IRIS FLOWERS CLASSIFICATION

Applied Machine Learning Project CCDS 322

TEAM MEMEBER

01

Khlood Alamoudi
2115339

02

Sara Alghamdi
1911173

03

Joud Aljehani
2111644

04

Lama Alghamdi
2006847

INTRODUCTION

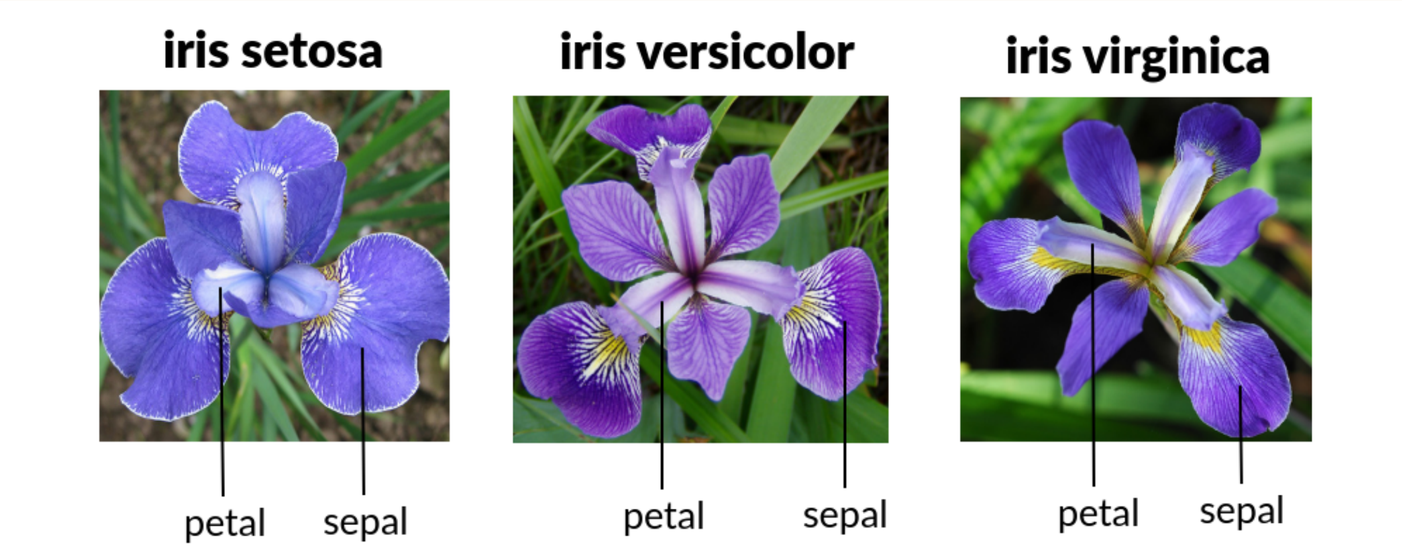
Develop a machine learning model to classify iris flowers based on their sepal and petal measurements. The model will be trained on a dataset of iris flower measurements and then used to predict the species of new iris flowers.

Goal :

The goal of this project is to train a machine learning model to accurately classify iris flowers into one of three species: Iris setosa, Iris versicolor, or Iris virginica.

ABOUT THE DATASET

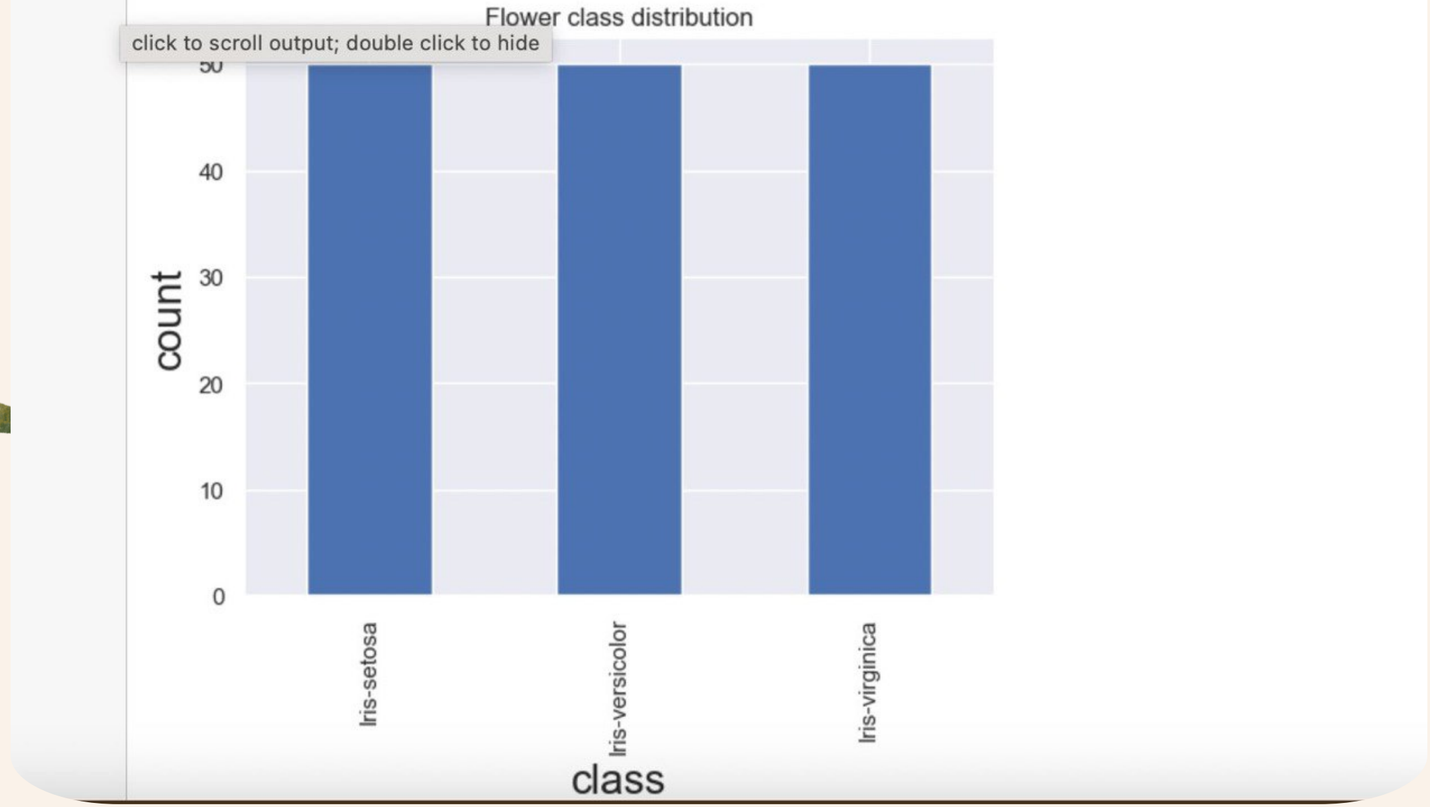
The Iris (or Iris) is a dataset that contains four features (length and width of sepals and petals) of 50 samples of three species of Iris (Iris setosa, Iris virginica, and Iris versicolor) in total 150 records. These measures were used to create a linear discriminant model to classify the species. The dataset is often used in data mining, classification and clustering examples and to test algorithms.



DATASET CLASS DISTRIBUTION

```
In [10]: nameplot = data['species'].value_counts().plot.bar(title='Flower class distribution')
         nameplot.set_xlabel('class',size=20)
         nameplot.set_ylabel('count',size=20)
```

```
Out[10]: Text(0, 0.5, 'count')
```



ABOUT THE DATASET

	sepal_length	sepal_width	petal_length	petal_width
count	150.000000	150.000000	150.000000	150.000000
mean	5.843333	3.054000	3.758667	1.198667
std	0.828066	0.433594	1.764420	0.763161
min	4.300000	2.000000	1.000000	0.100000
25%	5.100000	2.800000	1.600000	0.300000
50%	5.800000	3.000000	4.350000	1.300000
75%	6.400000	3.300000	5.100000	1.800000
max	7.900000	4.400000	6.900000	2.500000

QUESTIONS/HYPOTHESES

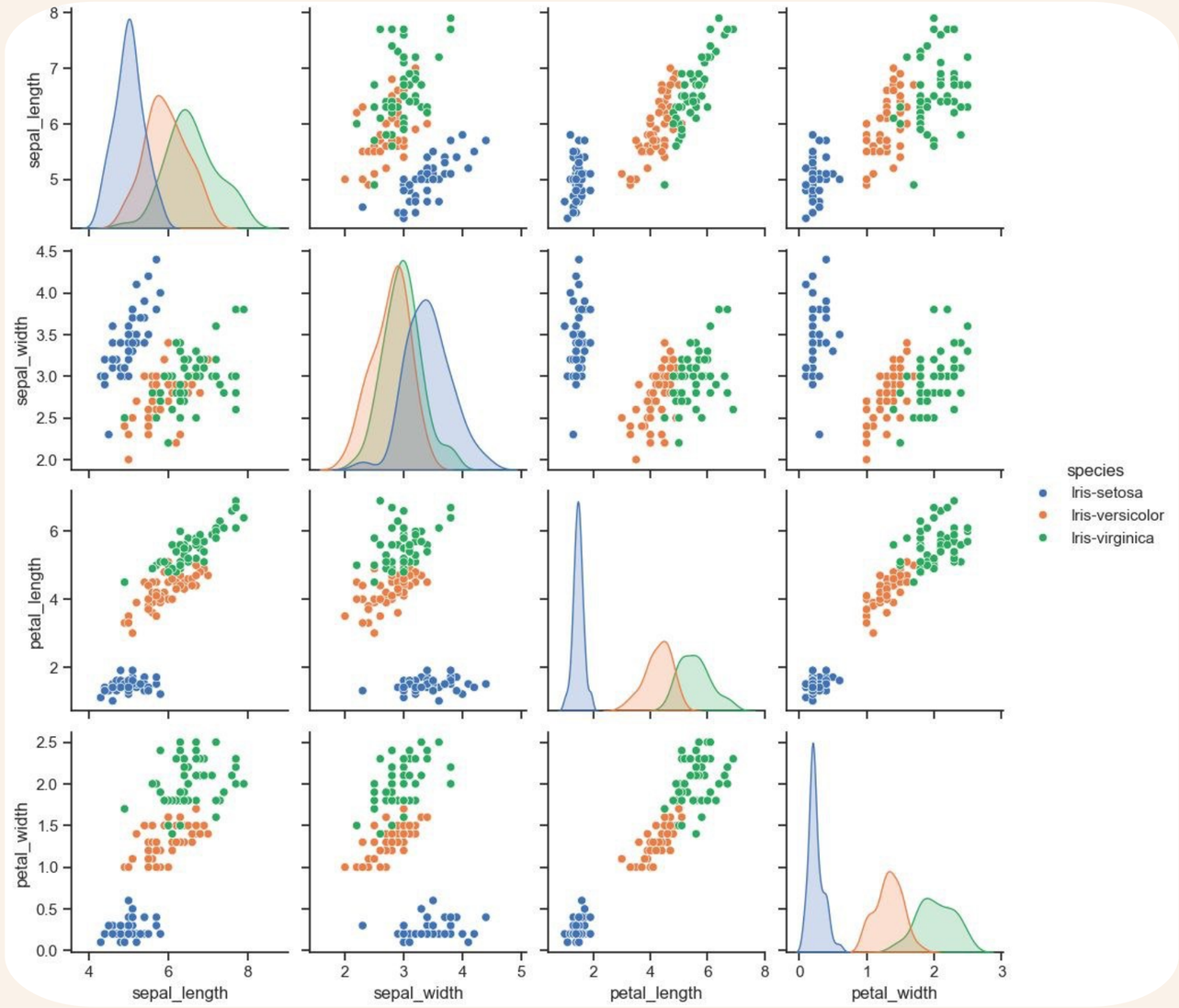
- Can a machine learning model achieve an accuracy of at least 95% in classifying Iris flowers?

- Hypothesis: A support vector machine (SVM) model will achieve an accuracy of at least 95% in classifying Iris flowers.

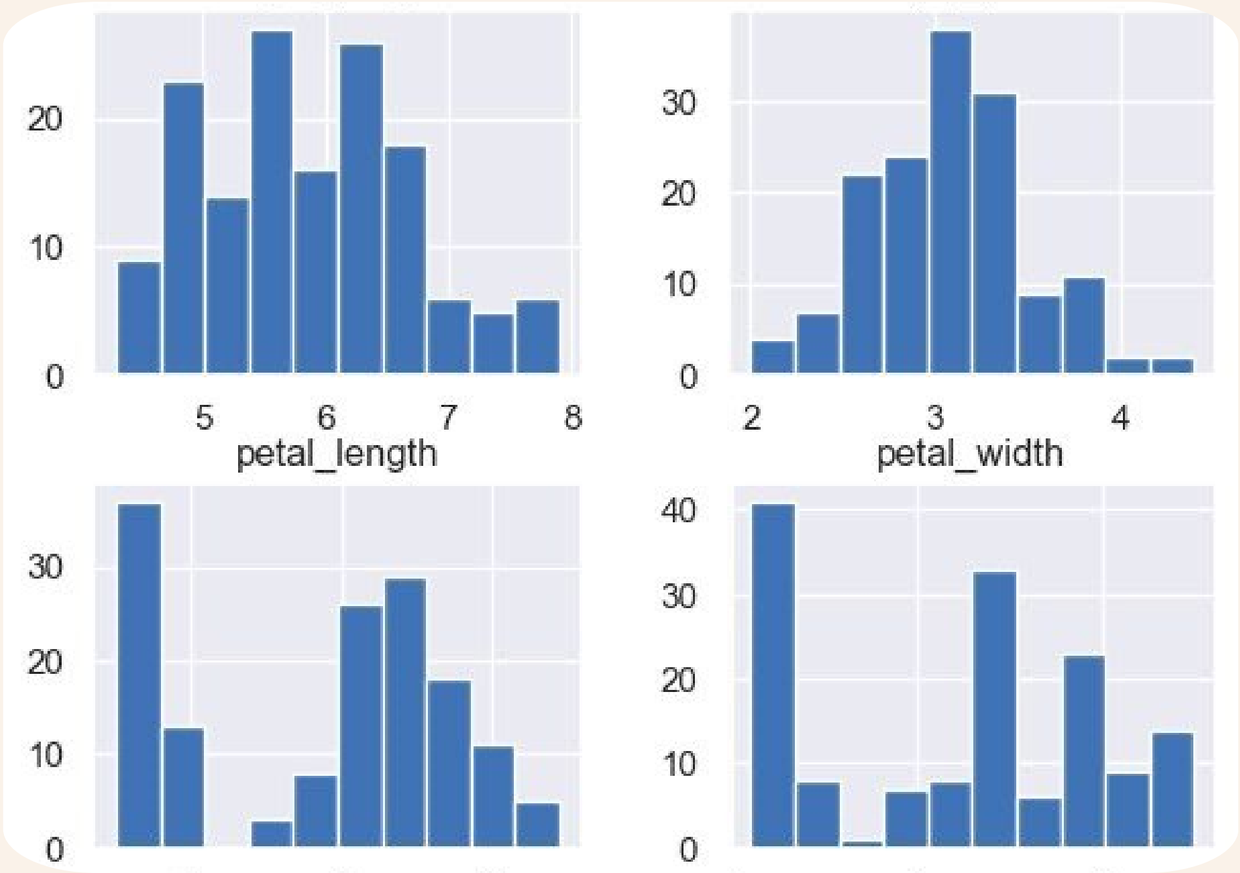
EXPLORATORY DATA ANALYSIS

Viewing Data Visualization and Insights

EDA


sepal_length
sepal_width
petal_length
petal_width
species
Iris-setosa
Iris-versicolor
Iris-virginica

DISTRIBUTIONS OF FEATURES AND TARGET

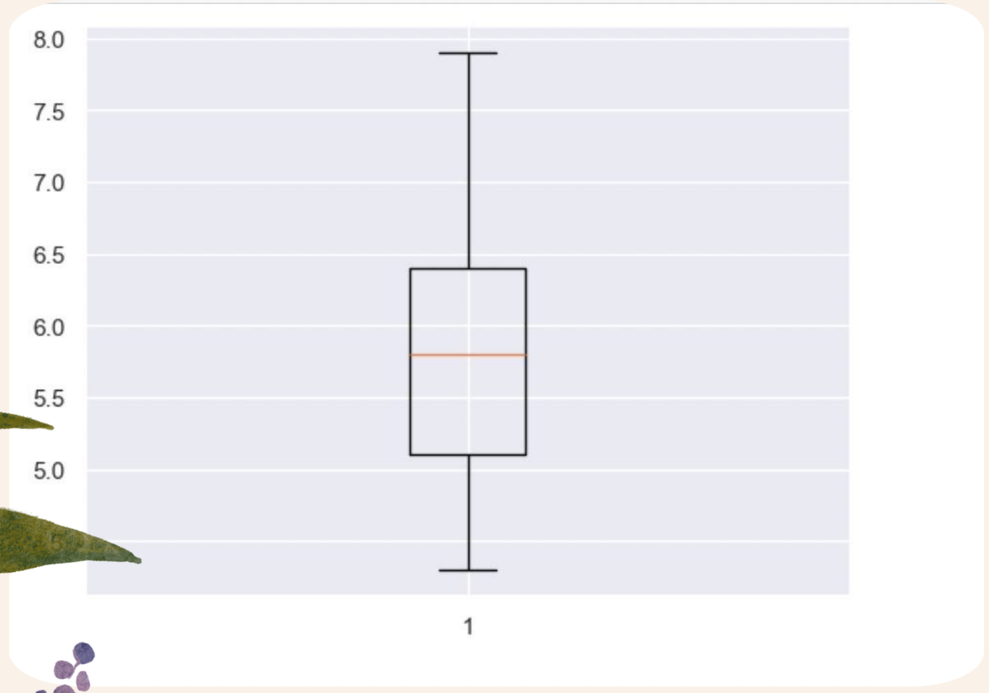


BOXPLOT

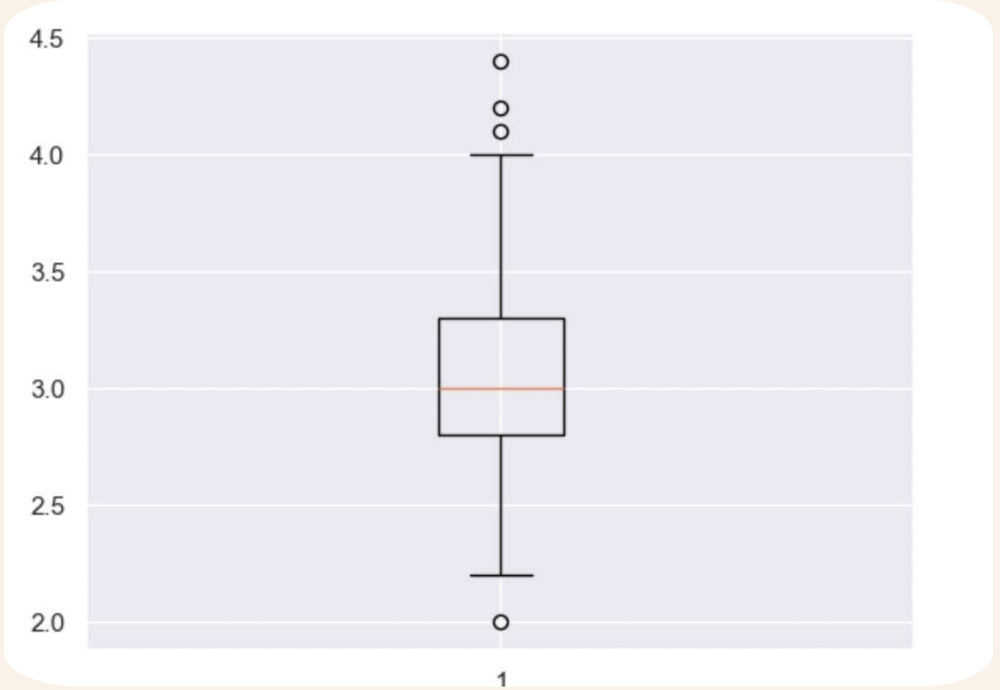
Here we draw a boxplot to visually inspect the presence of outliers in the dataset.

```
plt.figure(1)
plt.boxplot([data['sepal_length']])
plt.figure(2)
plt.boxplot([data['sepal_width']])
plt.show()
plt.figure(3)
plt.boxplot([data['petal_length']])
plt.figure(4)
plt.boxplot([data['petal_width']])
plt.show()
```

BOXPLOT



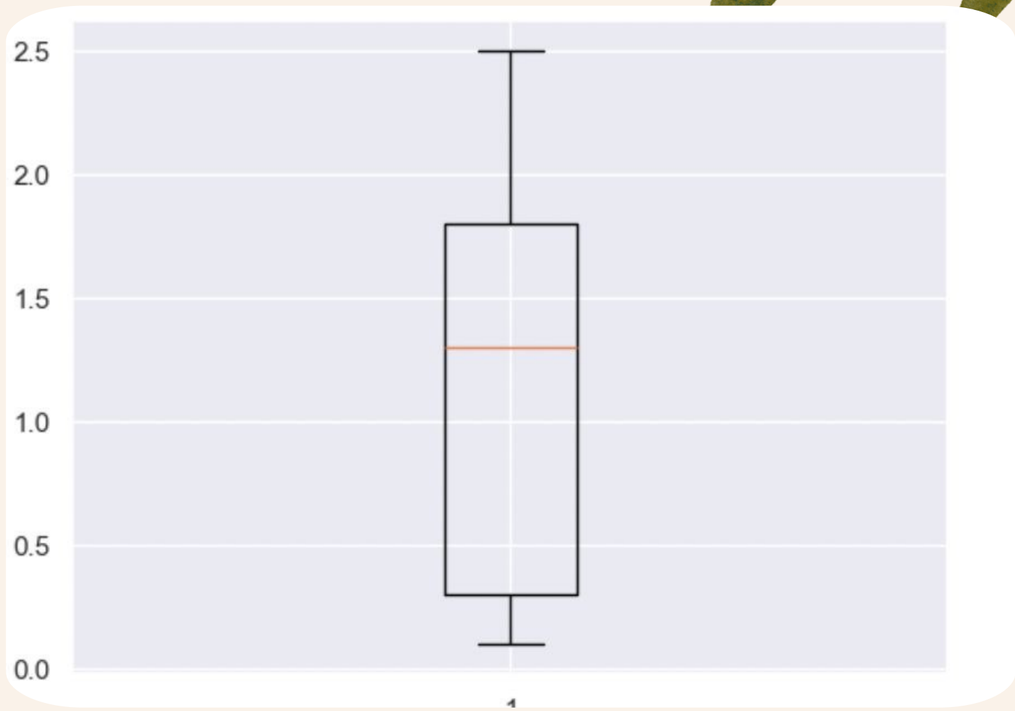
1



2

BOXPLOT

3



4

DATA PREPROCESSING

Checking for null values

```
]: #Checking for the null values
   data.isnull().sum()
```

```
]: sepal_length    0
   sepal_width     0
   petal_length    0
   petal_width     0
   species         0
   dtype: int64
```

This absence of missing data simplifies the preprocessing steps, allowing us to proceed with the analysis without the need for imputation or handling missing values.

PREPARE DATA FOR MODELING

Train test split

```
train, test = train_test_split(data, test_size = 0.3)
print(train.shape)
print(test.shape)
```

```
(105, 5)
(45, 5)
```

Prepare data for modeling

```
train_X = train[['sepal_length', 'sepal_width', 'petal_length',
                 'petal_width']]
train_y = train.species

test_X = test[['sepal_length', 'sepal_width', 'petal_length',
                 'petal_width']]
test_y = test.species
```

To assess the performance of machine learning models, the dataset was split into training and testing sets. The features (sepal_length, sepal_width, petal_length, and petal_width) were extracted for both the training and testing sets.

MACHINE LEARNING MODELS

Using some of the commonly used algorithms, we will be training our model to check how accurate every algorithm is. We will be implementing these algorithms to compare:

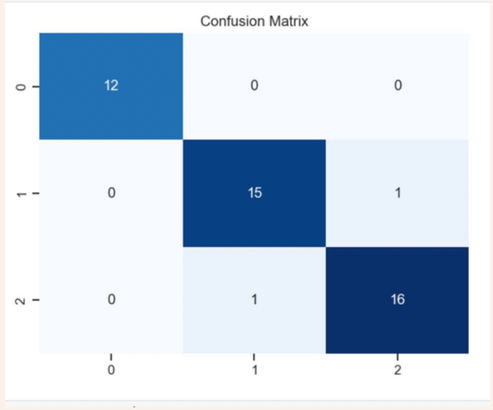
1] LOGISTIC REGRESSION

2] K – NEAREST NEIGHBOUR (KNN)

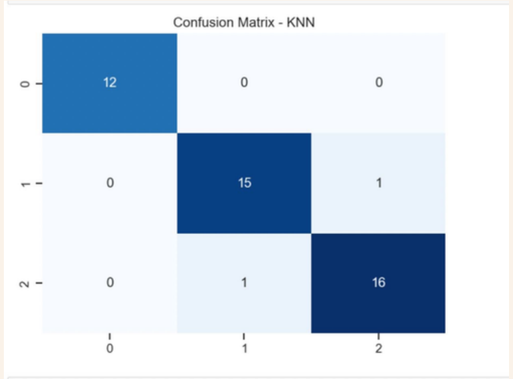
3] SUPPORT VECTOR MACHINE (SVM)

MACHINE LEARNING MODELS

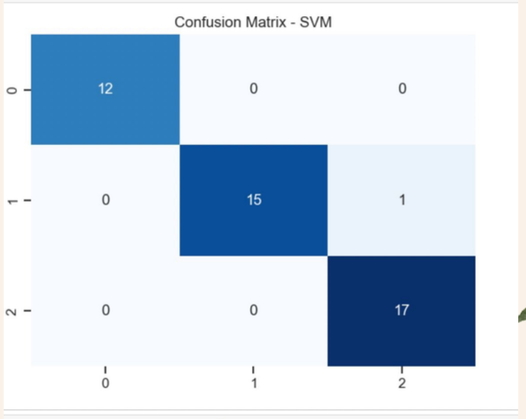
LOGISTIC REGRESSION



K-NEAREST NEIGHBOUR (KNN)



SUPPORT VECTOR MACHINE (SVM)



RESULTS

```
results = pd.DataFrame({
    'Model': ['Logistic Regression','Support Vector Machines','KNN'],
    'Score': [0.9777,0.9777,0.9555]})

result_df = results.sort_values(by='Score', ascending=False) $
result_df = result_df.set_index('Score')
result_df.head(9)
```

RESULTS

The resulting summary table provides a clear comparison of the models based on their accuracy scores:

Score	Model
0.9777	Logistic Regression
0.9777	Support Vector Machines
0.9555	KNN

ETHICAL CONSIDERATION OF DEVELOPING ML MODEL

- Accuracy: Higher accuracy is beneficial, but requires access to personal information.

- Bias: Depending on the applicable legislation, bias could be considered illegal discrimination. Algorithms developed to make judgments using historical data replicate the historical biases present in that data.

- Safety & Security: Safety is considered a big deal from a general perspective, but it's especially important when machine learning systems fail and cause actual harm

THNAKYOU FOR YOUR TIME

Interested?