RAJALAKSHMI ENGINEERING COLLEGE

Approved by AICTE | Affiliated to Anna University | Accredited by NAAC

Department of Computer Science and Engineering

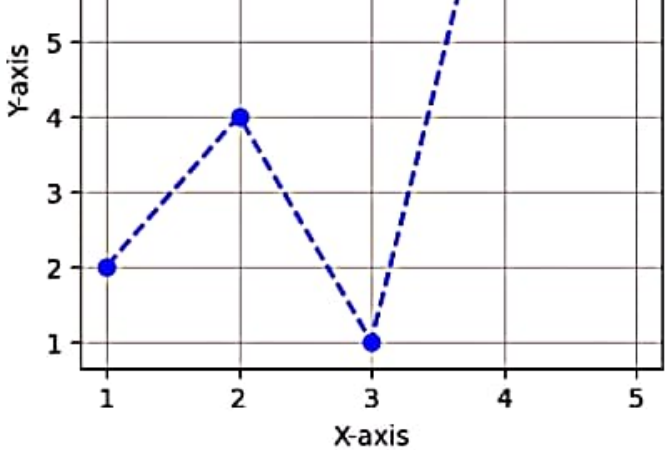
CS23334 Fundamentals of Data Science Lab

III semester II Year (2023R)

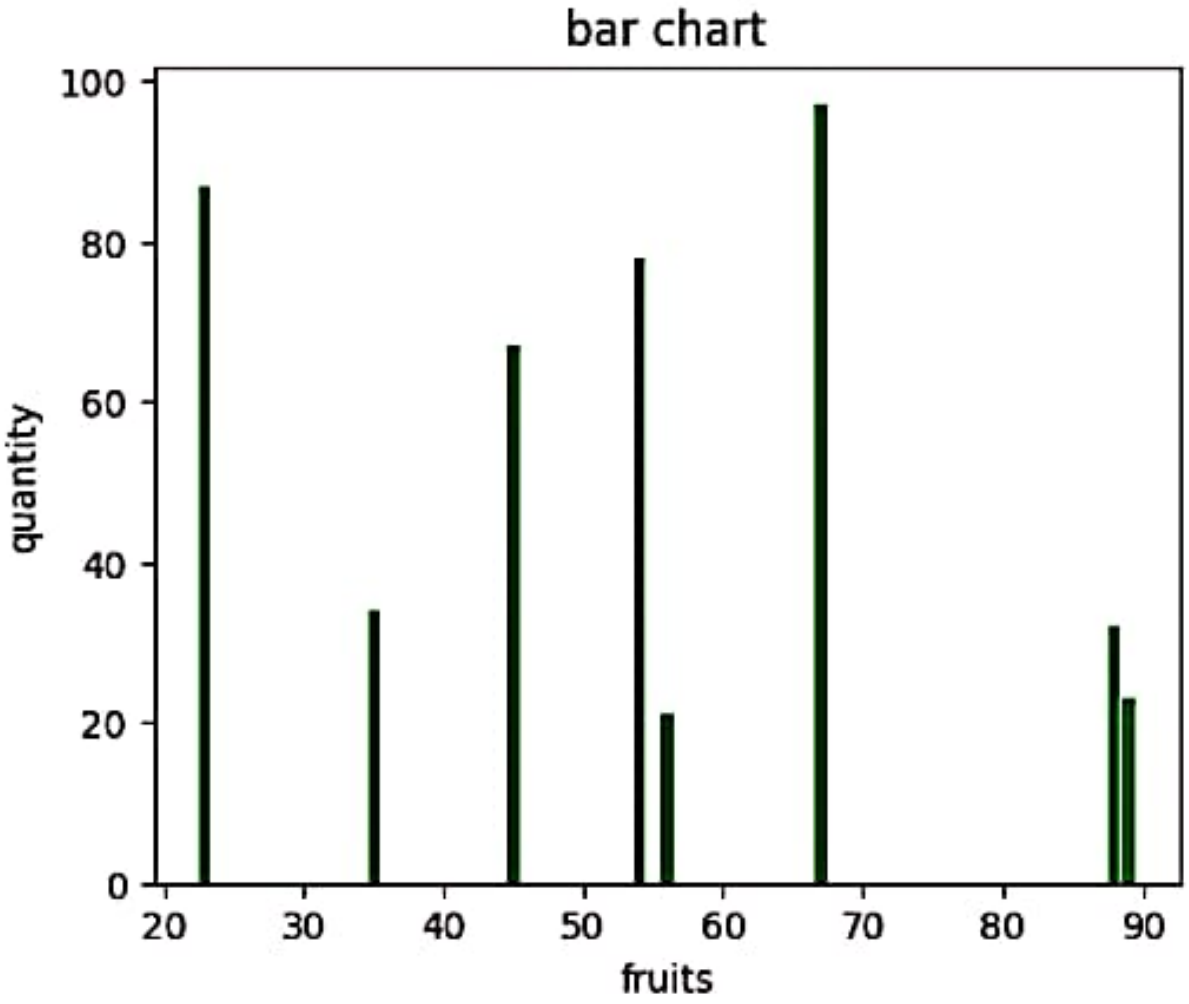
Name of the Student :SABARIPRABU M

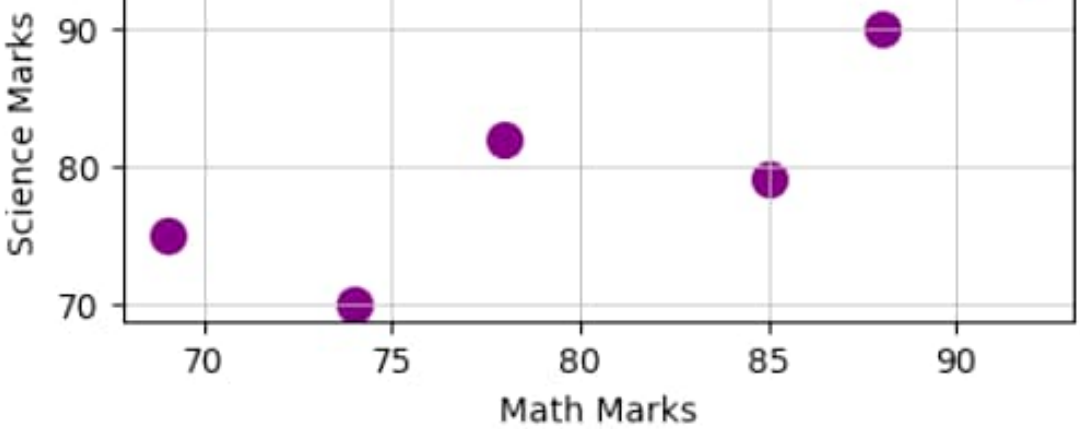
Register Number :240701446

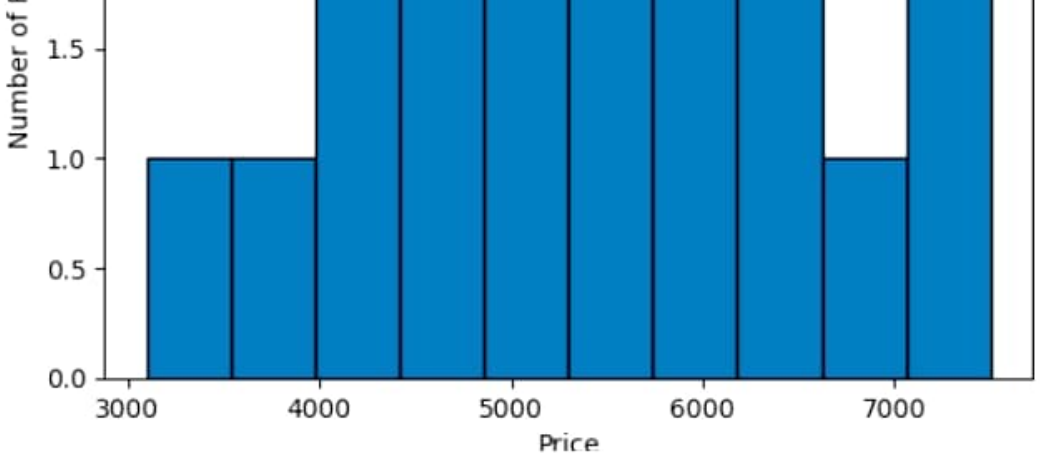

India
Pakistan
Average Team Score
320
300
280
2018
2019
2020
2021
2022
2023
2024
Year


Y-axis
1
2
3
4
5
1
2
3
4
5
X-axis

```
import matplotlib.pyplot as plt
x=[45,23,67,35,67,88,67,89,56,54]
y=[67,87,45,34,97,32,44,23,21,78]
plt.figure(figsize=(5,4))
plt.bar(x,y,color="green")
plt.title("bar chart")
plt.xlabel("fruits")
plt.ylabel("quantity")
plt.show()
```




Science Marks
90
80
70
70
75
80
85
90
Math Marks


Number of
1.5
1.0
0.5
0.0
3000
4000
5000
6000
7000
Price

```
plt.title('Distribution of Flipper Length (KDE)')
plt.show()

# Distribution plot without KDE - using color 'teal'
sns.displot(penguins['flipper_length_mm'].dropna(), kde=False, color='teal')
plt.title('Flipper Length Distribution (No KDE)')
plt.show()

# Jointplot between flipper length and body mass with custom colors
sns.jointplot(x='flipper_length_mm', y='body_mass_g', data=penguins, color='darkorange')
plt.show()
```

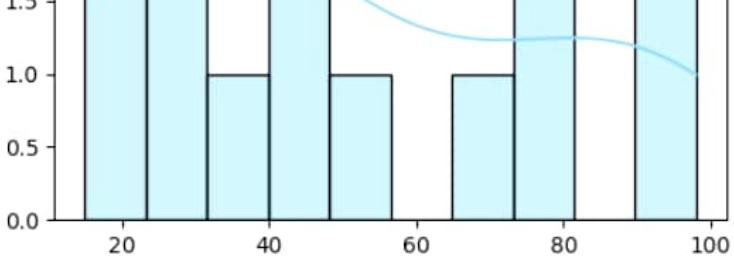
```
plt.title('Penguin Species Distribution')
plt.ylabel('')
plt.show()

# Bar chart of island counts with a dark palette
penguins['island'].value_counts().plot(kind='bar', color=['#D2691E', '#5F9EA0', '#9ACD32'])
plt.title('Number of Penguins by Island')
plt.xlabel('Island')
plt.ylabel('Count')
plt.show()
```

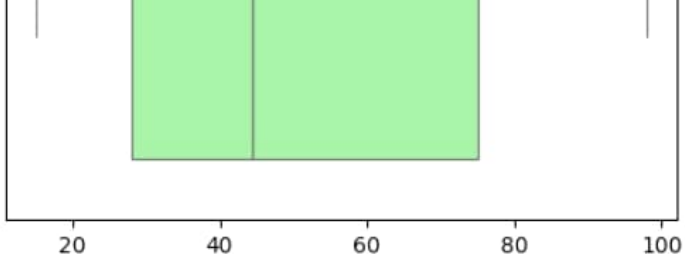
1	3800.0	Female
2	3250.0	Female
3	NaN	NaN
4	3450.0	Female

body_mass_g
0.6
-0.47
0.87
1
−0.4
bill_length_mm
bill_depth_mm
flipper_length_mm
body_mass_g

```
print("Cleaned data:", cleaned_data)

plt.figure(figsize=(12,5))

plt.subplot(1,2,1)
sns.histplot(data, bins=10, kde=True, color='skyblue')
plt.title('Histogram - Original Data')

plt.subplot(1,2,2)
sns.boxplot(x=data, color='lightgreen')
plt.title('Boxplot - Original Data')
```

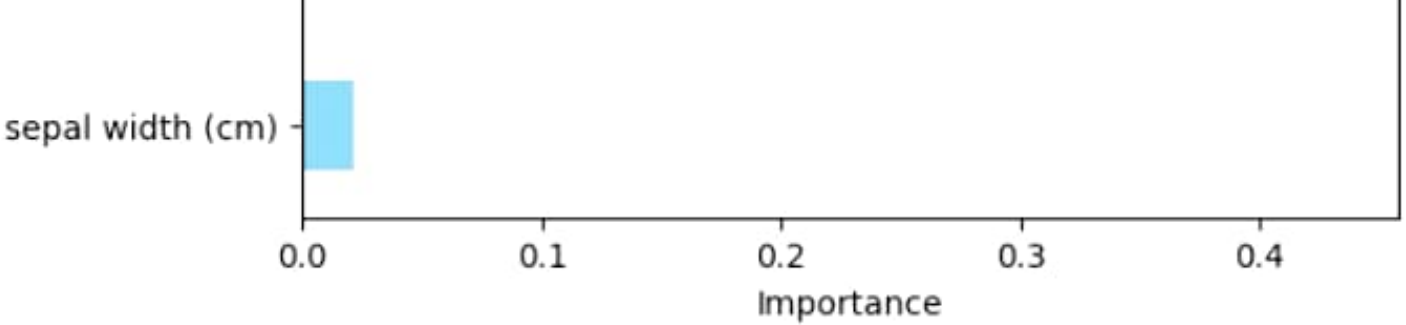
```
sns.boxplot(x=cleaned_data, color='lightcoral')
plt.title('Boxplot - Cleaned Data')

plt.show()
```


1.5
1.0
0.5
0.0
20
40
60
80
100


20
40
60
80
100

```
plt.title("Top 10 Feature Importances")
plt.xlabel("Importance")
plt.ylabel("Feature")
plt.show()
```

```
             Feature  Importance
2  petal length (cm)    0.436130
3   petal width (cm)    0.436065
0  sepal length (cm)    0.106128
1   sepal width (cm)    0.021678
```


sepal width (cm)
0.0
0.1
0.2
0.3
0.4
Importance

```
print("Testing Score:", model.score(x_test, y_test))
print("Coefficient:", model.coef_)
print("Intercept:", model.intercept_)
import pickle
pickle.dump(model, open('SalaryPred.model', 'wb'))
model = pickle.load(open('SalaryPred.model', 'rb'))
yr_of_exp = float(input("Enter Years of Experience: "))
yr_of_exp_NP = np.array([[yr_of_exp]])
Salary = model.predict(yr_of_exp_NP)
print(f"Estimated Salary for {yr_of_exp} years of experience is: {Salary[0][0]:.2f}")
```

```
25%            3.200000   56642.000000
50%            4.900000   63218.000000
75%            7.900000  101302.000000
max           10.500000  122391.000000
Training Score: 0.9393576307207374
Testing Score: 0.9170535247423002
Coefficient: [[9315.01199233]]
Intercept: [25125.45885762]
Enter Years of Experience: 4
Estimated Salary for 4.0 years of experience is: 62385.51
```

```
    if test_score > train_score:
        print("Test: {:.3f} | Train: {:.3f} | Random state: {}".format(test_score, train_score, i))
x_train, x_test, y_train, y_test = train_test_split(features, label, test_size=0.2, random_state=42)
final_model = LogisticRegression(max_iter=200)
final_model.fit(x_train, y_train)
print("Train Accuracy:", final_model.score(x_train, y_train))
print("Test Accuracy:", final_model.score(x_test, y_test))
print(classification_report(label, final_model.predict(features), target_names=iris.target_names))
```

accuracy			0.98	150
macro avg	0.98	0.98	0.98	150
weighted avg	0.98	0.98	0.98	150

```
print(classification_report(label, model_KNN.predict(features)))
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 150 entries, 0 to 149
Data columns (total 5 columns):
 #   Column        Non-Null Count  Dtype  
---  ------        --------------  -----  
 0   sepal_length  150 non-null    float64
 1   sepal_width   150 non-null    float64
 2   petal_length  150 non-null    float64
 3   petal_width   150 non-null    float64
 4   variety       150 non-null    object 
dtypes: float64(4), object(1)
memory usage: 6.0+ KB
```

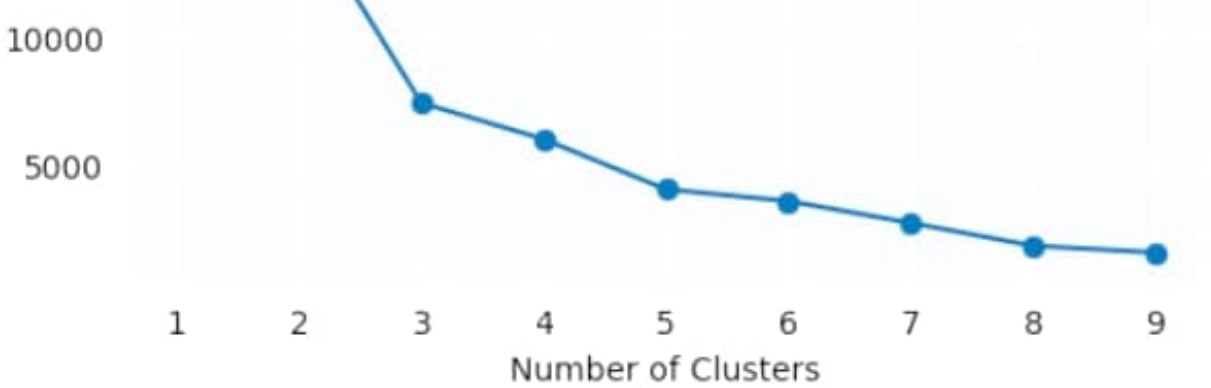
Classification Report:

	precision	recall	f1-score	support
Setosa	1.00	1.00	1.00	50
Versicolor	0.98	0.94	0.96	50
Virginica	0.94	0.98	0.96	50
accuracy			0.97	150
macro avg	0.97	0.97	0.97	150
weighted avg	0.97	0.97	0.97	150

```
final_df = df.copy()
final_df['Label'] = model.predict(features)
sns.set_style("darkgrid")
palette = sns.color_palette("Spectral", 5)
plt.figure(figsize=(8, 6))
for label in range(5):
    cluster = final_df[final_df['Label'] == label]
    plt.scatter(cluster['Annual Income (k$)'], cluster['Spending Score (1-100)'],
                s=80, color=palette[label], label=f'Cluster {label+1}', alpha=0.7, edgecolor='black')
plt.title("Customer Segments (K-Means Clustering)", fontsize=14)
plt.xlabel("Annual Income (k$)")
plt.ylabel("Spending Score (1-100)")
```

```
RangeIndex: 20 entries, 0 to 19
Data columns (total 5 columns):
 #   Column                  Non-Null Count  Dtype 
---  ------                  --------------  ----- 
 0   CustomerID              20 non-null     int64 
 1   Gender                  20 non-null     object
 2   Age                     20 non-null     int64 
 3   Annual Income (k$)      20 non-null     int64 
 4   Spending Score (1-100)  20 non-null     int64 
dtypes: int64(4), object(1)
memory usage: 932.0+ bytes
```


Spending Score
40
20


10000
5000
1
2
3
4
5
6
7
8
9
Number of Clusters

```
else:
    print("\nFail to Reject Null Hypothesis – no significant difference.")
```

```
Chi-Square Statistic: 14.0
P-Value: 0.002905152774267437

Reject Null Hypothesis – observed distribution differs from expected.
```

```
else:
    print("\nFail to Reject Null Hypothesis - no significant difference.")
```

```
T-Statistic: 4.330127018922191
P-Value: 0.000978488712899117

Reject Null Hypothesis - significant difference.
```

```
    print("\nReject Null Hypothesis - significant difference.")
else:
    print("\nFail to Reject Null Hypothesis - no significant difference.")
```

```
Z-Statistic: 4.1569219381653
P-Value: 3.225641456243845e-05

Reject Null Hypothesis - significant difference.
```

```
    print("\nReject Null Hypothesis - at least one group differs.")
else:
    print("\nFail to Reject Null Hypothesis - all groups are similar.")
```

```
F-Statistic: 123.12643678160978
P-Value: 1.006506307348831e-08

Reject Null Hypothesis - at least one group differs.
```