

Experiment 6: Classification Text Analysis

Name: Anuj Chavan

UID: 2019120012

Class: BE EXTC

```
In [ ]: import pandas as pd
        from sklearn.feature_extraction.text import CountVectorizer
        from sklearn.model_selection import train_test_split
        from sklearn.naive_bayes import MultinomialNB # suitable for discrete feature and Gaussian
        from sklearn.metrics import accuracy_score , f1_score , precision_score , recall_score
```

```
In [ ]: df = pd.read_csv('spam.csv', encoding='ISO-8859-1')
        df = df.iloc[:, :2]
        df.rename(columns={'v1': 'label', 'v2': 'sms'}, inplace=True)
```

```
In [ ]: df['label'] = df.label.map({'ham': 0 , 'spam': 1})
        df.head()
```

```
Out[ ]:      label      sms
0      0      Go until jurong point, crazy.. Available only ...
1      0      Ok lar... Joking wif u oni...
2      1      Free entry in 2 a wkly comp to win FA Cup fina...
3      0      U dun say so early hor... U c already then say...
4      0      Nah I don't think he goes to usf, he lives aro...
```

```
In [ ]: X_train, X_test, y_train, y_test = train_test_split(df['sms'], df['label'], test_size=0.2)
```

```
In [ ]: count_vector = CountVectorizer()
        train_data = count_vector.fit_transform(X_train)
        testing_data = count_vector.transform(X_test)
```

```
In [ ]: count_vector = CountVectorizer()
        col_name = count_vector.fit(df['sms']).get_feature_names_out()
        data = count_vector.transform(list(df['sms'])).toarray()
        BOW = pd.DataFrame(data, columns=col_name)
        BOW.head()
```

```
Out[ ]:      00  000  000pes  008704050406  0089  0121  01223585236  01223585334  0125698789  02  ...  6_
0      0   0   0       0       0   0   0       0       0       0   0   ...   0
```

	00	000	000pes	008704050406	0089	0121	01223585236	01223585334	0125698789	02	...	6_
1	0	0	0	0	0	0	0	0	0	0	...	0
2	0	0	0	0	0	0	0	0	0	0	...	0
3	0	0	0	0	0	0	0	0	0	0	...	0
4	0	0	0	0	0	0	0	0	0	0	...	0

5 rows × 8672 columns



```
In [ ]: naive_bayes = MultinomialNB()
naive_bayes.fit(train_data, y_train)
```

```
Out[ ]: ▾ MultinomialNB
MultinomialNB()
```

```
In [ ]: predaction = naive_bayes.predict(testing_data)
```

```
In [ ]: print('Accuracy score: {}'.format(accuracy_score(y_test, predaction)))
print('precision_score: {}'.format(precision_score(y_test, predaction)))
print('recall_score: {}'.format(recall_score(y_test, predaction)))
print('f1_score: {}'.format(f1_score(y_test, predaction)))
```

```
Accuracy score: 0.9829596412556054
precision_score: 0.95
recall_score: 0.9172413793103448
f1_score: 0.9333333333333332
```

CART Classifier

```
In [ ]: from sklearn.tree import DecisionTreeClassifier
```

```
In [ ]: cart = DecisionTreeClassifier()
cart.fit(train_data, y_train)
```

```
Out[ ]: ▾ DecisionTreeClassifier
DecisionTreeClassifier()
```

```
In [ ]: cart_predaction = cart.predict(testing_data)
```

```
In [ ]: print('Accuracy score: {}'.format(accuracy_score(y_test, cart_predaction)))
print('precision_score: {}'.format(precision_score(y_test, cart_predaction)))
print('recall_score: {}'.format(recall_score(y_test, cart_predaction)))
print('f1_score: {}'.format(f1_score(y_test, cart_predaction)))
```

```
Accuracy score: 0.9748878923766816
precision_score: 0.9090909090909091
recall_score: 0.896551724137931
f1_score: 0.9027777777777778
```

Random Forest

```
In [ ]: from sklearn.ensemble import RandomForestClassifier
```

```
In [ ]: rf = RandomForestClassifier()
        rf.fit(train_data, y_train)
```

```
Out[ ]: ▾ RandomForestClassifier
        RandomForestClassifier()
```

```
In [ ]: rf.predict(testing_data)
```

```
Out[ ]: array([0, 0, 0, ..., 0, 0, 0], dtype=int64)
```

```
In [ ]: print('Accuracy score: {}'.format(accuracy_score(y_test, cart_predection)))
        print('precision_score: {}'.format(precision_score(y_test, cart_predection)))
        print('recall_score: {}'.format(recall_score(y_test, cart_predection)))
        print('f1_score: {}'.format(f1_score(y_test, cart_predection)))
```

```
Accuracy score: 0.9748878923766816
precision_score: 0.9090909090909091
recall_score: 0.896551724137931
f1_score: 0.9027777777777778
```

Naive Bayes

```
In [ ]: from sklearn.naive_bayes import MultinomialNB
```

```
In [ ]: naive_bayes = MultinomialNB()
        naive_bayes.fit(train_data, y_train)
```

```
Out[ ]: ▾ MultinomialNB
        MultinomialNB()
```

```
In [ ]: predection = naive_bayes.predict(testing_data)
```

```
In [ ]: print('Accuracy score: {}'.format(accuracy_score(y_test, cart_predection)))
        print('precision_score: {}'.format(precision_score(y_test, cart_predection)))
        print('recall_score: {}'.format(recall_score(y_test, cart_predection)))
        print('f1_score: {}'.format(f1_score(y_test, cart_predection)))
```

```
Accuracy score: 0.9748878923766816
```

```
precision_score: 0.9090909090909091  
recall_score: 0.896551724137931  
f1_score: 0.9027777777777778
```