GLS University
FCAIT
IMCA SEM VI
222301605 PRACTICALS ON MACHINE LEARNING
Unit 2 Practical Assignment

1. You're working with a dataset containing the age of customers for a marketing campaign with 10 customers. Make a dataset of it, take into dataframe and display the histogram of it with the following conditions:

- Bins should be 5.
- Give title “Age distribution”
- Edge color should be black
- Color should be blue.
- Give x and y axis labels accordingly

2. A house price dataset contains a column with property prices. 100 Prices should be generated randomly. Display the histogram of it with the following conditions:

- Bins should be 10.
- Give title “Price distribution”
- Edge color should be gray
- Color should be pink.
- Give x and y axis labels accordingly.

3. {'Age': [25, 30, 35, None, 28, 32, 45, None, 30, 28]} Create a histogram after deleting or impute missing values.

4. models = ['Logistic Regression', 'SVM', 'Random Forest'] training_times = [2.1, 12.5, 8.4]
Create a bar graph with following conditions:

- Graph color should be green
- It should be horizontal graph.

5. Create a group bar graph and Stacked bar graph for the following data:
Shape = ['Square1', 'Rect', 'Square2']
Length = [10, 15, 20]
Height = [12, 18, 25]

6. Create a scatter plot for the following data

1. house_size = [1000, 1500, 2000, 2500, 3000] house_price = [200000, 250000, 300000, 350000, 400000]
2. actual_prices = [200000, 250000, 300000, 350000, 400000] predicted_prices = [210000, 245000, 290000, 360000, 390000]

7. Display box plot and show outliers for the following data

1. Transaction_amounts = [10, 15, 20, 25, 30, 35, 1000, 2000]
2. cv_scores_model_1 = [0.82, 0.84, 0.83, 0.81, 0.95]
   cv_scores_model_2 = [0.88, 0.87, 0.89, 0.86, 0.88]
3. daily_returns = [0.01, 0.02, 0.015, -0.03, 0.02, -0.1, 0.03, 0.08]

8. Display heatmap for the following data

1. data = { 'Feature1': np.random.rand(100), 'Feature2': np.random.rand(100) * 2, 'Feature3': np.random.rand(100) * 3, 'Target': np.random.rand(100) * 4, }
2. true_values = np.random.rand(10) predicted_values = true_values + np.random.normal(0, 0.1, 10)