We need to scrape twitter data for the first part of this project for the following categories :

>>--Entertainment
>>--Political
>>--Health
>>--Business
>>--Technology

Each category is expected to have a atleast a minimum of 5000 tweets .

The data for each category needs to be saved in csv files. Once the tweets have been downloaded it should be saved as CategoryName_Raw_Data.csv Example :- Entertainment_Raw_Data.csv

The dataset needs to be further sub divided into the following parts as stated below :

1. Divide the training set into two parts namely :

a> Training Set(70% of total data) should be saved as CategoryName_Raw_TrainingSet.csv
b> Testing set(30% of total data) should be saved as CategoryName_Raw_TestingSet.csv

The data set should be divided randomly and not sequentially. This process should be achieved using a java program as well.

2. Further divide the training set(1 a>) into two data sets :

a> Training set pure (80%) should be saved as CategoryName_Raw_TrainingSetPure.csv
b> Validation set (20%) should be saved as CategoryName_Raw_ValidationSetPure.csv

<u>First Deliverable</u>

- Raw Data Sets – Total five in number named as CategoryName_Raw_Data.csv
- Training Set – Total five in number named as CategoryName_Raw_TrainingSet.csv
- Testing set – Total five in number named as CategoryName_Raw_TestingSet.csv
- Training set pure – Total five in number named as CategoryName_Raw_TrainingSetPure.csv
- Validation set – Total five in number named as CategoryName_Raw_ValidationSetPure.csv

Total – 25 csv files

- Document stating the process used to scrape data.
- Script used to scrape data or java program for crawling tweeter data using tweeter4j api(preferred)- Should be placed under the java project name SupervisedTweetClassification/ DataScraping package with the file named as TweetScrapper.java(to scrape data for each category) and Organizer.java (to divide the files randomly).

- All Raw Data Sets (.csv files) should be placed under the ScrappedData folder on execution of TweetScrapper.java. The execution of Organizer.java should divide the data sets and place them under the DividedDataSet folder further subcateogirzed under Example

Raw Data Sets folder, Training Set folder , Testing set folder , Training set pure and Validation set folder .

  - All these folders should be created directly under the main project and not as part of the package.
- Details of twitter account created for use of twitter4japi

More information for the first deliverable -

- Tweet data saved in the csv file should be of the form :

User handle Date of tweet Tweet Data

- The data downloaded for each category should be related to the category and should have more keywords suggesting the category than just the keyword. For example : Consider the business category a good example of tweets to be downloaded would be :

Good Example : -

  - text" : "RT @RT_com: Britain ignores Bahrain's human rights record to pursue business interests with dictatorship (Op-Edge)…
  - text" : "Sr. Manager of Business Operations job at Perfectly Posh - Salt Lake City https://t.co/kraMEoOJTf #Indeed #jobs
  - text" : "Market cheers GDP data, Sensex at 6-month high\n https://t.co/5Cjtj0w9Af https://t.co/H7eg7pQbvU
  - To download business related data one can use the obvious keyword business, along with keywords like sensex, tweets by big companies, etc.
  - This example should be used as a tool to download twitter data for all the other categories picking out tweets which are relevant to the category by use of the cateogry name as keyword along with other keywords relevant with the category. (VERY IMPORTANT)
-

Bad Example: -

  - text" : "RT @everything4dad: Were the niggas that mind their business and don't be w allat drama tf ?
  - text" : "RT @lildurk: I mind my Business I don't know shit

- Preferred language : JAVA

Second Deliverable

- Once the data set is created we need to clean all the noise from the data using the following steps:

    - Outliers removal - To remove low frequent and high frequent words using Bag of words approach.
    - Stop words removal - To remove most common words, such as the, is, at, which, and on.
    - Keyword Stemming - To reduce inflected words to their stem, base or root form using porter stemming
    - Cleaning crawl data
    - Spelling Correction - To correct spellings using Edit distance method.
    - Named Entity Recognition- For ranking result category and finding most appropriate.
    - Synonym form - If feature(word) of test query not found as one of dimension in feature space than replace that word with its synonym. Done using WordNet.

- While writing code for cleaning data all steps should be represented under seperated methods with name similar to the process. For Example -

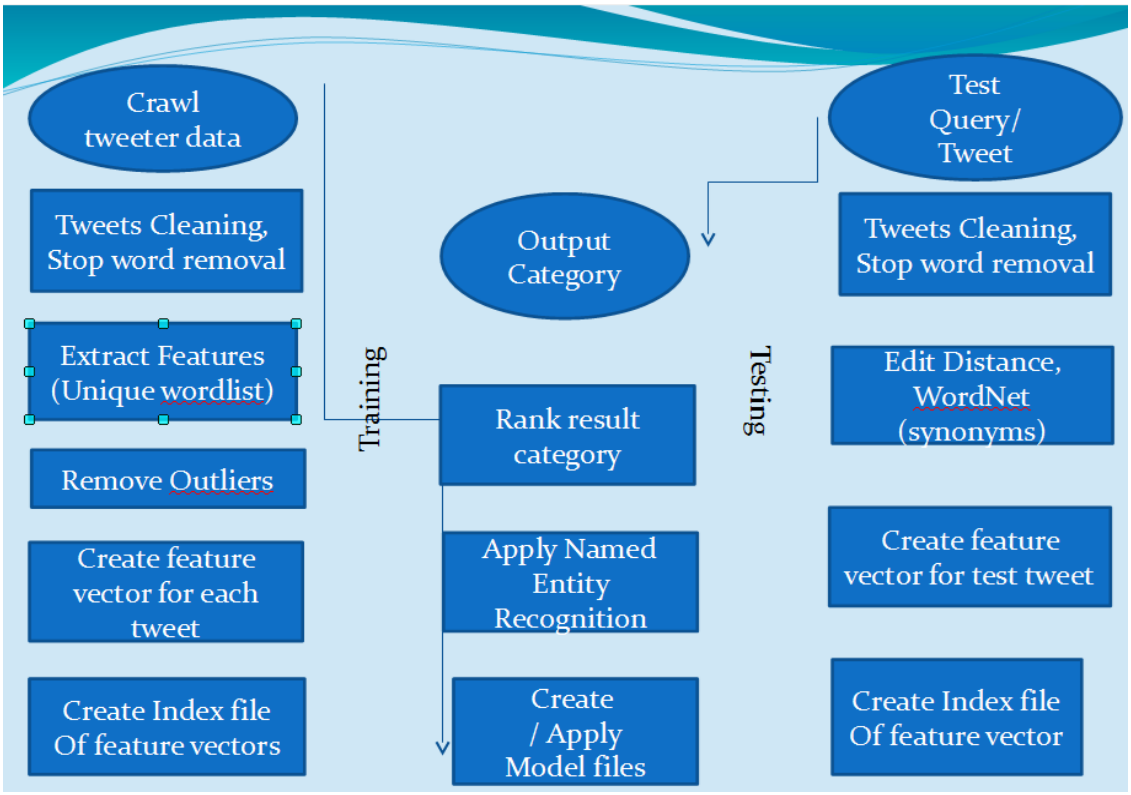
public void stopWordRemoval()

- All steps are important and none can be skipped.
- The Java Project should be created with the project name SupervisedTweetClassification with the second package named as DataPreprocessing. The java files expected under this package are :
    - NoiseRemoval.java – First java file with all the methods
    - PotterStemmer.java – With appropriate license of use
    - DataPreprocessingDriver – Containing the main function
    - Seperate file for stop words stored under Support folder
    - The input for this package should be taken from DividedDataSet folder. All the output files should be created automatically under the CleanData folder.
    - Clear comments for each method explaining the working of method in brief as well input and output to the method.
    - All data files (.csv files) after execution of DataPreprocessingDriver should be placed under the CleanData folder and further subcateogrized based on the notion of division used under first deliverable. Example

Raw Data Sets folder, Training Set folder , Testing set folder , Training set pure and Validation set folder .

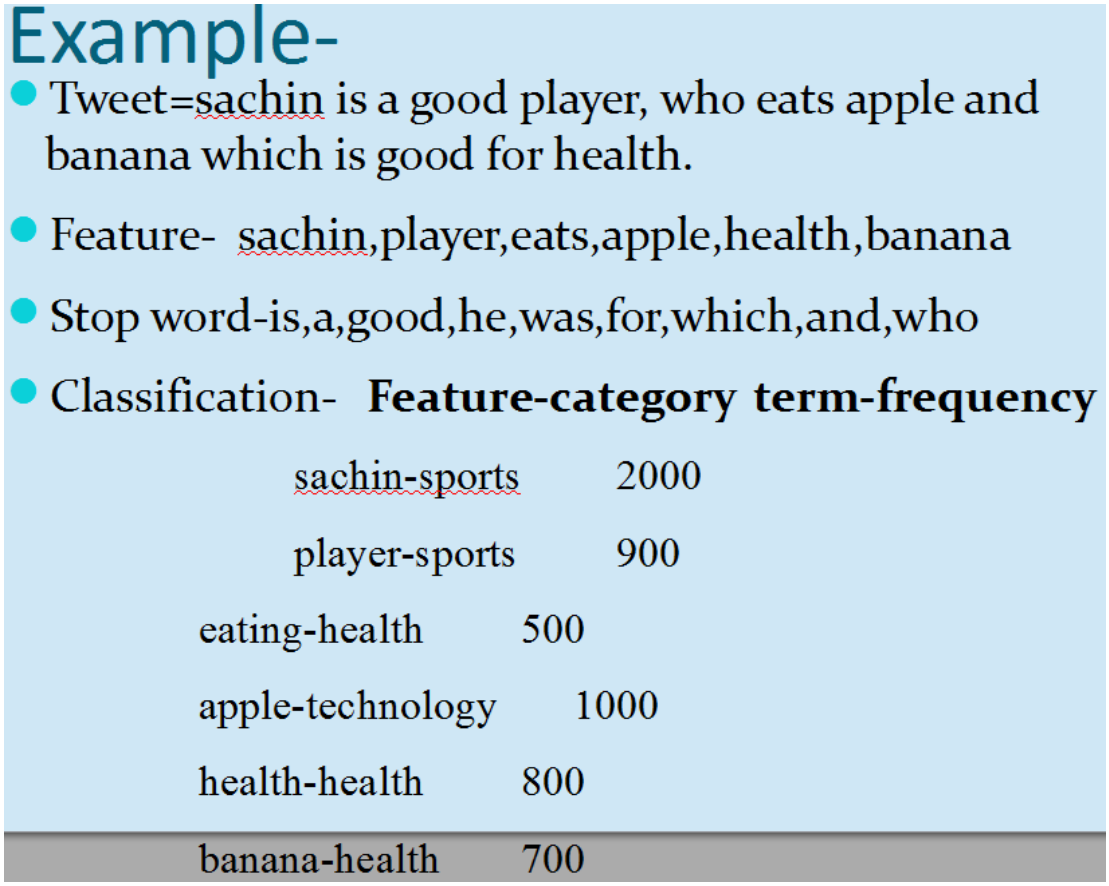
    - All these folders should be created directly under the main project and not as part of the package.
    - The preprocessed Input folder would be used as input for the third deliverable

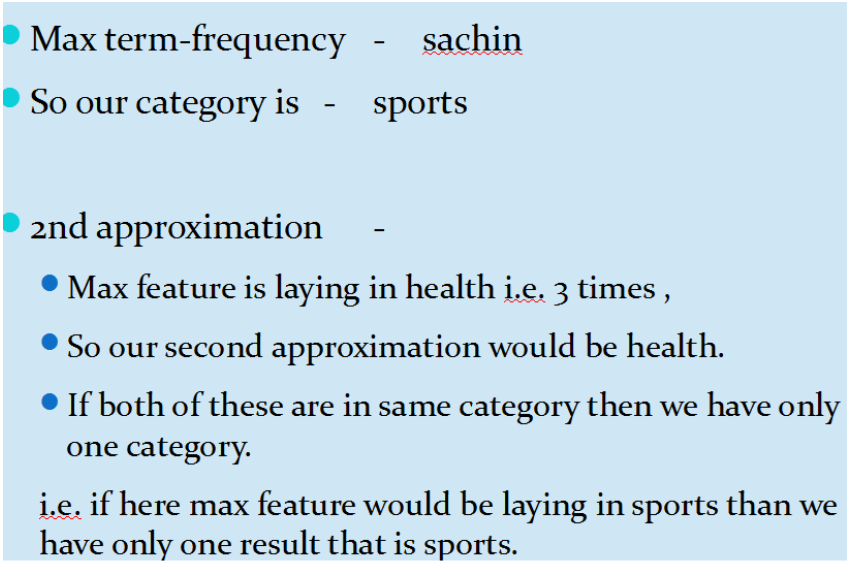
Third Deliverable



- Create feature vector for each tweet
- Create Index file of feature vectors
- Apply model files. As part of the third deliverable we need to classify the twitter data in each category using :
  - Naive Bayes
  - SVM Classifiers
  - Rule Based
- The java package containing these classifiers should be named as Classifiers . The java files expected under this package are :
  - NaiveBayesClassifier.java
  - SVMClassifier.java
  - RuleBasedClassifier.java
  - FeatureVector.java – Should contain methods to create feature vector for each tweet as well as index file of feature vectors . Below is an example of how a rule based classification should work .

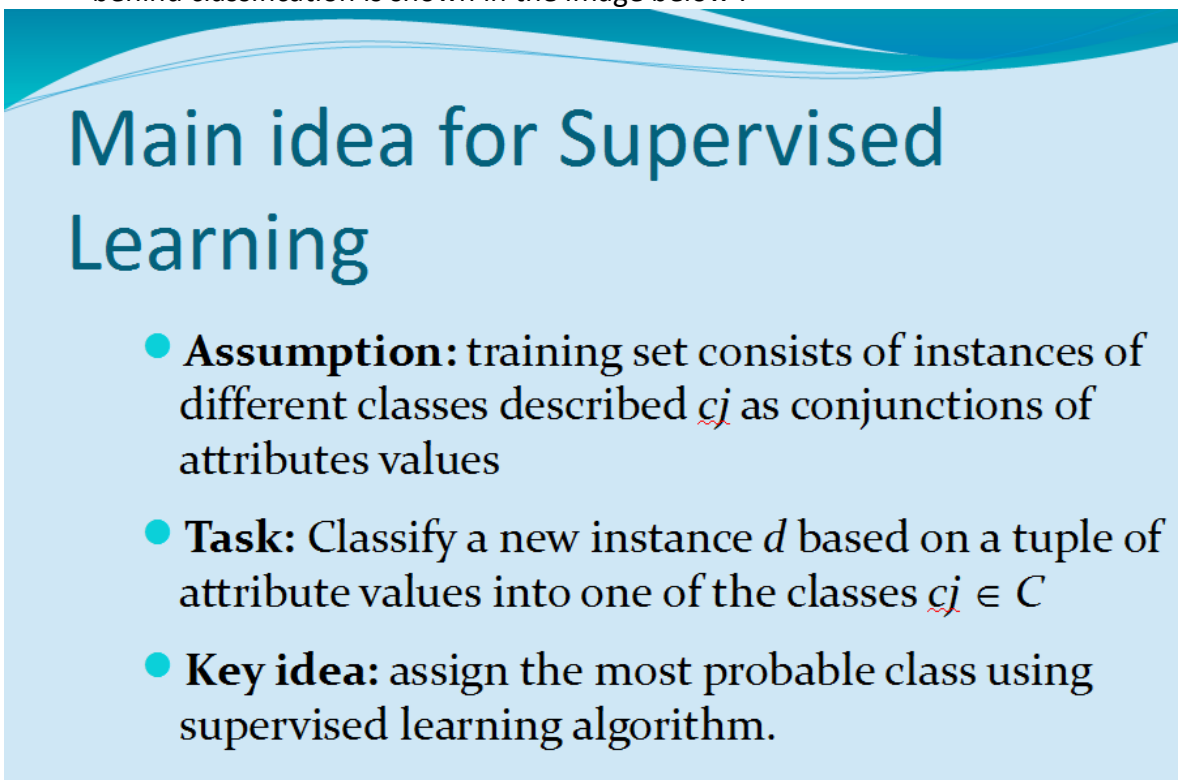
- The input for this package would be CleanData folder and output is to be stored under FeatureExtraction folder. It would contain index file of feature vectors as well feature vector for each text. Can be named appropriately
- In total there are four folders for input and output ScrappedData folder DividedDataSet folder CleanData folder FeatureExtraction folder





- FeatureVectorDriver.java -the main function to run the methods of FeatureVector.java

- ClassifierDriver.java – he main function to apply all the three classifiers. The main idea behind classification is shown in the image below :

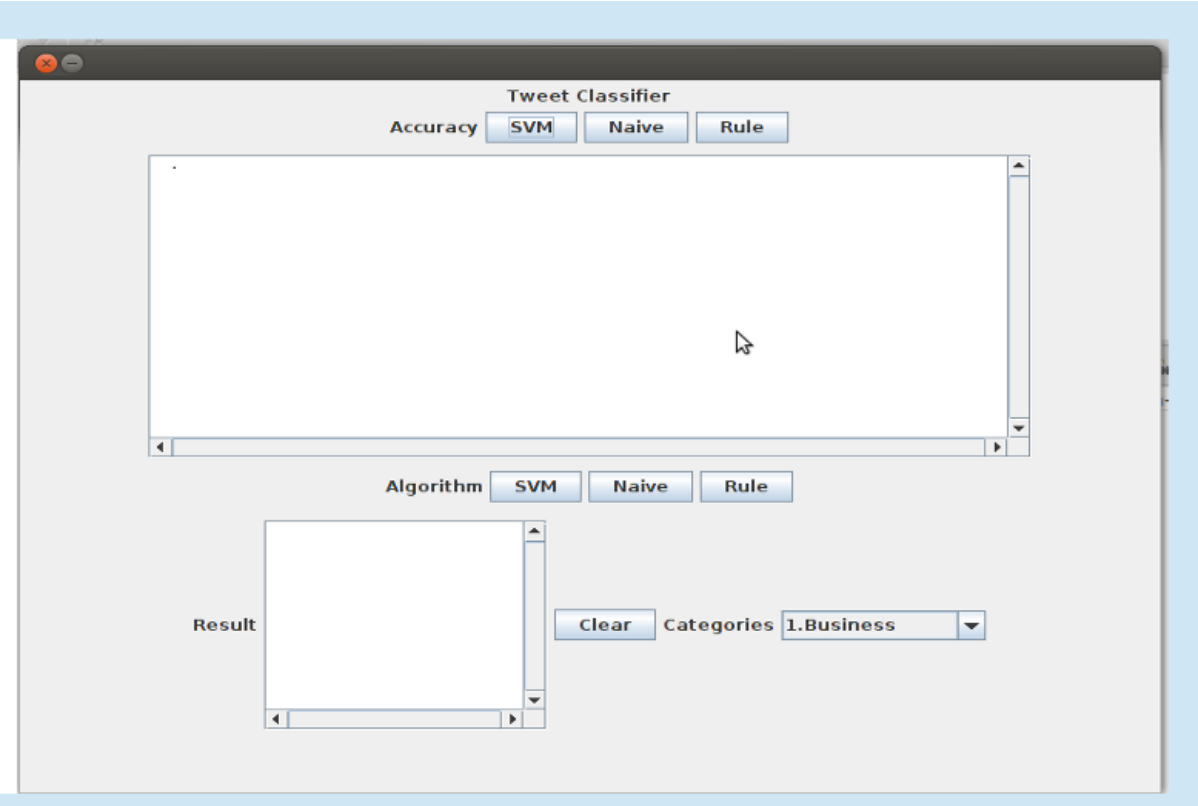


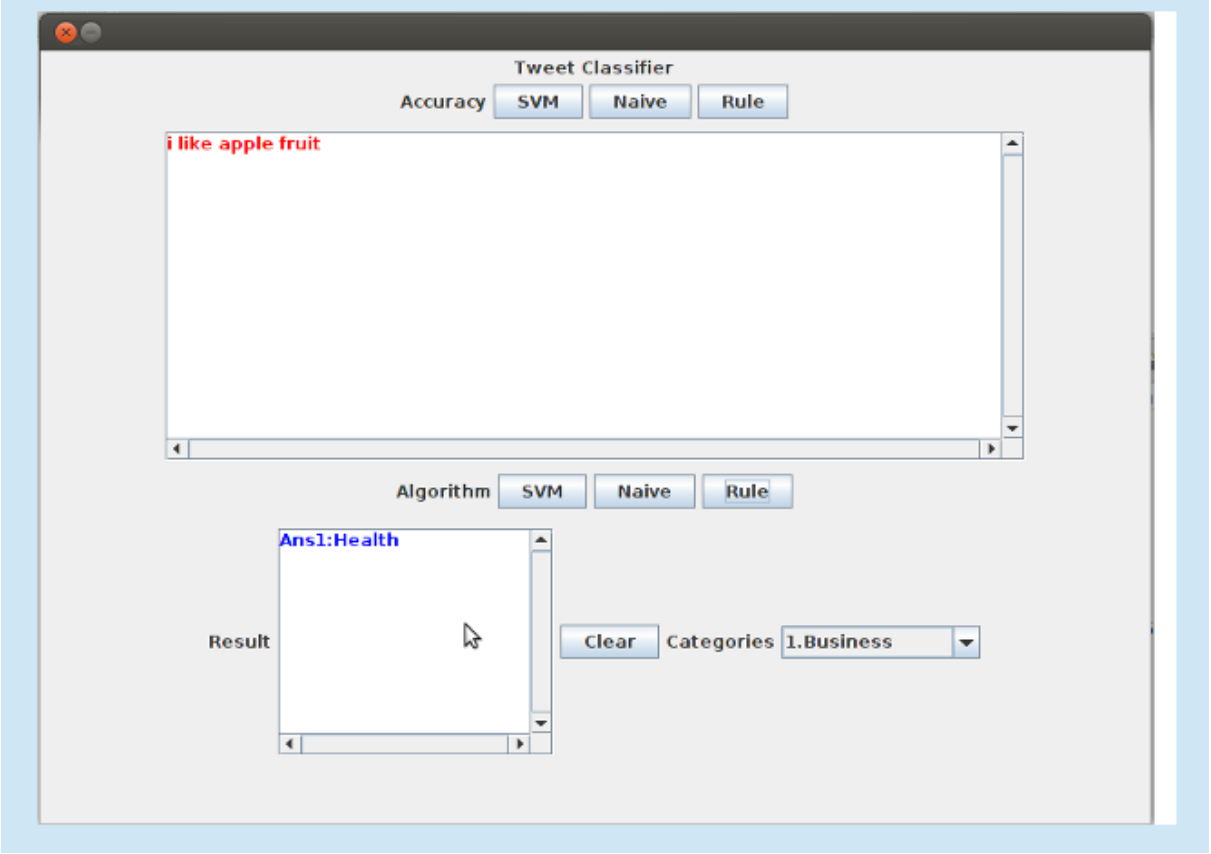
- The above driver files can be combined to be one single driver file based on your convenience.

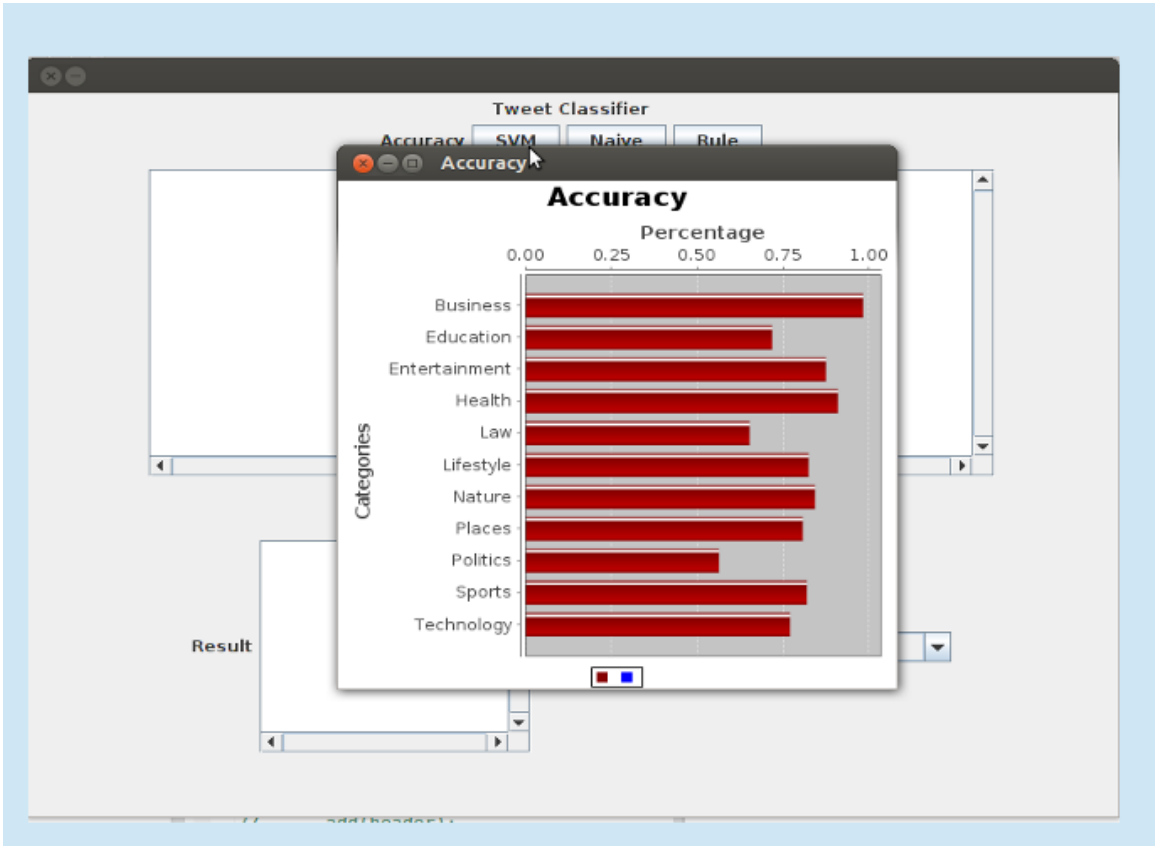
- The final output should be of the gui form and should give result of the classification in terms of accuracy :

Accuracy Results (10 folds)

Accuracy of Algorithm in %			
Categories\ Algo.	SVM	Naïve	Rule
Business	86.6	81.44	98.30
Education	85.71	76.07	81.8
Entertainment	86.8	79.1	87.49
Health	95.67	84.62	90.93
Law	81.17	73.38	75.25
Lifestyle	93.27	89.71	82.42
Nature	87.0	78.64	84.24
Places	81.01	75.35	80.73
Politics	81.91	81.88	76.31
Sports	87.11	83.57	81.87
Technology	83.64	82.44	77.05


Tweet Classifier
Accuracy
SVM
Naive
Rule
Algorithm
SVM
Naive
Rule
Result
Clear
Categories
1.Business


Tweet Classifier
Accuracy
SVM
Naive
Rule
i like apple fruit
Algorithm
SVM
Naive
Rule
Ans1:Health
Result
Clear
Categories
1.Business


Tweet Classifier
Accuracy
SVM
Naive
Rule
Accuracy
Accuracy
Percentage
0.00
0.25
0.50
0.75
1.00
Business
Education
Entertainment
Health
Law
Lifestyle
Nature
Places
Politics
Sports
Technology
Categories
Result