```
### Name: Afroz Quraishi
### Roll No: 2018IMT-009
### Course: Machine Learning Lab
### Course Code: ITIT - 4107
### Deadline : 25 October 2021
### Link: https://github.com/afroz23/ITIT-4103-2021/tree/main/Assignment-5
```

Name: Afroz Quraishi

Roll No: 2018IMT-009

Course: Machine Learning Lab

Course Code: ITIT - 4107

Deadline : 25 October 2021

Link: https://github.com/afroz23/ITIT-4103-2021/tree/main/Assignment-5

Problem Statement

Given iris dataset (https://archive.ics.uci.edu/ml/datasets/iris) with 3 classes and 4 features such as sepals/petals, Length, width etc. for each flower in the dataset. There are 50 instances per class in the dataset. Use Bayes Classifier as your base classifier model. Use 60% samples for training and 40% samples for testing.

1. Perform feature selection on this dataset using forward search.
2. As you select features, until 2 features, plot your right and incorrect classification instances for all classes.
3. For all the set of features selected, plot the accuracies to show the best subset of selected features

Importing libraries

```
import numpy as np
import scipy as sp
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from matplotlib import pyplot as plt
```

Loading Data

```
data_url = 'https://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.da
df = pd.read_csv(data_url, header = None)
df.columns = ['sepal_length', 'sepal_width', 'petal_length', 'petal_width', 'speci
X = df.iloc[:, :4].values
y = df['species'].values
df.head()
```

	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	Iris-setosa
1	4.9	3.0	1.4	0.2	Iris-setosa
2	4.7	3.2	1.3	0.2	Iris-setosa
3	4.6	3.1	1.5	0.2	Iris-setosa
4	5.0	3.6	1.4	0.2	Iris-setosa

Encoding the species type to integers values

```
le = LabelEncoder()
le.fit(df.species)
y = le.transform(df.species)
print(le.classes_)
```

```
['Iris-setosa' 'Iris-versicolor' 'Iris-virginica']
```

```
x_train, x_test, y_train, y_test = train_test_split(X, y, test_size=0.4, random_sta
```

Implementing bayesian model

```
from sklearn.naive_bayes import GaussianNB
from sklearn.metrics import confusion_matrix, accuracy_score
import seaborn as sns
classifier = GaussianNB()
```

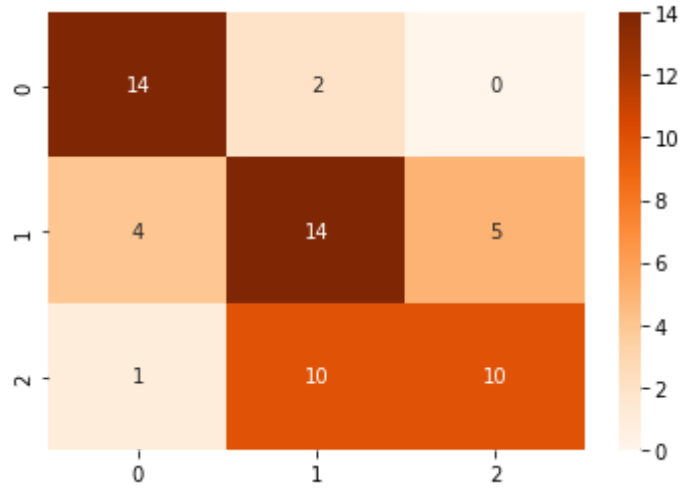
Here each feature was taken individually for the model evaluation

Training model using sepal_length feature and target variable is used

```
classifier.fit(x_train[:, 0].reshape(-1, 1), y_train)
y_pred = classifier.predict(x_test[:, 0].reshape(-1,1))
cm = confusion_matrix(y_test, y_pred)
```

```
print ("Accuracy when sepal length is chosen : ", accuracy_score(y_test, y_pred))
fig = plt.figure()
sns.heatmap(cm, annot=True, cmap="Oranges")
plt.plot()
print('=='*5)
```

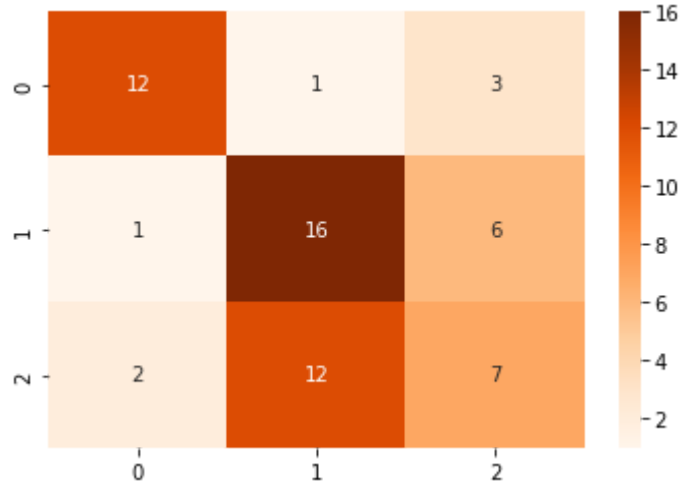
```
Accuracy when sepal length is chosen :  0.6333333333333333
==========
```



Training model using sepal_width feature and target variable is used

```
classifier.fit(x_train[:, 1].reshape(-1, 1), y_train)
y_pred = classifier.predict(x_test[:, 1].reshape(-1,1))
cm = confusion_matrix(y_test, y_pred)
print ("Accuracy when sepal width is chosen : ", accuracy_score(y_test, y_pred))
sns.heatmap(cm, annot=True, cmap="Oranges")
print('=='*5)
```

```
Accuracy when sepal width is chosen :  0.5833333333333334
==========
```

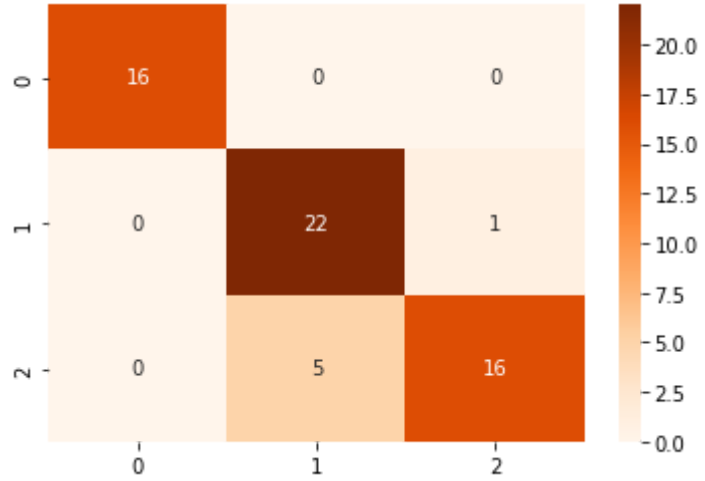


Training model using petal_length feature and target variable is used

```
classifier.fit(x_train[:, 2].reshape(-1, 1), y_train)
```

```
y_pred = classifier.predict(x_test[:, 2].reshape(-1,1))
cm = confusion_matrix(y_test, y_pred)
print ("Accuracy when petal length is chosen : ", accuracy_score(y_test, y_pred))
sns.heatmap(cm, annot=True, cmap="Oranges")
print('=='*5)
```

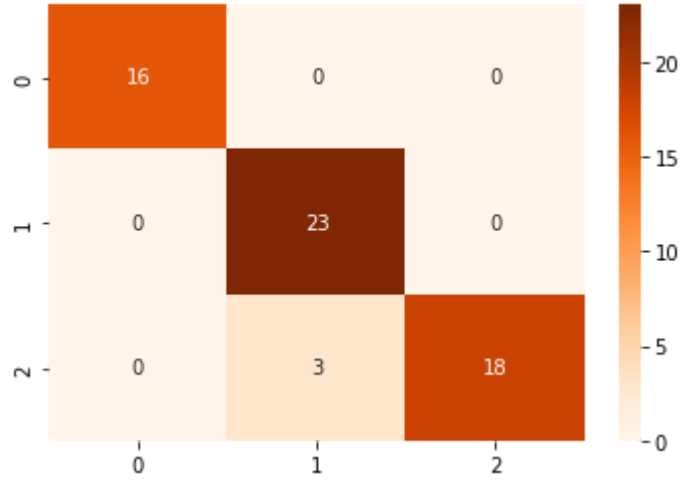
```
Accuracy when petal length is chosen :  0.9
==========
```



Training model using petal_width feature and target variable is used

```
classifier.fit(x_train[:, 3].reshape(-1, 1), y_train)
y_pred = classifier.predict(x_test[:, 3].reshape(-1,1))
cm = confusion_matrix(y_test, y_pred)
print ("Accuracy when petal width is chosen : ", accuracy_score(y_test, y_pred))
sns.heatmap(cm, annot=True, cmap="Oranges")
print('=='*5)
```

```
Accuracy when petal width is chosen :  0.95
==========
```



Multiple features was taken into consideration

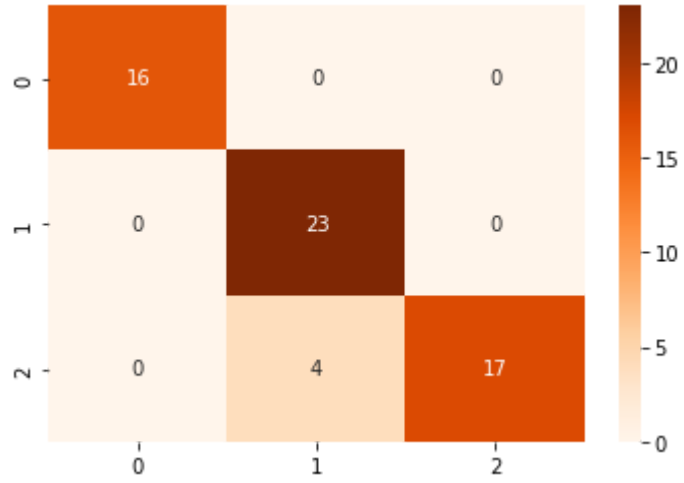
Here we can see that the petal length has the highest accuracy so we will take **petal width** as one of the feature for the model.

We will find other feature by trying different combination of the feature with petal width as one of the feature.

Training model using petal_width and sepal_length as input feature and target variable is used

```
x_input1 = np.array([[inp[0], inp[3]] for inp in x_train])
x_te = np.array([[inp[0], inp[3]] for inp in x_test])
classifier.fit(x_input1, y_train)
y_pred = classifier.predict(x_te)
cm = confusion_matrix(y_test, y_pred)
print ("Accuracy when petal_width and sepal_length is chosen : ", accuracy_score(y_
sns.heatmap(cm, annot=True, cmap="Oranges")
print('=='*5)
```

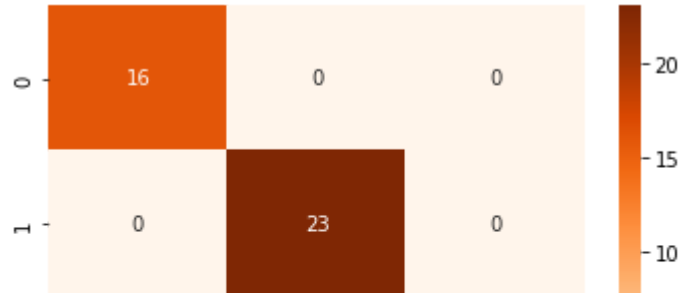
```
Accuracy when petal_width and sepal_length is chosen :  0.9333333333333333
==========
```



Training model using petal_width and sepal_width as input feature and target variable is used

```
x_input2 = np.array([[inp[1], inp[3]] for inp in x_train])
x_te = np.array([[inp[1], inp[3]] for inp in x_test])
classifier.fit(x_input2, y_train)
y_pred = classifier.predict(x_te)
cm = confusion_matrix(y_test, y_pred)
print ("Accuracy when petal_width and sepal_width is chosen : ", accuracy_score(y_
sns.heatmap(cm, annot=True, cmap="Oranges")
print('=='*5)
```

```
Accuracy when petal_width and sepal_width is chosen :  0.9333333333333333
==========
```



Training model using petal_width and petal_length as input feature and target variable is used


```
x_input3 = np.array([[inp[2], inp[3]] for inp in x_train])
x_te = np.array([[inp[2], inp[3]] for inp in x_test])
classifier.fit(x_input3, y_train)
y_pred = classifier.predict(x_te)
cm = confusion_matrix(y_test, y_pred)
print ("Accuracy when petal_width and petal_length is chosen : ", accuracy_score(y_
sns.heatmap(cm, annot=True, cmap="Oranges")
print('=='*5)
```

```
Accuracy when petal_width and petal_length is chosen :  0.9333333333333333
==========
```