Parshvanath Charitable Trust's
A. P. SHAH INSTITUTE OF TECHNOLOGY
(Approved by AICTE New Delhi & Govt. of Maharashtra, Affiliated to University of Mumbai)
(Religious Jain Minority)

Department of Information Technology

Academic Year: 2018-19 **Name of Student:Pratiksha Patil**
Semester: VIII **Student ID: 16204011**

Project Title : **Sentiment Analysis Framework for Social Media**

Group No : **14**

Group Members : **Riddhi Prajapati**

Aafreen Shaikh

Pratiksha Patil

Guide : **Prof. Sunil N. Sushir**

Co Guide : **-**

Naive Bayes

In [56]:
```
from sklearn.metrics import accuracy_score
model = MultinomialNB()
model.fit(X_train_vectorized, y_train)
pred = model.predict(vect.transform(X_val))

print('MODEL fitting:', f1_score(y_val, pred))
print("accuracy of prediction",accuracy_score(y_val, pred))
```

```
MODEL fitting: 0.572528883184
accuracy of prediction 0.958328119134
```

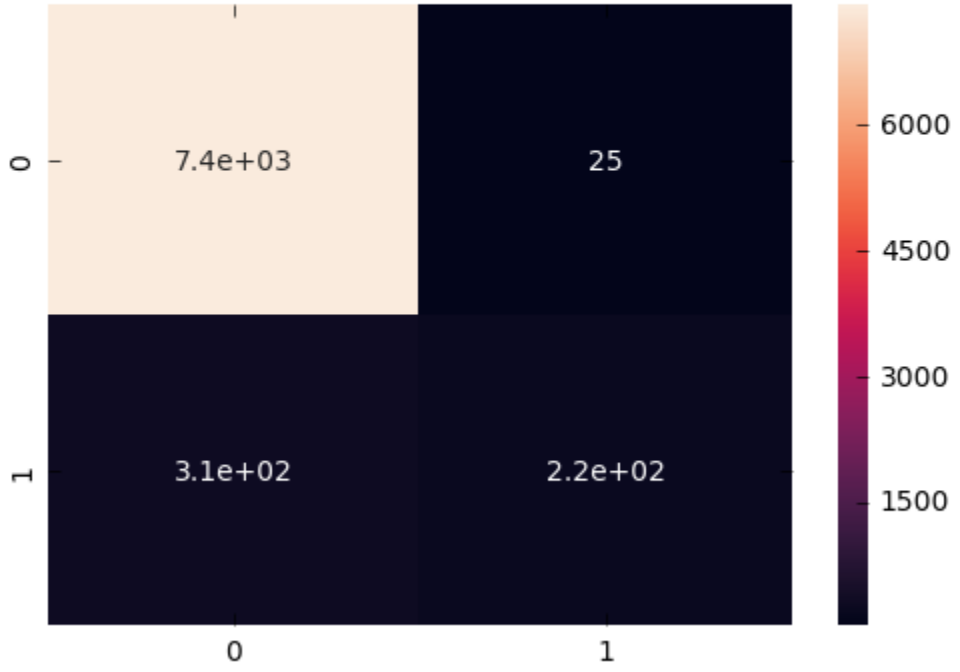
In [57]:
```
from sklearn.metrics import confusion_matrix
cm=confusion_matrix(y_val, pred)
cm
```

Out[57]:
```
array([[7435,   25],
       [ 308,  223]])
```

Here we have used three algorithms – Naïve Bayes, Logistic Regression and Random Forest. All the three algorithms are compared amongst themselves and we have found which algorithm gives highest accuracy. Firstly, we have used Naïve Bayes in which we have imported the required libraries for Naïve Bayes and that model is fitted to x and y train data set. Here it gives the accuracy of 0.95%. Then we have generated confusion matrix.

```
Out[57]: array([[7435,   25],
                [ 308,  223]])
```

```
In [58]: sns.heatmap(cm,annot=True)
         plt.show()
```



This figure shows confusion matrix of Naïve Bayes model. Here it shows how many number of tweets are classified accurately and how many are misclassified.

Logistic Regression

In [59]:
```
model = LogisticRegression()
model.fit(X_train_vectorized, y_train)
pred = model.predict(vect.transform(X_val))
```

In [60]:
```
print('MODEL fitting:', f1_score(y_val, pred))
print("accuracy of prediction",accuracy_score(y_val, pred))
```

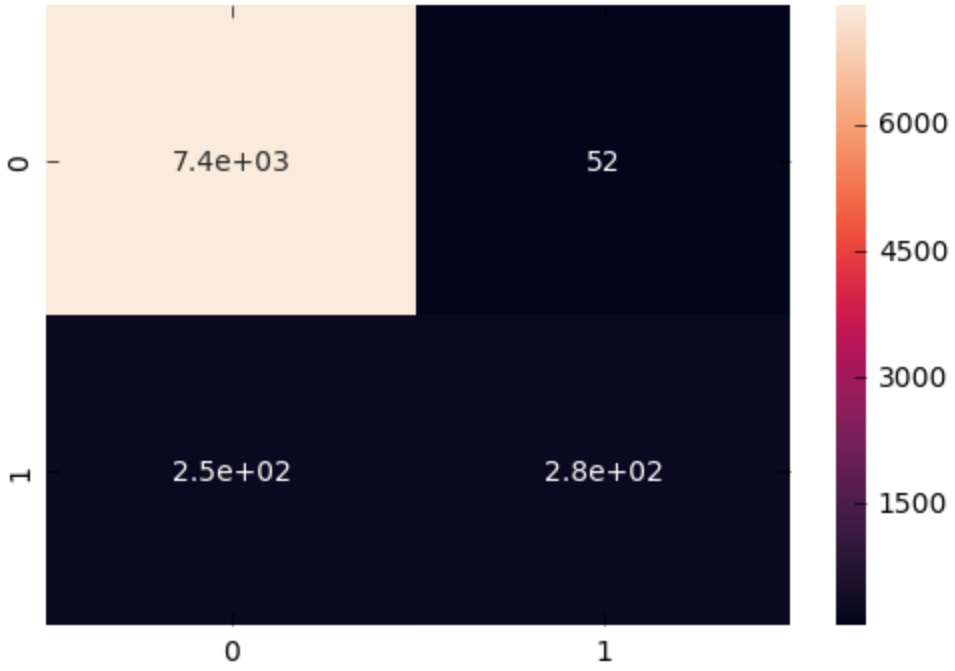
```
MODEL fitting: 0.647331786543
accuracy of prediction 0.961957201852
```

In [61]:
```
from sklearn.metrics import confusion_matrix
cm=confusion_matrix(y_val, pred)
cm
```

Out[61]:
```
array([[7408,   52],
       [ 252,  279]])
```

The second algorithm is Logistic Regression onto which the training data set are applied. The accuracy of this model is highest amongst all three of them which is 0.96%.

```
In [62]: sns.heatmap(cm,annot=True)
         plt.show()
```



Also the confusion matrix is plotted for Logistic Regression showing the number of classified and miss-classified tweets.

[Paste your screenshots here]