Project No #7

Spam Email Detection

	Label	EmailText
0	ham	Go until jurong point, crazy.. Available only ...
1	ham	Ok lar... Joking wif u oni...
2	spam	Free entry in 2 a wkly comp to win FA Cup fina...
3	ham	U dun say so early hor... U c already then say...

Importing needful libraries

```
import pandas as pd
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.model_selection import train_test_split
from sklearn.svm import SVC
from sklearn.model_selection import GridSearchCV
from sklearn.model_selection import KFold
from sklearn.metrics import accuracy_score
from imblearn.over_sampling import SMOTE
```

Importing data

```
data = pd.read_csv("D:\\Workshops\\Python for Data Science Comprehensiv
data.head()
```

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

class imbalance

```
data["Label"].value_counts()
```

```
ham     4825
spam     747
Name: Label, dtype: int64
```

Separating independent & dependent data

```
x = data["EmailText"]
y = data["Label"]
```

Creating a matrix with frequencies of email text

```
cvec = CountVectorizer()
cx = cvec.fit_transform(x)
```

Using SMOTE for balancing the response data

```
smt=SMOTE()
x_sm,y_sm=smt.fit_resample(cx,y)
```

```
y_sm.value_counts()
```

```
ham     4825
spam    4825
Name: Label, dtype: int64
```

Splitting training & testing data

```
x_train,x_test,y_train,y_test=train_test_split(x_sm,y_sm,test_size=0.2,random_state=0)
```

Grid Search for identifying best hyperparameters

```
params = {'kernel': ['rbf','linear']}
cv=KFold(n_splits=5)
```

```
model = GridSearchCV(SVC(), params,cv=cv)
```

```
model.fit(x_train,y_train)
print(model.best_params_)
```

```
{'kernel': 'linear'}
```

Defining the model with best hyperparameters

```
bmodel=SVC(kernel="linear")
```

Training the model with training data

```
bmodel.fit(x_train,y_train)
```

```
SVC(kernel='linear')
```

Predictions and accuracy of the model

```
: 1 y_pred=bmodel.predict(x_test)
```

```
: 1 y_pred
```

```
: array(['ham', 'spam', 'spam', ..., 'spam', 'spam', 'spam'], dtype=object)
```

```
: 1 y_test
```

```
: 1070     ham
  4488     ham
  8763    spam
  7372    spam
  7633    spam
          ...
  212      ham
  4546     ham
  6411    spam
  7916    spam
  6712    spam
  Name: Label, Length: 1930, dtype: object
```

```
1 accuracy_score(y_test,y_pred)
```

```
0.9569948186528497
```

```
1 print(classification_report(y_test,y_pred))
```

```
              precision    recall  f1-score   support

         ham       0.98      0.93      0.95       936
        spam       0.94      0.98      0.96       994

    accuracy                           0.96      1930
   macro avg       0.96      0.96      0.96      1930
weighted avg       0.96      0.96      0.96      1930
```

Checking the model with new sample emails

```
1 emails=["Hey, you have won a car !!!!. Conrgratzz","Dear applicant, Your CV has been recieved. Best regards"]
```

```
1 yp=bmodel.predict(cvec.transform(emails))
```

```
1 yp
```

```
array(['spam', 'ham'], dtype=object)
```