PROTOTYPE PROJECT REPORT

Submitted By

ARUN KUMAR RANA

FEYNN LABS

2023/07/09

FAKE NEWS DETECTION BUSSINESS PROTOTYPE

Step 1: Prototype Selection

ABSTRACT

Fake news can be interpreted as falsified information or wrong facts presented in the newspapers, television news, online media with a spiteful intent to damage the reputation of a person, organization or any other society for personal or professional benefits. Few media outlets present incorrect narrative about prominent personalities to gain Television Rating Points (TRP's), to increase viewership to yield higher revenue, to smear a person to settle previous conflicts etc. Given the serious consequences that result a fake news article and its impact on formulations of key decisions in everyday life, it is necessary to come up with a resolution plan to stop the spread of fake news and handle it in a suitable manner. In this project, we try to resolve the problem of fake news by utilizing Natural Language Processing (NLP) which is a Data Science sub-branch to come up with a solution that can be used to classify the fake news.

Problem Statement

In variety of fields, including computer science, fake news has become the key research topic. Today's troublesome issue is that press, particularly social media, has become the place for false information that impacts the integrity of entire news environment. This problem is rising because any one can enroll on social media as a news source without any cost (e.g., anyone can create a Facebook page pretending to be a news media organization). There are increasing concerns about fake news outlets publishing "true" news stories, and often adding "fake" followers widely to those news stories. Since the widespread dissemination of fake news can have a significant adverse impact on individuals and society, the lack of robust fact-checking techniques is particularly worrying.

Target Specifications and Characterization

1. **Boost of Sales**: Using this technique, vendors and shopkeepers can visualize which products or items are more profitable for them, so they can focus on which items to buy more for their inventory and which items to buy less according to customer needs so their management of loadout and finance is improved.

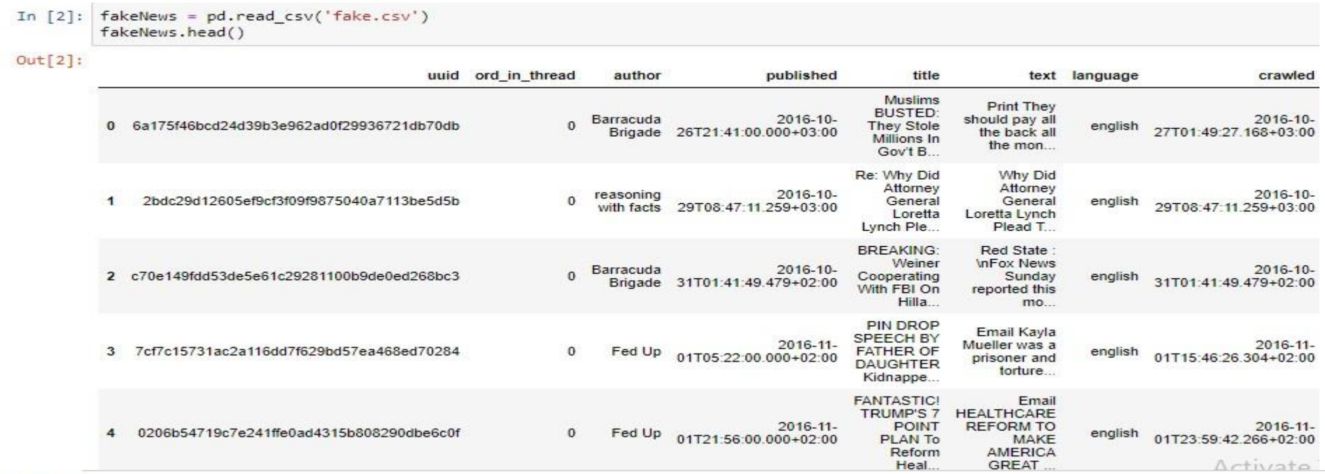
2. **Customer Retention**: With usage of analysis created by models, media and vendors can group certain items together for easier shopping for customers and keep them happy and satisfied as they don't have to look for things bought in combination differently. For eg: keeping best and reliable news closer to each other as they are bought in combination most of the times. For vendors, analysis aims to suggest to them, frequency of bought vegetables and fruits, so they can focus on that cultivation more to increase their sales and maximize their profits.

DATA

The source of this dataset is Kaggle. The dataset consists of text and metadata which is scrapped from 12,999 post on 244 different websites. The dataset has 20 columns and 12999 rows in the dataset where each row corresponds to a new article. For the classification problem under consideration we have used 'title' and 'text' as input columns and column 'type' as output variable. The title, text and type are of string data type. The data is classified into following types - bias, conspiracy, fake, satire, hate, junksci and state. However, in some of the rows some values were missing for some of the features and value for some of the features are 'NaN'. The data was cleaned in the Data pre-processing steps mentioned below, before using it for training different models.

Dataset URL: https://www.kaggle.com/mrisdal/fake-news

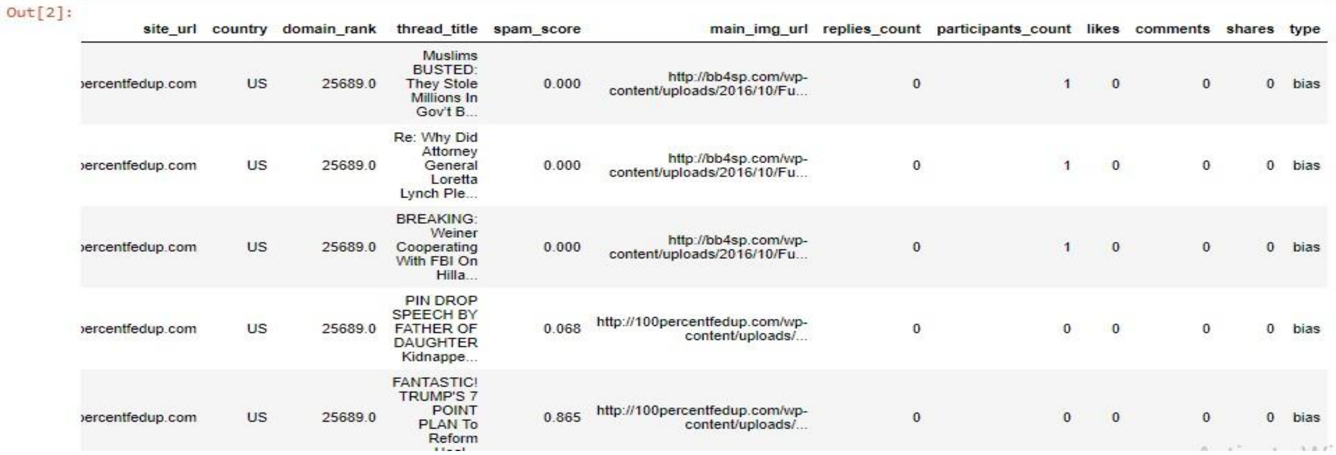
Columns: uuid, ord_in_thread, author, published, title, text, language, crawled, site_url, country, domain_rank, thread_title, spam_score, main_img_url, replies_count, likes, participants_count, comments, shares, type.



```
In [2]: fakeNews = pd.read_csv('fake.csv')
        fakeNews.head()
```

Out[2]:

	uuid	ord_in_thread	author	published	title	text	language	crawled
0	6a175f46bcd24d39b3e962ad0f29936721db70db	0	Barracuda Brigade	2016-10-26T21:41:00.000+03:00	Muslims BUSTED: They Stole Millions In Gov't B...	Print They should pay all the back all the mon...	english	2016-10-27T01:49:27.168+03:00
1	2bdc29d12605ef9cf3f09f9875040a7113be5d5b	0	reasoning with facts	2016-10-29T08:47:11.259+03:00	Re: Why Did Attorney General Loretta Lynch Ple...	Why Did Attorney General Loretta Lynch Plead T...	english	2016-10-29T08:47:11.259+03:00
2	c70e149fdd53de5e61c29281100b9de0ed268bc3	0	Barracuda Brigade	2016-10-31T01:41:49.479+02:00	BREAKING: Weiner Cooperating With FBI On Hilla...	Red State : \nFox News Sunday reported this mo...	english	2016-10-31T01:41:49.479+02:00
3	7cf7c15731ac2a116dd7f629bd57ea468ed70284	0	Fed Up	2016-11-01T05:22:00.000+02:00	PIN DROP SPEECH BY FATHER OF DAUGHTER Kidnappe...	Email Kayla Mueller was a prisoner and torture...	english	2016-11-01T15:46:26.304+02:00
4	0206b54719c7e241ffe0ad4315b808290dbe6c0f	0	Fed Up	2016-11-01T21:56:00.000+02:00	FANTASTIC! TRUMP'S 7 POINT PLAN To Reform Heal...	Email HEALTHCARE REFORM TO MAKE AMERICA GREAT ...	english	2016-11-01T23:59:42.266+02:00



Out[2]:

site_url	country	domain_rank	thread_title	spam_score	main_img_url	replies_count	participants_count	likes	comments	shares	type
ercentfedup.com	US	25689.0	Muslims BUSTED: They Stole Millions In Gov't B...	0.000	http://bb4sp.com/wp-content/uploads/2016/10/Fu...	0	1	0	0	0	bias
ercentfedup.com	US	25689.0	Re: Why Did Attorney General Loretta Lynch Ple...	0.000	http://bb4sp.com/wp-content/uploads/2016/10/Fu...	0	1	0	0	0	bias
ercentfedup.com	US	25689.0	BREAKING: Weiner Cooperating With FBI On Hilla...	0.000	http://bb4sp.com/wp-content/uploads/2016/10/Fu...	0	1	0	0	0	bias
ercentfedup.com	US	25689.0	PIN DROP SPEECH BY FATHER OF DAUGHTER Kidnappe...	0.068	http://100percentfedup.com/wp-content/uploads/...	0	0	0	0	0	bias
ercentfedup.com	US	25689.0	FANTASTIC! TRUMP'S 7 POINT PLAN To Reform	0.865	http://100percentfedup.com/wp-content/uploads/...	0	0	0	0	0	bias

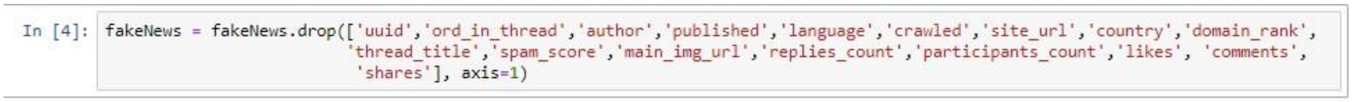
This is a multi-class classification problem.

METHODS

- **Data Pre-processing Steps**

Step 1: Feature Engineering

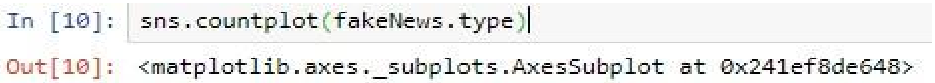
Features considered- Title and Test (Title feature used for handling missing text feature values).



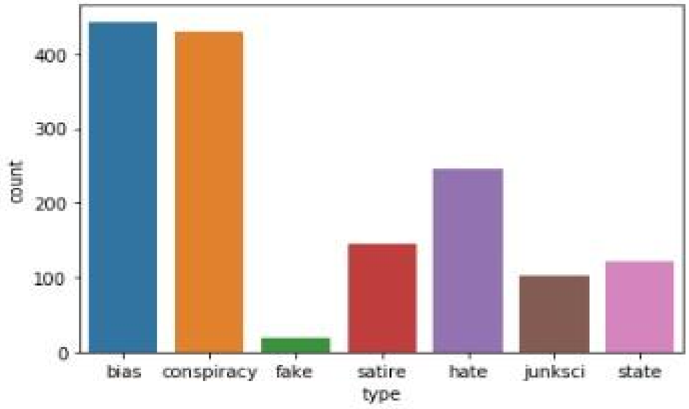
```
In [4]: fakeNews = fakeNews.drop(['uuid','ord_in_thread','author','published','language','crawled','site_url','country','domain_rank',
                                  'thread_title','spam_score','main_img_url','replies_count','participants_count','likes', 'comments',
                                  'shares'], axis=1)
```

Step 2: Data distribution based on output class.

The dataset is imbalanced.



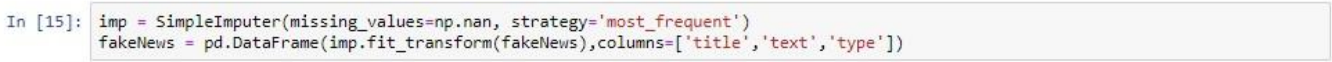
```
In [10]: sns.countplot(fakeNews.type)
Out[10]: <matplotlib.axes._subplots.AxesSubplot at 0x241ef8de648>
```



Step 3: Handling missing text values.

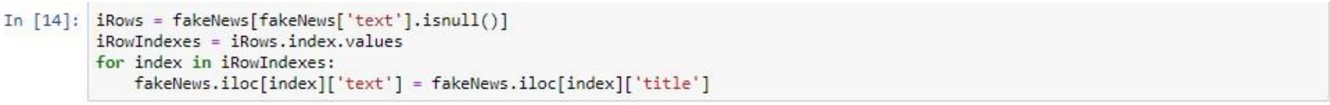
Missing text values can be handled in two ways:

1. Simple Imputing missing values with most frequent strategy



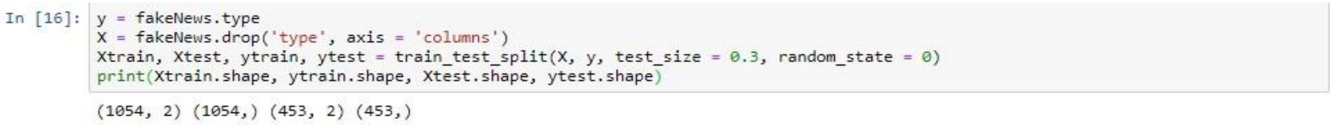
```
In [15]: imp = SimpleImputer(missing_values=np.nan, strategy='most_frequent')
         fakeNews = pd.DataFrame(imp.fit_transform(fakeNews),columns=['title','text','type'])
```

2. Replacing the NaN values of text column with the corresponding title value



```
In [14]: iRows = fakeNews[fakeNews['text'].isnull()]
         iRowIndexes = iRows.index.values
         for index in iRowIndexes:
             fakeNews.iloc[index]['text'] = fakeNews.iloc[index]['title']
```

Step 4: Train Test Split. Keeping aside thirty percent of the data for test set.



```
In [16]: y = fakeNews.type
         X = fakeNews.drop('type', axis = 'columns')
         Xtrain, Xtest, ytrain, ytest = train_test_split(X, y, test_size = 0.3, random_state = 0)
         print(Xtrain.shape, ytrain.shape, Xtest.shape, ytest.shape)

         (1054, 2) (1054,) (453, 2) (453,)
```

Step 5:

a. **Stemming**: Stemming is a method of text standardization(or word standardization) in the Natural Language Processing area that is used to prepare text, words and documents for further processing

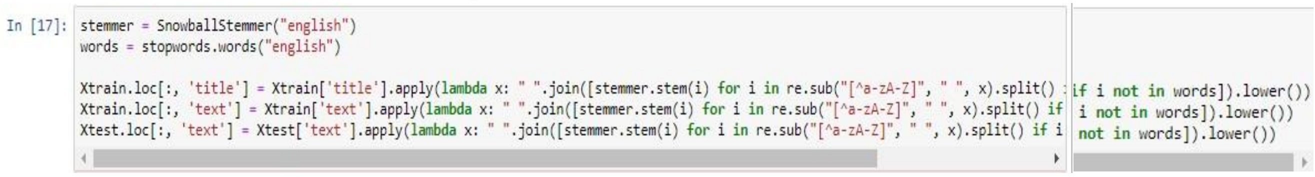
   Stemming is a method of reducing word inflexion to its root forms such as mapping a group of words to the same stem even though the stem itself is not a valid word in the language.

   We use Snowball stemmer for this purpose. This algorithm is also known as the Porter2 stemming algorithm.

b. **Stop words:**

   Stop Words are words used to be used in search queries that do not contain important meaning. These words are typically filtered out of search queries because they return a vast quantity of unnecessary information.

c. Removing numerical and special characters and converting the words to lower case.



```
In [17]: stemmer = SnowballStemmer("english")
         words = stopwords.words("english")

         Xtrain.loc[:, 'title'] = Xtrain['title'].apply(lambda x: " ".join([stemmer.stem(i) for i in re.sub("[^a-zA-Z]", " ", x).split() if i not in words]).lower())
         Xtrain.loc[:, 'text'] = Xtrain['text'].apply(lambda x: " ".join([stemmer.stem(i) for i in re.sub("[^a-zA-Z]", " ", x).split() if i not in words]).lower())
         Xtest.loc[:, 'text'] = Xtest['text'].apply(lambda x: " ".join([stemmer.stem(i) for i in re.sub("[^a-zA-Z]", " ", x).split() if i not in words]).lower())
```

Step 6: Bag of Words

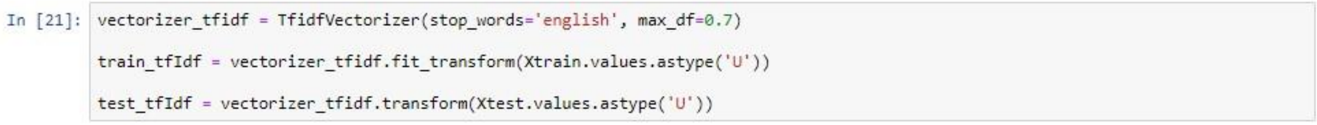
CountVectorizer: Convert a collection of text documents to a matrix of token counts

TfidfTransform: Transform a count matrix to a normalized tf or tf-idf representation

TF-IDF: Vector representation of Text. TF-IDF is an abbreviation for Term Frequency-Inverse Document Frequency and is a very common algorithm to transform text into a meaningful representation of numbers.

Convert a collection of raw documents to a matrix of TF-IDF features.

TfidfVectorizer: Equivalent to CountVectorizer followed by TfidfTransformer.



```
In [21]: vectorizer_tfidf = TfidfVectorizer(stop_words='english', max_df=0.7)

         train_tfIdf = vectorizer_tfidf.fit_transform(Xtrain.values.astype('U'))

         test_tfIdf = vectorizer_tfidf.transform(Xtest.values.astype('U'))
```

We conduct our experiment by implementing the following classification models:

1. Logistic Regression
2. Multinomial Naïve Bayes
3. Random Forest
4. Support Vector Machine
5. XGBoost

Approach:

Creating a model pipeline from the imblearn package. It takes care to upsample using the upsampling method specified in the pipeline. It performs upsampling when fit() is called on the pipeline, and does not upsample test data (when called predict()).

We use RandomizedSearchCV for tuning the hyper parameters. In comparison to GridSearchCV, it uses random combinations of hyper parameters in every iteration and less number of parameter settings are tried, but achieves a better coverage due to randomness and consumes less computational time.

Perform oversampling using SMOTE (Synthetic Majority Oversampling Technique) technique. We are specifying SMOTE upsampling method in the pipeline and performing oversampling as part of parameter tuning using Randomized Search Cross Validation.

For evaluation metrics, we use macro-averages of precision, recall, f1-score, accuracy score and hamming loss for each class. A macro average metric measures the score for each class separately and then takes the average by treating all the classes equally.

For XGBoost, we first encode the output labels using Label Encoder and then convert the dataframe into XGBoost’s Dmatrix object prior to fitting the model. We use XGBClassifier with tuned parameters.

For tuning the hyper parameters, we have used Bayesian Optimization technique as it takes less number of steps in finding the optimal parameters for XGBoost model when compared to randomized search. The computational time is expensive for random search in XGBoost model when compared to other classification models. Bayesian methods use the outcomes of previous assessment results to decide the next values for evaluation.

DISCUSSION:

We can improve the model in the following ways:

1. Adding more data

   Using huge volumes of data to train the models will help improve performance, leading to more accurate models.

2. Enhancing the quality of data by collecting news articles published in various domains as the vocabulary varies with domain. The news articles collected from social media

platform will contain improper words like “awsm, fyn, baaad” etc are not used in news articles by news agencies.

3. Using an Exhaustive Stopword List:

   Apart from language stopwords, there are some other supporting words as well which are of lesser importance than any other terms. These includes: Location stopwords – Country names, Cities names etc. Time stopwords and Numerical stopwords

4. Eliminating features with extremely low frequency:

   There are words that rarely occur in many news articles and these words usually do not play much role in the text classification. Removing features for the words that rarely occur in news dataset can result in improving in the performance for different models.

5. Use Complex Features:

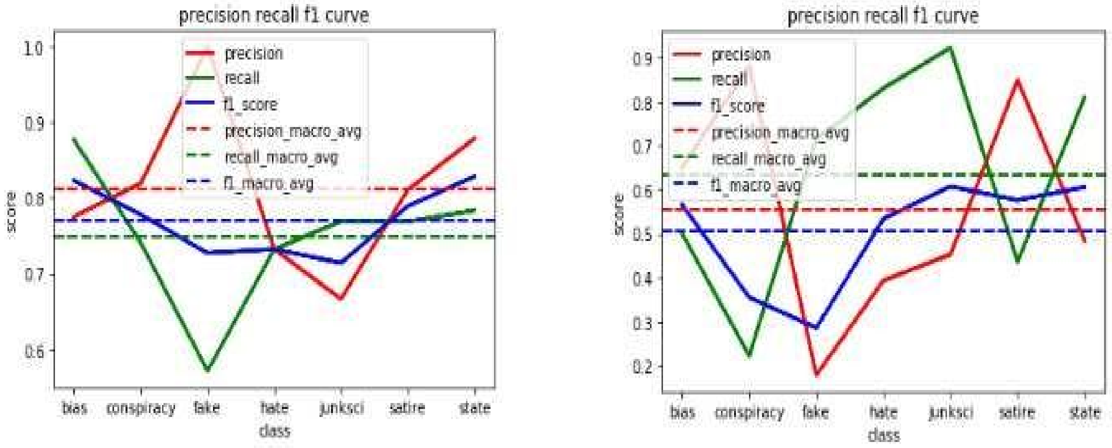
   N-grams and part of speech tags. Joining multiple words together to form a single feature and using these features along with single words for features can help in improving the model accuracy. Combination of N words into a single feature are known N-grams.

6. Using domain specific knowledge and human insight can help in choosing better features for the prediction task under consideration.

7. Multiple Approaches:

   Using different classification algorithms for the news classification task can help in choosing the best solution. Hence, using multiple approaches and comparing their performance on validation dataset can help in choosing the correct approach to solve the classification problem under consideration.

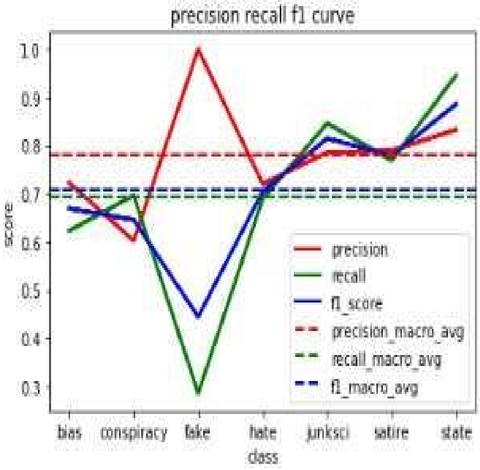
Figure 3: Precision Recall F1 Score



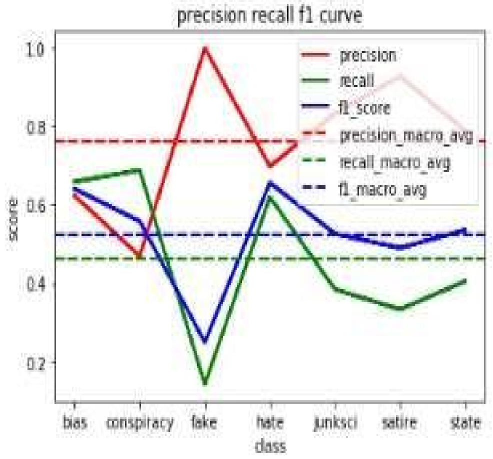
Logistic Regression

Multinomial Naive Bayes Classifier

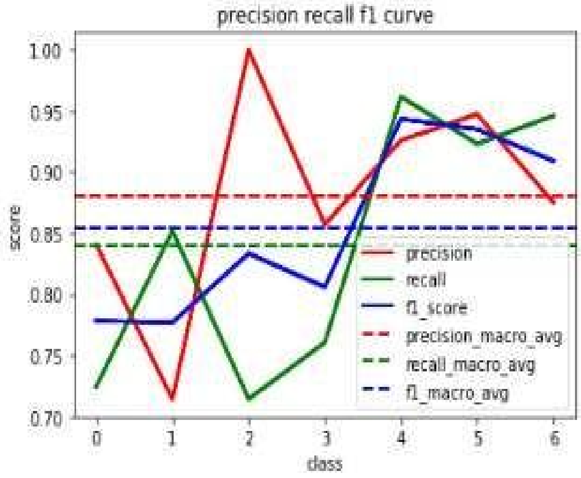
Random Forest Classifier



Support Vector Machine



XGBoost



Direction of Development

At present, the detection of false news and rumors has made good progress, and at the same time, it is also facing some challenges.

First, the lack of universally accepted benchmark data sets must be addressed, especially data sets related to fake news and related social media posts. This is the basis for evaluating the effectiveness of each method and comparing it among them.

In terms of detection technology, they tend to use supervised classification methods to detect false news and rumors. In fact, deep learning technology has achieved the desired results, and the latest methods are dedicated to the development of this framework to some extent. The hybrid model is an important development direction. It can simultaneously model different aspects of fake news.

From a model point of view, the use of semisupervised or unsupervised models will be an important research direction. Their main advantage is that they can learn from unlabeled data, thereby reducing the time cost and accuracy of labeled data.

Business Opportunity

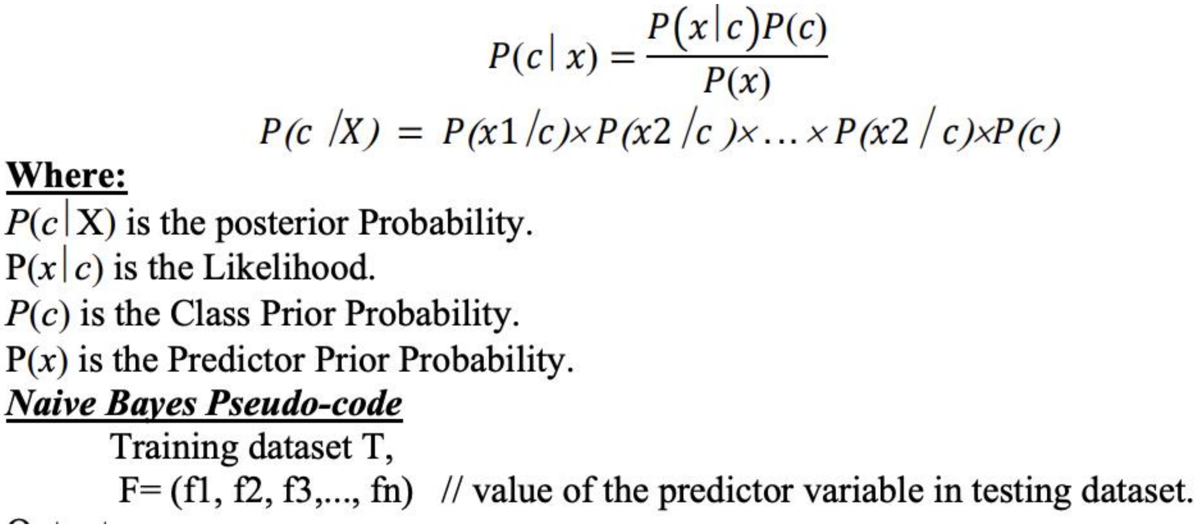
This technique of using Association Rule Mining to group Product Combinations and recommendations is extensively used by larger companies and it is still improving day by day. When small shop owners and vendors start using these techniques, they will not only improve their sales but they will also have an in-depth analysis of what things customers are buying and what they are not buying. That will also help them with maintaining their budget and which will eventually help them increase their reach and have growth in their business.

Final Product Prototype/ Product Details

The final product provides service to operators about the most bought combinations of products for them to analyze customer shopping patterns and helps them manage their inventory and also create new strategies and schemes to increase their appropriate news.The service implements the Market Basket Analysis, i.e Association Rule Mining technique on the dataset of transactions collected from the news providers.

Naive Bayes Equation

Regardless of whether these functions depend on each other or on different functions, and even if these functions depend on each other or on other functions



$$P(c \mid x) = \frac{P(x \mid c)P(c)}{P(x)}$$

$$P(c\ /X) = P(x1\,/c)\times P(x2\,/c\,)\times\ldots\times P(x2\,/\,c)\times P(c)$$

Where:

$P(c \mid X)$ is the posterior Probability.

$P(x \mid c)$ is the Likelihood.

$P(c)$ is the Class Prior Probability.

$P(x)$ is the Predictor Prior Probability.

Naive Bayes Pseudo-code

```
    Training dataset T,
    F= (f1, f2, f3,..., fn)   // value of the predictor variable in testing dataset.
```

REFERENCES

1. Fake News Dataset, https://www.kaggle.com/mrisdal/fake-news

2. Fatemeh Torabi Asr, Maite Taboada, 2019. *Big Data and quality data for fake news and misinformation detection*, https://journals.sagepub.com/doi/full/10.1177/2053951719843310
3. Jillian Tompkins, 2018. *Disinformation Detection: A review of linguistic feature selection and classification models in news veracity assessments,* https://www.albany.edu/~jt972467/tompkins_669final.pdf
4. Miriam Seoane Santos, Jastin Pompeu Soares, Pedro Henriques Abreu, Helder Araujo and Joao Santos, 2018. *Cross-Validation for Imbalanced Datasets: Avoiding Overoptimistic and Overfitting Approaches,* https://www.researchgate.net/publication/328315720_Cross-Validation_for_Imbalanced_Datasets_Avoiding_Overoptimistic_and_Overfitting_Approaches
5. Trevor Hastie, Robert Tibshirani and Jerome Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, Second Edition, 2008.
6. Randomized Search on hyper parameters, https://scikitlearn.org/stable/modules/generated/sklearn.model_selection.RandomizedSearchCV.ht ml
7. Stemming and Lemmatization in Python – by Hafsa Jabeen, https://www.datacamp.com/community/tutorials/stemming-lemmatization-python
8. Imbalanced learn pipeline, https://imbalancedlearn.readthedocs.io/en/stable/generated/imblearn.pipeline.Pipeline.html
9. Metrics and scoring: quantifying the quality of predictions, https://scikitlearn.org/stable/modules/model_evaluation.html
10. Oversampling of imbalanced data, https://www.researchgate.net/post/should_oversampling_be_done_before_or_within_cross-validation
11. Boosting Algorithms Simplified – by Tavish Srivastava, https://www.analyticsvidhya.com/blog/2015/05/boosting-algorithms-simplified/
12. XGBoost Python Package API Reference, https://xgboost.readthedocs.io/en/latest/python/python_api.html
13. Bayesian Optimization Documentation,

    https://github.com/fmfn/BayesianOptimization#bayesian-optimization