1. Document Categorization

The task is categorize an article to an appropriate topic.

Dataset

The dataset was collected from https://scdnlab.com/corpus/ . The dataset contains articles of 12 categories in 12 different folders. The number of articles each folder contains is given below.

1. Accident : 6350 articles
2. Art : 2669 articles
3. Crime : 8840 articles
4. Economics : 5351 articles
5. Education : 12389 articles
6. Entertainment : 10139 articles
7. Environment : 6852 articles
8. International : 5922 articles
9. Opinion : 8116 articles
10. Politics : 20479 articles
11. Science_tec : 2906 articles
12. Sports : 12086 articles

Feature Extraction

We used TF-IDF values of each unique words in a document as a feature vector for a single document.

Experiment Procedure

The task is mainly classification. For each categories we took 500 articles in our train dataset and 50 articles in our test dataset. We fed the TF-IDF vectors of train dataset in a classifier and checked how the classifer worked on test dataset. We used Different Classifiers for this task. The results of our experiments is given below.

Experiment Results

Naïve Bayes:

We achieved accuracy rate of 82.0 using Naïve Bayes Classifier. We used ***sklearn***'s MultinomialNB as classifier. The confusion matrix is given below.

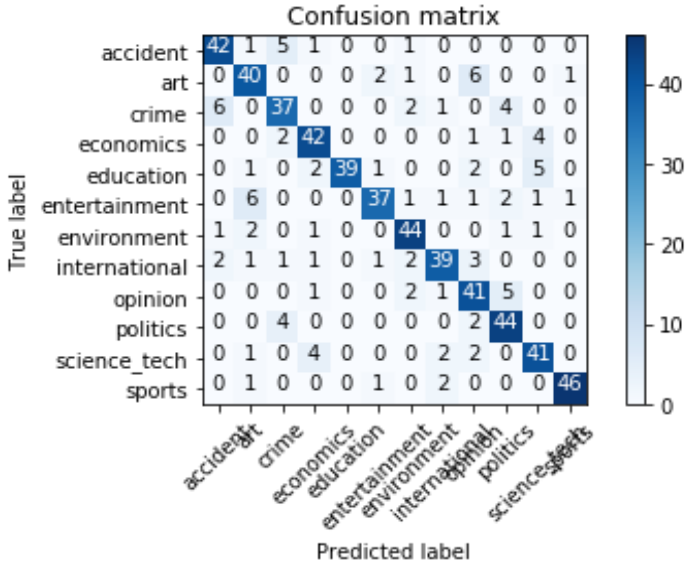


Figure: Confusion matrix of Naïve Bayes Classifier.

Multinomial Naive Bayes has a parameter ***alpha,*** which acts as a additive smoothing parameter. We tuned this value and got different accuracy rates.

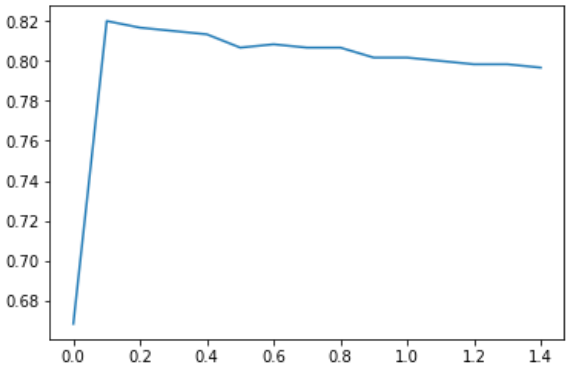


*Figure: Accuracy rate vs **alpha** value*

Passive Aggressive Classifier:

Using passive aggressive classifier, we got an accuracy rate of 77.3%. Below is the confusion matrix.

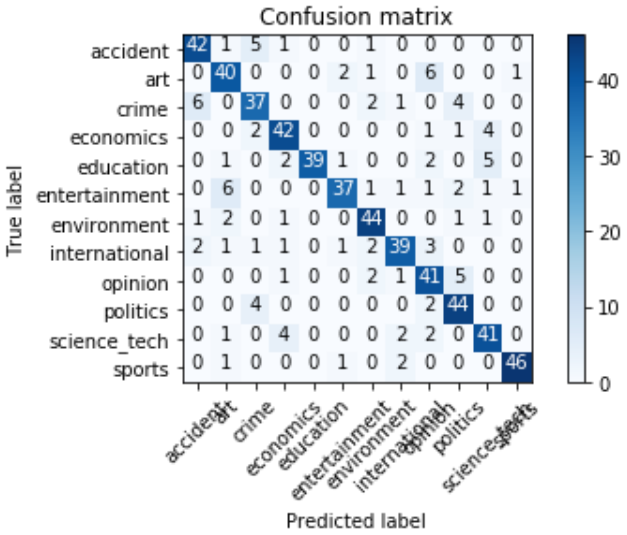


Fig: Confusion matrix PAC

Random Forest Classifier:

We achieved an accuracy rate of 76.3% using Random Forest Classifier setting number of estimator trees to 58.

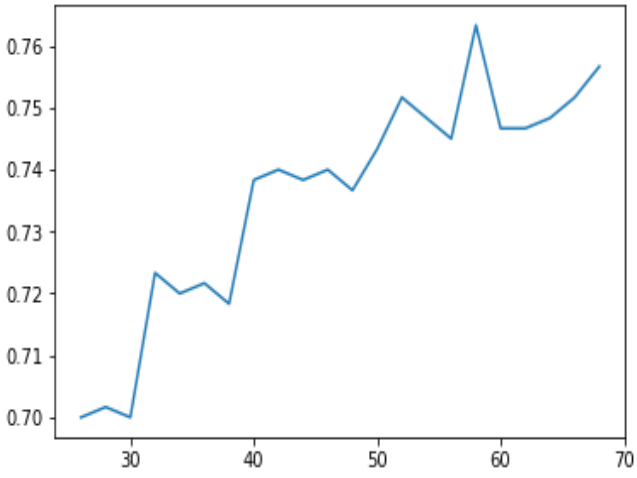


Fig: Accuracy rate vs number of tree

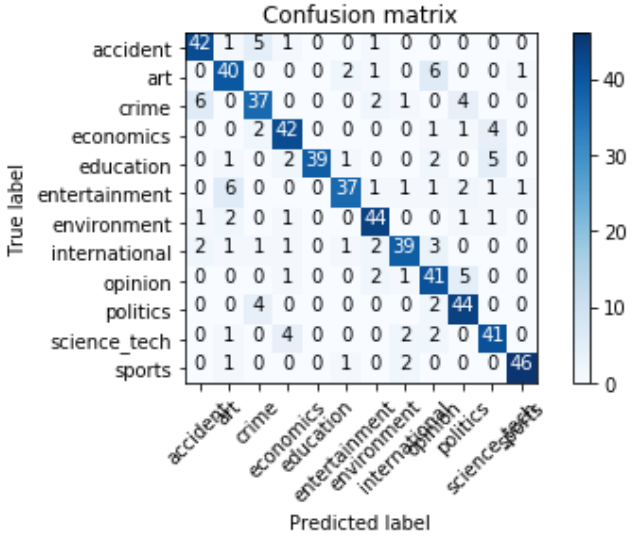


Fig: Confusion matrix Random Forest

XGBoost:

We achieved accuracy rate of 75.2% using XGBoost classifier. The confusion matrix is given below.

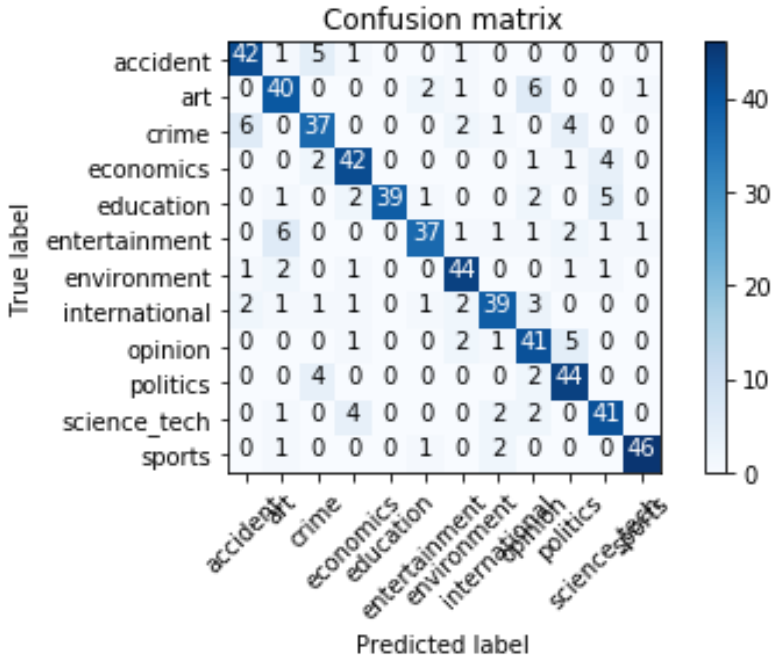


Fig: Confusion matrix XGBoost

2. Sentiment Analysis

The task is to detect the sentiments an article expresses.

Dataset

The dataset contains 27731 rows, each containing a sentence/article and a **sentiment** separated by semicolon (;). The dataset contains data of 18 different sentiments. The sentiments are given below:

```
['Love', 'Like ', 'Consciousness ', 'Protestant ', 'Smiley ', 'Angry ', 'Blush', 'Skip ', 'Rocking ', 'Fail ', 'Shocking ', 'WOW', 'Bad ', 'HaHa', 'Sad ', 'Skeptical ', 'Evil ', 'Provocative ']
```

Dataset Analysis

In our analysis the dataset is very bad to work with for the task at hand.

1. The dataset claimed that document , sentiment pairs are separated by semicolon (;), but actually it contained semicolon (;) inside documents too. An example is given below.

   জীবনের নিরাপত্তা ও জীবনের মুল্য ২টিই বেশী; বাংলাদেশে গরুর মুল্য বোধ হয় মানুষ থেকে বেশী হওয়ার কথা।;HaHa(হা হা)

   This made reading the dataset a bit hard for us.

2. We found the dataset inconsistant and meaning-less. The same sentence is tagged with different sentiments at different times.

থ্যাংকস সুমন ভাই।;Like (ভাল)
থ্যাংকস সুমন ভাই।;Smiley (স্মাইলি)
থ্যাংকস সুমন ভাই।;Angry (রাগান্বিত)
থ্যাংকস সুমন ভাই।;Consciousness (চেতনাবাদ)
থ্যাংকস সুমন ভাই।;Skip (বোঝতে পারছি না)
থ্যাংকস সুমন ভাই।;HaHa(হা হা)
থ্যাংকস সুমন ভাই।;WOW(কি দারুন)
থ্যাংকস সুমন ভাই।;Love(ভালবাসা)

থ্যাংকস সুমন ভাই।;Blush(গোলাপী আভা)

.

.

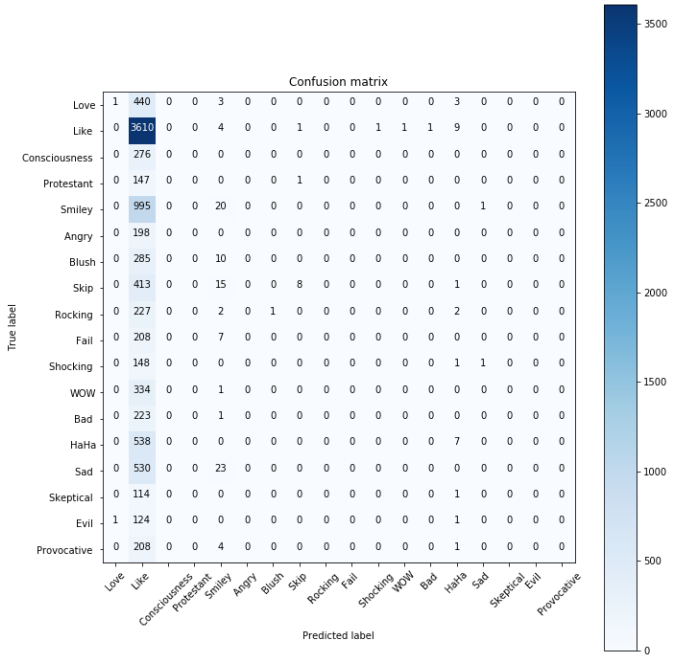
.

What sentiment exactly the sentence ‘থ্যাংকস সুমন ভাই।’ represents???

3. The same document-sentiment tuple occurs multiple times. The tuple থ্যাংকস সুমন ভাই।;Like (ভাল)
occurs 17 times in this dataset !!!

Experiment

Despite the shortcomings of the dataset we actually did some experiments on it. Our feature was TFIDF value of the dataset. We used Naïve Bayes as our classifier. As expected the accuracy was very low, giving us an accuracy rate of only 39.8%. The confusion matrix is given below.



Then we thought that, may be the task is not a classification one. Then we built 18 separate classifiers, one for each of the sentiments to check whether an article expresses the sentiment or not. The results were still not very convincing.

Bangla Handwritten Digits

Training Images : 23824
Testing Images : 2653

Feature Extraction

1*785 array for a single 28px * 28px image where the 785th element is the label.

Classifiers

1. Convolutional Neural Network (CNN) :

- 2D convolutional layer with 32 filters
- Pooling layer of 5*5 (Activation function - *ReLU)*
- Another 2D convolution layer
- Another pooling layer of 5*5
- Hidden layer of 512 neurons
- Dropout of 0.25 for avoiding over-fitting
- Output layer: Number of classes (Activation function - *Softmax)*
- Loss: Categorical Cross Entropy (Optimizer – Adam)

2. Random Forest :

- *RandomForestClassifier of sklearn*

3. Naive Bayes :

- *MulinomialNB of sklearn*

Results

1. **CNN – 98.87%**

2. **Random Forest :**

- 5 Trees – 91.18%
- 10 Trees – 93.29%
- 15 Trees – 94.61%
- 25 Trees – 94.91%

- 25 Trees – 95.51%
- 30 Trees – 95.63%
- **35 Trees – 96.00%**
- 40 Trees – 95.63%

3. Naive Bayes – 86.58%