**Problem Definition:**Classify the articles into one of the five sections: Business, Entertainment, Politics, Sport, Technology.

**Solution Design:**  
The problem is divided into below parts:

* Features\Variables Generation
* Creation\Training of Model using Machine Learning Algorithm
* Validation of the model

**Features\Variables Generation**

The data provided has two variables which is “Category” and “Text”. The objective is to classify the text into one of the 5 categories (Business, Entertainment, Politics, Sport, Technology) thus we will consider Category as the Dependent variable and Text as an independent variable.  
Target\Dependent Variable(DV): Category  
Independent Variable\Features (IDV): Text

The Independent variable here is Text which is very long and as a whole they don’t provide any information. The Independent variable has to be broken into many small independent variables or features in order for the algorithm to process and make a sense of it. The small independent variables are nothing but the words\terms contained in the text. All the words contained in the text across all the observations in the data set will be considered as the independent variable or features. And the value for each independent variable will be its count of occurring in the document. For e.g the word “Actor” appears 7 times in observation 2 & 3 times in observation 3 then “Actor” will become a feature whose value in ‘observation2’ will be 7 and 3 in ‘observation3’.

|  |  |
| --- | --- |
| Document\Observation | Actor(Feature1) |
| 1 | 0 |
| 2 | 7 |
| 3 | 3 |

This method of generating features is called ‘Bag of words’ which assigns the value of features as the frequency of it occurring in the document. The disadvantage of the ‘Bag of words’ is it only looks at the frequency of the term occurring in one document which may or may not be useful for classifying the document to one of the categories. For e.g. the word ‘said’ occurs in every document multiple times since almost every news article will have an interview with some people and it will contain ‘Mr.X said “this is not good for the economy”’,’Mr.Y said “this is the future of Automation Technology”’. Such words don’t provide much information about the nature\concept of document. Thus the value for these words which are common across all the documents should have less value and the words which are unique to one set of documents should have more value like ‘Automation’, ’Republican Party’, ’Baseball’, ‘Movie’. This is done using the algorithm ‘TF-IDF’. For this problem the Features are generated using the TF-IDF algorithm.  
  
Feature generation Algorithm: TF-IDF

**Creation\Training of Model using Machine Learning Algorithm**  
After the features are generated a model has to be fit on the data set using the machine learning algorithms. The below algorithms will be explored for this problem and the algorithm which gives the highest accuracy on the validation data set will be the final algorithm:  
1. Support Vector Machine (SVM)  
2. Multinomial Logistic Regression  
If the validation accuracy of the above models is not as expected then below algorithms will be explored further:  
1. Neural Networks  
2. K-means Clustering  
3. K-nearest neighbors  
These algorithms are choses since they belong to classification algorithm and they fit with high accuracy (they may lead to overfitting but).  
  
Model Algorithm:  
Support Vector Machine (SVM)  
Multinomial Logistic Regression

**Validation of the model**  
In some cases the algorithm fits the model perfectly in the data set which is used to train the model, but it fails to do the same for data sets other than the training data set. This is because the algorithm failed to generalize the data and it over fits the training data set (high variance) and does not work well when given a new data set. To check this a validation data set will be used. The entire data set will be divided into 2 parts: training & validation. First the algorithm will be trained using the training data set and then it will be used on the validation data set to check for overfitting.   
  
For this problem multiple validation will be done as below:  
1. Randomly divided data set: Training :70%, Validation:30%  
2. Another randomly divided data set: Training :70%, Validation:30%  
3. Randomly divided data set: Training :75%, Validation:25%

**Workflow for the solution**1.Problem Definition  
2.Solution Design  
3.Data Extraction  
4.Data Cleaning\Preparation  
5.Data Exploration  
6.Model building  
7.Validation

**Data Extraction**Data is already extracted and provided in a csv format. The csv file has to be loaded in the program in a format which is suitable for analyzing, processing. Notable points for the data set is:  
1. The file has headers  
2. The columns are separated by ‘,’  
3. The Text column has ‘,’ and quotes ‘ ” ’  
Once the data is loaded to the program the data will be randomly sorted as the file may be prepared according to some pattern which will cause problem to the algorithm.  
  
The column Text is unstructured and thus we will convert it to a corpus which is a structured set of documents.  
 **Data Cleaning\Preparation**The data in the corpus is still a raw data and it has to be changed to a suitable way for processing\analyzing. First all the text will be converted to one particular case (either upper\lower). Since the data will contain unnecessary blank those will be removed. Punctuations, Special characters and numbers are not useful in explaining the context of the documents and will be removed from the data.  
  
Words like ‘the’,’a’,’is’,’if’,’of’,’on’ offer no information about the content in the document. These are called stop words and will be removed.  
  
Some of the words in the documents will have the same meaning. For e.g. words like respond-responded, play-playing-played, history-historical refer to the same thing in terms of machine. These words will be converted to their root word so that it will be easy for the algorithm to get the context of the document. The words respond-responded-response will be converted to ‘respon’ and play-playing-played will be converted to ‘play’. This process is called stemming the document.  
  
**Data Preparation\Feature Generation**Once the data is cleaned it can be used for feature generation. As mentioned earlier the algorithm ‘TF-IDF’ will be used to generate the features. First the corpus created will be created to a DTM (document Term Matrix) which will look like below.

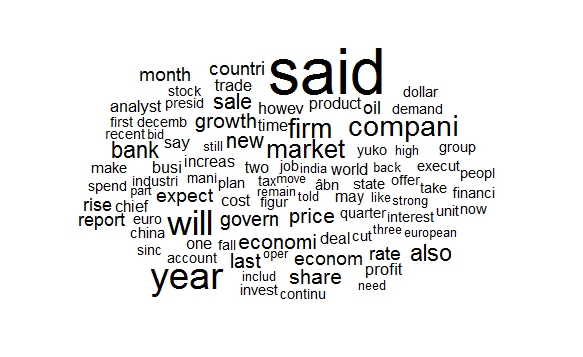
Documents

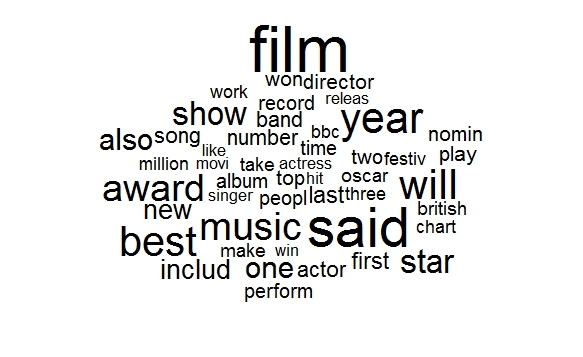
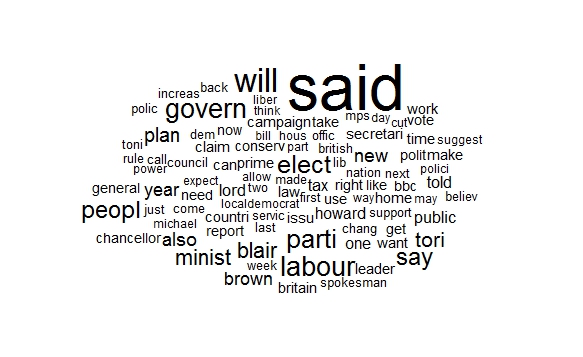
Terms

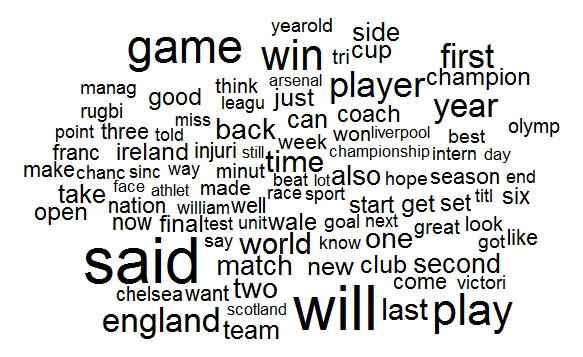
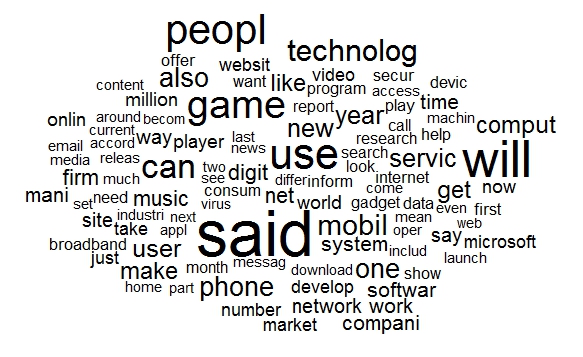
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Documents** | **Republic** | **NASA** | **Economy** | **Rugby** |
| Document1 | 1 | 3 | 0 | 0 |
| Document2 | 0 | 0 | 0 | 2 |
| Document3 | 2 | 0 | 0 | 5 |
| Document4 | 3 | 1 | 8 | 0 |

**Data Exploration**Once the features are generated, the data set is studied\explored in detail. The total number of documents is 2225 and each category has below number of documents.

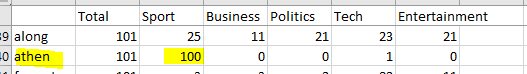
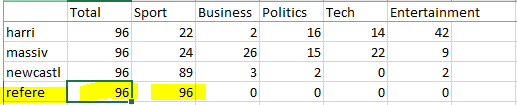
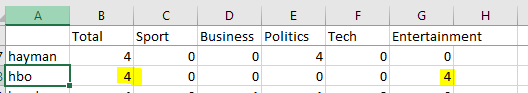
|  |  |
| --- | --- |
| Category | #document |
| business | 510 |
| entertainment | 386 |
| politics | 417 |
| sport | 511 |
| tech | 401 |
| **Grand Total** | **2225** |

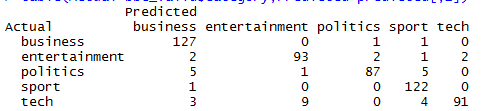
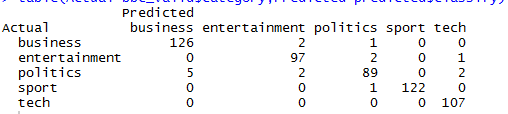
A mapping between each category and all the terms were generated so as to understand what all terms occur most for each category. Refer below for the frequency mapping.  
  
Understanding the relation between terms and the categories through excel will be somewhat difficult compared to graphs. Thus a graph of terms and the documents is plotted for a better understanding of the data set.  
  
Terms most occurring in the category business:  
  
The words which occur most have a larger size compared to words which occur less number of times.

**Entertainment:**  
 **Politics:**

**Sport:** **Tech:**

As seen from the above plots words like “said”,” can”,” will” occur across all the category of documents which is not useful in classifying and thus the weights of these terms should be reduced.  
The words like ‘game’,’film’,’music’,’sofwar’,’econom’,’labour’ occur only in specific category and thus the weights for these terms should be increased. Thus TF-IDF will be used to assign the inverse frequency.  
The final data will have large number of features against the documents. A lot of terms will occur only once\twice in the entire corpus offering little information to classify and will only be a bottle neck to process (computationally expensive). These terms are called sparse terms and can be removed.

**Modelling**The modelling will be done as per the below flow:  
1. First Model SVM with 96% sparse terms and training set 70% and validation set 30%  
2. SVM with only significant terms rather removing all sparse terms (99% sparse terms). Training Set 75%, Valid:25%  
3. Multinomial logistic regression(MLR) with 96% sparse terms. Training set 75%, validation set 25%  
4. MLR with different Random Training Set, Validation set. Training set 75%, validation set 25%  
5. MLR with different Random Training Set, Validation set. Training set 70%, validation set 30%  
6. MLR with only significant terms rather removing all sparse terms (99% sparse terms). Training Set 70%, Valid:30%  
7. MLR with only significant terms rather removing all sparse terms (98% sparse terms). Training Set 70%, Valid:30%  
8. MLR with only significant terms rather removing all sparse terms (99% sparse terms). Training Set 75%, Valid:25%  
  
**Choosing significant terms**  
Most of the features in the document term matrix are not useful for classifying the documents. Thus the sparse terms were removed. However, it may also be the case that some of the terms removed are actually significant in classifying and some of the terms retained are not helpful in classifying. For e.g. the word ’will’,’can’,’said’ are not helpful in classifying. Also for the first model when 96% sparse terms were removed even below terms were removed:  
  
‘athen’ useful in recognizing sports article.  
  
  
‘refere’ useful in recognizing sports article.  
  
  
  
‘hbo’ useful in recognizing Entertainment  
  
  
Thus only the significant terms has to be used as features. Again the terms occurring once or twice should be remove as they are not significant.  
Chi square test of goodness of fit is used to determine the significant terms.  
To determine the significant terms first the expected probability count of each terms has to be determined.  
  
If there are equal number of sports, business, politics, tech, entertainment documents in the data set then for each term the expected probability of it occurring in each of the category will be equal (=1\5)  
  
  
  
But the number of documents for each category is not equal and thus the expected probability of a term occurring in each category of the document will be different and as below.  
  
  
  
which is nothing but the (number of documents in particular category/total documents).  
Expected probability of term for sport = 511/2555  
Expected probability of term for business = 510/2555  
Expected probability of term for tech = 401/2555  
Expected probability of term for entertainment = 386/2555  
Expected probability of term for politics = 417/2555  
  
Once the expected probability of each term is known, the chi square test can be run to get the p-value for each terms.  
To choose the significant terms we will only chose terms with p-value less than 0.05 and rerun the model for both SVM and multiple logistic regression.

**Validation and conclusion:**  
The models are run and the training error and validation error are noted down. Below is the model summary for the problem.  
Confusion matrix for the model#2:  
  
Confusion matrix for the model #8:  
****

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Model No | Model Name | Remark | #Train Observation | #Valid Observation | Training Error | AIC | Deviance | Validation Error | Accuracy on Validation Set |
| 1 | SVM | First Model | 1557 | 668 | 1 | NA | NA | 38 | 94.31137725 |
| 2 | SVM | Model 2 choosing only significant features<=0.05, sparse terms=0.99 | 1669 | 556 | 0 | NA | NA | 37 | 93.34532374 |
| 3 | Multinomial Logistic Regression | First Model | 1669 | 556 | 0 | 6832.0002 | 0.0001581 | 24 | 95.68345324 |
| 4 | Multinomial Logistic Regression | Random training Valid Split\_2 | 1669 | 556 | 0 | 6832.0002 | 0.0001581 | 24 | 95.68345324 |
| 5 | Multinomial Logistic Regression | Random training Valid Split: Train:70, Valid:30 | 1557 | 668 | 0 | 6832.0001 | 0.0001276 | 27 | 95.95808383 |
| 6 | Multinomial Logistic Regression | P\_Value:0.05, sparse:0.98 | 1557 | 668 | 0 | 11288 | 0.0001548 | 21 | 96.85628743 |
| 7 | Multinomial Logistic Regression | P\_Value:0.05, sparse:0.99 | 1557 | 668 | 0 | 12008 | 0.0001353 | 20 | 97.00598802 |
| 8 | Multinomial Logistic Regression | Train:75, Valid:25, P\_Value:0.05, sparse:0.99 | 1669 | 556 | 0 | 12848 | 0.0001335 | 16 | 97.12230216 |

As shown the last model has the highest accuracy on validation model and thus it will be used for modelling.