Machine Learning Week – 2 Programs

7. Write a program to create and visualize a histogram using Matplotlib?
8. Write a program to plot a line graph to show trends over time with Matplotlib?
9. Write a program to create a scatter plot to visualize the relationship between two variables using Matplotlib?
10. Write a program to use Seaborn's pairplot for exploratory data analysis?
11. Write a program to create a boxplot to visualize the distribution of numerical data with Seaborn?

Assessment 1:
Write a program to convert the values in column A to a NumPy array and perform element-wise multiplication with the values in column B of the DataFrame, then add the result as a new column C in the DataFrame.

df = pd.DataFrame({'A': [1, 2, 3, 4, 5], &{'B': [6, 7, 8, 9, 10]}).

7. Write a program to create and visualize a histogram using Matplotlib?

```
# Import necessary libraries
import matplotlib.pyplot as plt
import numpy as np
# Generate some random data
data = np.random.randn(1000)
print(data)
# Create a histogram with the data
plt.hist(data, bins=30, color='blue', edgecolor='black')
# Add titles and labels
plt.title('Histogram Example')
plt.xlabel('Value')
plt.ylabel('Frequency')
# Show the plot
plt.show()
```

8. Write a program to plot a line graph to show trends over time with Matplotlib?

```
# Import necessary libraries
import matplotlib.pyplot as plt
# Sampledata:Time (e.g., months) and corresponding values (eg: sales,temperature, etc.)
time = ['January', 'February', 'March', 'April', 'May', 'June', 'July', 'August', 'September',
'October', 'November', 'December']
values = [5, 6, 7, 8, 10, 9, 11, 12, 14, 15, 17, 20]  # Example data
# Create a line plot
plt.plot(time, values, marker='o', color='b', linestyle='-', markersize=6, label='Trends')
# Add titles and labels
plt.title('Trends Over Time')
plt.xlabel('Month')
plt.ylabel('Values')
# Show the legend
plt.legend()
# Rotate x-axis labels for better readability
plt.xticks(rotation=45)
# Display the plot
plt.tight_layout()  # Adjust layout to prevent clipping of x-axis labels
plt.show()
```

9. Write a program to create a scatter plot to visualize the relationship between two variables using Matplotlib?

```
# Import necessary libraries
import matplotlib.pyplot as plt
import numpy as np
# Sample data: x and y represent two variables
x = np.random.rand(50)  # 50 random values for x
y = 2 * x + np.random.randn(50) * 0.5  # y is linearly dependent on x with some noise
# Create the scatter plot
plt.scatter(x, y, color='blue', edgecolor='black', alpha=0.7)
# Add titles and labels
plt.title('Scatter Plot Example')
plt.xlabel('X Variable')
plt.ylabel('Y Variable')
# Display the plot
plt.show()
```

10. Write a program to use Seaborn's pairplot for exploratory data analysis?

```
# Import necessary libraries
import seaborn as sns
import matplotlib.pyplot as plt
# Load a sample dataset (e.g., Iris dataset)
iris = sns.load_dataset('iris')
# Create a pairplot to visualize relationships between variables
sns.pairplot(iris, hue='species', diag_kind='hist', markers='o')
# Show the plot
plt.show()
```

11. Write a program to create a boxplot to visualize the distribution of numerical data with Seaborn?

```
# Step 1: Import necessary libraries
import seaborn as sns
import matplotlib.pyplot as plt
# Step 2: Load the Iris dataset
iris = sns.load_dataset('iris')
# Step 3: Create a boxplot to visualize the distribution of numerical features
plt.figure(figsize=(10, 6))
sns.boxplot(data=iris[['sepal_length', 'sepal_width', 'petal_length', 'petal_width']])
# Step 4: Customize the plot (optional)
plt.title('Boxplot of Numerical Features in Iris Dataset')
plt.xlabel('Features')
plt.ylabel('Value')
# Step 5: Show the plot
plt.show()
```

Assessment 1: (This Program must write by students)
Write a program to convert the values in column A to a NumPy array and perform element-wise multiplication with the values in column B of the DataFrame, then add the result as a new column C in the DataFrame.

df = pd.DataFrame({'A': [1, 2, 3, 4, 5], &{'B': [6, 7, 8, 9, 10]}).

Machine Learning Week – 3 Programs

12. Write a program to create a heatmap to visualize data intensity or correlations using Seaborn?

13. Write a program to plot a bar chart to compare categories of data with Seaborn?

14. Write a program to create a pie chart to represent proportions of categories in a dataset with Matplotlib?

15. Write a program to create a stacked bar chart to show composition of categories in a dataset with Matplotlib?

16. Write a program to plot a normal distribution curve and add a Kernel Density Estimate (KDE) using Seaborn?

17. Write a program to create a grid of subplots to display multiple visualizations in a single figure using Matplotlib?

12. Write a program to create a heatmap to visualize data intensity or correlations using Seaborn?

```
# Step 1: Import necessary libraries
import seaborn as sns
import matplotlib.pyplot as plt
import pandas as pd
# Step 2: Load your exercise dataset (replace with actual path or data)
# For the sake of example, we're creating a small sample dataset
data = {
    'exercise_time': [30, 45, 25, 60, 40, 50, 20],
    'calories_burned': [250, 350, 200, 450, 300, 375, 180],
    'steps': [5000, 7500, 4000, 10000, 6000, 8000, 3500],
    'heart_rate': [130, 140, 125, 150, 135, 145, 120],
    'age': [25, 30, 35, 40, 45, 50, 55]
}
# Create a DataFrame
exercise_df = pd.DataFrame(data)
```

 # Step 3: Compute the correlation matrix for numerical columns

corr_matrix = exercise_df.corr()

 # Step 4: Create the heatmap

plt.figure(figsize=(8, 6)) # Set the figure size

sns.heatmap(corr_matrix, annot=True, cmap='coolwarm', fmt='.2f', linewidths=0.5, cbar=True)

 # Step 5: Add title and labels (optional)

plt.title('Correlation Heatmap of Exercise Dataset')

 # Step 6: Show the plot

plt.show()

13. Write a program to plot a bar chart to compare categories of data with Seaborn?

 # Step 1: Import necessary libraries

import seaborn as sns

import matplotlib.pyplot as plt

 # Step 2: Load the Titanic dataset (a built-in Seaborn dataset)

titanic = sns.load_dataset('titanic')

 # Step 3: Create a bar chart comparing the average age by passenger class

plt.figure(figsize=(8, 6))

sns.barplot(x='class', y='age', data=titanic, ci=None)

 # Step 4: Add labels and title

plt.title('Average Age by Passenger Class')

plt.xlabel('Passenger Class')

plt.ylabel('Average Age')

 # Step 5: Show the plot

plt.show()

14. Write a program to create a pie chart to represent proportions of categories in a dataset with Matplotlib?

 # Step 1: Import necessary libraries

```
import matplotlib.pyplot as plt

import pandas as pd
```

 # Step 2: Create or load the dataset (for example purposes, we're creating a simple one)

```
data = ['Apple', 'Banana', 'Apple', 'Orange', 'Banana', 'Apple', 'Orange', 'Apple', 'Grapes', 'Banana']
```

 # Convert the list to a pandas Series

```
fruit_series = pd.Series(data)
```

 # Step 3: Count the frequency of each category (fruit)

```
fruit_counts = fruit_series.value_counts()
```

 # Step 4: Create a pie chart to show proportions

```
plt.figure(figsize=(7, 7))  # Optional: Adjust the figure size

plt.pie(fruit_counts, labels=fruit_counts.index, autopct='%1.1f%%', startangle=90, colors=['#ff9999','#66b3ff','#99ff99','#ffcc99'])
```

 # Step 5: Add a title

```
plt.title('Proportions of Different Fruits')
```

 # Step 6: Display the plot

```
plt.show()
```

15. Write a program to create a stacked bar chart to show composition of categories in a dataset with Matplotlib?

 # Step 1: Import necessary libraries

```
import matplotlib.pyplot as plt

import pandas as pd
```

 # Step 2: Create or load the dataset (for example purposes, creating a simple dataset)

```
data = {

    'Product': ['Product A', 'Product B', 'Product C', 'Product D'],

    'North': [100, 150, 200, 250],

    'South': [120, 160, 180, 220],
```

```
    'East': [110, 130, 170, 210],
    'West': [90, 140, 160, 180]
}
```

 # Step 3: Convert the dataset into a pandas DataFrame

```
df = pd.DataFrame(data)
```

 # Step 4: Set the product names as the index

```
df.set_index('Product', inplace=True)
```

 # Step 5: Plot the stacked bar chart

```
ax = df.plot(kind='bar', stacked=True, figsize=(10, 7), color=['#ff9999', '#66b3ff', '#99ff99', '#ffcc99'])
```

 # Step 6: Add labels and title

```
plt.title('Sales Composition by Region for Each Product')
plt.xlabel('Product')
plt.ylabel('Sales')
```

 # Step 7: Display the plot

```
plt.show()
```

16. Write a program to plot a normal distribution curve and add a Kernel Density Estimate (KDE) using Seaborn?

 # Step 1: Import necessary libraries

```
import seaborn as sns
import matplotlib.pyplot as plt
import numpy as np
```

 # Step 2: Generate synthetic data (normally distributed)

```
np.random.seed(42)  # For reproducibility
data = np.random.normal(loc=0, scale=1, size=1000)  # Normal distribution (mean=0, std=1)
```

 # Step 3: Create the plot with a histogram and KDE

```
plt.figure(figsize=(8, 6))
sns.histplot(data, kde=True, color='skyblue', stat='density', bins=30, edgecolor='black')
```

 # Step 4: Add a title and labels

```
plt.title('Normal Distribution with KDE', fontsize=16)
```

```
plt.xlabel('Value', fontsize=12)
plt.ylabel('Density', fontsize=12)
# Step 5: Show the plot
plt.show()
```

17. Write a program to create a grid of subplots to display multiple visualizations in a single figure using Matplotlib?

```
# Step 1: Import necessary libraries
import matplotlib.pyplot as plt
import numpy as np
import seaborn as sns
# Step 2: Generate some sample data
np.random.seed(42)
data1 = np.random.normal(loc=0, scale=1, size=1000)  # Normal distribution data
data2 = np.random.uniform(low=-2, high=2, size=500)  # Uniform distribution data
categories = ['A', 'B', 'C', 'D']
values = [5, 7, 3, 8]
# Step 3: Create a 2x2 grid of subplots
fig, axs = plt.subplots(2, 2, figsize=(10, 8))  # 2 rows and 2 columns of subplots
# Step 4: Plot on each subplot
# First subplot: Histogram of data1
axs[0, 0].hist(data1, bins=30, color='skyblue', edgecolor='black')
axs[0, 0].set_title('Histogram of Normal Distribution')
# Second subplot: Histogram of data2
axs[0, 1].hist(data2, bins=20, color='salmon', edgecolor='black')
axs[0, 1].set_title('Histogram of Uniform Distribution')
# Third subplot: Bar chart for categories and values
axs[1, 0].bar(categories, values, color='lightgreen')
axs[1, 0].set_title('Bar Chart of Categories')
# Fourth subplot: Boxplot using Seaborn
sns.boxplot(data=[data1, data2], ax=axs[1, 1], palette="Set2")
```

```
axs[1, 1].set_title('Boxplot of Two Datasets')
# Step 5: Adjust layout to prevent overlap
plt.tight_layout()
# Step 6: Show the plot
plt.show()
```