

```
In [25]: ### Date - 18/10/2023
### Team ID -
### Project Title -
```

IMPORTING THE REQUIRED MODULES

```
In [26]: import pandas as pd
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
from sklearn.naive_bayes import MultinomialNB
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
```

Load the data

```
In [27]: data = pd.read_csv('spam.csv')
```

Split the data into training and testing sets

```
In [28]: X_train, X_test, y_train, y_test = train_test_split(data['v1'], data['v2'], test_size=0.2)
```

Text vectorization using TF-IDF (Term Frequency-Inverse Document Frequency)

```
In [29]: vectorizer = CountVectorizer()
X_train = vectorizer.fit_transform(X_train)
X_test = vectorizer.transform(X_test)
```

Make predictions

Evaluate the classifier

Naive Bayes Classifier

```
In [30]: naive_bayes = MultinomialNB()
naive_bayes.fit(X_train, y_train)
nb_predictions = naive_bayes.predict(X_test)
nb_accuracy = accuracy_score(y_test, nb_predictions)
```

Decision Tree

```
In [31]: decision_tree = DecisionTreeClassifier()
decision_tree.fit(X_train, y_train)
dt_predictions = decision_tree.predict(X_test)
dt_accuracy = accuracy_score(y_test, dt_predictions)
```

Random Forest

```
In [32]: random_forest = RandomForestClassifier()
random_forest.fit(X_train, y_train)
rf_predictions = random_forest.predict(X_test)
rf_accuracy = accuracy_score(y_test, rf_predictions)
```

Results

```
In [33]: print("Naive Bayes Accuracy:", nb_accuracy)
print("Decision Tree Accuracy:", dt_accuracy)
print("Random Forest Accuracy:", rf_accuracy)
```

```
Naive Bayes Accuracy: 0.008976660682226212
Decision Tree Accuracy: 0.008976660682226212
Random Forest Accuracy: 0.008976660682226212
```