Université d'Ottawa
Faculté de génie

École d'ingénierie et de
technologie de l'information

uOttawa
L'Université canadienne
Canada's university

University of Ottawa
Faculty of Engineering

School of Information
Technology and Engineering

CSI 6900: Intensive Graduate Project

Political News Classification into Sensationalist and Objective

Project Report

Gurpreet Singh

Student Number: 8908872

Supervisor: Prof. Diana Inkpen

Acknowledgement

I would like to express my sincere gratitude and thanks to Dr. Diana Inkpen for giving me the opportunity to work on the task of Classifying the political news into Objective and Sensationalist Categories as a part of my Master's graduate project. She was always very helpful and provided me with all the necessary guidance and direction throughout the project. She always ensured her availability whenever I needed her guidance. Without her expertise on the Natural language processing and her continuous support and guidance, it would not have been possible to complete the project on time.

Lastly, I would like to thank the NLP group members for sharing their knowledge throughout our monthly meetings which definitely helped me a lot in getting a lot of exposure to multiple concepts of Natural language processing.

Table of Contents

Abstract

In the age of digital news, the sensationalist news has been spreading in the huge amount and has generated bulk of misinformation, best example being the presidential elections of the United States conducted in 2016, during which a significant amount of the false or modified information was circulated before the voting. This definitely provided an upper hand to Donald trump over the rival candidate Hilary Clinton [3] [4]. It is the modern day trend to make the news sensational to get the focus of the users online with the motive of generating money online whether through ads or getting the support from the political parties. Hence, it is very important to categorize the news into Sensationalist and Objective. Sensationalist news is written in the format with the intention of attracting the focus of the user [2] [9]. It might not be 100% fake but is definitely modified a lot compromising the real facts. While Objective news is the one which is delivered without any manipulation of the content and solely based on the facts. This project explores the usage of natural language processing techniques for classifying the news as Sensationalist or Objective. Using the dataset from the politifact.com which is extracted and cleaned by the Computational Linguistics Research Group at the University of Antwerp, we apply the term frequency-inverse document frequency (TF-IDF) and DictVectorizer of multiple text statistic features to a corpus of around 12000 articles [9]. We test the two classification algorithms – Support Vector Machine and Bernoulli Naive Bayes on the test set. Using TF-IDF of multiple features and feeding them to the SVM helps in achieving the accuracy of 92% to classify the news as Sensationalist or Objective.

1. Introduction

1.1 Motivation & Prior Work

The sensationalist news has become a big concern these days, particularly to describe the factually incorrect and misleading articles published mainly with the motive of making money through page views. Sensationalism is a type of editorial bias in mass media in which news stories are overhyped to present the biased impressions on the events, which definitely cause the manipulation to the truth of the story. The effect of sensationalism could be very well observed during the 2016 presidential elections in the United States when the modified news gave an advantage to Donald Trump in winning the elections [3][4]. The Computational Linguistics Research group at the University of Antwerp has done some work on this problem. They had collected the dataset using the list of resources from the opensources.co. Mainly, the data was retrieved from the politicat.com and was processing by the researchers in the research group [9]. They got decent results of getting the accuracy of 92% using the Support Vector Machine Classifier for detecting whether the news is sensationalist or Objective.

1.2 Main Contribution

We have tried to do the experiments with the motive of increasing the accuracy of the model by using additional features and additionally testing the results using different combinations of the list of features. Some of the unique features added are Modality and words with first letter upper case. Modality refers to the degree of certainty as a value between -1 and 1.0 [1]. Moreover, we used TfidfVectorizer rather than CountVectorizer as CountVectorizer just counts the word frequencies whereas with the TfidfVectorizer, the value increases proportionally to count, but is offset by the frequency of the word in the corpus. The experiments with the changes in the ratio of the train and test data were also done in order to check if we could increase the accuracy of the model. Addition of the new features increased the precision score for the objective news while there was not much impact on the recall. The maximum accuracy we attained almost remained the same as that of the results from the researchers at the University of Antwerp [8]. Finally, we tested the accuracy of multiple models including random forest classifier, SVM and Bernoulli Naïve Bayes. The results from the SVM came out to be the best at 92% for this tasking of classifying the news as Sensationalist and Objective with an increase in the precision to 93% for classifying the news as Objective.

2. Dataset

There are multiple sources which publish information about the news domain like OpenSources which is available for public use. Other sources like Politifact and MediaBiasFactCheck.com which contain 1600+ media sources and categorize them into classes such as objective, sensationalist, questionable, right, right-center, least biased, left, left-center, conspiracy, pseudoscience etc [8] [9] [10].

2.1 Data Pre-processing:-

The CLIPS (Computational linguistics & Psycholinguistics) research group at the University of Antwerp, Belgium has done a great job of collecting the dataset and organizing it into the proper shape. They aggregated and compiled a list of multiple sources of the news with approximately 2000 in number. This list had the following details: - name of the source, URL of the source, fake or real [8].

Then, the research team handpicked some of the sources with the major emphasis on the politics and the world news. The team scrapped the articles from approximately 200 resources from the above compiled list for multiple days and hence was able to generate a dataset of roughly 16000 articles for the training with around 8000 of them classified as sensationalist and the other approximate 8000 as objective. The test data had the size of approximately 1165 articles in total. The details of this dataset are mentioned in the figure 1 below [8].

Removal of ads, social links, html tags were taken care of by the team using the following modules of the pattern library:-

a) Datasheet
b) Pd
c) Newsfeed
d) URL
e) DOM
f) plaintext

	Headlines	Body	Source	Label
Training	16657	16657	16657	16657
Test	1165	1165	1165	1165

Fig 1 – Format and the size of the training and test data

Each of the articles is classified as either Sensationalist or Objective with the label 1 for the sensationalist and 0 for the objective. Below mentioned is the distribution of the sensationalist and objective count of the articles.

	Sensationalist	Objective
Training	7999	7977
Test	498	497

Fig 2 – Number of Sensationalist and Objective news in training and testing data

Data Format – Below mentioned is the format of the dataset with a sample example of each header.

Headline - Breaking! NY Times is devastated, Trump Jr. just broke them!

Body - Donald Trump Jr. released his emails this morning shutting down the fake news conspiracy on his meeting with a Russian lawyer.

A writer for the NY Times and Politico is absolutely devastated that Trump Jr. ruined a story he had spent the last year working on.

Have a closer look:

You gotta love what this American patriot wrote in response to this:

If you havenâ€™t checked out and liked our Facebook page, please go here and do so.

Source - The Washington Feed

Label – 1 which means it is labeled as the Sensationalist News

Example of Sensationalist News - Despite Voting to Keep Obamacare, John McCain Promised to Repeal It During 2016 Campaign

Example of Objective News - In big move, Narendra Modi government empowers Army, Navy, IAF, hands over financial power to boost security at installations

The Sensationalist News is given a label of 1 whereas the objective news is given a label of 0.

Clearly, there is a difference in the way of writing between the Sensationalist and the objective news. The user tries to highlight each word using the first letter as the capital.

3. System Overview

We used four tools primarily in the development of the sensationalism classifier: Python 3.6, Scikit-learn, Natural Language Toolkit (NLTK). I was comfortable with python and the reason for choosing other tools are mentioned below.

3.1 Scikit-learn

Scikit–learn is a Python package which provides a multiple variety of machine learning methods and algorithms for feature extraction and normalization. Further, it provides the ability to model selection by comparing, validating and improving the accuracy by tuning the parameters [6].

My experience with Scikit-learn before beginning this project was not too much and I had to spend a decent amount of time in understanding the different modules of Scikit-learn and the best way to use these modules.

- **sklearn.feature_extraction:** It has the methods to normalize the data before passing it to the classification model. I used DictVectorizer and TfidfVectorizer primarily from the feature_extraction module.
- **sklearn.pipeline:** It primarily manages the flow for preprocessing, feature extraction, normalization and classification with a better accessibility for tuning. I used the functionalities FeatureUnion and Pipeline from the pipeline module.
- **Sklearn.base:** Used BaseEstimator and TransformerMixin as they are useful to make the model grid searchable with GridSearchCV for automated parameters tuning. It behaves well when combined in a pipeline with others.
- **Sklearn. naive_bayes:** Used Bernoulli Naïve Bayes as the baseline model.
- **Sklearn.svm:** It was considered to be the best model during testing.
- **Sklearn.metrics:** This module includes the score functions, performance metrics and pairwise metrics and distance. The API classification report builds a text report showing the main classification metrics.

3.2 Natural Language Toolkit

NLTK is a python package which provides text processing libraries for classification, tokenization, stemming, tagging, parsing and semantic reasoning [5]. Using NLTK, we were able to perform a number of data preprocessing tasks on data which we will discuss in section 4.3. Particularly, we used the following modules:

- **nltk.word_tokenize:** Helps in dividing the string into lists of substrings specifically words and punctuations. For example -

  ['Good', 'muffins', 'cost', '$', '3.88', 'in', 'New', 'York', '.', 'Please', 'buy', 'me', 'two', 'of', 'them', '.', 'Thanks', '.'] This list is formed using the word tokenizer.

- **nltk.sent_tokenize:** Helps in tokenizing the string at the sentence level. It tokenizes each sentence separately. For example –

  ['Good muffins cost $3.88\nin New York.', 'Please buy me\ntwo of them.', 'Thanks.'] This list is formed using the sentence tokenizer

- **nltk.pos_tag:** Helps in assigning the POS tags to the tokens of the sentence. It can be used in combination with the word tokenizer.

  [('Good', 'JJ'), ('muffins', 'NNS'), ('cost', 'VBP'), ('$', '$'), ('3.88', 'CD'), ('in', 'IN'), ('New', 'NNP'), ('York', 'NNP'), ('.', '.'), ('Please', 'NNP'), ('buy', 'VB'), ('me', 'PRP'), ('...', ':'), ('two', 'CD'), ('of', 'IN'), ('them', 'PRP'), ('.', '.'), ('Thanks', 'NNS'), ('.', '.')] This is a sample list in which pos tags are assigned to the tokens.

4. Model

It was realized that to create a baseline model was very important to start off with some results. Then, using those results, we tried other approaches/models to come out with better results. In this section, we will discuss about the baseline classifier and the final model we opted to achieve the results to classify the news into sensationalist and objective.

4.1 Baseline

We decided to proceed with Bernoulli Naïve Bayes as our baseline model. The only feature we used for the baseline model was the map having the punctuation count for each type of the punctuation symbol which we have discussed in detail in the section 4.3.

$$P(x_i \mid y) = P(i \mid y)x_i + (1 - P(i \mid y))(1 - x_i)$$

Fig 3 - Decision Rule for Bernoulli naïve Bayes [6]

BernoulliNB implements the Bayes training and classification algorithms for data that is distributed according to multivariate Bernoulli distributions which mean that there could be multiple features but each one of them is assumed to be a binary-valued variable i.e the Boolean variable. The decision rule of the Bernoulli Naive Bayes explicitly penalizes the non-occurrence of a feature i which is an indicator for class ɣ, where the multinomial variant would simply ignore a non-occurring feature [6] [11].

4.2 Model Selection

We decided to proceed with the SVM classifier which is a supervised machine learning algorithm and is considered to be one of the best for the classification problems. It works really well with clear margin of separation and uses a subset of training points in the decision function (called support vectors), which makes it memory efficient too [6].

4.3 Features

- **Text Statistics** - refer to the combination of average sentence length, all-caps, profanity, polarity and subjectivity features.
  - **Average sentence length** – It is calculated by dividing the summation of the number of tokens in each sentence by the number of sentences in the paragraph [5].

    av_sent_len = float(sum(sent_lengths)) / len(sent_lengths)
  - **Profanity** – The number English swear words like anus, arse, arsehole, ass, bitch, bitchass, bitches, bitchtits etc. It is calculated by dividing the number of swear words in the paragraph by the total number of word tokens in the paragraph [1].

num_prof = float(len([w for w in tok_text if w.lower() in PROFANITY]))/len(tok_text)

- **Polarity & Subjectivity –** The pattern.en module provides a lexicon of adjectives like good, bad, amazing, irritating etc. which occur very frequently in the opinion of the people. It is annotated with scores for sentiment polarity, which is either a positive or a negative value, and subjectivity, which is either objective or subjective. Basically, the sentiment() function from the Pattern.en module returns a (polarity, subjectivity) tuple for the given sentence. The value of the polarity ranges from -1.0 to +1.0, whereas that of subjectivity ranges from 0.0 to 1.0 [1].

  Example – from pattern.en import sentiment

  print sentiment("The way elections were conducted last year were pathetic and its negative effect can be seen even till now")

  (-0.34,1.0) where -0.34 is the polarity and 1.0 is the subjectivity of the sentence.

- **Punctuation** – to get the frequency count of each punctuation mark.

  Here is the list of the punctuations which are considered to get the frequency of each one of them [5].

  '"!()-[]{};:'"\,<>./?@#$%^&*_~'"

- **POS unigrams** – It involves POS tagging of the individual words [5].
- **POS bigrams** – It involves tagging the words with context to its neighbors [5].

4.4 Final Model

We ended up using an SVM classifier as the final model for our task of classifying the news as Objective or Sensationalist. Our classifier uses the linear kernel classifier with the random state of 0. Our final model achieved an accuracy of **92 %** using all the features described above. Using the above features combined provided better results than using the features individually.

In SVM, the kernel functions can be any of the following:-

- Linear
- Polynomial
- Rbf
- Sigmoid

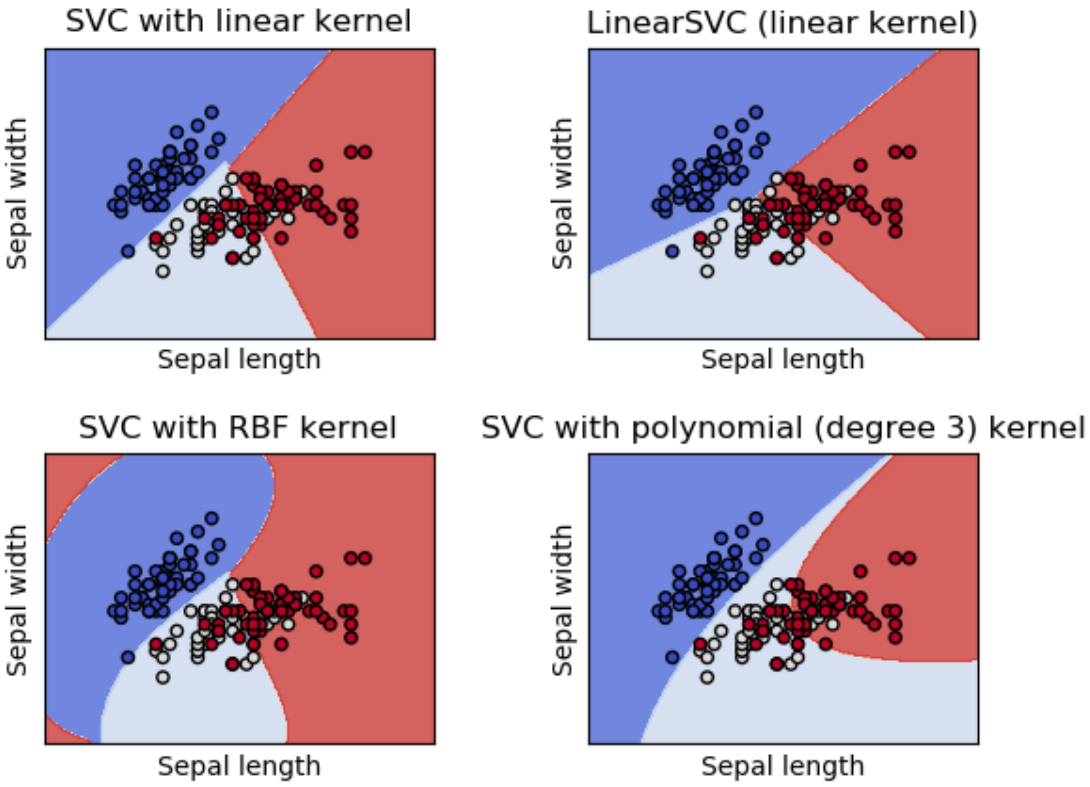

Fig 4 – Classification visualizations of different kernels of SVC and LinearSVC [6]

5. Evaluation Measures

The performance of each classifier is evaluated based on the accuracy with which it recognizes whether the news is sensationalist or objective in the test data. F-measure is considered to be the best way to evaluate the performance which requires Precision and Recall to be calculated.

Precision: - It is the percentage of selected items that are correct.

$P = TP / (TP + FP)$ *where*

P = Precision TP = True Positives FP = False Positives [7]

Recall: - It is the percentage of correct items that are selected.

$R = TP / (TP + FN)$ *where*

R = Recall TP = True Positives FN = False Negatives [7]

F-measure: - F-score or F-measure is used in statistical analysis to measure the test's accuracy. It uses Precision and Recall as its parameters to compute the score. It can also be interpreted as weighted average of the precision and recall where score of 1 indicates the best value and that of 0 states the worst value. It can also be considered as harmonic mean of the precision and recall.

$F1 = (2 \times P \times R) / (P + R)$ *[1]*

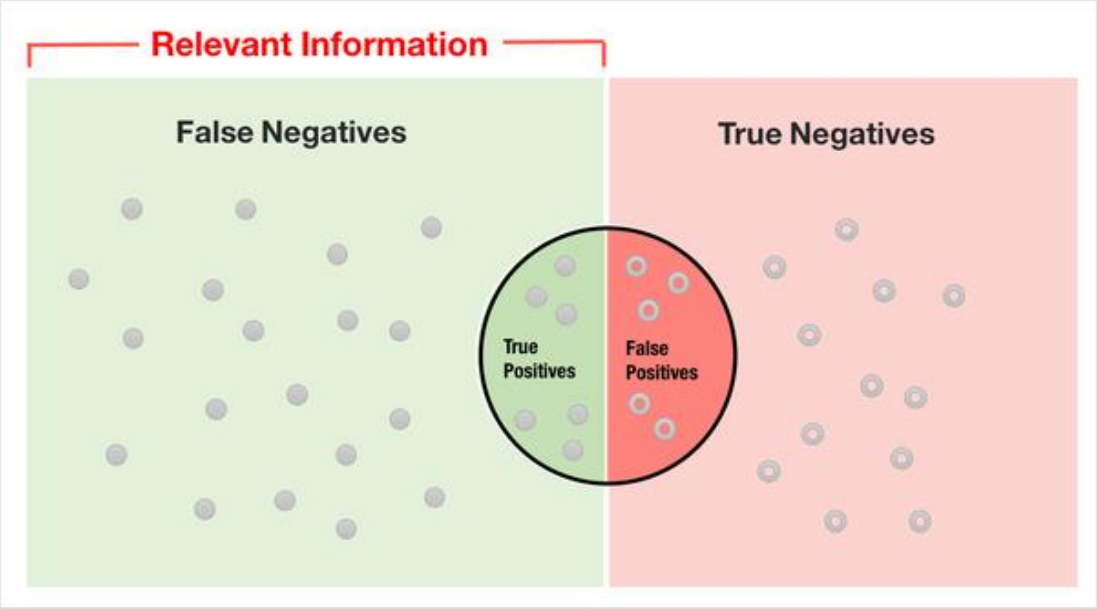

Fig 5 - Overview of Precision and Recall components [7]

6. Results & Discussion

6.1 Baseline

We used Bernoulli Naïve Bayes classifier to test the results for the baseline model. The following features were used for the baseline:-

- **Punctuation count** - The punctuation count for the news headlines and the news body was being used and convert to the vector to be used as the feature for the baseline model.

The Accuracy of the baseline model comes out to be 75%. It would have been lower if we would not have used the punctuation count as the feature. Below are the confusion matrix and the classification report for the same results of the baseline model. The AUC of this model come out to be 0.76 as demonstrated in the fig 8. Fig 6 represents the confusion matrix for this model while the classification report can be observed in the fig 7.

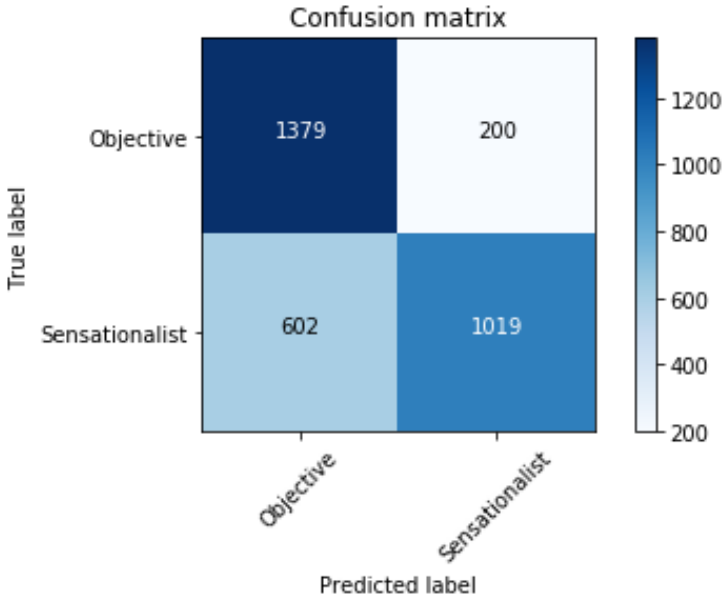

Fig 6– Confusion matrix for the Baseline Model

	precision	recall	f1-score	support
0	0.87	0.70	0.77	1981
1	0.63	0.84	0.72	1219
micro avg	0.75	0.75	0.75	3200
macro avg	0.75	0.77	0.75	3200
weighted avg	0.78	0.75	0.75	3200

Fig 7 – Classification report of the Baseline Classifier

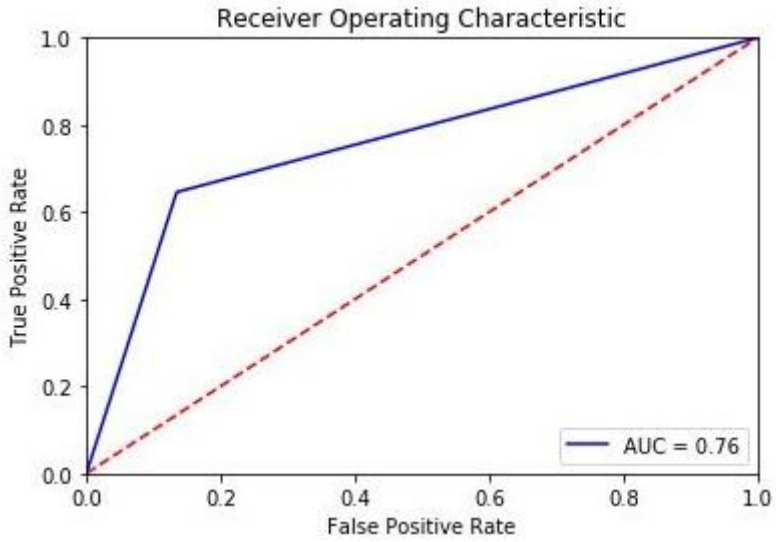

Fig 8 – ROC curve of the Naïve Bayes with AUC = 0.76

6.2 Final Model

We used SVM classifier with the linear kernel as the final model to achieve our classification task. This classifier achieved an accuracy of 92% on the test dataset. The details for the precision and recall can be found in the classification report Fig 10. which is obtained using the metrics package of the Scikit learn library. The AUC of this model come out to be 0.92 as demonstrated in the ROC curve in the fig 11.

Error Analysis – Fig 9 shows the confusion matrix for our final classifier for the task to classify the news as sensationalist or objective. In this confusion matrix, the labels on the rows indicate the true labels and the labels on the columns indicate our classifier's predicted labels. Samples found along the diagonal indicate they were correctly classifier. Darkened areas indicate that more samples fell into that area in comparison to those in the lighter areas. It is good to know that with the accuracy of 92%, our classifier is correctly able to predict the news as objective and Sensationalist which can be observed in the dark blue boxes of the confusion matrix. From the lighter boxes, it can be observed that the classifier is more aligned towards falsely predicting the Sensationalist news as Objective than the other way around as it is predicting 148 Sensationalist news as Objective while 120 of the Objective news as Sensationalist. It could be due to lack of some more features or inability to distinguish between the normal text and the text pattern based upon the current set of text statistics used. Still we should appreciate the level of accuracy achieved.

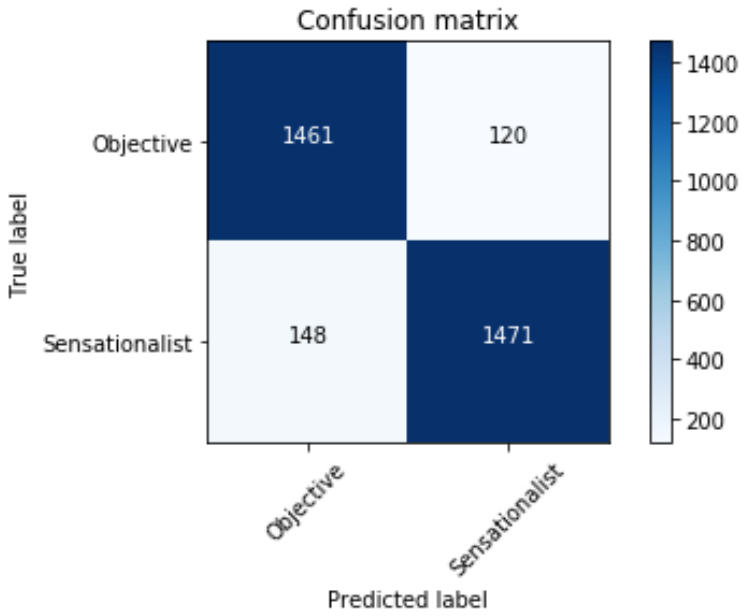

Fig 9 – Confusion matrix for the SVM Classifier Results [6]

	precision	recall	f1-score	support
0	0.93	0.91	0.92	1652
1	0.91	0.92	0.92	1548
micro avg	0.92	0.92	0.92	3200
macro avg	0.92	0.92	0.92	3200
weighted avg	0.92	0.92	0.92	3200

Fig 10 – Classification report of the SVM Classifier [6]

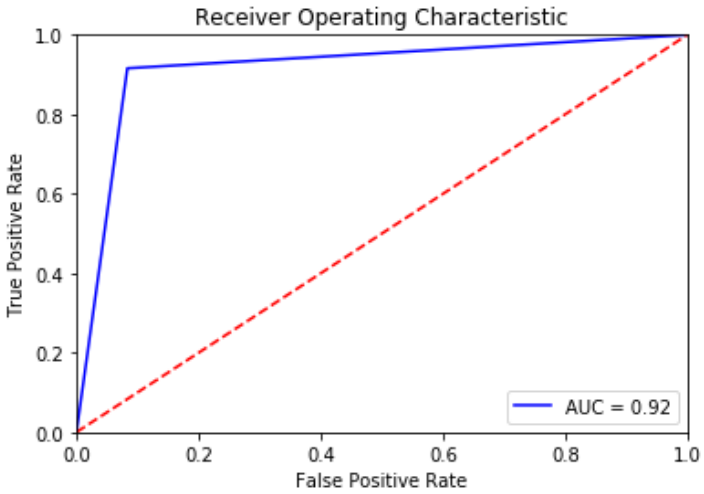

Fig 11 – ROC curve of the SVM Classifier with AUC = 0.92

7. Conclusion

We took a supervised machine learning approach in predicting the Sensationalism of the political news data from the politifact.com. We have shown our baseline classifier and then the improvements made to get the better model. As the dataset was balanced in terms of the number of the ratio of the number of the sensationalist news and the objective news, we didn't have to spend too much time on processing the data set as the major processing part was already handled by the researchers at the University of Antwerp, Belgium [8]. Using the SVM model with the linear classifier and the additional features like modality and upper case words, we were able to achieve our most accurate classifier for this task.

The Scikit-learn helped us a lot by providing the implementations of many machine learning algorithms, which triggered us to research and finalize the best model for our task. The different packages it provides especially sklearn.metrics, helped us in visualizing the results in the much better way. Overall, this project was a great opportunity for me to explore the different possibilities of Machine Learning.

8. Future Work

Our model and results were good but still further improvements could be done moving forward. Firstly, we would be interested in finding more features and try different combinations of the current features and the new one to further improve the accuracy of the model. Secondly, we would like to do some more research and compare our model with that of other researchers to evaluate their models and possibly improve the accuracy using their inputs. Lastly, we would be interested in exploring some other datasets which could belong to some different language and would like to train and test our model on that to see the classification accuracy.

References

[1] Clips.uantwerpen.be. (2018). pattern.en | CLiPS. [online] Available at: https://www.clips.uantwerpen.be/pages/pattern-en [Accessed 12 Dec. 2018].

[2] En.wikipedia.org. (2018). Sensationalism. [online] Available at: https://en.wikipedia.org/wiki/Sensationalism [Accessed 12 Dec. 2018].

[3] Gilda, S. (2017). Evaluating machine learning algorithms for fake news detection. In: 2017 IEEE 15th Student Conference on Research and Development (SCOReD). IEEE.

[4] Agudelo G.E.R., Parra O.J.S., Velandia J.B. (2018) Raising a Model for Fake News Detection Using Machine Learning in Python. In: Al-Sharhan S. et al. (eds) Challenges and Opportunities in the Digital Era. I3E 2018. Lecture Notes in Computer Science, vol 11195. https://agostini.tech/2017/01/04/implementing-a-stack-using-a-linked-list-data-structure/Springer, Cham

[5] Nltk.org. (2018). Natural Language Toolkit — NLTK 3.4 documentation. [online] Available at: https://www.nltk.org/ [Accessed 12 Dec. 2018].

[6] Scikit-learn.org. (2018). scikit-learn: machine learning in Python — scikit-learn 0.16.1 documentation. [online] Available at: https://scikit-learn.org [Accessed 12 Dec. 2018].

[7] A. Yedidia, Against the F-score.

[8] GitHub. (2018). clips/news-audit. [online] Available at: https://github.com/clips/news-audit [Accessed 12 Dec. 2018].

[9] PolitiFact. (2018). Fact-checking U.S. politics | PolitiFact. [online] Available at: https://www.politifact.com/ [Accessed 12 Dec. 2018].

[10] Media Bias/Fact Check. (2018). Media Bias/Fact Check - Search and Learn the Bias of News Media. [online] Available at: https://mediabiasfactcheck.com/ [Accessed 12 Dec.2018].

[11] C.D. Manning, P. Raghavan and H. Schütze (2008). Introduction to Information Retrieval. Cambridge University Press, pp. 234-265.