

## Model Performance Test

Date	10 November 2022
Team ID	PNT2022TMID22986
Project Name	Project - Web Phishing Detection
Maximum Marks	10 Marks

### Model Performance Testing:

S.No	Parameter	Values
1.	Metrics	Random Forest Classifier Accuracy score-96.96
2.	Tune the Model	Hyperparameter Tuning - Validation Method -

### 1. METRICS

#### Classification Report:

```
In [25]: score = [log_reg, ran_for, des_class, kn_class, supp_vec]
Models = pd.DataFrame({
    'Classification Algorithms': ["Logistic Regression", "Random Forest Classifier", "Decision Tree Classifier", "K Neighbors", "Support Vector Machine"],
    'Accuracy': score})
Models.sort_values(by='Accuracy', ascending=False)
```

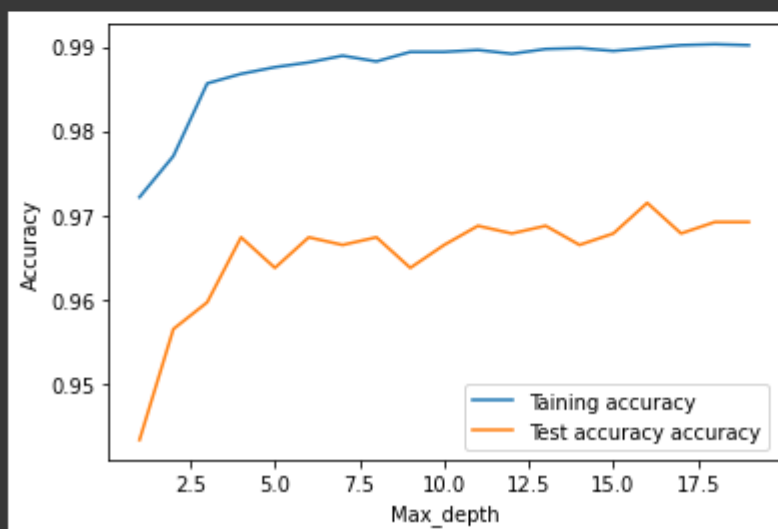
```
Out[25]:
```

	Classification Algorithms	Accuracy
1	Random Forest Classifier	0.969697
2	Decision Tree Classifier	0.963817
3	K Neighbors Classifier	0.943464
4	Support Vector Machine	0.940751
0	Logistic Regression	0.916780

## Performance:

```
[25] import matplotlib.pyplot as plt

training_accuracy=[]
test_accuracy=[]
depth = range(1,20)
for n in depth:
    rfc = RandomForestClassifier(n_estimators=n)
    rfc.fit(x_train,y_train)
    training_accuracy.append(rfc.score(x_train,y_train))
    test_accuracy.append(rfc.score(x_test,y_test))
plt.figure(figsize=None)
plt.plot(depth,training_accuracy,label="Taining accuracy")
plt.plot(depth,test_accuracy,label="Test accuracy accuracy")
plt.ylabel("Accuracy")
plt.xlabel("Max_depth")
plt.legend();
```



## 2. Tune the model

```
from sklearn.metrics import r2_score
from sklearn.metrics import mean_absolute_error
from sklearn.metrics import mean_squared_error

print ('R Squared =',r2_score(y_test, y_pred2))
print ('Mean Absolute Error =',mean_absolute_error(y_test, y_pred2))
print ('Mean Square Error =',mean_squared_error(y_test, y_pred2))
```

```
R Squared = 0.8761301676281433
Mean Absolute Error = 0.06151062867480778
Mean Square Error = 0.12302125734961555
```