TITLE PAGE

Project Title : Classification texts into opinion and facts.

Group No : 19

Group leader :
Abhinav Sharma 13IE10030

Group Members :
Somrup Chakraborty 13EX20027
Shubham Pachouri 13EX20026
Anum Raza 13AE10003

Main Objectives :

The objective of our project is to be able **to classify each sentence of an article into either events** (which can be proved true or false**) or expression** of feeling of people or opinions (which can not be proved true or false). The opinions are then to be further classified into Reporting Judgement, Advice and Sentiment Expressions.

The essence behind this binary classification is that it can be extended for use in a number of related projects like for an **opinion question answering system** where sentence classification is a necessary step. This may also be used to implement an alternate way for **document summarisation** by mapping the re-comments on the articles with these facts and opinions and using the generated data to summarise the article.

Dataset(s) :

Availed crawled datasets of newspaper articles and comments from newspapers such as **'The Guardian'.** The dataset thus availed from our mentor had been pre-divided into categories viz., **Business, Technolgy, Lifestyle etc**. The text in some of these sub-categories required cleaning.

Data Pre-processing :

After formulating our data set, for which we used crawled newspaper articles from the Guardian with a fair mix of articles with a preponderance of opinions and those without, like articles from Business, Environment, Lifestyle, Technology, Sports and Politics sections.
The sentences of those articles were then manually indexed to separate the facts and opinions and the opinion subcategories. In total**, manual classification was done for around 2000 sentences** which were then used for training. The indexing, however depended upon our discretion as facts and opinions **aren't exactly mutually exclusive** and multiple opinions may arise on whether a particular statement should be a fact or an opinion.

The manually indexed articles were stored in csv format which were used by our python script for feature detection so that a classification may be possible.

Feature Extraction :

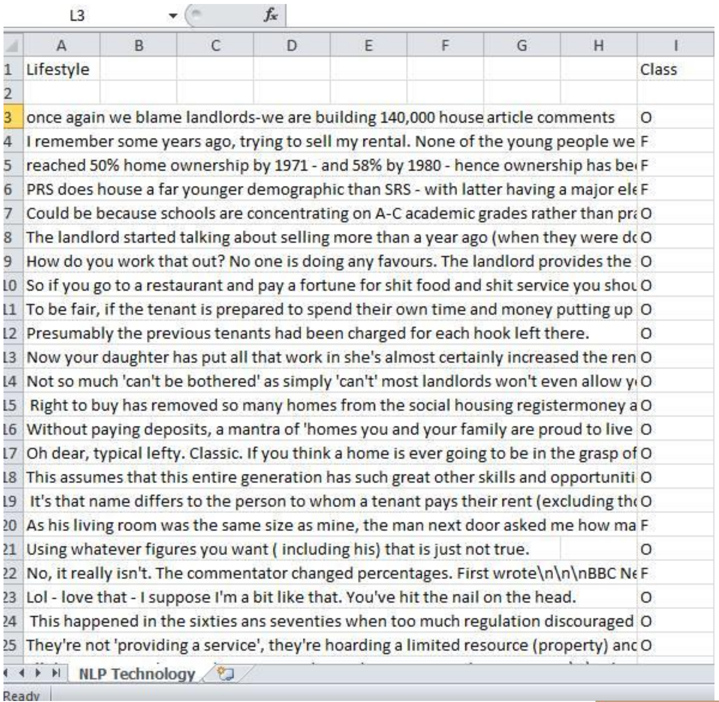
The features chosen were dependent on a common understanding as to **what grammatical features were more likely** to be present in a fact and which were more likely to be present in an opinion. **Facts** would have **more data and numbers** while **opinions** would have a higher presence of **strong and weak adjectives**.

A number of features, predominantly number of times a certain grammatical expression appears in a sentence, were identified on the basis of which the sentences were to be classified. These expressions are:

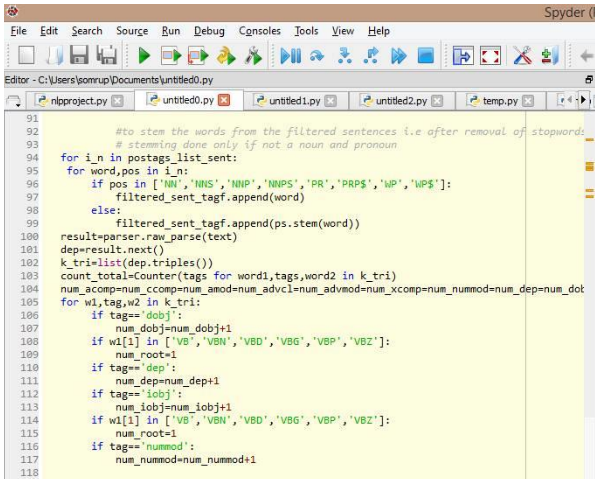
- acomp – Adjectivial complement
- advcl- Adverbial clause
- amod: adjectival modifier
- cc: coordination
- ccomp: clausal complement
- Presence of root verb
- Number of nouns
- Presence of strong and weak words
- Presence of number

The parts of speech were identified using the Stanford POS Tagger and the dependencies were extracted using the Stanford Dependency Parser.

Feature Extraction Snippets :



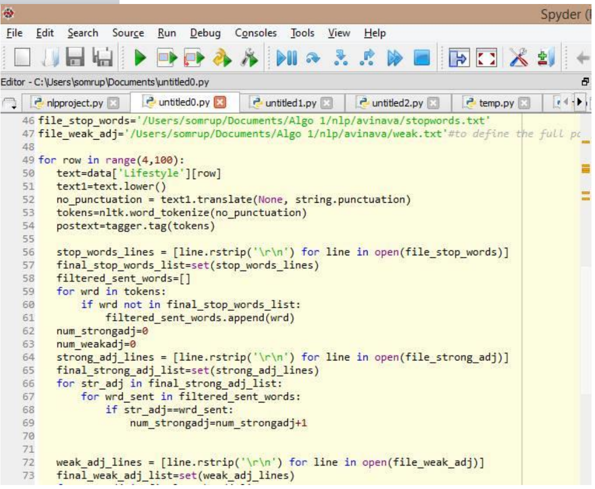
	A	B	C	D	E	F	G	H	I
1	Lifestyle								Class
2									
3	once again we blame landlords-we are building 140,000 house article comments								O
4	I remember some years ago, trying to sell my rental. None of the young people we								F
5	reached 50% home ownership by 1971 - and 58% by 1980 - hence ownership has be								F
6	PRS does house a far younger demographic than SRS - with latter having a major ele								F
7	Could be because schools are concentrating on A-C academic grades rather than pra								O
8	The landlord started talking about selling more than a year ago (when they were do								O
9	How do you work that out? No one is doing any favours. The landlord provides the								O
10	So if you go to a restaurant and pay a fortune for shit food and shit service you shou								O
11	To be fair, if the tenant is prepared to spend their own time and money putting up								O
12	Presumably the previous tenants had been charged for each hook left there.								O
13	Now your daughter has put all that work in she's almost certainly increased the ren								O
14	Not so much 'can't be bothered' as simply 'can't' most landlords won't even allow yo								O
15	Right to buy has removed so many homes from the social housing registermoney a								O
16	Without paying deposits, a mantra of 'homes you and your family are proud to live								O
17	Oh dear, typical lefty. Classic. If you think a home is ever going to be in the grasp of								O
18	This assumes that this entire generation has such great other skills and opportuniti								O
19	It's that name differs to the person to whom a tenant pays their rent (excluding th								O
20	As his living room was the same size as mine, the man next door asked me how ma								F
21	Using whatever figures you want (including his) that is just not true.								O
22	No, it really isn't. The commentator changed percentages. First wrote\n\n\nBBC Ne								F
23	Lol - love that - I suppose I'm a bit like that. You've hit the nail on the head.								O
24	This happened in the sixties ans seventies when too much regulation discouraged								O
25	They're not 'providing a service', they're hoarding a limited resource (property) anc								O



```

              #to stem the words from the filtered sentences i.e after removal of stopwords
              # stemming done only if not a noun and pronoun
    for i_n in postags_list_sent:
     for word,pos in i_n:
        if pos in ['NN','NNS','NNP','NNPS','PR','PRP$','WP','WP$']:
            filtered_sent_tagf.append(word)
        else:
            filtered_sent_tagf.append(ps.stem(word))
    result=parser.raw_parse(text)
    dep=result.next()
    k_tri=list(dep.triples())
    count_total=Counter(tags for word1,tags,word2 in k_tri)
    num_acomp=num_ccomp=num_amod=num_advcl=num_advmod=num_xcomp=num_nummod=num_dep=num_dob
    for w1,tag,w2 in k_tri:
        if tag=='dobj':
            num_dobj=num_dobj+1
        if w1[1] in ['VB','VBN','VBD','VBG','VBP','VBZ']:
            num_root=1
        if tag=='dep':
            num_dep=num_dep+1
        if tag=='iobj':
            num_iobj=num_iobj+1
        if w1[1] in ['VB','VBN','VBD','VBG','VBP','VBZ']:
            num_root=1
        if tag=='nummod':
            num_nummod=num_nummod+1

```



```
file_stop_words='/Users/somrup/Documents/Algo 1/nlp/avinava/stopwords.txt'
file_weak_adj='/Users/somrup/Documents/Algo 1/nlp/avinava/weak.txt'#to define the full pc

for row in range(4,100):
    text=data['Lifestyle'][row]
    text1=text.lower()
    no_punctuation = text1.translate(None, string.punctuation)
    tokens=nltk.word_tokenize(no_punctuation)
    postext=tagger.tag(tokens)

    stop_words_lines = [line.rstrip('\r\n') for line in open(file_stop_words)]
    final_stop_words_list=set(stop_words_lines)
    filtered_sent_words=[]
    for wrd in tokens:
        if wrd not in final_stop_words_list:
            filtered_sent_words.append(wrd)
    num_strongadj=0
    num_weakadj=0
    strong_adj_lines = [line.rstrip('\r\n') for line in open(file_strong_adj)]
    final_strong_adj_list=set(strong_adj_lines)
    for str_adj in final_strong_adj_list:
        for wrd_sent in filtered_sent_words:
            if str_adj==wrd_sent:
                num_strongadj=num_strongadj+1


    weak_adj_lines = [line.rstrip('\r\n') for line in open(file_weak_adj)]
    final_weak_adj_list=set(weak_adj_lines)
```

Machine Learning for classification :

The feature extraction was possible for nearly 1036 sentences. 70% of this data was used to train a eXtreme Gradient Boosting model, for binary classification.

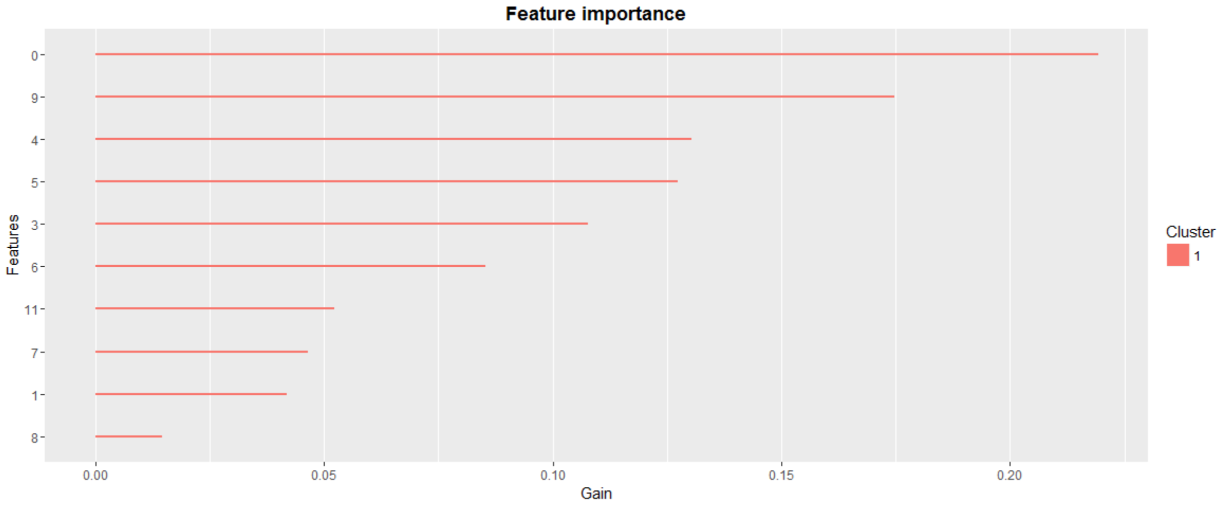
Before running XGboost, we must set three types of parameters: general parameters, booster parameters and task parameters.

The parameters used for modelling are as under :

After trying out multiple approaches like “multi:softprob” the objective of the xgboost was set as “reg:logistic”. Other parameters include -
Eval_Metric : “merror” identifies the rate of mis classification
Min child weight : 4
Max_depth : 10
The total number of trees used for boosting = 1000.
Eta = 0.1 to avoid overfitting.

Feature Importance :

The below graph shows which features are important in the model.



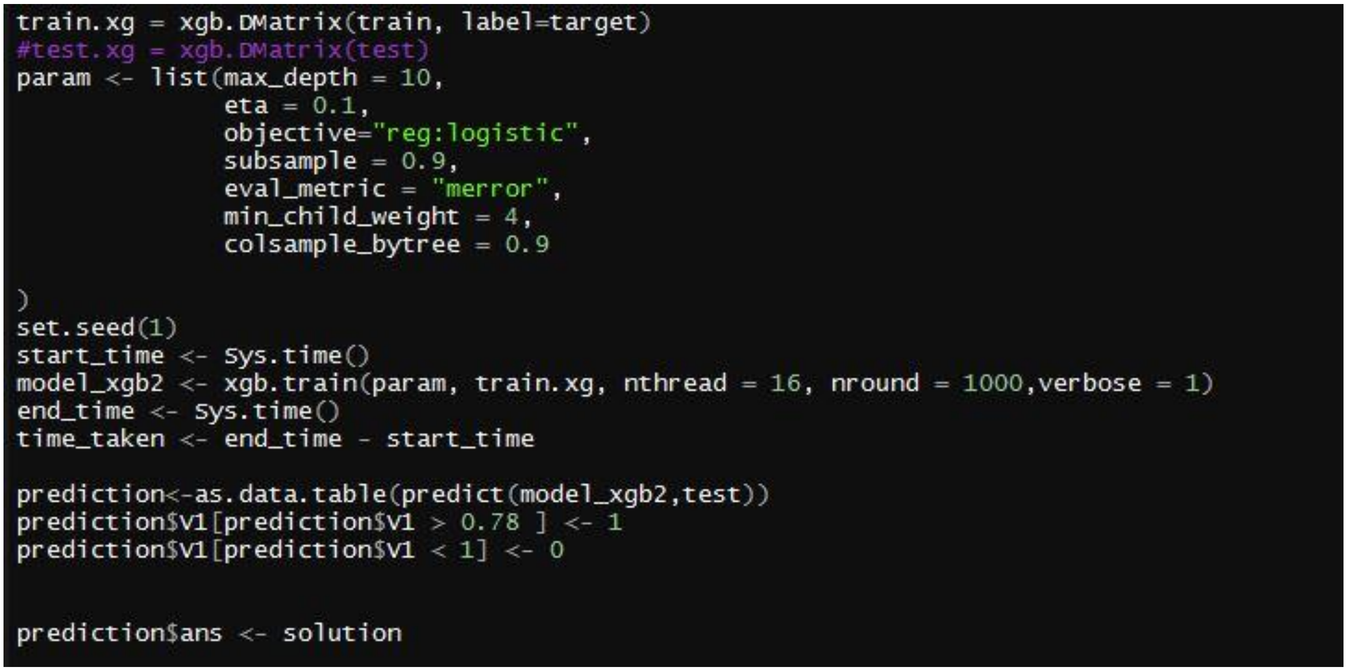
Below graph shows the feature importance plot ; the most important features being number of **strong adjectives**, number of **numeric ordinals**, number of **adverb modifier** and number of **adverb clause modifier** respectively.

Results :

From the **317 sentences** tested, **243** were identified correctly with a reasonable accuracy of **77.91%**. This accuracy may be improved further by validation and by adding
more features to our model.

Machine Learning Code Snippets :

Model in R



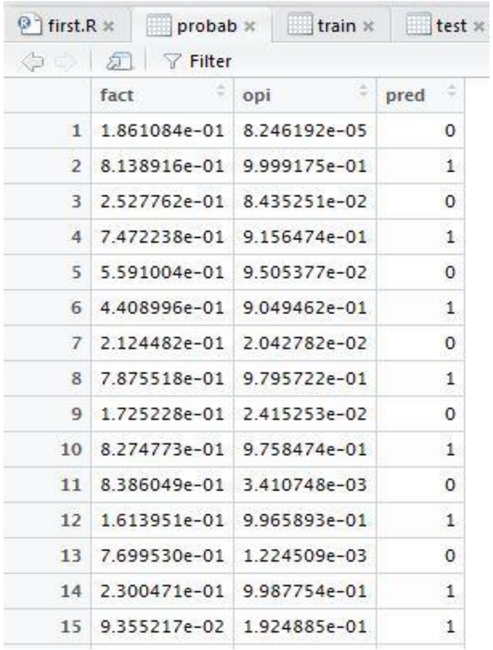
```r
train.xg = xgb.DMatrix(train, label=target)
#test.xg = xgb.DMatrix(test)
param <- list(max_depth = 10,
              eta = 0.1,
              objective="reg:logistic",
              subsample = 0.9,
              eval_metric = "merror",
              min_child_weight = 4,
              colsample_bytree = 0.9

)
set.seed(1)
start_time <- Sys.time()
model_xgb2 <- xgb.train(param, train.xg, nthread = 16, nround = 1000,verbose = 1)
end_time <- Sys.time()
time_taken <- end_time - start_time

prediction<-as.data.table(predict(model_xgb2,test))
prediction$V1[prediction$V1 > 0.78 ] <- 1
prediction$V1[prediction$V1 < 1] <- 0


prediction$ans <- solution
```

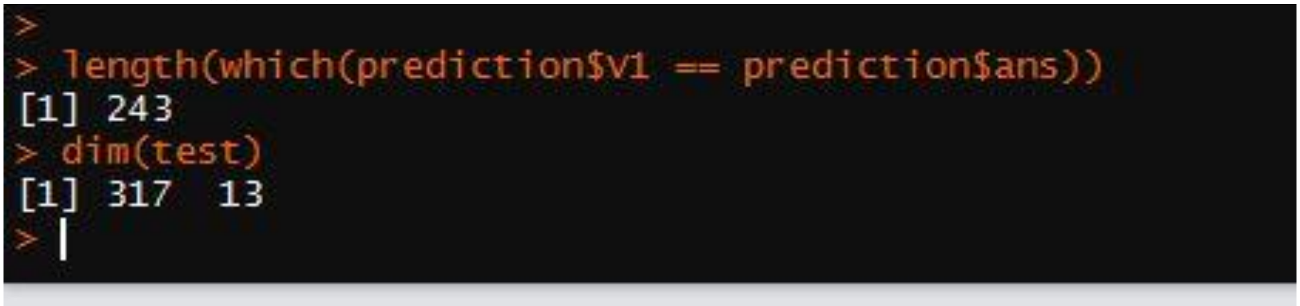
Probabilities by multi:softprob



	fact	opi	pred
1	1.861084e-01	8.246192e-05	0
2	8.138916e-01	9.999175e-01	1
3	2.527762e-01	8.435251e-02	0
4	7.472238e-01	9.156474e-01	1
5	5.591004e-01	9.505377e-02	0
6	4.408996e-01	9.049462e-01	1
7	2.124482e-01	2.042782e-02	0
8	7.875518e-01	9.795722e-01	1
9	1.725228e-01	2.415253e-02	0
10	8.274773e-01	9.758474e-01	1
11	8.386049e-01	3.410748e-03	0
12	1.613951e-01	9.965893e-01	1
13	7.699530e-01	1.224509e-03	0
14	2.300471e-01	9.987754e-01	1
15	9.355217e-02	1.924885e-01	1

Appendix :

(A) Snippet for result :



```
>
> length(which(prediction$v1 == prediction$ans))
[1] 243
> dim(test)
[1] 317  13
> |
```

(B) Work done by individual team mates :

Sr. no.	Name	Roll no	Work Done
1	Abhinav	13IE10030	Manual annotation to Business & Environment sub-fields Machine Learning modelling and testing in R
2	Somrup	13EX20027	Manual annotation to technology and world sub-fields Feature Extraction from individual sentences in Python
3	Shubham	13EX20026	Manual annotation to politics and sports sub-fields Report and Presentation
4	Anum	13AE10003	Manual annotation to lifestyle and opinion subfields Contribution in Report