PROJECT – 20 NEWSGROUP DATASET

Cleaning, Prediction and Performance analysis of text dataset.

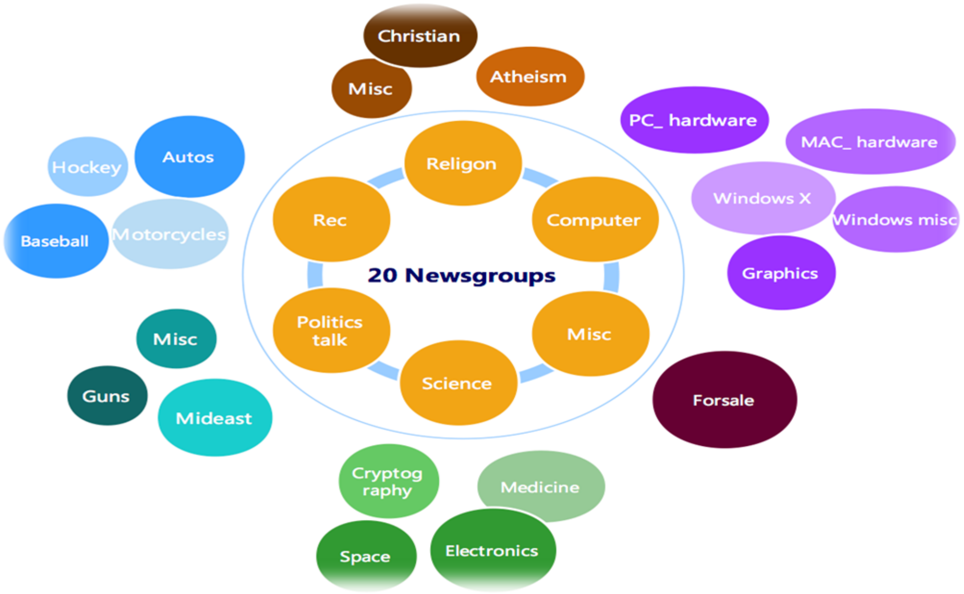
Submitted By: Silky (B.Tech IT)

OBJECTIVES

- Introduction of Dataset
- Preparing Dataset
- Data Cleaning
- Vectorization: tf_idf
- Text Classifier
- Prediction on Random Test Data
- Algorithm Performance
- Model Evaluation

INTRODUCTION

- The 20 newsgroups dataset comprises around 18000 newsgroups posts on 20 topics split in two subsets: one for training (or development) and the other one for testing (or for performance evaluation). The split between the train and test set is based upon a messages posted before and after a specific date.
- The 20 newsgroups collection has become a popular data set for experiments in text applications of machine learning techniques, such as text classification and text clustering.
- The data is organized into 20 different newsgroups, each corresponding to a different topic. Some of the newsgroups are very closely related to each other.



PREPARING DATASET

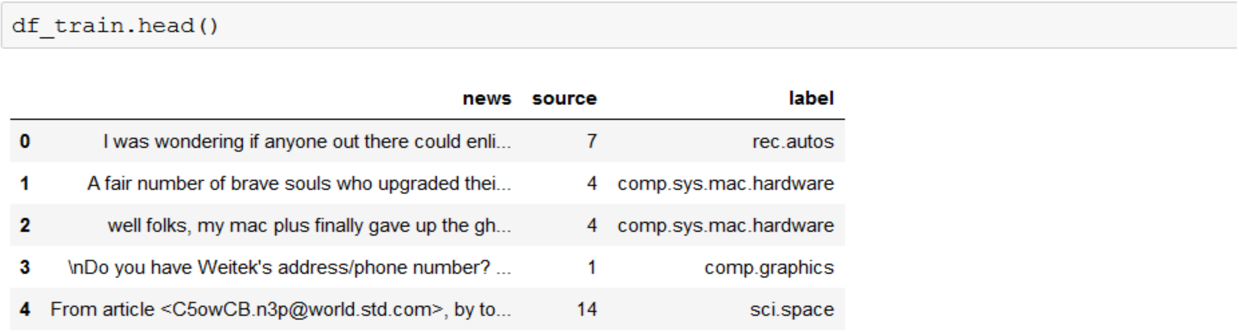
- Fetching and Displaying Dataset

The sklearn.datasets.fetch_20newsgroups function is a data fetching / caching functions that downloads the data archive from the original 20 newsgroups website .

- The whole dataset would be divided in 2 parts:

1. Train Dataset: We can fetch train dataset by using command :

```
train = fetch_20newsgroups(subset='train’)
```

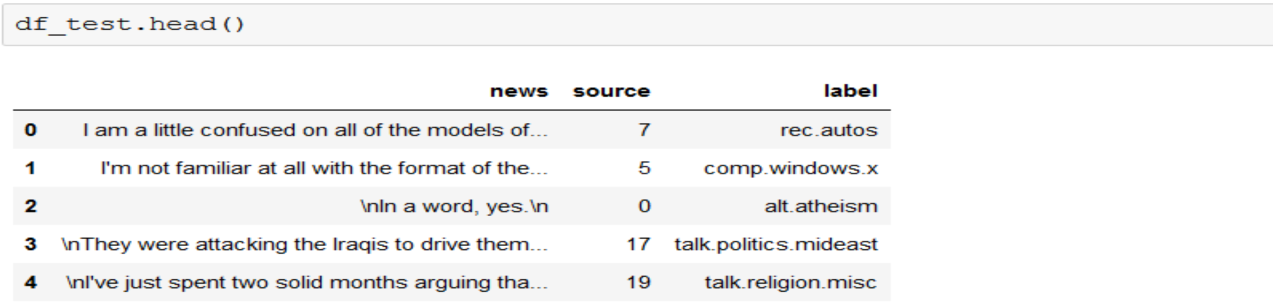


```
df_train.head()
```

	news	source	label
0	I was wondering if anyone out there could enli...	7	rec.autos
1	A fair number of brave souls who upgraded thei...	4	comp.sys.mac.hardware
2	well folks, my mac plus finally gave up the gh...	4	comp.sys.mac.hardware
3	\nDo you have Weitek's address/phone number? ...	1	comp.graphics
4	From article <C5owCB.n3p@world.std.com>, by to...	14	sci.space

2. Test Dataset We can fetch test dataset by using command:

```
test = fetch_20newsgroups(subset='test’)
```



```
df_test.head()
```

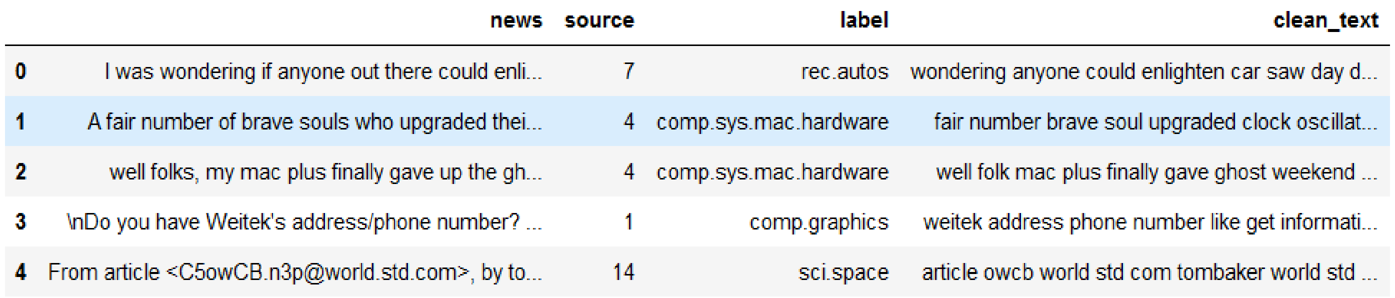
	news	source	label
0	I am a little confused on all of the models of...	7	rec.autos
1	I'm not familiar at all with the format of the...	5	comp.windows.x
2	\nIn a word, yes.\n	0	alt.atheism
3	\nThey were attacking the Iraqis to drive them...	17	talk.politics.mideast
4	\nI've just spent two solid months arguing tha...	19	talk.religion.misc

DATA CLEANING

- In the test dataset, the data contains text with newlines, punctuation, misspellings, and other items common in text documents. To build a model, we will clean up the text by removing some of these issues.
- Data cleaning will include:

  1. Removing stopwords
  2. Filter out short words
  3. Lowercase and removing everything except words
  4. Removing of special characters
  5. Applying lemmatization to the text

- Using nltk library the clean data in dataframe will look like:



	news	source	label	clean_text
0	I was wondering if anyone out there could enli...	7	rec.autos	wondering anyone could enlighten car saw day d...
1	A fair number of brave souls who upgraded thei...	4	comp.sys.mac.hardware	fair number brave soul upgraded clock oscillat...
2	well folks, my mac plus finally gave up the gh...	4	comp.sys.mac.hardware	well folk mac plus finally gave ghost weekend ...
3	\nDo you have Weitek's address/phone number? ...	1	comp.graphics	weitek address phone number like get informati...
4	From article <C5owCB.n3p@world.std.com>, by to...	14	sci.space	article owcb world std com tombaker world std ...

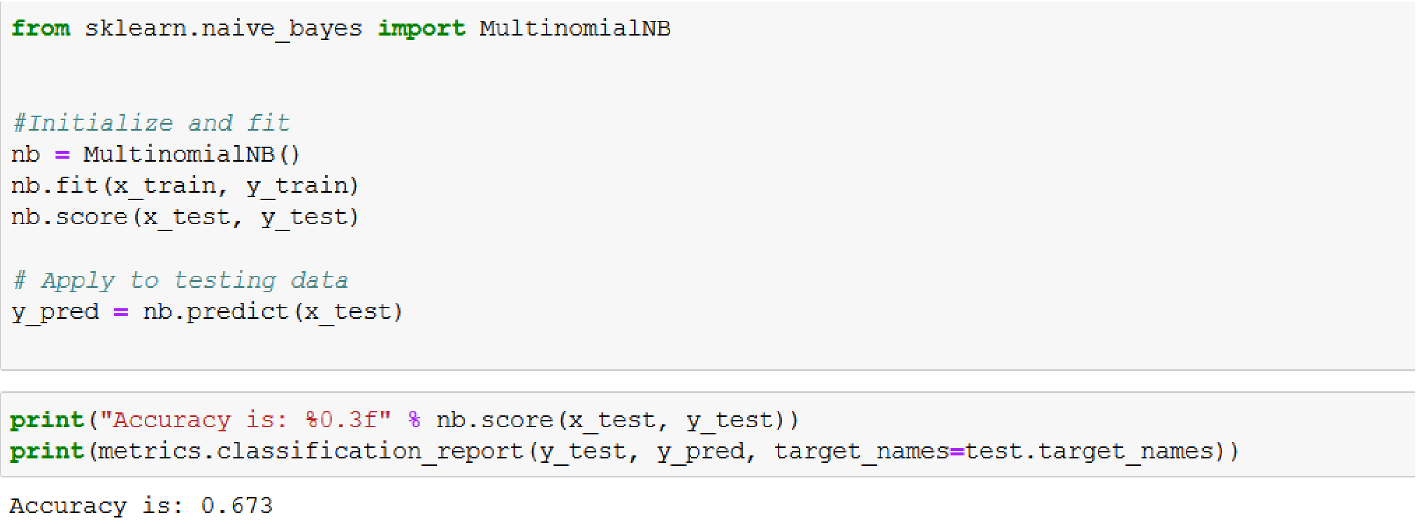
CONVERTING TEXT TO VECTORS

- In order to feed predictive or clustering models with the text data, one first need to turn the text into vectors of numerical values suitable for statistical analysis.
- This can be achieved with the utilities of the sklearn.feature_extraction.text that extract TF-IDF vectors of unigram tokens from a subset of 20news.
- We can import from sklearn by using command:

```
from sklearn.feature_extraction.text import
CountVectorizer,TfidfVectorizer

vectorizer = TfidfVectorizer(min_df=5,
strip_accents='ascii', analyzer='word', lowercase=True)
```

TEXT CLASSIFIER

- The multinomial Naive Bayes classifier which is suitable for discrete classification.
- Scikit-learn has a great Class called *Pipeline*, which allows us to a create pipeline for a classifier, i.e. we can just add the functions that we want to use on our input data.
- Here, we are using a *TfidfVectorizer()* as vectorizer and *Multinomial* as classifier:
- It is easy for a classifier to overfit on particular things that appear in the 20 Newsgroups data, such as newsgroup headers.



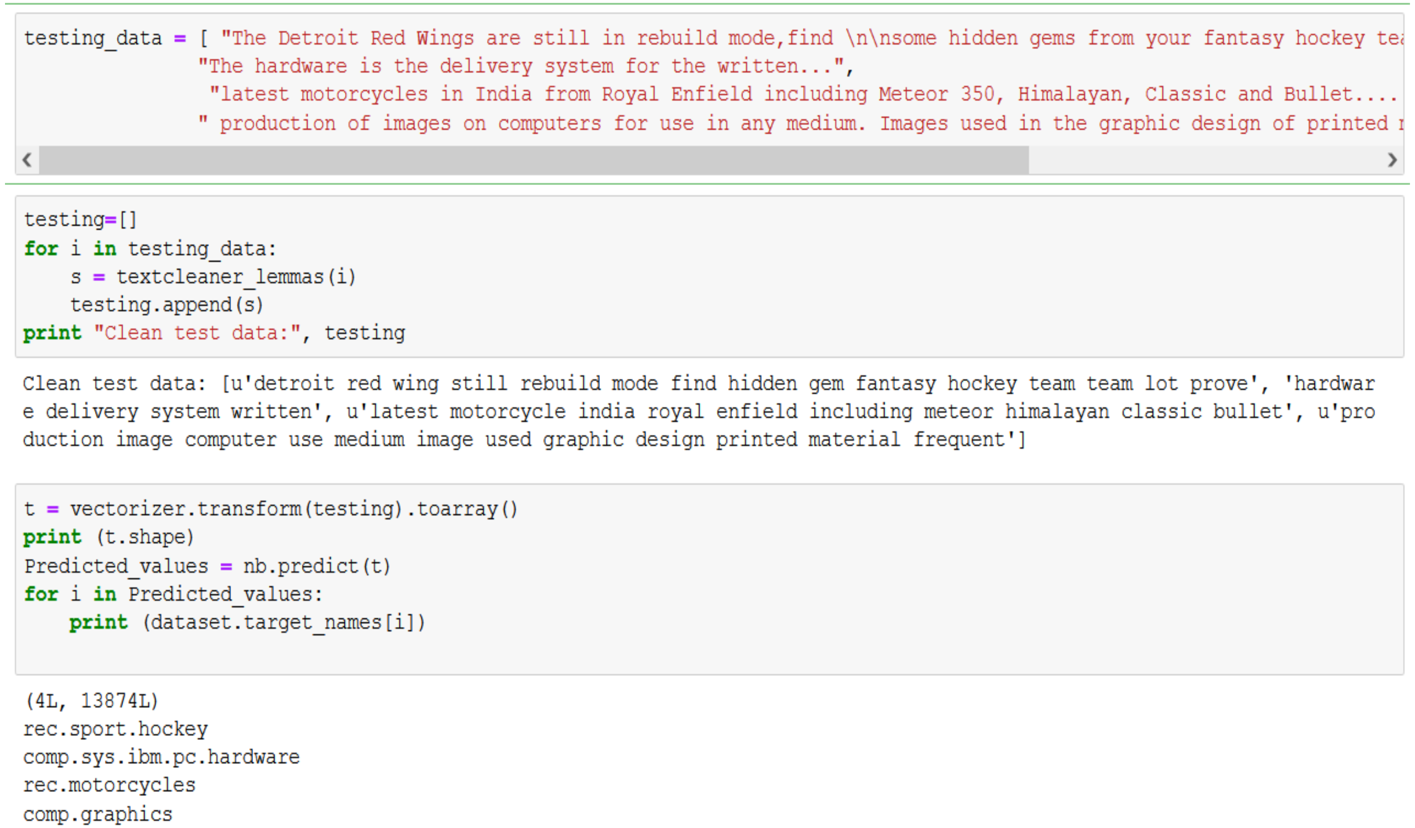
```
from sklearn.naive_bayes import MultinomialNB


#Initialize and fit
nb = MultinomialNB()
nb.fit(x_train, y_train)
nb.score(x_test, y_test)

# Apply to testing data
y_pred = nb.predict(x_test)
```

```
print("Accuracy is: %0.3f" % nb.score(x_test, y_test))
print(metrics.classification_report(y_test, y_pred, target_names=test.target_names))
```

```
Accuracy is: 0.673
```

PREDICTION ON RANDOM TEST DATA

- After training, testing and cleaning the data the model is now ready to predict the random values and in which category the data belong.
- We can provide a list of different data and test the model for its accurate prediction.
- This can be done by the following method in code:



```
testing_data = [ "The Detroit Red Wings are still in rebuild mode,find \n\nsome hidden gems from your fantasy hockey tea
               "The hardware is the delivery system for the written...",
                "latest motorcycles in India from Royal Enfield including Meteor 350, Himalayan, Classic and Bullet....
               " production of images on computers for use in any medium. Images used in the graphic design of printed
```

```
testing=[]
for i in testing_data:
    s = textcleaner_lemmas(i)
    testing.append(s)
print "Clean test data:", testing
```

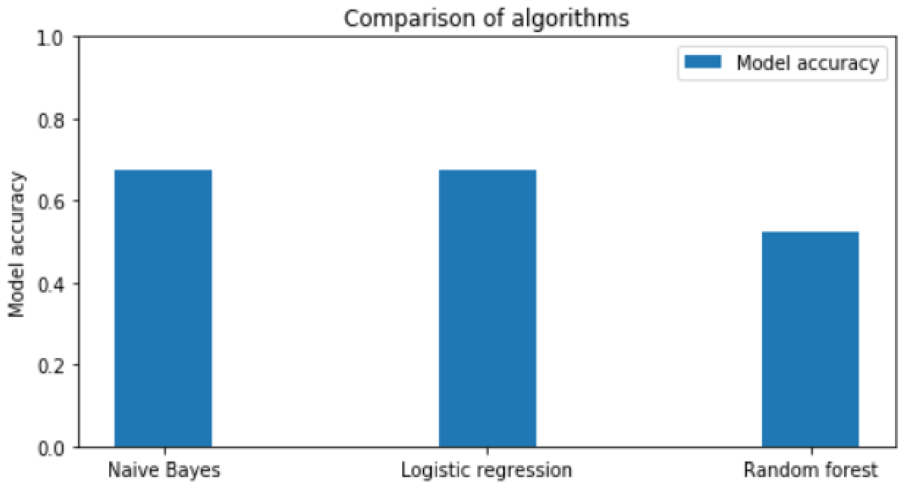
```
Clean test data: [u'detroit red wing still rebuild mode find hidden gem fantasy hockey team team lot prove', 'hardwar
e delivery system written', u'latest motorcycle india royal enfield including meteor himalayan classic bullet', u'pro
duction image computer use medium image used graphic design printed material frequent']
```

```
t = vectorizer.transform(testing).toarray()
print (t.shape)
Predicted_values = nb.predict(t)
for i in Predicted_values:
    print (dataset.target_names[i])
```

```
(4L, 13874L)
rec.sport.hockey
comp.sys.ibm.pc.hardware
rec.motorcycles
comp.graphics
```

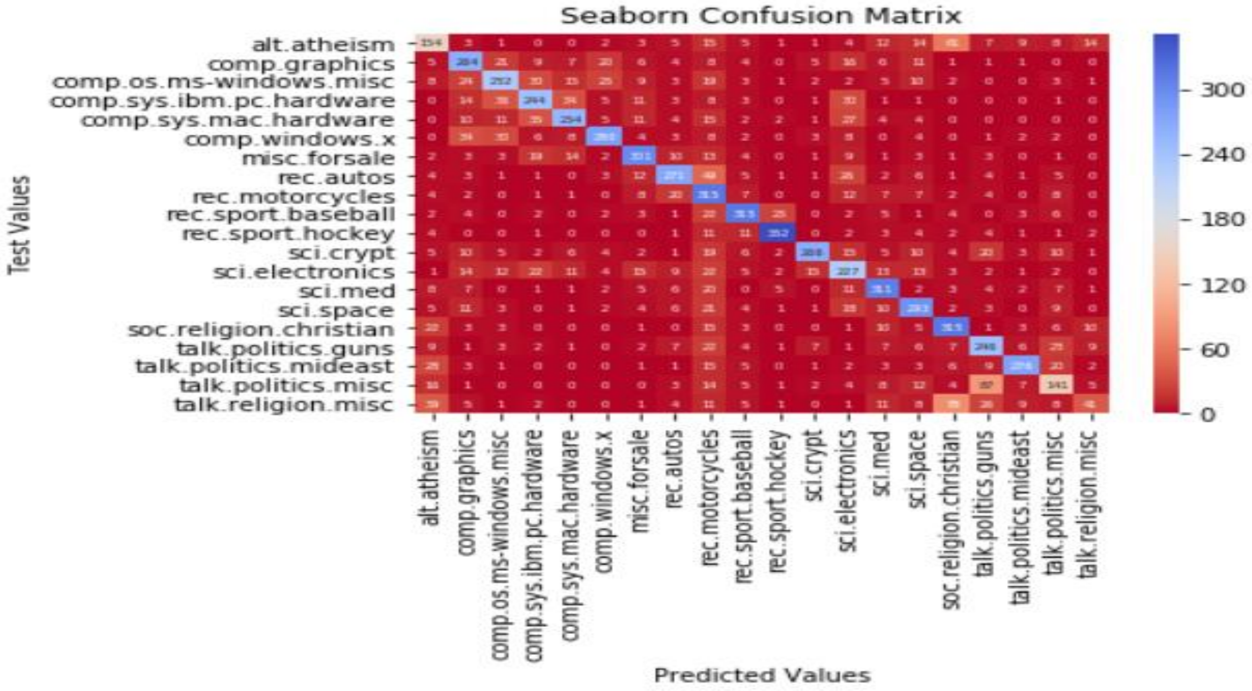
ALGORITHM PERFORMANCE

- In this analysis we can use different algorithms to get the better accuracy score.
- The algorithms used are:
  1. Naïve Bayes
  2. Logistics Regression
  3. Random Forest
- Each algorithm gives the different accuracy score which helps in the analysis of performance of each algorithm and the optimal performance can be found.
- Performance can be shown in the form of graph:



MODEL EVALUATION

- We will use *confusion_matrix()* from sckit-learn to compare real and predicted categories.
- A confusion matrix is a technique for summarizing the performance of a classification algorithm.
- Confusion matrices are useful because they give direct comparisons of values like True Positives, False Positives, True Negatives and False Negatives.
- The output of Seaborn Confusion Matrix is:



THANK YOU.