News Ranking/Recommendation

Hammad Ali
Samiul Ehsan Chowdhury
University of Stavanger, Norway
hammadd.aali@gmail.com
samiul.ehsan@live.com

ABSTRACT

We are trying to address the problem of news article being popular enough to land in WCEP (Wikipedia Current Event Portal) on the basis of relevant information gained for it from online sources. Recommendation engine have been hot topic for a while and several literature have addressed the topic previously. Among all the approaches that can be found from these works the idea of popularity of news article is still not explored to the fullest. Contributions Incorporated by us in this paper is to find the patterns that involves making a news popular.

State of the art algorithm are able to find the similarity between different articles. While curating the similarity of different articles information is gained which is useful to predict popularity of the article. Different Experimental evaluation have been purposed on the real-world news corpora. Articles in the data set we used to find patterns span to almost 2 years.

KEYWORDS

News recommendation, Popularity, Wikipidea, Classification, Ten-sorflow, WCEP,Similarity, Clustering, Recommendation, sklearn

1 INTRODUCTION

News and newspaper has been a vital source of information for a long time. The amount of news readers have increased exponentially since the providers migrated from print media to digital media. In every minute, news are generated by various news agencies, independent journalists, bloggers etc. As the amount of news providers is growing, all the portals tries to keep user online by providing news that will interest them. They try to do so by relevant news they are reading or news that might be interesting. Popularity of a news event has been appraised by frequency of its reporting by the news medias around. This has been used by many news ranking and recommendation approaches apart from the article contents.

Popular news recommendation system is to build a aggregation that automates news articles and organize them by rank. Most of the news agencies like Google News, Yahoo News use this kind of recommendation of ranking and clustering although there are some portals such as Wikipedia Current News Portal shows articles based on current world events.

Wikipedia Current News Portal has listing of important events such as 'Disasters and accidents', 'Armed conflicts and attacks','Business and economy','Politics and elections'. If any article that is related to these events, will be nominated to be published in the portal. Moreover, although the articles are current news events, the articles with historical importance will have higher probability of submission. For example, articles such as 'Yemen on brink of 'world's worst famine in 100 years' and 'More than 70,000 killed in Yemen's civil war' are related to Yemen crisis. As two articles have historical and ongoing events, they are more likely to be published in the portal.

Generally finding datasets for this type of project requires a lot of text mining and time due the vast amount of new portals.

Our dataset contains articles that are taken from Gdelt.[1] project .Gdelt takes news from thousands of sources from more than 150 countries making it a diverse corpora.Thus, it is crucial to have mixture of events as the unit of ranking that are represented as a cluster of news articles. Hence, getting a mined and modeled reduced the risk unprocessed data.

We started text mining from the given data set and collect information that can be user in our cause.The parameters taken to find the similarity of the news has throughout been the relevance between the topics.

Prediction of WCEP has been done using Built in Neural Network from tensorflow library and random forest, SVM, PCA, Logistic regression using scikit-learn libraries . A brief description of their accuracy and comparison will be discussed later pages.

The major difficulty was to train for the whole data set as only 1 percent of the articles had positive submission in WCEP.

Supervised by Vinay Jayarama Shetty

News/Ranking recommandation (DAT550),
IDE, UiS 2018.

2 BACKGROUND

Probability of successful submission depend many factors. It depends on historical importance, popularity, relation to events.

In our project we consider several algorithms.They are disscussed below.

SVMs are supervised learning models with associated learning algorithms that analyze data used for classification analysis. It is called linear classifier.

Neural network itself is not an algorithm, but rather a framework for many different machine learning algorithms to work together and process complex data inputs.Neural networks are universal approximators, and they work best if the dataset is used to model has a high tolerance to error. However, as articles are published in WCEP as a pattern, neural network can still be used.

Principal component analysis (PCA) is a statistical procedure that uses an orthogonal transformation to convert a set of observations of possibly correlated variables.

Random forest are an ensemble learning method for classification, regression and other tasks that operates by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees.

3 APPROACH

The data set we were provided with had several features related to news articles. All these features were curated already, which are being used as a parameter to find the popularity of the article. To make the data useful for this particular task we had to do required adjustments. Since all the machine learning tasks require the data to be in a certain format, transforming the data to these required pre-sets had to be done carefully without the loss of the information.

The corpus that we worked on was divided into several json files where each file held the news articles of few days. We read all the JSON files and converted them into one single large dataframe so the later computations could be performed easily without losing the information.

3.1 Pre-processing

For pre-processing we convereted the data to the desired shape and remove all the categorical values after mining enough information from them. Since the textual data would not be able to provide any information regarding this task and information required has already been computed we drop the data in form of text. Some Numerical data was present in the dictionary and was contributing in providing information regarding the classification, this data was fetched and appended on the data set as an extra feature.

Since this data was multi-dimensional it was impossible to visualize the data on a 2-D graph. We had to reduce the dimension of the data and for that we used PCA and converted the data into 2 dimensions. Upon visualization we were able to spot some outliers in the data and removed

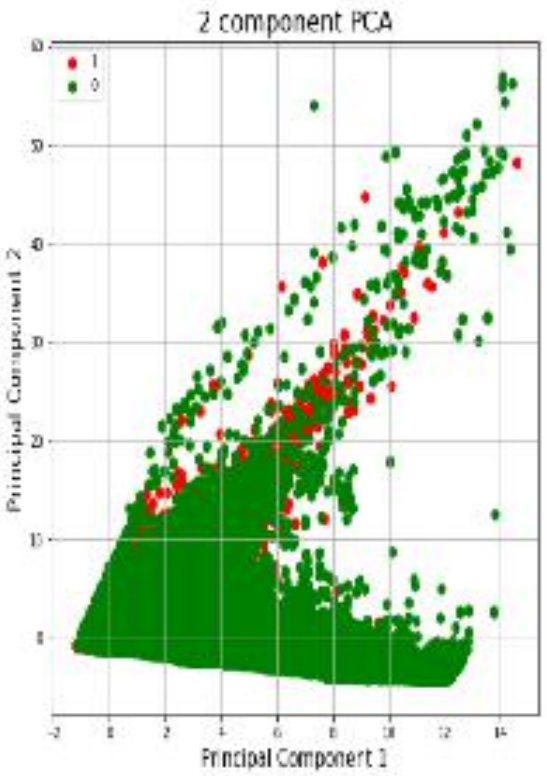

Figure 1: Principal Component Analysis Representation. After removing the outliers completely

them accordingly. We repeated this step several times to completely remove the outliers from the data.

Later we standardized the data using the sklearn library since almost all the machine learning libraries need data to be in standardized form to perform smoothly.

3.2 Visualization

We used the seaborn library to get the hold of data that how it is distributed and which variables contribute the most information to the target variable which we are predicting. Plotting the data as pair and heatmapping the variables provided us with relevant infor-mation of which variables to choose by looking at the correlation among them. As shown in Figure 2

4 MODELS

For the purpose of classification several models have been used and different accuracy have been observed along every model. In this section we will provide brief over view of each model and the accuracy gained along with other statistical inferences which would provide brief over view. Since the values provided for inferring if the article would land in WCEP comprise only 1% of the whole data our approach was to train the model on a small batch since if the model is trained on the full data set it will result in bad prediction due to the fact that data is not divided in a good proportion.

Better results were predicted by taking a chunk of data rather than full data it self. Following this approach we took almost 220000 entries from our data set and applied the models listed bellow

4.1 Random Forest

Since this is a supervised learning algorithm and provides more flexibility with ease of implementation, we started off by classi-fying using random forest. number of estimators were set to 100 for maintaining equilibrium between precision and computation. Stated below are some of the results acquired from the random forest.

Random Forest

	Positive	Negative
Positive	58879	1098
Negative	2348	2656

Using the confusion matrix stated above we can compute the precision, recall and f1-score

	precision	recall	f1-score
0	0.96	0.98	0.97
1	0.71	0.53	0.61

4.2 Support Vector Machine

Being a non-probabilistic linear classifier we tried to find the hyper plane between our classes. Since our data in not linearly separable which can also be inferred from the diagrams above SVM was unable to perform better than the rest of algorithm proposed in this paper Some of the results that we acquired by applying SVM on our subset of data are as follow.

	Positive	Negative
Positive	59774	206
Negative	4365	636

Classification report

	precision	recall	f1-score
0	0.93	1.00	0.96
1	0.76	0.13	0.22

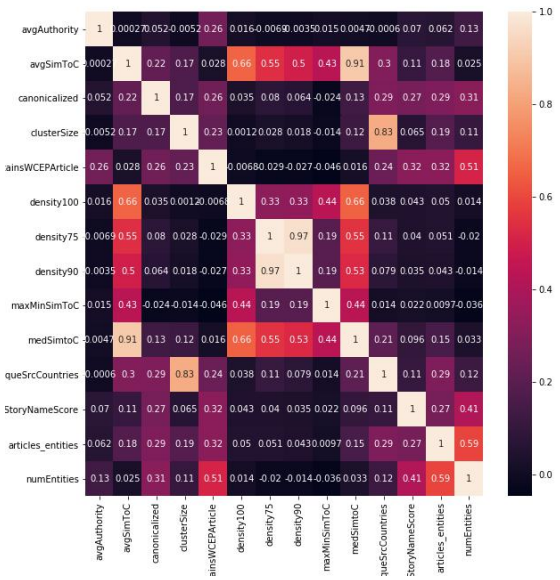

Figure 2: HEATMAP of numerical variables present

4.3 TensorFlow

In the ever evolving world of technology and information we also implemented approaches that are fairly new in the area of computer science. Deep learning model was implemented to learn the weights and find out hidden patterns from the subset we are working on. The model used to predict the out come can be visualized by the looking at Figure 3.

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense_8 (Dense)              (None, 50)                700
_________________________________________________________________
dense_9 (Dense)              (None, 128)               6528
_________________________________________________________________
dense_10 (Dense)             (None, 128)               16512
_________________________________________________________________
dense_11 (Dense)             (None, 1)                 129
=================================================================
Total params: 23,869
Trainable params: 23,869
Non-trainable params: 0
_________________________________________________________________
None
```

Figure 3: TensorFlow Model Summary

TensorFlow

	Positive	Negative
Positive	59247	733
Negative	2788	2213

	precision	recall	f1-score
0	0.96	1.00	0.96
1	0.75	0.44	0.56

4.4 Logistic Regression

Since it is considered as one of the favorite among data scientists when it comes to binary classification. Same amount of data as used for the other Machine learning algorithms was used for this approach as well and the following results were curated.

Logistic Regression

	Positive	Negative
Positive	59399	581
Negative	3392	1609

	precision	recall	f1-score
0	0.95	0.99	0.97
1	0.73	0.32	0.45

5 FUTURE WORKS

The paper is unable to set the precedent to be considered as the benchmark since the generated predictions are fair but not good. Different techniques were thought for the betterment of the algorithm and to overcome the constraints that were faced by it. Among these the best possible solution according to our knowledge, few are discussed below

- Inoculating Ensemble Learning methods would be able to de-crease variance or improve predictions of the algorithm. since one of the major constraint working on this algorithm is that data for the purpose of classification is divided in a much bizarre manner which makes it hard for the algorithm to learn the patterns. Learning different models and combining them might be able to over come this problem
- Accumulate more data for the web and running these algo-rithms would be able to produce decent results, as this technique was mirrored by us as well by taking a small amount of data where the classification parameters were almost equally spread

6 CONCLUSION

Performing the predictions based on the algorithm discussed throughout this paper we conclude that our approach is different from the algorithms suggested that algorithms are performing well when the dataset is evenly distribution of 0 and 1.

However, implementing the algorithms in the whole dataset gives very low accuracy since there are not many positive submission of WCEP.

One possible solution of this problem can be training the algorithm with larger dataset

7 SOURCE CODE

https://github.com/samiul11/assignments/tree/master/miniproject

8 DATASET

The dataset we worked on taken from Gdletproject.org

- 5GB of 340 JSON files.
- Each file contains thousands of articles as rows and 18 columns.
- Target column: **containsWCEPArticle**
- Mixture of categorical and numerical values.

9 REFERENCES

- https://Gdeltproject.org
- https://en.wikipedia.org/wiki/Artificial_neural_network
- https://en.wikipedia.org/wiki/Support-vector_machine
- https://en.wikipedia.org/wiki/Random_forest
- infolab.stanford.edu/~ullman/mmds/ch9.pdf
- https://towardsdatascience.com/building-a-recommendation-system-using-neural-network-embeddings-1ef92e5c80c9
- https://dataaspirant.com/2017/01/25/svm-classifier-implemenation-python-scikit-learn/
- https://towardsdatascience.com/pca-using-python-scikit-learn-e653f8989e60