CS 521 Final Project Paper:

Team Member: Zongqi Lyu, Yiqing Zhang

Date: Dec 16th, 2021

Source:

- **https://medium.com/analytics-vidhya/randomforest-classifier-vs-multinomial-naive-bayes-for-a-multi-output-natural-language-2426381a5217**
- **https://muthu.co/understanding-the-classification-report-in-sklearn/**
- **https://scikit-learn.org/stable/index.html**
- **https://medium.com/axum-labs/logistic-regression-vs-support-vector-machines-svm-c335610a3d16**
- **https://en.wikipedia.org/**

Introduction:

For this project, we chose the topic: Fake News Detector. The libraries we used including pandas, sklearn as the skeleton libraries, and some other important libraries included in sklearn which will be mentioned later in this paper including SVC, LogisticRegression, RandomForestClassifier and MultinomialNB. The majority purpose for this project is to use the given news data to build and train a model, and then use that model to test the realisticness of the input news, also find some interesting relationships inside each data set.

Making a network request for data:

To request the data from internet is pretty essential thing for data analysis, we have to hardcode all the data to local files then analyze the file if we are unable to dynamically retrieve data from internet. That would be an extremely high cost of both time and money. To retrieve the data from internet, we used the requests library, which allows us to send a http request to the destination web page and getting the corresponding response of it. The code to use the request is simple, by using the get method in requests library we can get the given data from GitHub page and write to the local csv file. This is an important step to start with, since we need to use this dataset to build and train our model.

Using Pandas:

With the local csv file, our next step is to figure out how to use pandas properly. Pandas is a powerful tool to help us achieve our final goal. Pandas can create data frame for the given data, and use that data frame to reshape data including add, edit and delete data for each column or rows, join and merge multiple data sets, filter data, slice data using label and index etc. Pandas can read multiple types of data, including text file, spread sheet, JSON, LaTex etc. So, for our project, we used the read_csv method to read the local csv file which is created by retrieve data from the GitHub repo. Inside the created data frame, we have three columns first we have title column, which is the title of the news, then we have the text column, which is the main content of the news, and finally we have the label column, which indicates whether the news is real or fake.

Using Sklearn:

As soon as we have all the data in our data frame, we can start to train our model. We used different types of methods to train our model, including SVC, LogisticRegression, RandomForestClassifier and MultinomialNB.

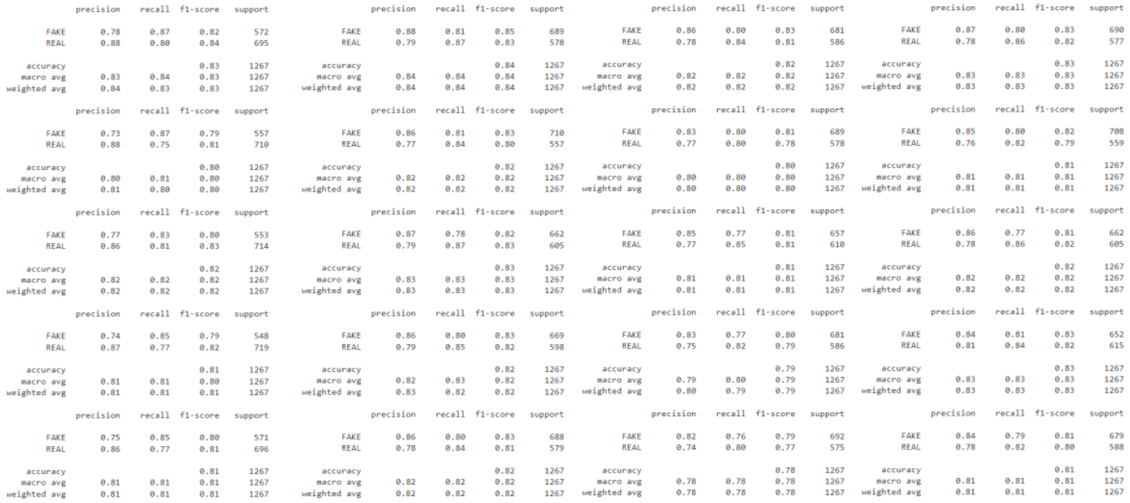
SVC stands for support vector classification, a support-vector machine constructs hyperplane that are in a high-dimensional space. They can be used for various tasks, such as classification and regression. The problem of discriminating sets in a finite-dimensional space is usually not linearly separable. In order to make the separation easier, the original space should be mapped into a higher-dimensional space. The higher-dimensional space hyperplanes are set of points whose dot product has a vector in that space. They are defined by vectors that are orthogonal to each other.

Naive Bayes is a technique for constructing class labels for problem instances. It assumes that the value of a feature is independent of the other feature values indicated by the class variable. Naive Bayes classifiers can be easily trained in a supervised learning environment. In most cases, they can be estimated with the method of maximum likelihood. Despite their apparent oversimplification, naive Bayes classifiers can still work well in complex real-world situations. An analysis in 2004 showed that the use of Bayes classification was not as implausible as previously believed. And in 2006 showed that Bayes classification is outperformed by other approaches, such as boosted trees or random forests.

In various fields, such as machine learning, logistic regression is commonly used. It is also used to develop medical scales and predict the mortality of injured patients. Logistic regression is a technique that can be used to predict the likelihood of a given disease developing. For instance, it could predict the age, sex, race, and other details of a person to determine which political party will win in the next election. It can also be used to predict the likelihood that a customer will purchase a product or a service in the future. In economics, it can be applied to predict the likelihood that a person will enter the labor force.

Random forests are learning systems that are composed of several decision trees that are trained at the same time. For classification and regression tasks, the outputs of the random forests are computed by the trees' individual predictions. Tin Kam Ho developed the random decision forest algorithm in 1995. It was based on the random subspace method. An extension of the Random Forests algorithm was created by Leo Breiman and Adele Cutler. It combines the idea of random selection with the bagging idea. Random forests are often used as black boxes in businesses to generate reasonable predictions across various datasets.

With all 4 classifiers presented, we have showed the classification report for each of them. The classifier used from left to right are MultinomialNB, SVC, RandomForestClassifier and LogisticRegression respectively.



```
              precision    recall  f1-score   support                 precision    recall  f1-score   support                 precision    recall  f1-score   support                 precision    recall  f1-score   support

        FAKE       0.78      0.87      0.82       572           FAKE       0.88      0.81      0.85       689           FAKE       0.86      0.80      0.83       681           FAKE       0.87      0.80      0.83       690
        REAL       0.88      0.80      0.84       695           REAL       0.79      0.87      0.83       578           REAL       0.78      0.84      0.81       586           REAL       0.78      0.86      0.82       577

    accuracy                           0.83      1267       accuracy                           0.84      1267       accuracy                           0.82      1267       accuracy                           0.83      1267
   macro avg       0.83      0.84      0.83      1267      macro avg       0.84      0.84      0.84      1267      macro avg       0.82      0.82      0.82      1267      macro avg       0.83      0.83      0.83      1267
weighted avg       0.84      0.83      0.83      1267   weighted avg       0.84      0.84      0.84      1267   weighted avg       0.82      0.82      0.82      1267   weighted avg       0.83      0.83      0.83      1267
              precision    recall  f1-score   support                 precision    recall  f1-score   support                 precision    recall  f1-score   support                 precision    recall  f1-score   support

        FAKE       0.73      0.87      0.79       557           FAKE       0.86      0.81      0.83       710           FAKE       0.83      0.80      0.81       689           FAKE       0.85      0.80      0.82       708
        REAL       0.88      0.75      0.81       710           REAL       0.77      0.84      0.80       557           REAL       0.77      0.80      0.78       578           REAL       0.76      0.82      0.79       559

    accuracy                           0.80      1267       accuracy                           0.82      1267       accuracy                           0.80      1267       accuracy                           0.81      1267
   macro avg       0.80      0.81      0.80      1267      macro avg       0.82      0.82      0.82      1267      macro avg       0.80      0.80      0.80      1267      macro avg       0.81      0.81      0.81      1267
weighted avg       0.81      0.80      0.80      1267   weighted avg       0.82      0.82      0.82      1267   weighted avg       0.80      0.80      0.80      1267   weighted avg       0.81      0.81      0.81      1267
              precision    recall  f1-score   support                 precision    recall  f1-score   support                 precision    recall  f1-score   support                 precision    recall  f1-score   support

        FAKE       0.77      0.83      0.80       553           FAKE       0.87      0.78      0.82       662           FAKE       0.85      0.77      0.81       657           FAKE       0.86      0.77      0.81       662
        REAL       0.86      0.81      0.83       714           REAL       0.79      0.87      0.83       605           REAL       0.77      0.85      0.81       610           REAL       0.78      0.86      0.82       605

    accuracy                           0.82      1267       accuracy                           0.83      1267       accuracy                           0.81      1267       accuracy                           0.82      1267
   macro avg       0.82      0.82      0.82      1267      macro avg       0.83      0.83      0.83      1267      macro avg       0.81      0.81      0.81      1267      macro avg       0.82      0.82      0.82      1267
weighted avg       0.82      0.82      0.82      1267   weighted avg       0.83      0.83      0.83      1267   weighted avg       0.81      0.81      0.81      1267   weighted avg       0.82      0.82      0.82      1267
              precision    recall  f1-score   support                 precision    recall  f1-score   support                 precision    recall  f1-score   support                 precision    recall  f1-score   support

        FAKE       0.74      0.85      0.79       548           FAKE       0.86      0.80      0.83       669           FAKE       0.83      0.77      0.80       681           FAKE       0.84      0.81      0.83       652
        REAL       0.87      0.77      0.82       719           REAL       0.79      0.85      0.82       598           REAL       0.75      0.82      0.79       586           REAL       0.81      0.84      0.82       615

    accuracy                           0.81      1267       accuracy                           0.82      1267       accuracy                           0.79      1267       accuracy                           0.83      1267
   macro avg       0.81      0.81      0.80      1267      macro avg       0.82      0.83      0.82      1267      macro avg       0.79      0.80      0.79      1267      macro avg       0.83      0.83      0.83      1267
weighted avg       0.81      0.81      0.81      1267   weighted avg       0.83      0.82      0.82      1267   weighted avg       0.80      0.79      0.79      1267   weighted avg       0.83      0.83      0.83      1267
              precision    recall  f1-score   support                 precision    recall  f1-score   support                 precision    recall  f1-score   support                 precision    recall  f1-score   support

        FAKE       0.75      0.85      0.80       571           FAKE       0.86      0.80      0.83       688           FAKE       0.82      0.76      0.79       692           FAKE       0.84      0.79      0.81       679
        REAL       0.86      0.77      0.81       696           REAL       0.78      0.84      0.81       579           REAL       0.74      0.80      0.77       575           REAL       0.78      0.82      0.80       588

    accuracy                           0.81      1267       accuracy                           0.82      1267       accuracy                           0.78      1267       accuracy                           0.81      1267
   macro avg       0.81      0.81      0.81      1267      macro avg       0.82      0.82      0.82      1267      macro avg       0.78      0.78      0.78      1267      macro avg       0.81      0.81      0.81      1267
weighted avg       0.81      0.81      0.81      1267   weighted avg       0.82      0.82      0.82      1267   weighted avg       0.78      0.78      0.78      1267   weighted avg       0.81      0.81      0.81      1267
```

To understand the table, we need to understand that there are 4 different ways to check if the predictions are right or wrong: TN for True Negative which means the predicted result is negative and the case is truly negative, TP for True Positive which means the predited result is positive and the case is truly positive, FN for False Negative which means the predited result is negative but the case is actually positive, FP for False Positive which means the predited result is positive but the case is actually negative. The precesion column means what percentage of the predict result are correct, which is calculated by using TP/(TP + FP). The recall column means what percentage of the positive case did the predict result catch. The F1 score indicates the percent of positive predictions were correct. There are 5 results for each classifier is because we have diveded the data into 5 groups. As the result shown in the table, we can say that with this perticular data set, SVC is the best method since it has average of 0.82 on precesion and recall.

Using model test random news on internet:

After the last step, we have a brief overview of each classifier, so the next step is to test our models. For the testing data, we used a news API to retrieve the new dynamically from internet. Which is done by using the news API configuration, so we can edit the API URL based on the area we are interested. We have tested 4 groups of news in total, the first group is the news from Apple, second group is the news from tesla, third group is tech news, and fourth group is business news. Although the results of each group are varied, but they only differs in 1 or 2 news.

Conclusion:

After building, training and testing the model, we noticed that all the 4 methods we used are working perfectly, based on the model given, it can predict the news quickly and precisely.