of regular business hours, leading to increased customer satisfaction.

2. Instant Responses:Chatbots offer immediate responses to customer queries, reducing wait times. This speed enhances the overall customer experience and prevents frustration that may arise from delayed responses.

3. Cost Efficiency: Implementing a chatbot can be more cost-effective than maintaining a large customer support team. Chatbots can handle a high volume of queries simultaneously, reducing the need for additional manpower.

4. Consistent Communication: Chatbots provide consistent and standardized responses, ensuring that customers receive the same level of service regardless of the time of day or the workload on the customer support team.

5. Efficient Information Retrieval: Chatbots can quickly access and retrieve information from databases or knowledge bases, allowing them to provide accurate and relevant information to customers.

6. **Task Automation:** Chatbots can automate routine tasks, such as order tracking, appointment scheduling, or account management. This frees up human agents to focus on more complex and high-value interactions.

7. Personalization: Advanced chatbots can analyze customer data and preferences to provide personalized recommendations or solutions. This personal touch enhances the customer experience and builds stronger relationships.

8.Multi-Platform Support: Chatbots can be integrated into various communication channels, including websites, social media, and messaging apps. This ensures that customers can interact with businesses through their preferred platforms.

In summary, chatbots contribute to improved customer services by enhancing efficiency, reducing response times, providing consistent support, and offering a personalized and convenient experience for customers.

1.b)Create two series as shown using pd.series() function.
Series_A = [1, 2, 3, 4, 5] & Series_B = [2,4,6,8,10] Get the items common to both.
CODE :

```
import pandas as pd
import numpy as np

series_A=pd.Series([1,2,3,4,5])
series_B=pd.Series([2,4,6,8,10])
cmn=series_A[series_A.isin(series_B)]
df=np.abs(cmn)
print(df.values)
```

OUTPUT :

```
[2 4]
```

2.a) List and Explain the Python Packages for machine learning.
Here's a list of some prominent ones along with brief explanations:

1. **NumPy:**
   - Purpose: NumPy is a fundamental package for scientific computing in Python. It provides support for large, multi-dimensional arrays and matrices, along with mathematical functions to operate on these arrays.

2. Pandas:
   - Purpose: Pandas is a powerful library for data manipulation and analysis. It provides data structures like DataFrame for efficient handling and analysis of structured data.

3. Matplotlib:
   - Purpose: Matplotlib is a 2D plotting library that produces static, animated, and interactive visualizations in Python. It is commonly used for data visualization and exploration.

4. Seaborn:
- **Purpose:** Seaborn is a statistical data visualization library based on Matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphics.

5. Scikit-learn:
- **Purpose:** Scikit-learn is a simple and efficient tool for data mining and data analysis. It provides simple and efficient tools for data mining and data analysis, built on NumPy, SciPy, and Matplotlib.

6. TensorFlow:
- **Purpose:** TensorFlow is an open-source machine learning framework developed by the Google Brain team. It is widely used for building and training deep learning models.

7. Keras:
- **Purpose:** Keras is a high-level neural networks API, written in Python and capable of running on top of TensorFlow, CNTK, or Theano. It allows for easy and fast prototyping of deep learning models.

8. SciPy:
- **Purpose:** SciPy is an open-source library used for mathematics, science, and engineering. It builds on NumPy and provides a large number of functions that operate on NumPy arrays and are useful for different types of scientific and engineering applications.

12. XGBoost:
- **Purpose:** XGBoost is an efficient and scalable implementation of gradient boosting. It is widely used for supervised learning tasks and is known for its speed and performance.

These libraries collectively provide a comprehensive ecosystem for machine learning and cover tasks such as data preprocessing, feature engineering, model training, and evaluation.

b) Suppose you have a dataset containing information about different products in a store. The dataset includes the following columns:

Product Name
Category (e.g., Electronics, Clothing, Home & Kitchen)
Price
Quantity Sold
Now, Create a DataFrame with sample data for this product dataset.

Perform the following operations:
1. Create a pivot table that displays the average price of each category of products.
2. Create a pivot table that shows the total revenue (price multiplied by quantity sold) and themean price of products in each category.

CODE :

```
#Create a pivot table that displays the average price of each category of products.
import pandas as pd
data={'productname'   :['mobile','tshirts','detergent','tv','pant'],
    'category'      :['electronics','clothing','home & kitchen','elkectronics','clothing'],
    'price'         :[5000,150,50,10000,300],
    'quatntity sold':[2,3,5,2,3],
}
df=pd.DataFrame(data)
pd.pivot_table(df,index='category',values='price',aggfunc='mean')
```

OUTPUT :

category	price
clothing	225
electronics	5000
elkectronics	10000
home & kitchen	50

CODE :

#Create a pivot table that shows the total revenue (price multiplied by quantity sold) and themean price of products in each category.

```
import pandas as pd
data={'productname'   :['mobile','tshirts','detergent','tv','pant'],
    'category'      :['electronics','clothing','home & kitchen','elkectronics','clothing'],
    'price'        :[5000,150,50,10000,300],
    'quantity_sold':[2,3,5,2,3],
}
df=pd.DataFrame(data)
df['total_revenue']=df['price']*df['quantity_sold']
pd.pivot_table(df,index='category',values=['total_revenue','price'],aggfunc={'total_revenue':'sum','price':'mean'})
```

OUTPUT :

category	price	total_revenue
clothing	225	1350
electronics	5000	10000
elkectronics	10000	20000
home & kitchen	50	250

Section-2 (Practical) - 20 marks

3.Data from an online platform has been collected. This data contains the passengers information. Dataset - 'mtcars.csv'

Create the following plots to visualize/summarize the data and customize appropriately.

1. Histogram to check the frequency distribution of the variable 'mpg' (Miles per gallon) and
note down the interval having the highest frequency.

2. Scatter plot to determine the relation between weight of the car and mpg

3. Bar plot to check the frequency distribution of transmission type of cars.

4. Box and Whisker plot of mpg and interpret the five number summaries.

5. Create a git repository and push source code to repo.

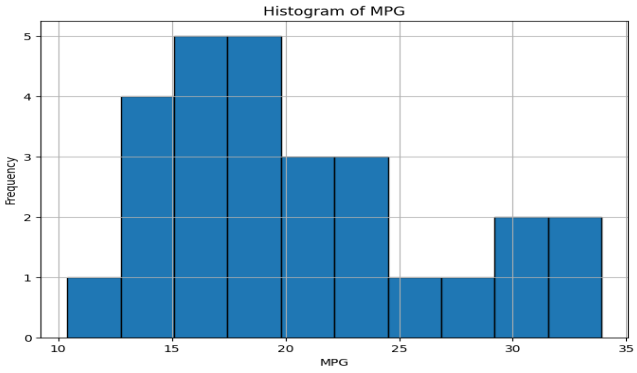
i.CODE :

#Histogram to check the frequency distribution of the variable 'mpg' (Miles per gallon)

```
import pandas as pd
import matplotlib.pyplot as plt

df=pd.read_csv('mtcars.csv')

plt.figure(figsize=(8,6))
plt.hist(df['mpg'],bins=10,edgecolor='k')
plt.title("Histogram of MPG")
plt.xlabel("MPG")
plt.ylabel("Frequency")
plt.grid(True)
plt.show()
```



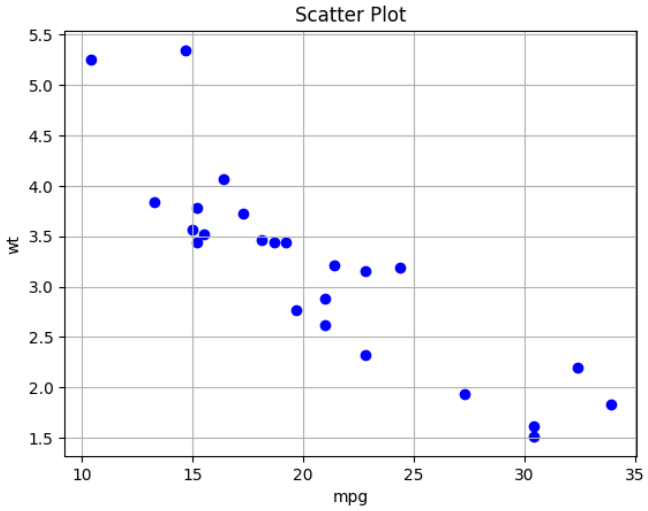
ii.CODE :

 # Scatter plot to determine the relation between weight of the car and mpg

```
import pandas as pd
import matplotlib.pyplot as plt

df=pd.read_csv('mtcars.csv')
plt.scatter(df['mpg'],df['wt'],color='blue')
plt.title('Scatter Plot')
plt.xlabel('mpg')
plt.ylabel('wt')
plt.grid(True)
plt.show()
```

OUTPUT :

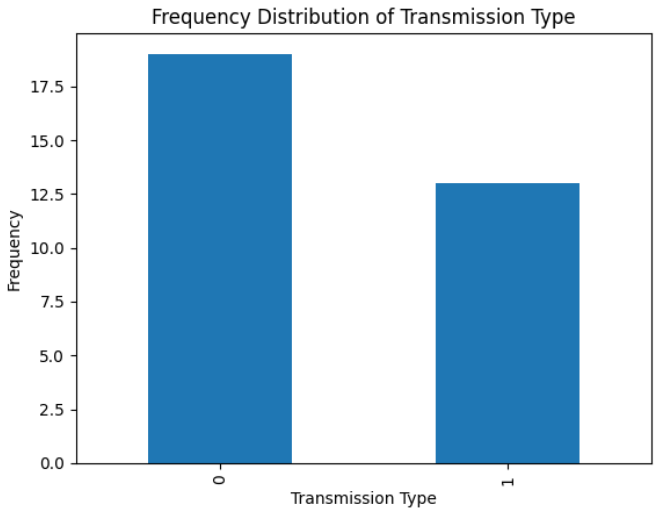


#Bar plot to check the frequency distribution of transmission type of cars.

```
import pandas as pd
import matplotlib.pyplot as plt

pd.read_csv('mtcars.csv')
transmission_counts = df['am'].value_counts()
transmission_counts.plot(kind='bar')
plt.title('Frequency Distribution of Transmission Type')
plt.xlabel('Transmission Type')
plt.ylabel('Frequency')
plt.show()
```

OUTPUT :



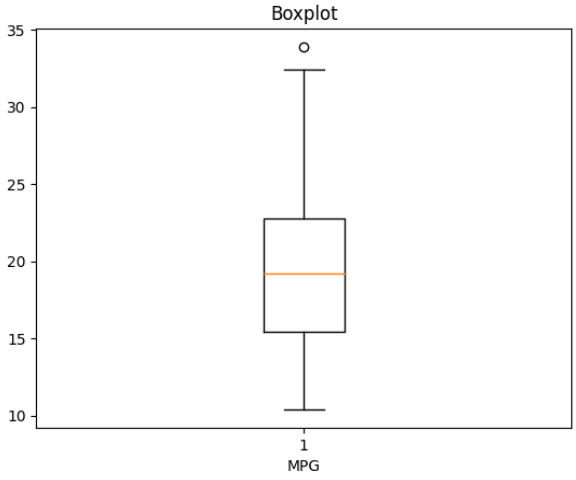
iv.CODE :

Box and Whisker plot of mpg and interpret the five number summaries.

```
import pandas as pd
import matplotlib.pyplot as plt

df=pd.read_csv('/content/mtcars.csv')
plt.boxplot(df['mpg'])
plt.title("Boxplot")
plt.xlabel('MPG')
plt.show()
```

OUTPUT :



V.Pushed source code to Repository.