Fake News Detection

Stop the spread of misinformation with fact-checking

MIE1624 – Introduction to Data Science and Analytics

University of Toronto – Fall 2019

Course Project

news_Buster ©

Gurtej Bhasin - 1001103940

Elise Lagacé – 1006598530

Alex Kwan – 1001559057

Jacob Bulir – 1002260698

Hao (Aaron) Tan – 999735728

November 28th, 2019

1. Introduction

The "fake news" issue is currently a hot topic since the 2016 U.S. presidential elections, but has been a problem in the business world for much longer. Incorrect news stories can not only cause plummeting stock prices or destroy the reputation of a business; it can also cause unreasonable customer expectations and lead to substantially weaker customer engagement. There are many reasons unethical businesses would generate fake news or reviews: 1) in order to boost their own stature or profits, 2) to create widely shared online content for generating more ad revenue/web traffic 3) discrediting a public figure, political movement, company, etc. [1]

Definition of Fake News: A false, or partly false story, advertised as factual, usually sensational in nature.
How can we detect it and prevent its spread of misinformation – protecting your business and customers? The purpose of this algorithm is aims to address this growing concern – the detection and labelling of fake news, true news and partly true news.

1.1. Objective & Scope

The team's goal was to develop an artificial intelligence algorithm that can rate a claim as TRUE, PARTLY TRUE or FALSE without human intervention [2]. The input to the algorithm is a claim, along with supporting information, such as who made it, when, supporting articles, etc. The output of the algorithm is a "truth rating" that indicates whether the claim is true (rating of 2), partly true (rating of 1) or false (rating of 0).

2. Exploratory Data Analysis

An important part of developing a good algorithm start with properly understanding the data. Exploratory data analysis (EDA) is done to recognize patterns and understand potential trends between key features from the claims and their truth ratings. Visualizing key features can help in the process of feature selection and provide a basis for refining the algorithm. The following figures describe key trends found.

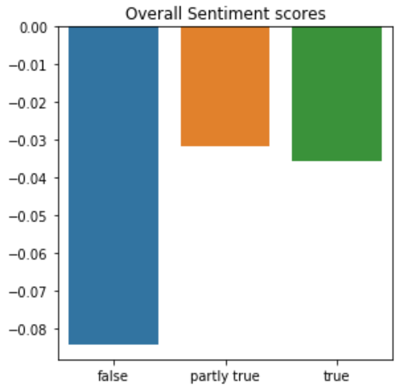

Figure 1: Overall mean sentiment for each claim truth rating

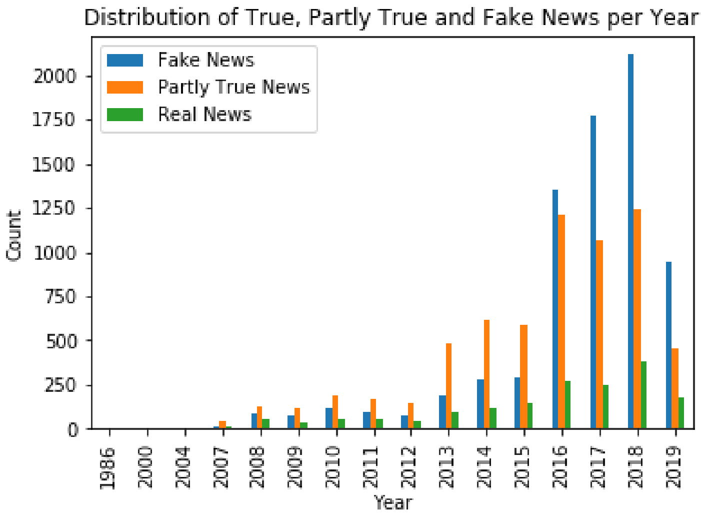

Figure 2: Distribution of Truth Ratings per Year

2.1. Observations from EDA

Through EDA, we were able to visualize major trends in the data. The major trends we discovered are as follows:

- There is some correlation between a negative sentiment and a "false" claim and almost neutral sentiment and "true/partly true" claims. The sentiment scores for related articles were also explored, however these did not results in any significant trends.
- The fraction of truth rating between claims with claimants as non-politicians and politicians is equivalent. When looking at the top 15 claimants, we see more a correlation between false claims and claims made by bloggers, various websites and Facebook posts/users. This indicated there is potentially a good correlation between the

truth rating and the claimant. Due to the obvious trends here, we explored clustering claimants in various ways to create additional claimant features.

- There is commonality between the most frequent words appearing in each truth rating (see Word Clouds in Appendix), meaning there are no obvious words that directly correlate to the truth rating. However, a collection of words or other features from the words in the claims may be good features for predicting truth rating. We explored these parts of speech features further in Feature Engineering.
- There is no significant trend between the truth rating and month the claim was made, whoever, there is a jump in “false” claims made starting in 2016, where the number of false claims surpasses partly true and true claims significantly.

From these findings, we created new features by clustering common features in multiple ways and creating new features to further explore other trends. These will be explored in Section 3.2 – Feature Engineering. Additional EDA figures can be found in Appendix 7: Additional EDA.

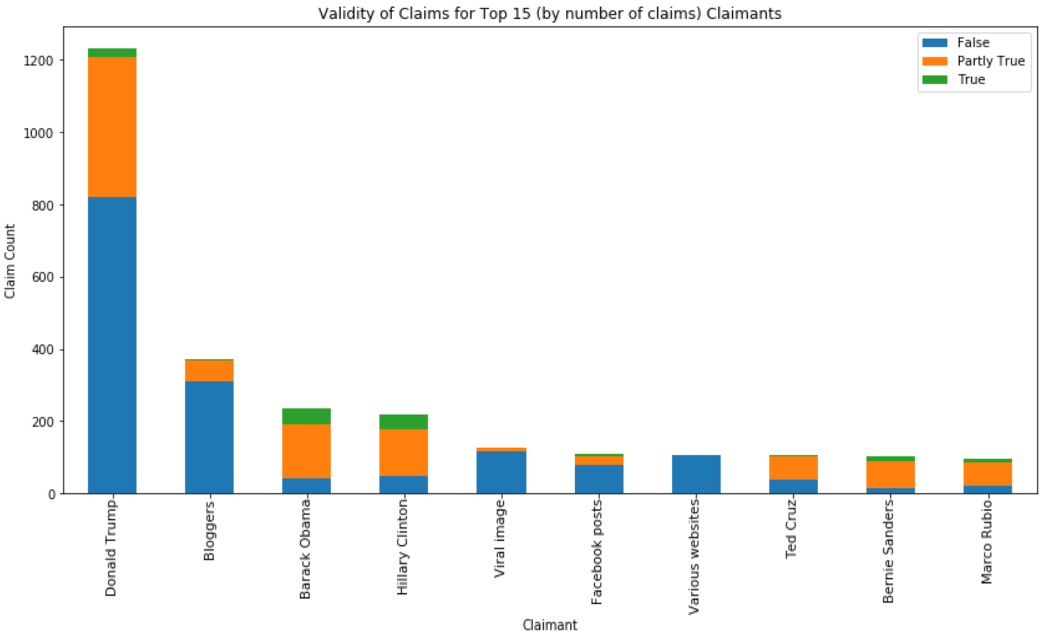

Figure 3: Validity of Claims made by the top 15 claimants

3. Methodology

The algorithm was developed using sample claims that were already associated to a known truth rating. The steps below were followed to develop our “Fake News Detection” algorithm.

3.1. Data Cleaning

The text data and other features in the dataset must be cleaned in order for the machine learning algorithms to perform properly. The following steps were the major tasks in data cleaning:

- Related article text was imported into the dataset to be incorporated as a feature in the analysis
- Claims and Related Articles text were cleaned by removing punctuations, Unicode hex, white space, URLs, stop words and other items that are not words – everything was made lowercase and stemmed to the root of each word
- Dates were converted to the proper date format
- Any missing claimant cell is replaced with “unknown”
- The “ID” column (features) was removed from the dataset as it is not useful for the model; any rows missing claims or related articles are also removed because they do not contain enough information for proper processing.

At the end of this step we are left with standardized text, simple common words, for both claims and related articles from which we can create additional features for our model.

3.2. Feature Engineering

Algorithms require features with some specific characteristic to work properly. Feature engineering consists of preparing proper input data that provide meaningful information from the input data about the outputs, which aims to improve the performance of the model. Algorithms are unable to directly interpret text or derive semantic meaning. In our case, knowing the importance of strong features, we spent a considerable amount of time creating new features to describe the data meaningfully. We created the following features based on our insights from EDA:

Feature Category	Considerations	Feature Created
Date Features	Since we saw some correlation with the data and the truth rating in EDA, we further develop "date" features to explore the following ideas: • Is there a correlation between truth rating and day of the week (pay days, days more news is published)? Weekends vs. weekdays? Seasons? • Consecutive claim days (does the truth rating change after a few days of consistent claims?). We aimed to understand if different clusters of dates/time features help reveal additional trends.	➢ Raw dates (original data) ➢ Month the claim was made ➢ Year the claim was made ➢ Days of the week/month the claim was made ➢ Consecutive days ➢ Weekday vs. Weekend ➢ Seasons
Claimant Features	We also saw significant trends between the claimant and the truth rating, as expected. Since this is also an important feature, we developed additional "Claimant" features to explore the following: • Do politicians result in more fake news or not? • Do individuals vs. organizations results in specific truth ratings – for example do websites, social media and bloggers spread more fake news • Does gender of the individual contribute to truth rating?	➢ Raw Claimant names (original data) ➢ Individual vs. Group/Organization ➢ Gender of claimant ➢ Politicians vs. non-politicians
Similarity Scores	We wanted to see if there was a correlation between the similarity of the claim to its related articles and its truth rating. We calculated similarity scores between the claim and every single sentence of all related articles. From this, we extracted the top 5 most similar sentences to add to the dataset as text features. The similarity scores were also added as features. A high score meant the claim was very similar to what was said in the article.	➢ Top similar related article sentences (text) ➢ Similarity scores between claim and related articles
Sentiment Scores	There was some trend between sentiment scores and claim truth rating. Using existing sentiment analysis packages, we calculated the sentiment score (positive, neutral, negative and overall) for each claim, this is based on the number of positive, negative and neutral words that show up in the text. We used all sentiment types in the analysis for both claims and related articles.	Positive, neutral, negative and overall sentiment scores for: ➢ Claims ➢ Related articles
Parts of Speech	We were also curious to see if the claim text contained any features that would correlate to truth rating: • Do more punctuations, adjectives and special characters indicate "false" claims? • Does the number of words per claim correlate the truth rating? When people lie don't they tend to use more descriptive words? • Do stop words correlate to truth rating? • If the related articles frequently refers to words stated in the claim – does that result in a more truthful claim?	➢ Weighted sum of overlapping words between claim and related articles ➢ Number of words per claim ➢ Number of stop words per claim ➢ Number of special characters per claim ➢ Number of character per claim ➢ Number of nouns, verbs, adjectives and punctuations in each Claim

Once all features were obtained, seeing as they are all on a different scale and some are categorical in nature, all features were encoded and/or scaled to fit between 0 and 1. Many machine learning algorithms perform better or converge faster when features are on a relatively similar scale and/or close to normally distributed. This allows the model to understand the categorical data without introducing errors due to magnitude of data.

3.3. Feature Selection

The purpose of this step was to determine the features that are most significant in predicting the truth rating of the claim. Only those important features will be used to build the machine-learning algorithm, significantly improving its performance. Irrelevant or partially relevant features can negatively impact model performance.

We used Extra Tree Classifier to get the feature importance score of each feature, using the “feature importance” property of this classifier. Feature importance gives you a score for each feature: the higher the score more important or relevant is the feature towards the output variable (i.e truth rating). With that score, only the features above mean importance score are selected and kept. The number of features drops from 7100 to approximately 996 features, greatly simplifying our data to the most important elements.

The most important features for predicting claim truth ratings are: claimant group (individuals), claimant gender – male/female, specific years, specific claimants (names), sentiment scores and some claim/related article words. These features and their associated correlation with the truth rating can be found illustrated in Appendix. The top features do align well with the trends observed in EDA. Most of the top features have been engineered and are not inherent in the data, meaning our feature engineering was a necessary endeavor for the performance of the algorithm.

3.4. Algorithm Set-up

The algorithm used to predict our output is an “Ensemble of models”, which improves machine learning by combining multiple models. In our case we used Logistic Regression, K-NN, Decision Trees Classifier and Support Vector Machines (SVM) as our models, each predicting a truth rating for each claim. These models were selected based on their historical performance for classifying data.

The “Ensemble” works as follows:

- Each model is tuned (optimized) using Gridsearch, a common hyperparameter tuning method. In our case we are only tuning one hyperparameter per model due to the large cost associated with tuning: for logistic regression, we tuned learning rate; for k-NN the number of neighbors; for Decision Trees, the max depth; and for SVM, number of cross validation folds. More details on models selected and hyperparameters can be found in Appendix or in the Jupyter Notebook.
- For this step we create 10 models, including the optimal model, for each type of model (Logistic, k-NN, Decision Trees and SVM) resulting in 40 models total, each providing a prediction for each claim.
- The outputs from each model are compared and the most frequent prediction is selected as the final prediction. The output from the optimal (tuned) models are weighted more heavily than the other outputs in this comparison.

The output from the ensemble model is the best prediction of the truth rating for a claim.

4. Results

Once the ensemble of models is training on a training set, it is then applied to a test dataset to determine its performance on un-seen data. The results and algorithm performance are illustrated below.

Overall the algorithm performs adequately, with an accuracy of 63%. The algorithm is better at detecting False and Partly True labels, making it conservative, preferential to labelling “False” or “Partly True” over “True”. This ensures that claims that are not “true” would not easily make it through. As you can see, model implementation takes a long time, and this can be reduced significantly by only using optimized models and avoiding tuning every time new data is introduced.

Performance Metric	***Algorithm Score***
Accuracy	63%
Precision	70%
Recall	47%
f1 Score	45%
Run time (cost)	85min (Data Cleaning); 400 min (Model Implementation)

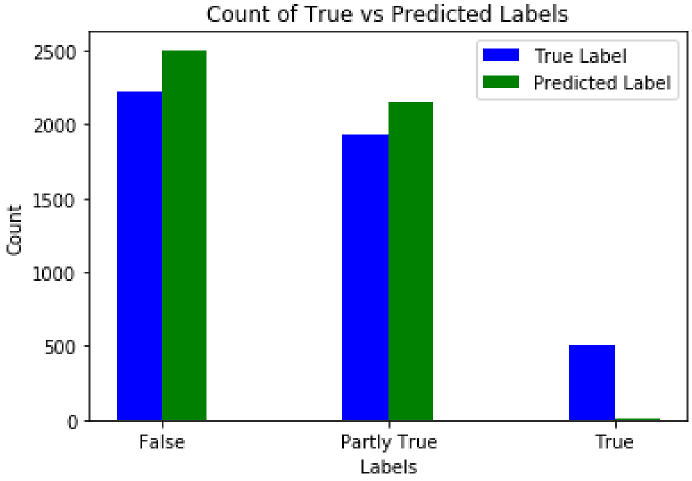

Figure 4: Algorithm Results comparing true versus algorithm predicted truth ratings

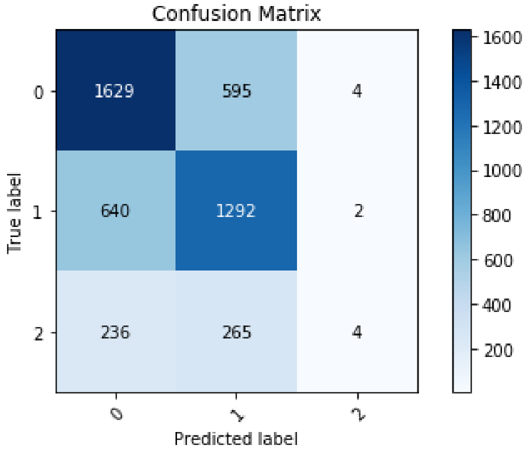

Figure 5: Algorithm Results Confusion Matrix illustrating numerical values of true versus predicted truth ratings

5. Conclusions

As discussed, the stakes are high for ensuring your content remains factually accurate. Unfortunately, current content-management algorithms are designed to maximize user engagement but may inadvertently promote content of dubious quality—including fake news. Adding this "fake news detection" algorithm to your content management algorithm could significantly reduce the number of fake content shared or contained on your platform, by at least 63%. Even with this performance, the algorithm can at least distinguish well between fake and true news, providing some barrier against fake news making it onto your platform, a step in the right direction.

The benefits of limiting fake news are vital to your long-term success: 1) improved customer trust in your content, which leads to increased traffic/engagement; 2) higher rates of other sites, platforms sharing your content, further increasing your visibility and finally 3) stable/growing and robust reputation and market value.

For a technical review of our algorithm and methodology, please refer to the jupyter notebook, which contains explanations of each step taken throughout the development process.

5.1. Further Work

If you choose to adopt this algorithm to manage your content, we have the following suggestions to further improve performance and increase the accuracy and performance of the algorithm for your purposes:

- We can optimize the models to improve precision over accuracy, to ensure returns only relevant true news instances and that no false positives sneak through (fake news categorized as true/partly true news).
- We can perform more hyperparameter tuning for each model on multiple parameters to end up with the true optimal models.
- Seeing as most of the top features are engineered features, and not from the raw dataset, it may be worthwhile to develop additional features. Examples to explore are social media sites vs. non-social media, assigning a credibility rating to the claimants and trying different types of word embedding methods for text features. One important feature to try to engineer is populating the name of missing claimants, seeing as a few claimant features are some of the most important features.
- Adding a neural net model to our ensemble, such as Long Short Term Memory Networks (LSTM) that takes into account the ordering of words in a sentence, as opposed to just the words themselves.

We look forward to discussing further developments and optimizing this algorithm further to suit your needs.

6. References

[1] Madison College Libraries. Fake News: Why is Fake News Created? https://libguides.madisoncollege.edu/fakenews (accessed November 16, 2019).

[2] DataCup. Leaders Prize: Fact or Fake News? https://datacup.ca/main/competitions/leadersprize2019/about (accessed October 20, 2019).

[3] Company Logo Reference: https://www.stockunlimited.com/vector-illustration/brain-shape-circuit_1648303.html

7. Appendix: Additional EDA

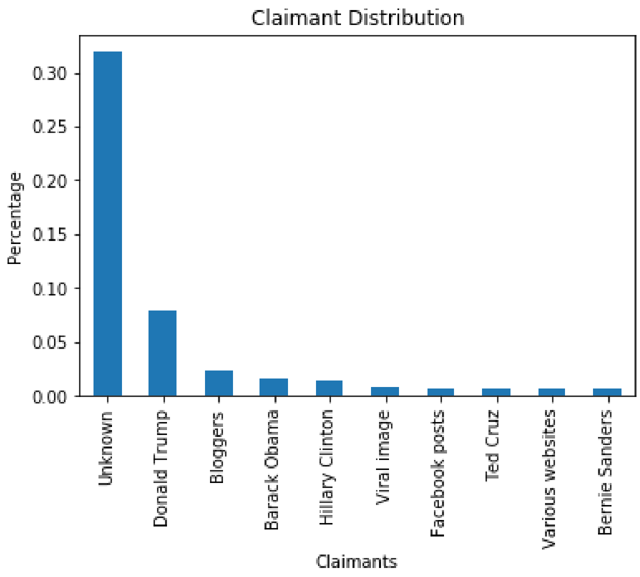

Figure 6: Distribution of original claimant names

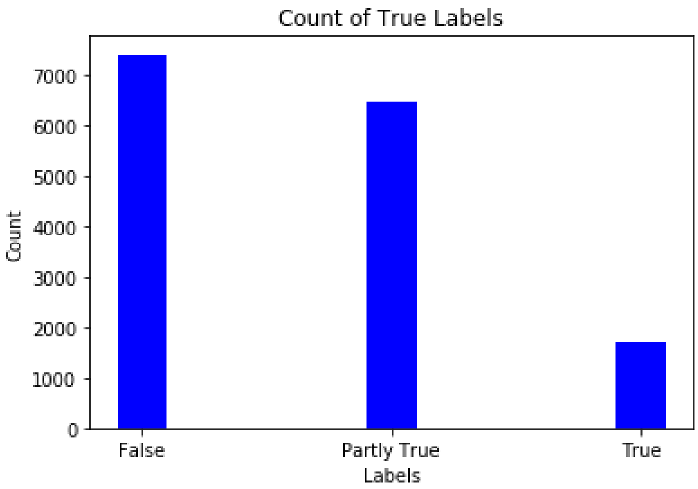

Figure 7: Raw count for Truth Ratings

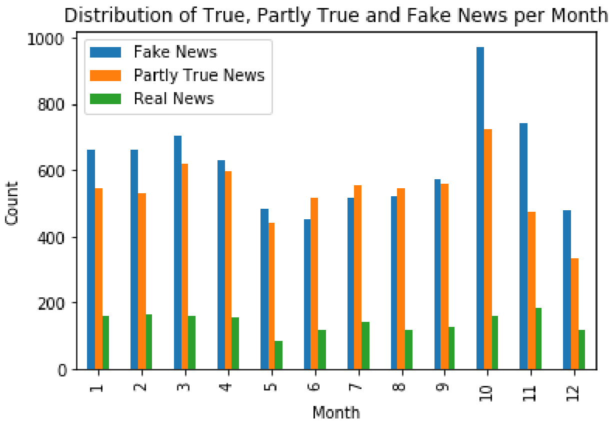

Figure 8: Distribution of truth rating per month

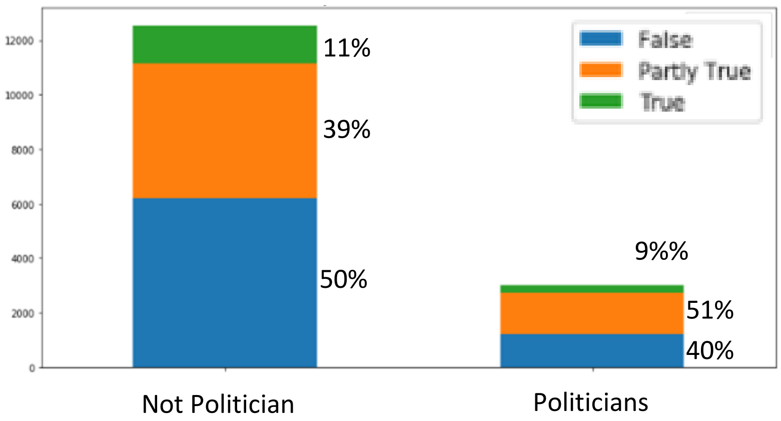

Figure 9: Validity of claims made by politicians versus non-politicians

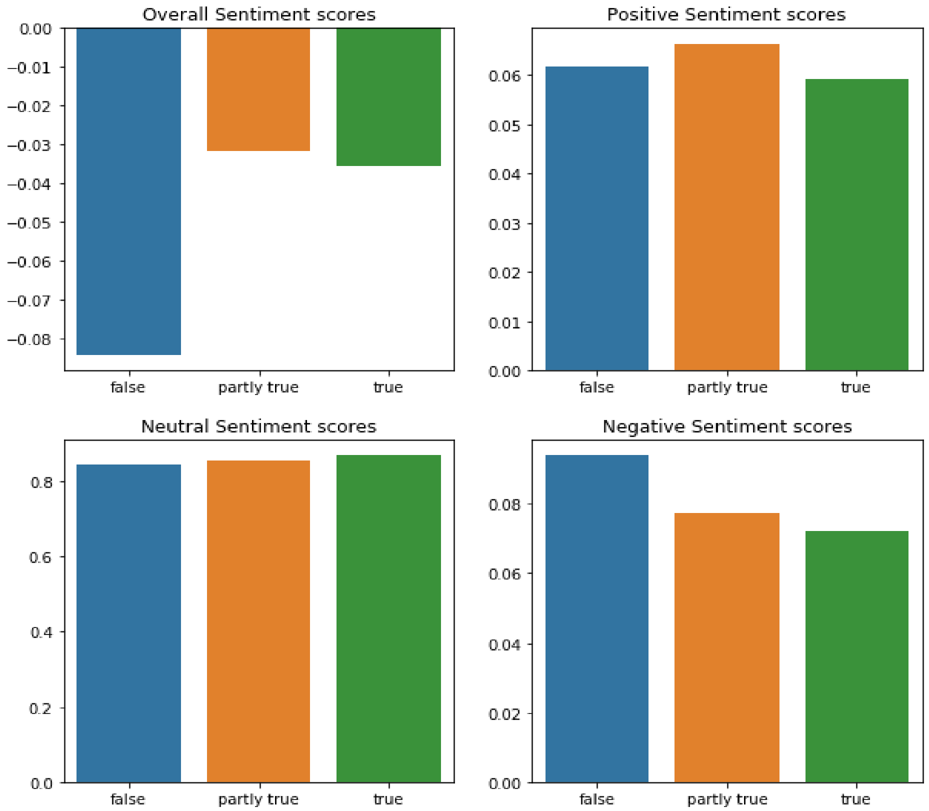

Figure 10: Results of Sentiment Analysis on the claims using the "Vader" python package. Illustrating the trends between the claim sentiment and its truth rating.

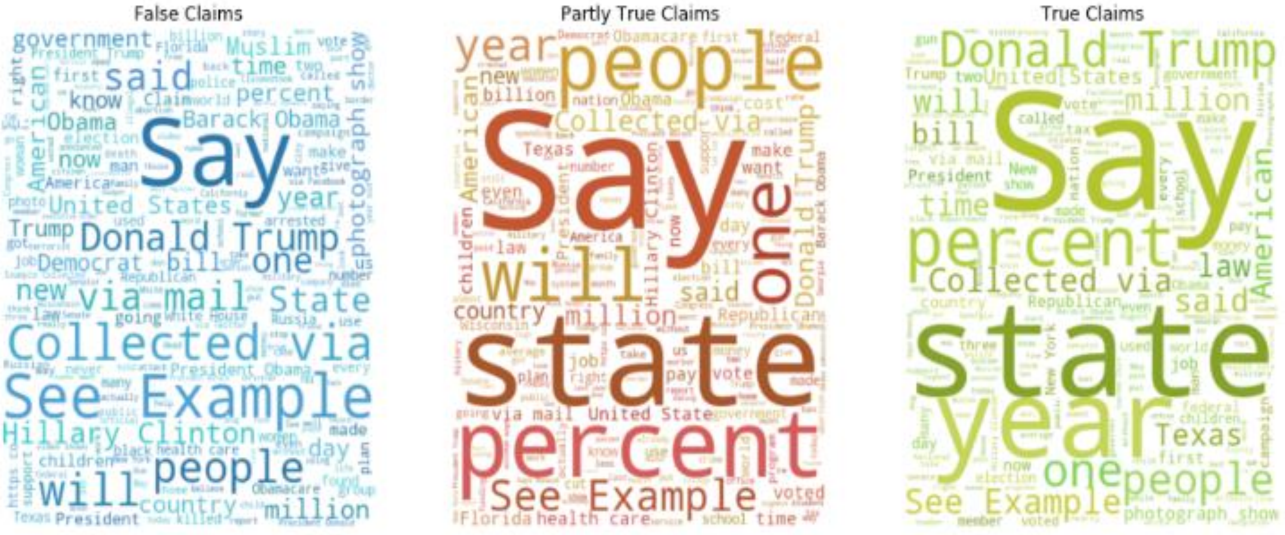

Figure 11: Word cloud (most frequent words) for each claim type

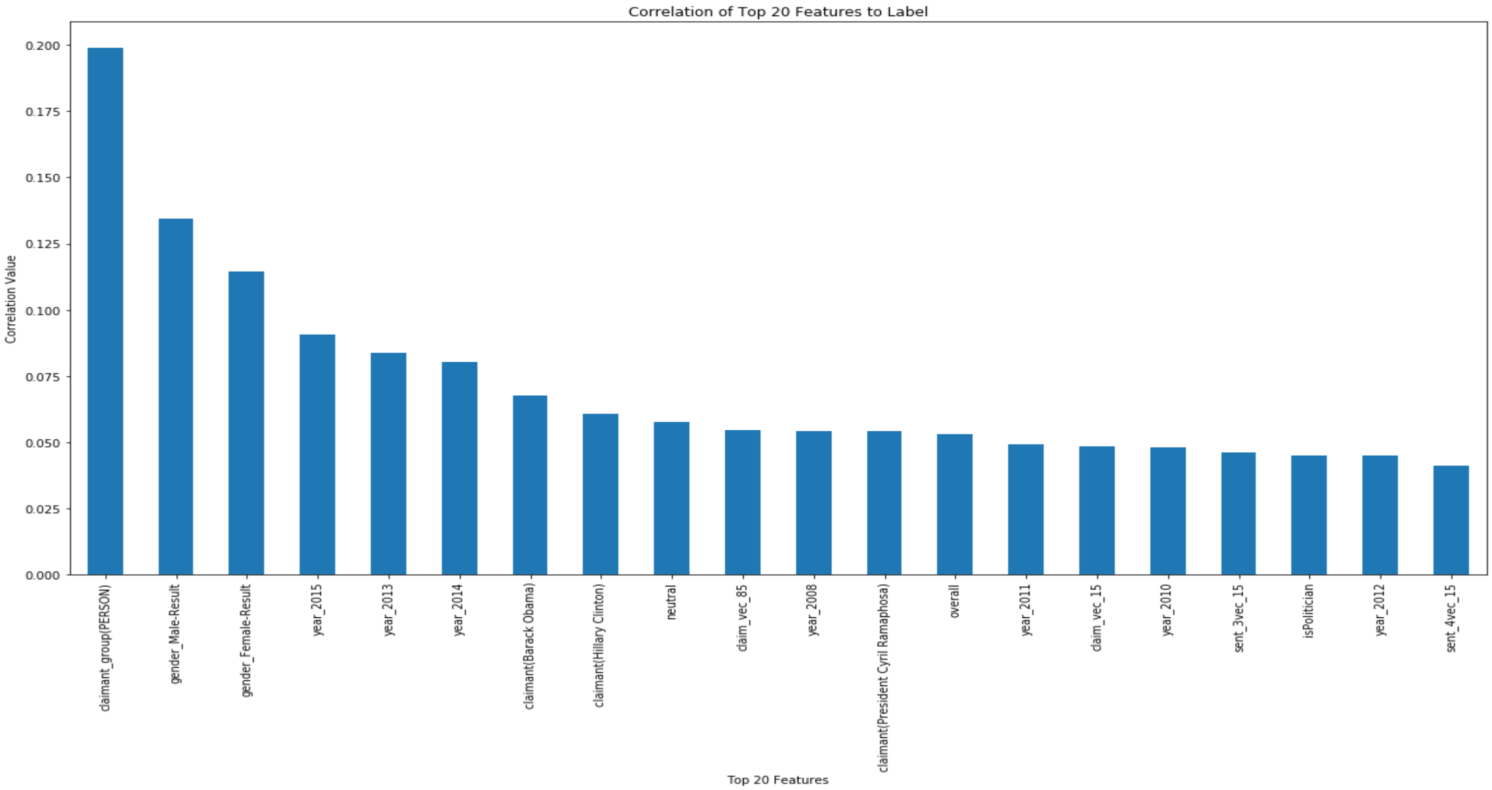

Figure 12: Correlation value between the top 20 features and the truth rating (output)

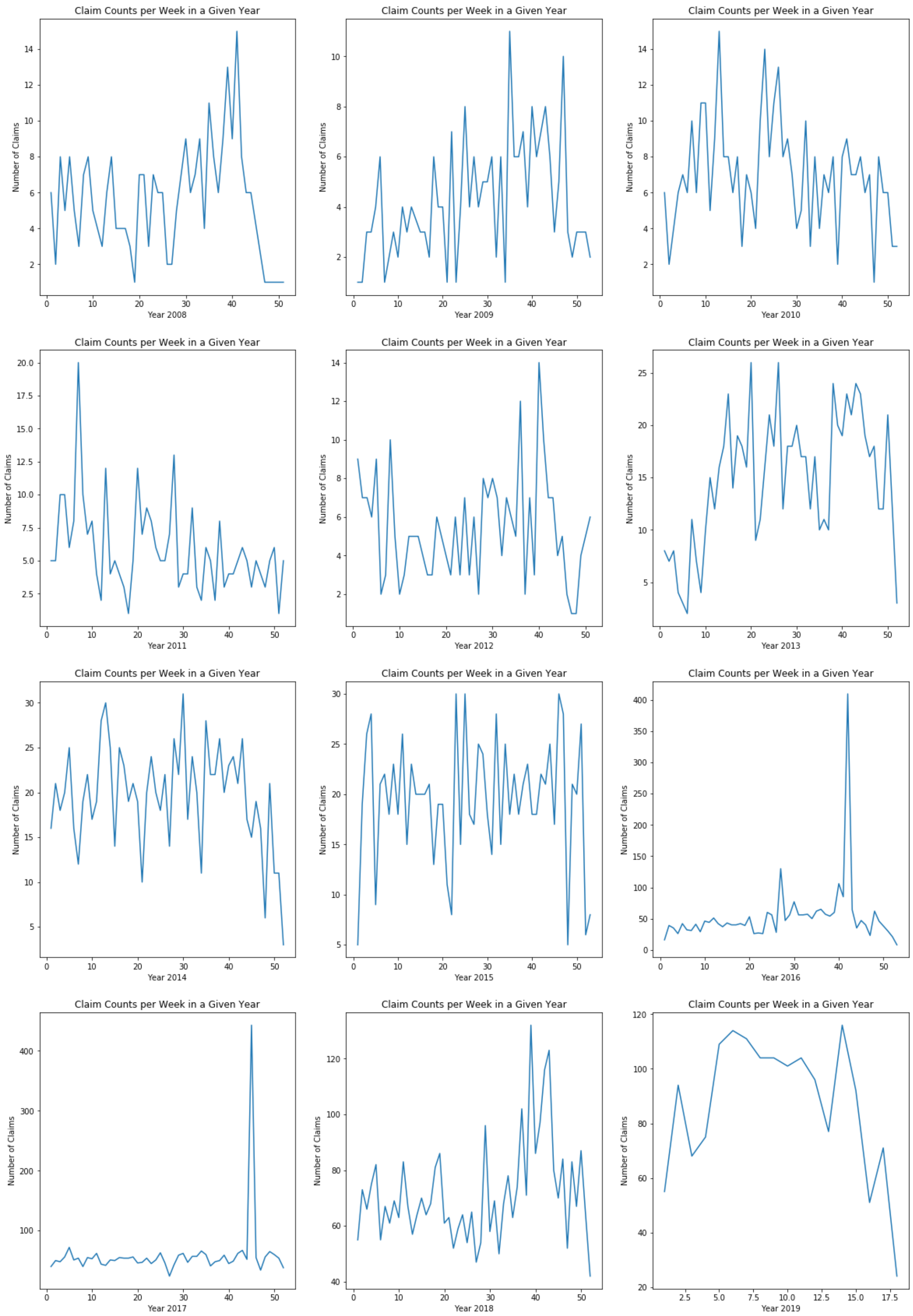

Figure 13: Number of Claims made weekly for each year in the dataset