# Karanjot Singh

In [16]:

```
import pandas as pd
import nltk
from nltk.corpus import stopwords
import string
```

In [17]:

```
email = pd.read_csv('emails.csv')
email.head(5)
```

Out[17]:

| | text | spam |
|---|---|---|
| 0 | Subject: naturally irresistible your corporate... | 1 |
| 1 | Subject: the stock trading gunslinger fanny i... | 1 |
| 2 | Subject: unbelievable new homes made easy im ... | 1 |
| 3 | Subject: 4 color printing special request add... | 1 |
| 4 | Subject: do not have money , get software cds ... | 1 |

In [18]:

```
email.drop_duplicates(inplace = True)
```

In [19]:

```
def process_text(text):
    nopunc = [char for char in text if char not in string.punctuation]
    nopunc = ' '.join(nopunc)

    clean_words = [word for word in nopunc.split() if word.lower() not in stopwords.wor
ds('english')]

    return clean_words
```

In [20]:

```
from sklearn.feature_extraction.text import CountVectorizer
message_bow = CountVectorizer(analyzer=process_text).fit_transform(email['text'])
message_bow.shape
```

Out[20]:

```
(5695, 41)
```

In [26]:

```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test = train_test_split(message_bow,email['spam'],test_size =
0.20,random_state = 0)
```

In [27]:

```
from sklearn.naive_bayes import MultinomialNB
classifier = MultinomialNB()
classifier.fit(x_train,y_train)
```

Out[27]:

```
MultinomialNB(alpha=1.0, class_prior=None, fit_prior=True)
```

In [28]:

```
print('Predicted Values: ',classifier.predict(x_test))
print('Actual value: ',y_test.values)
```

```
Predicted Values:  [1 0 0 ... 0 0 0]
Actual value:  [1 0 0 ... 0 0 0]
```

In [30]:

```
from sklearn.metrics import classification_report,confusion_matrix,accuracy_score
pred = classifier.predict(x_test)
print(classification_report(y_test,pred))

print('Confusion Matrix :\n',confusion_matrix(y_test,pred))
print()
print('Accuracy: ',accuracy_score(y_test,pred))
```

```
              precision    recall  f1-score   support

           0       0.90      0.76      0.82       870
           1       0.48      0.72      0.57       269

    accuracy                           0.75      1139
   macro avg       0.69      0.74      0.70      1139
weighted avg       0.80      0.75      0.76      1139

Confusion Matrix :
 [[660 210]
 [ 76 193]]

Accuracy:  0.7489025460930641
```