Assignment Machine Learning
Parth Saboo 2019A4PS0457P

Dataset file name: **category(1).csv**

Data Collection: For obtaining a variety of data, we had collected data from more than one email ID's including the personal and Bits mail id. We have used Subject, Body and sender as three of the mail for classification into labels. We had chosen six labels to distribute the collected data. Steps followed for data collection are as follows

Step1 - Created six labels in gmail labels.

Step2 - Classified mails under the selected labels manually

Step3 - Created python code to extract subject, body, sender's address into a CSV file from the mails.

Step4 - We used Gmail API to access the emails and extracted it by using a python script.

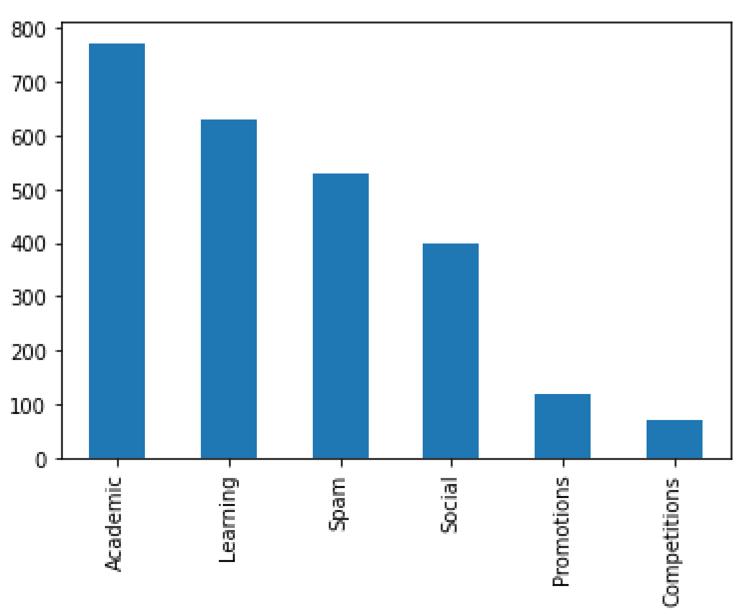
Data Cleaning:

We cleaned data at several layers

1) Removed html tags
2) Removed emojis
3) Removed characters excluding A-Z and 1-9
4) Lemmatized text of the mails
5) Removed stop words and converted everything to lower case.

Exploratory Data Analysis

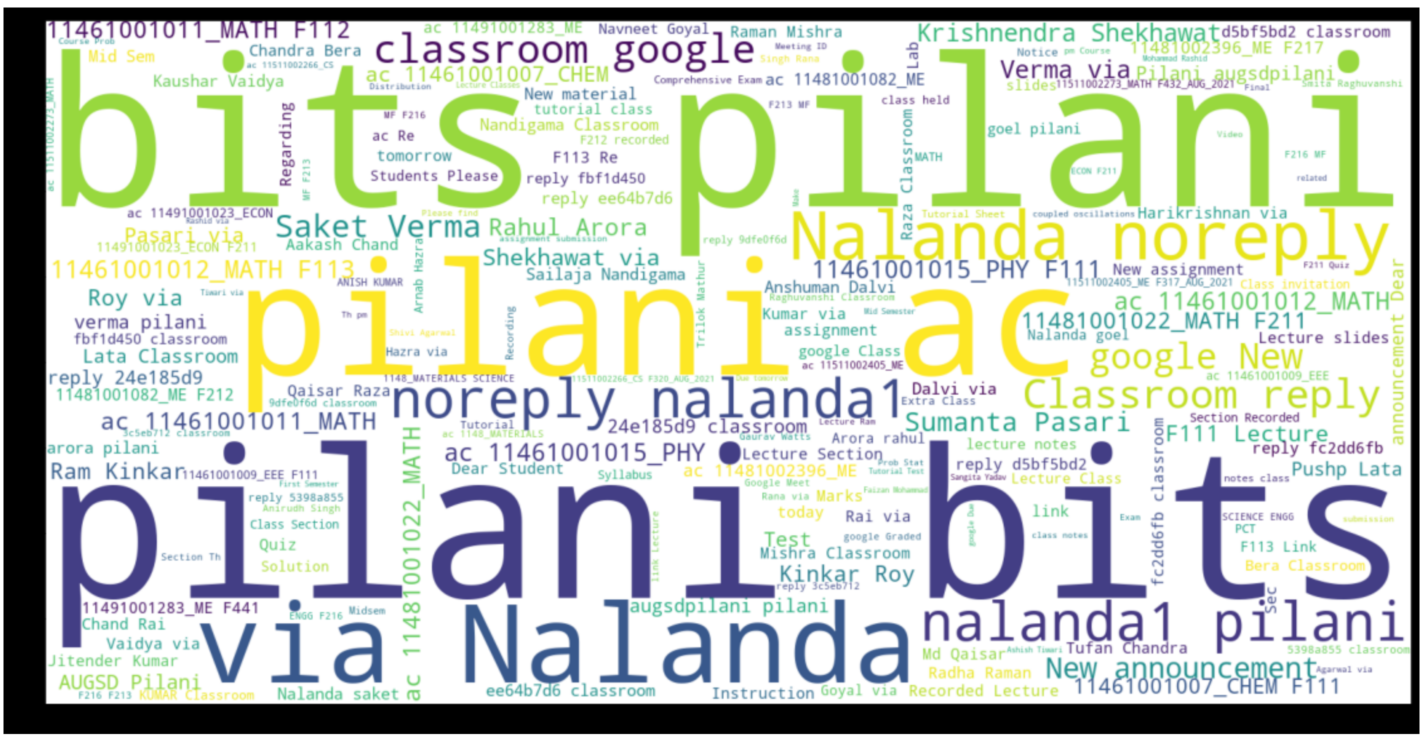
- **Value counts of different categories**



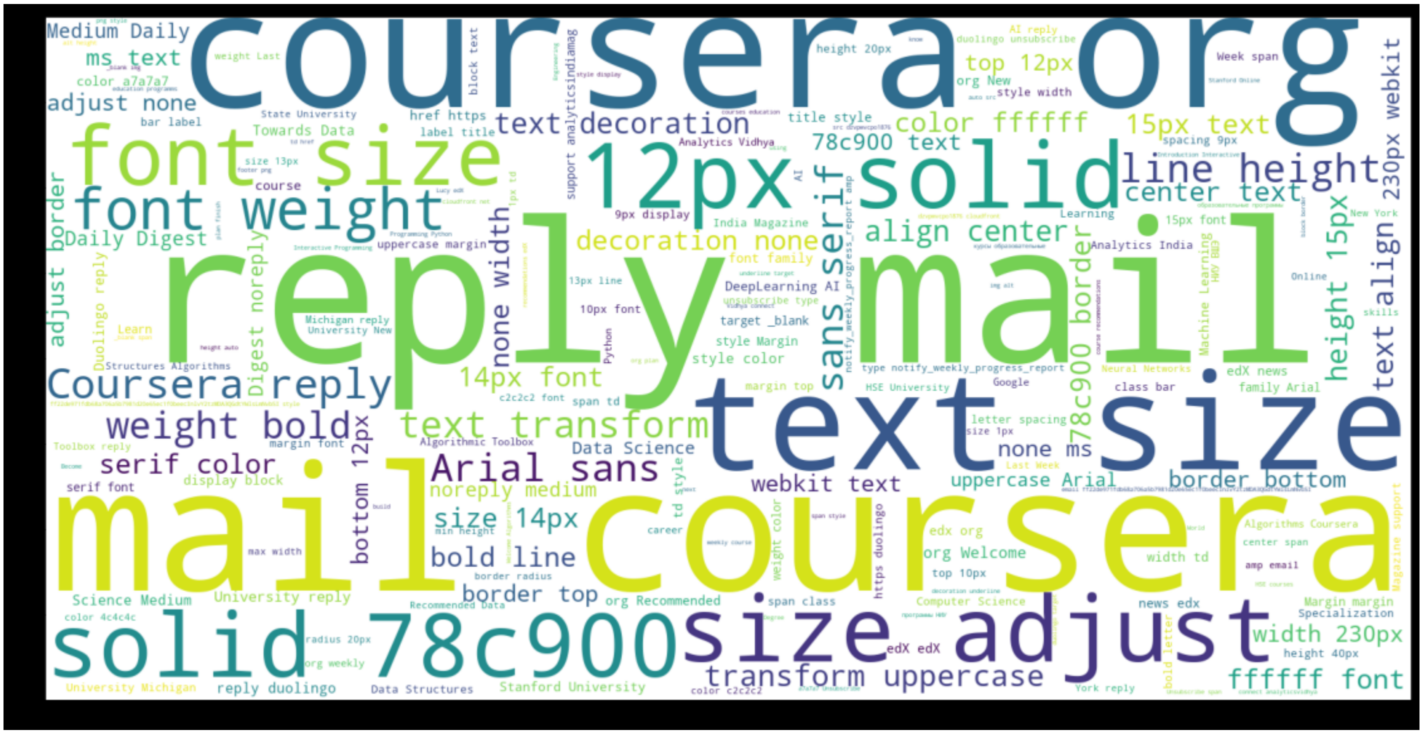
-We used Word clouds to depict key senders, words and body in different label emails

1) **Spam**

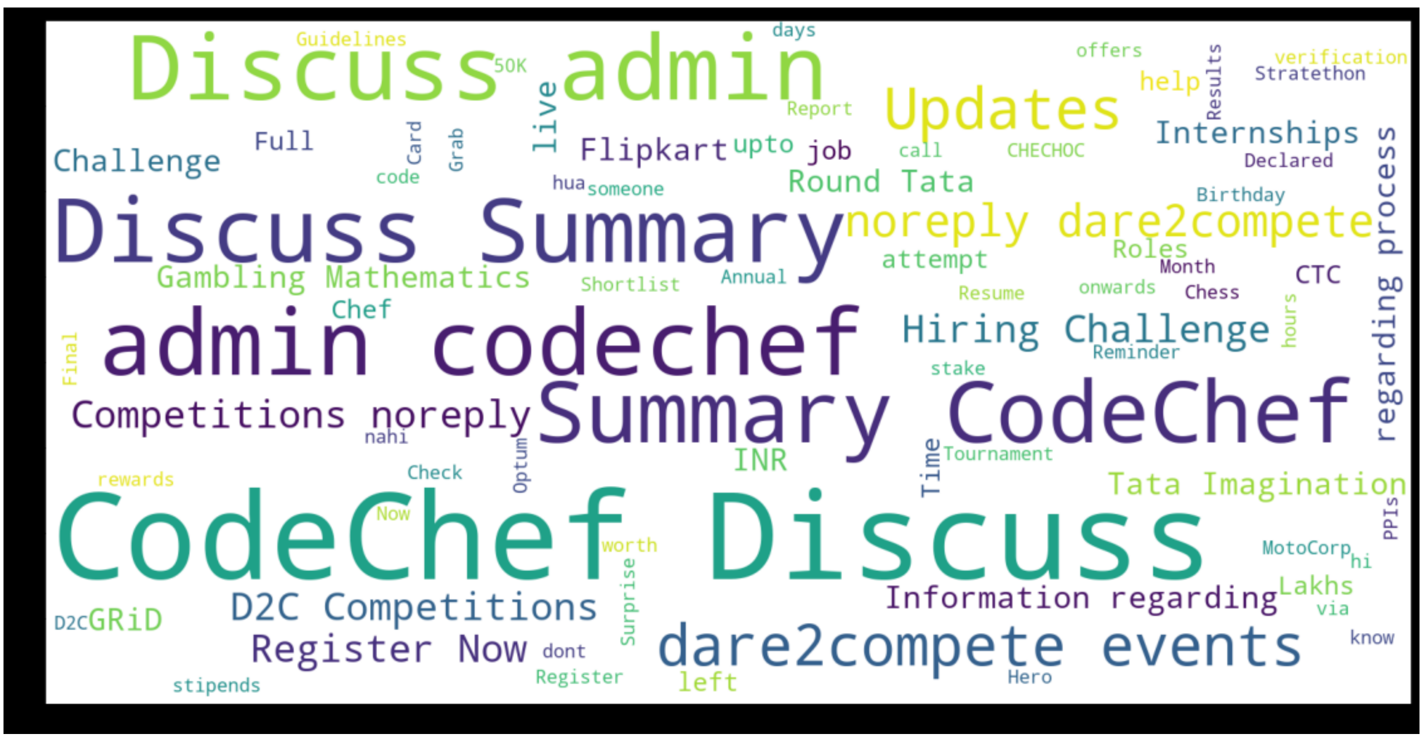
2) **Academic**



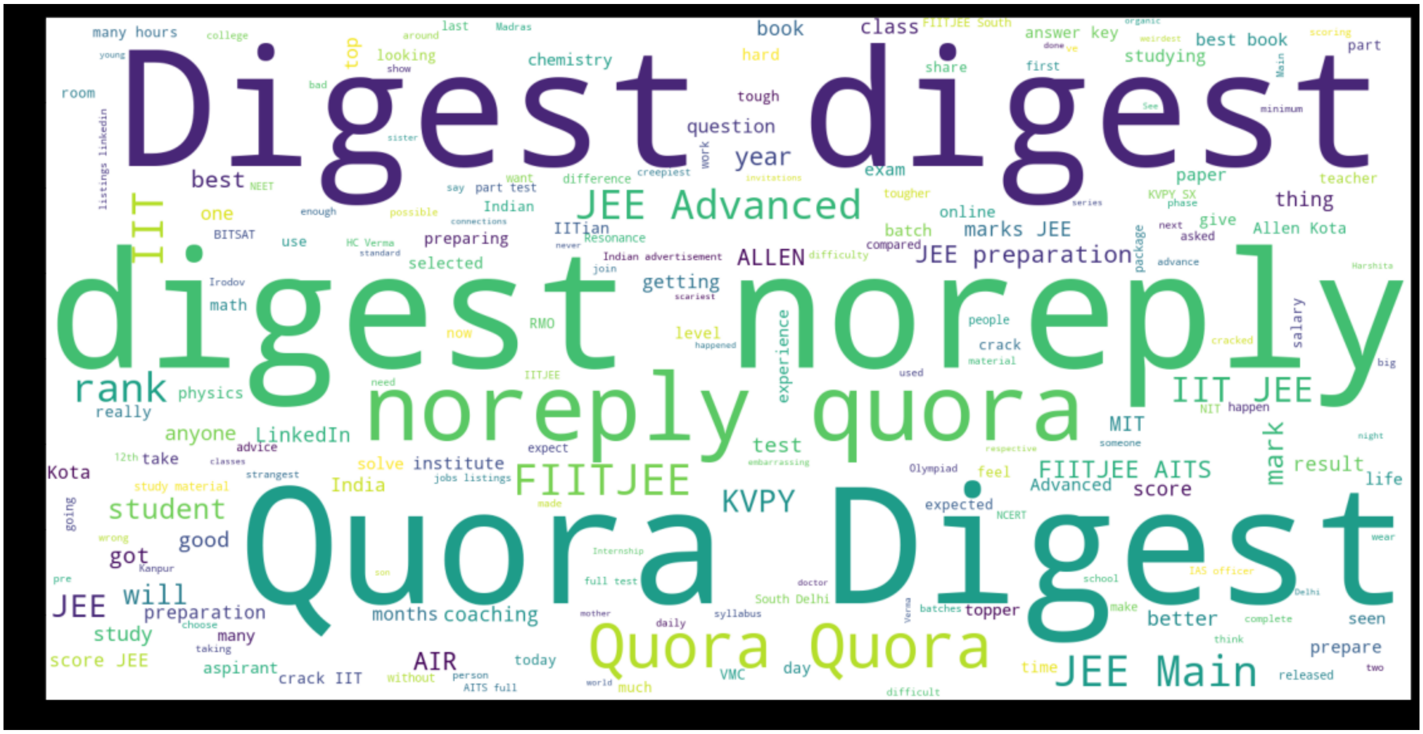
3) **Learning**



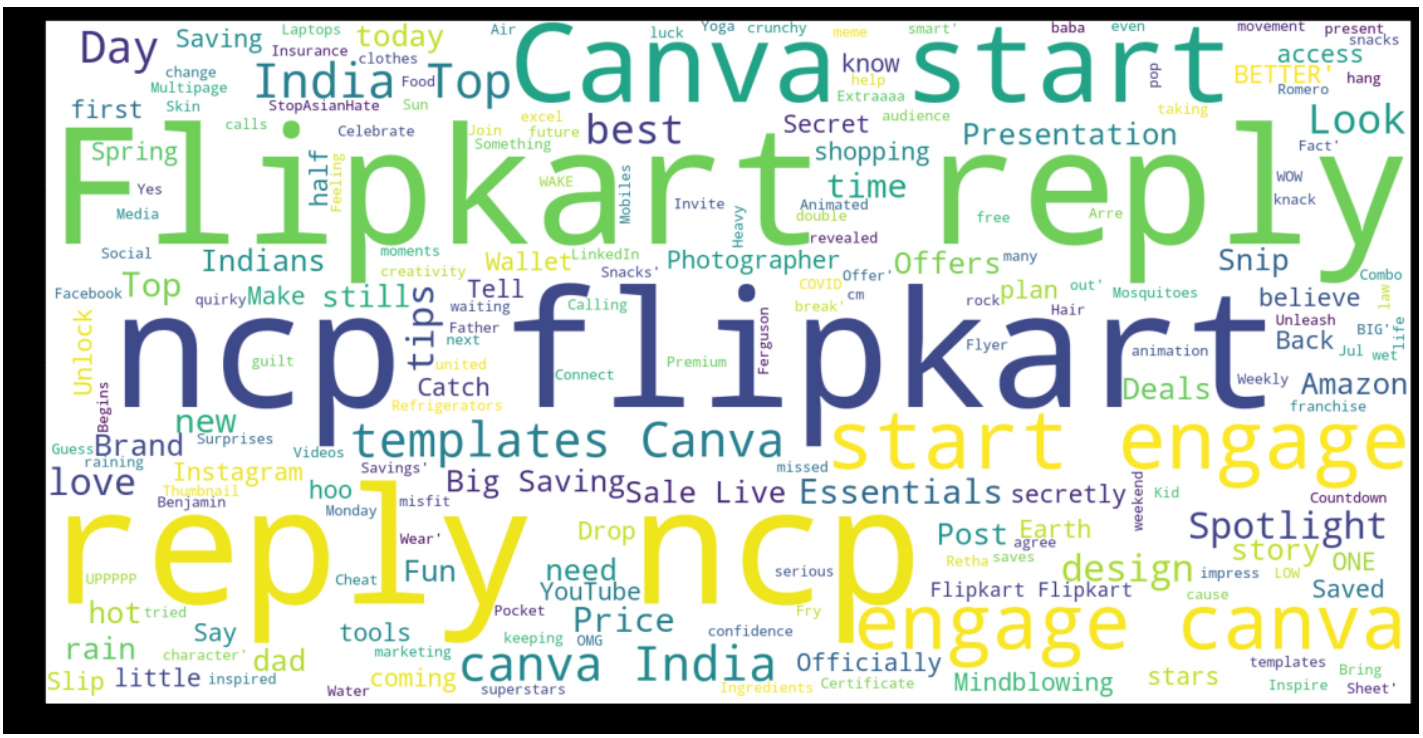
4) Competitions



5) Social



6) Promotions



Feature Extraction

We concatenated cleaned Senders address, Subject and Body columns of the CSV file to a single column. We splitted data into training and testing data in the ratio of 0.75 and 0.25 respectively.

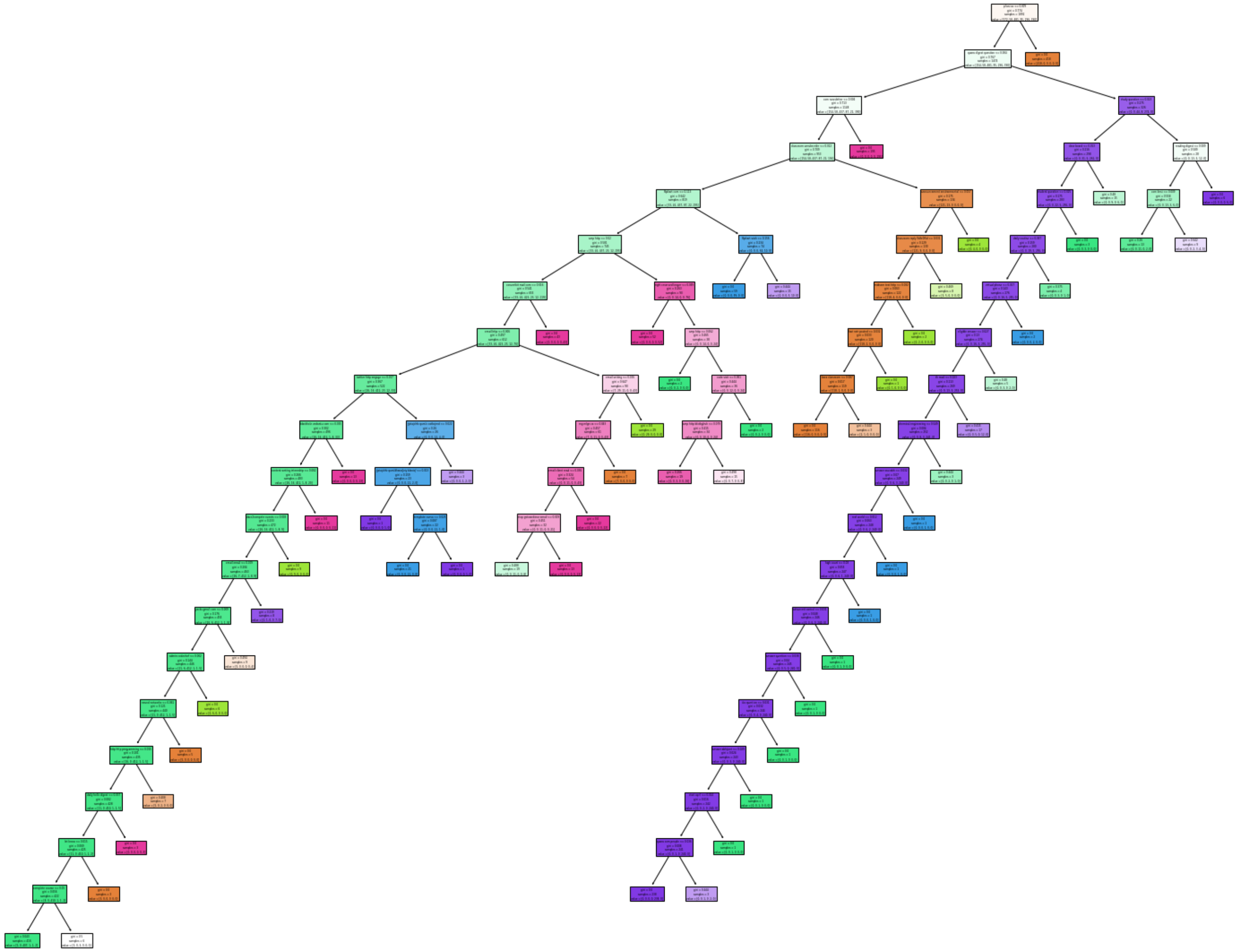
We used Term Frequency Document-Inverse Frequency (Tf Dif) to extract features from the mails using n-gram range 2,3. We didn't used CountVectorizer to extract features as it doesn't account for relative importance of words. As there were a very high number of features, we selected the top 10% of the features using **sklearn's select percentile** function. It gave us a feature in the form of a numpy array with each element ranging from 0-1.

Models Used to Classify

1)Decision trees :- After a few iterations, min_samples_split =20 , max_depth = 50 was chosen to save computation time and also preserve accuracy.

Obtained decision tree :

```
import matplotlib.pyplot as plt
fig = plt.figure(figsize=(25,20))
_ = tree.plot_tree(clf, feature_names=feature_names,
                   filled=True)
```



a) Classification Report

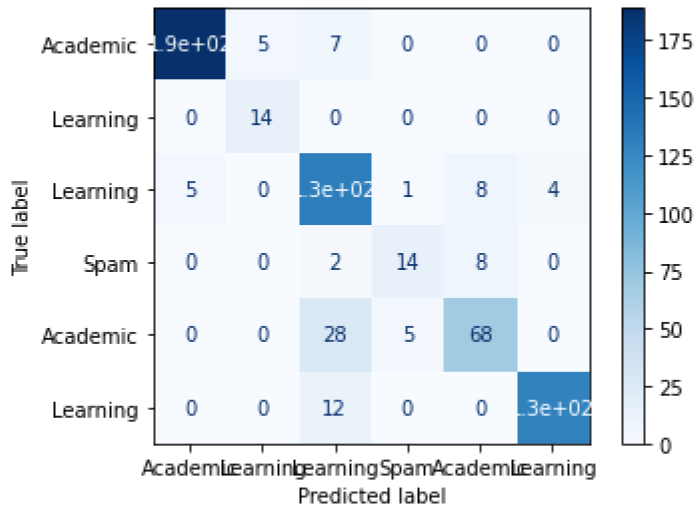
```
print("Other Metrics:")
from sklearn.metrics import classification_report
print(classification_report(y_val, pred))
```

```
Other Metrics:
              precision    recall  f1-score   support

    Academic       0.97      0.94      0.96       201
Competitions       0.74      1.00      0.85        14
    Learning       0.72      0.88      0.79       150
  Promotions       0.78      0.58      0.67        24
      Social       0.82      0.66      0.73       101
        Spam       0.97      0.92      0.95       141

    accuracy                           0.87       631
   macro avg       0.83      0.83      0.82       631
weighted avg       0.87      0.87      0.87       631
```

b) Confusion Matrix



2)Gaussian Naive bayes

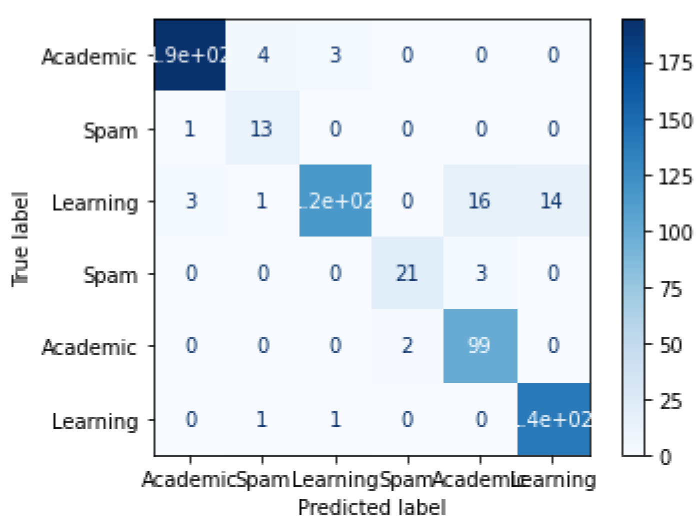
a) Classification Report

```
Other Metrics:
              precision    recall  f1-score   support

    Academic       0.98      0.97      0.97       201
Competitions       0.68      0.93      0.79        14
    Learning       0.97      0.77      0.86       150
  Promotions       0.91      0.88      0.89        24
      Social       0.84      0.98      0.90       101
        Spam       0.91      0.99      0.95       141

    accuracy                           0.92       631
   macro avg       0.88      0.92      0.89       631
weighted avg       0.93      0.92      0.92       631
```

b) Confusion Matrix



3)SVC

a) Classification Report

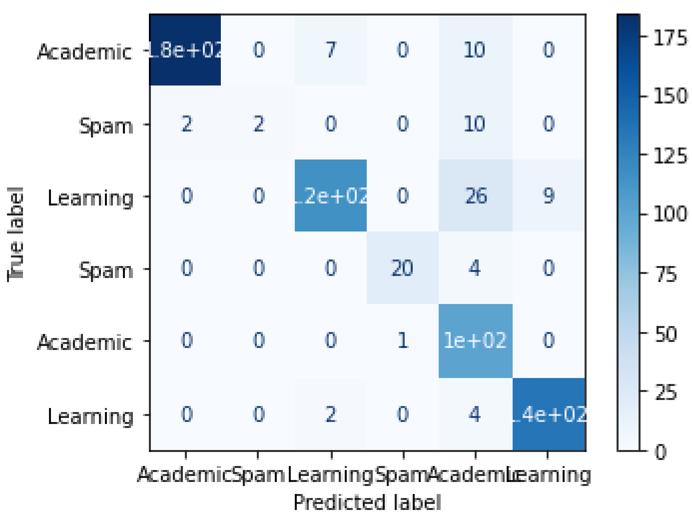
```
Other Metrics:
               precision    recall  f1-score   support

     Academic       0.99      0.92      0.95       201
 Competitions       1.00      0.14      0.25        14
     Learning       0.93      0.77      0.84       150
   Promotions       0.95      0.83      0.89        24
       Social       0.65      0.99      0.78       101
         Spam       0.94      0.96      0.95       141

     accuracy                           0.88       631
    macro avg       0.91      0.77      0.78       631
 weighted avg       0.91      0.88      0.88       631

0.8811410459587956
```

b) Confusion Matrix



4)Random Forest

a) Classification Report

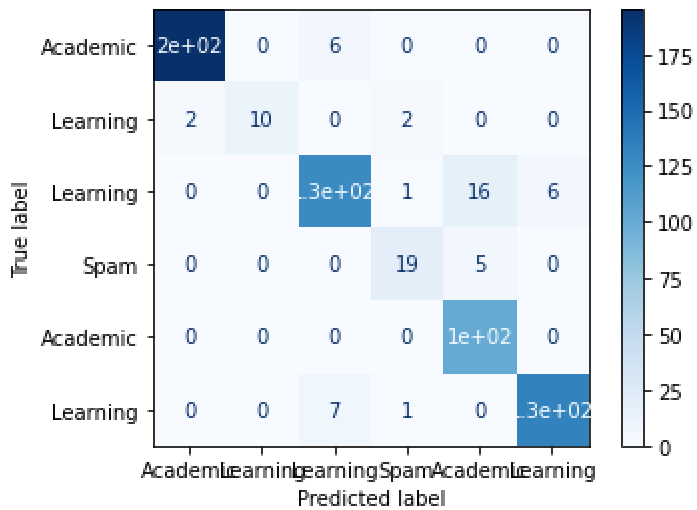
```
Other Metrics:
               precision    recall  f1-score   support

    Academic       0.99      0.97      0.98       201
Competitions       1.00      0.71      0.83        14
    Learning       0.91      0.85      0.88       150
  Promotions       0.83      0.79      0.81        24
      Social       0.83      1.00      0.91       101
        Spam       0.96      0.94      0.95       141

    accuracy                           0.93       631
   macro avg       0.92      0.88      0.89       631
weighted avg       0.93      0.93      0.93       631

0.9270998415213946
```

b) Confusion Matrix



Accuracy model wise

Models	Accuracy (in %)
SVC	88.11
Gaussian NB	92.23
Decision Tree	86.53
Random Forest	92.71