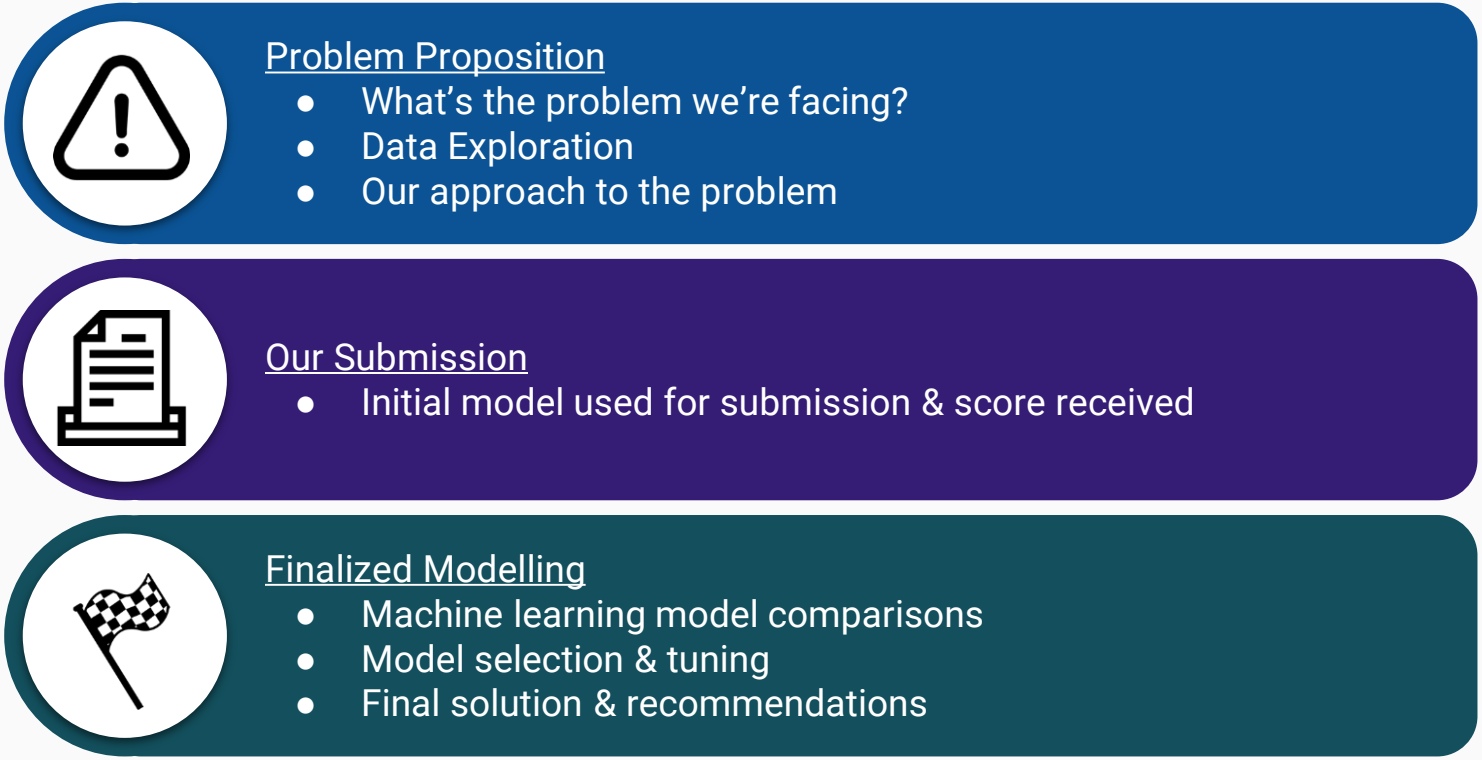
Fake News Detection

Group 19 - Kernel Sanders

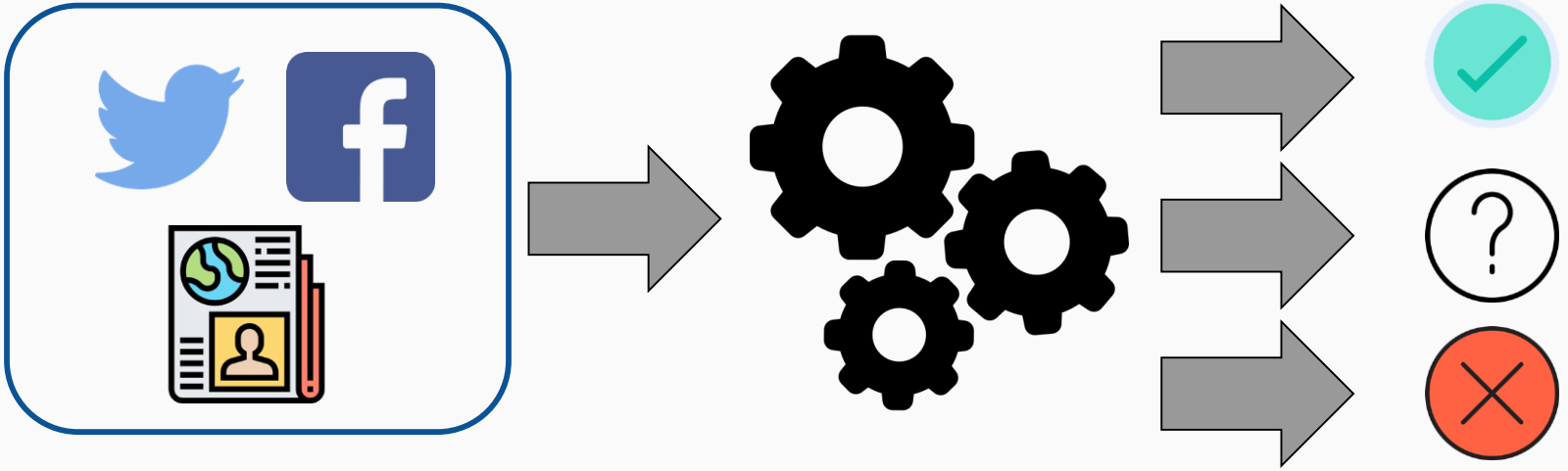
Syed Hamdan | Matthew Vassov | Michael Attong | Muhammad Farhan Riaz

Agenda


Problem Proposition
What's the problem we're facing?
Data Exploration
Our approach to the problem
Our Submission
Initial model used for submission & score received
Finalized Modelling
Machine learning model comparisons
Model selection & tuning
Final solution & recommendations

Problem Proposition

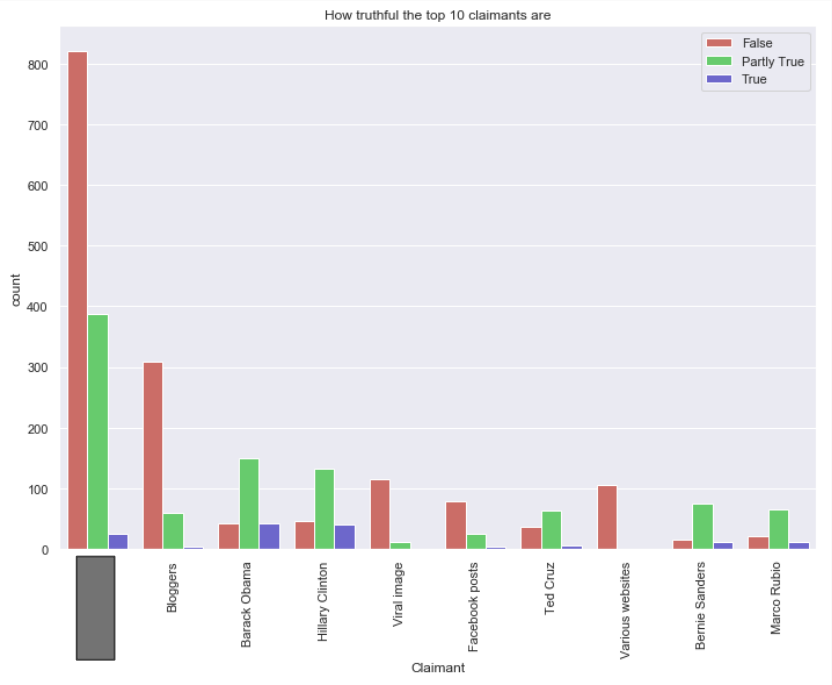
In the recent years there has been a significant increase of information and news in which society is presented. This means we need to be able to **trust what we're reading**, especially since stories can be manipulated to depict a different narrative.

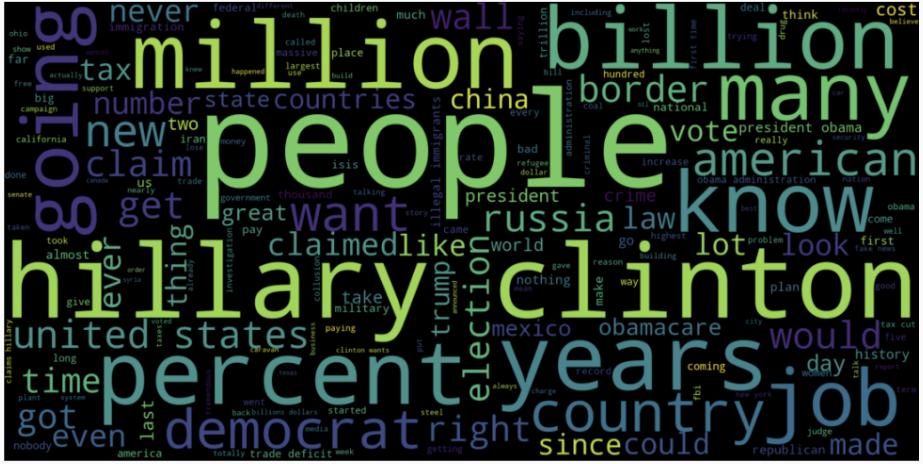


New Sources **Prediction Model** **News Validity**

Data Exploration - Who's lying?

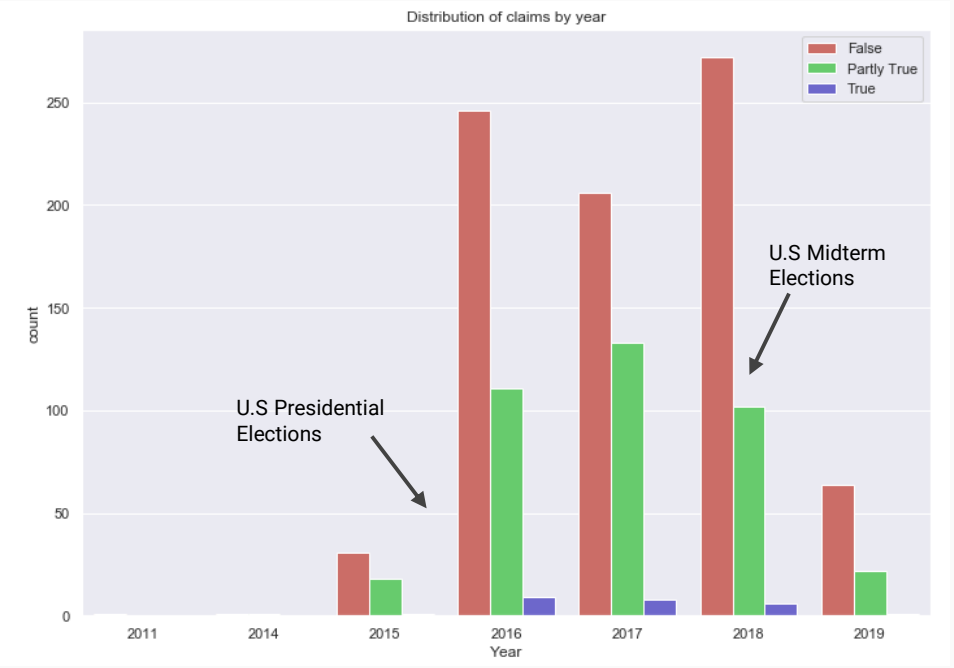
In reviewing the data we've noticed a couple different interesting trends. Below we would like to highlight some of the insights from the data. There is one claimant who has more lies than the rest.
Can you guess who it is?

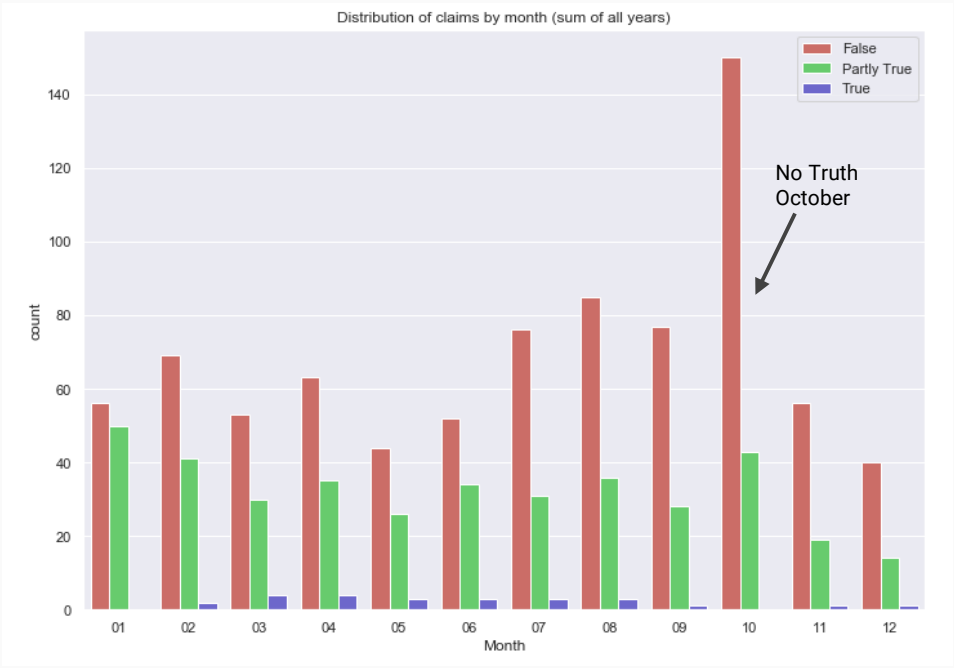




Data Exploration - When are they lying?

In case you were not able to guess it, perhaps looking at *when* they are untruthful may give you a hint as to who they are.






“
Fake news, folks.”
PRESIDENT TRUMP
January 26, 2018

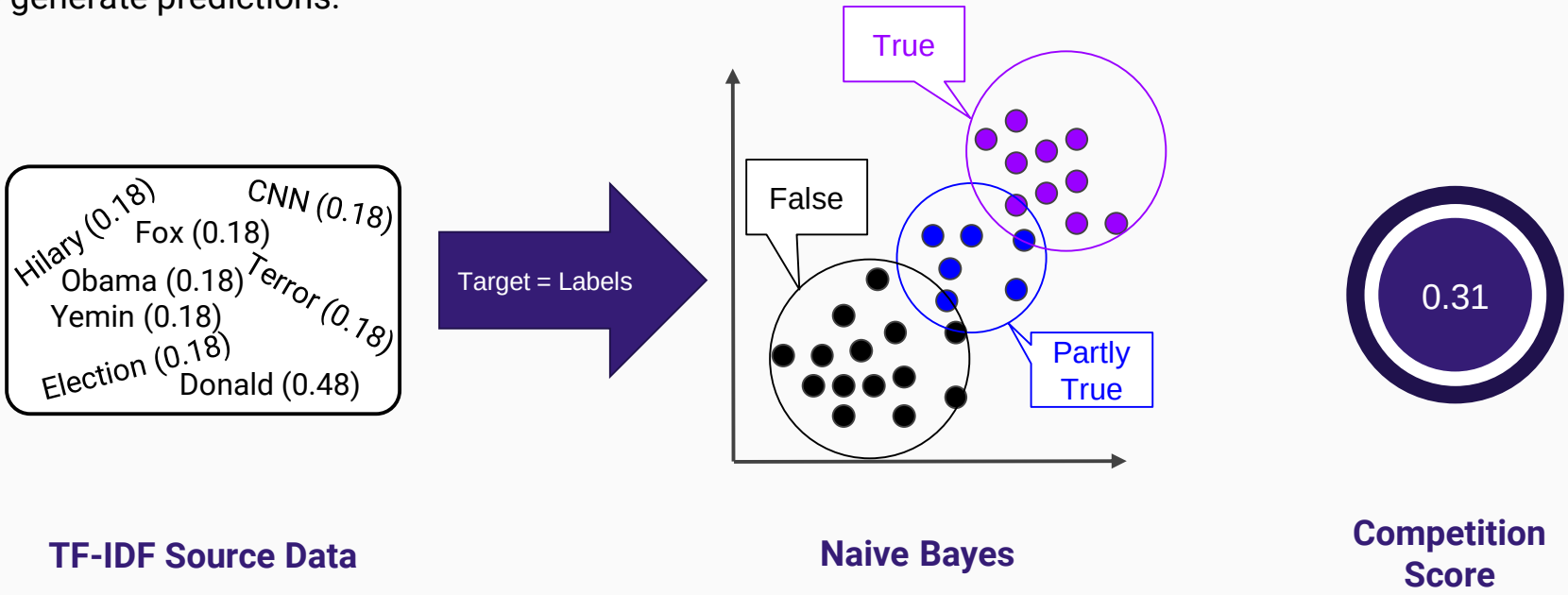
Overall Approach

Data Preparation	TF-IDF	Test Train Split	Model Testing	Hyperparameter Tuning
Constructing a new dataframe that contains both claims and articles	Using the term frequency within the document as numerical inputs	Separating the train (70%) and test (30%) data within the TF-IDF features	Using the split data to train the different models: - Decision Trees - Multinomial Naive Bayes - Logistic Regression - Neural Nets	After selecting the best model we will tune hyperparameters. Which could be: - Input Features - Alpha - Neurons Layers

Key assumption: Related article's truth label = Claim's label

Initial Model & Score

Our initial model takes the information from the claims database, converting the **claims into features** using TF-IDF. This numerical frequency value is fed into a trained multinomial **Naive Bayes algorithm** to generate predictions.



TF-IDF Source Data

Naive Bayes

Competition Score

Note: No pickles were harmed in the process

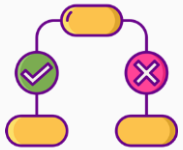
Initial Model & Score (continued)

At the time of preparing this presentation, we were #44 on the leaderboard.

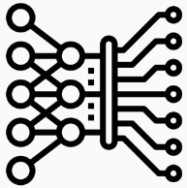
40	YAMM	0.320116
41	teamUW	0.317473
42	Threed	0.317145
43	UcanUup_1624	0.312152
44	Kernel Sanders	0.312117
45	G12	0.307378
46	MIE1624 Group 5	0.302275
47	justDoIt	0.299227
48	gra3017	0.298193

Model Comparison

There are many different models available to be able to manipulate and predict the validity of claims the model reviews. With considerations around **F1-Score** and **Accuracy** we have chosen to select **Logistic Regression** for the prediction generation.



Decision Trees



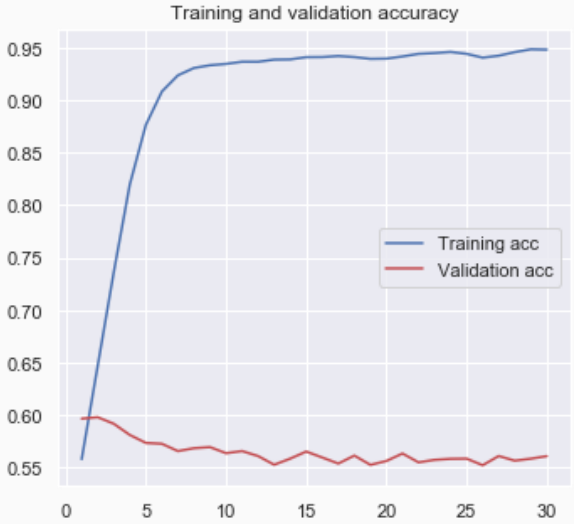
Neural Nets

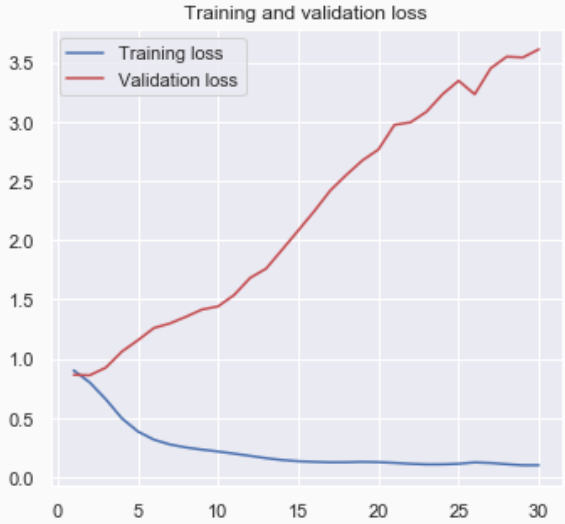
Logistic Regression

Model	F1-Score	Accuracy (Train / Test)	Comments
Random Forest	0.41	70% / 59%	Entropy Criterion
Naive Bayes	0.42	59% / 59%	Simple and quick
Logistic Regression	**0.44**	**65% / 62%**	**Best all round**
Neural Nets	0.45	96% / 56%	CNN utilized

Best Model Selection

Keras Sequential model was utilized to create a **Dense Neural Network** and a **Convolutional Neural Network - CNN** (which produced better results). Hyperparameter tuning via Random Search technique was performed for the CNN to obtain the best parameters.





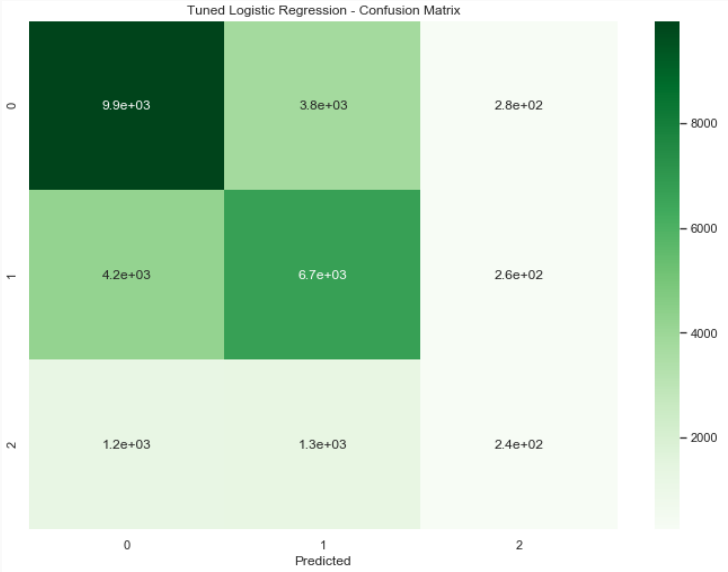
In the end, we decided to stick with **Logistic Regression** since Neural Networks are computationally expensive and they both (Logistic Regression and Neural Network) generated the similar results

Model Tuning

After selecting Logistic Regression as the model of choice, we used **Grid Search** in combination with **K-Fold** to identify the learning rate and solver methodology. From this we saw an average F1-Score of **0.47** which can be highlighted by the **confusion matrix**.

	ovr	multi	auto
C =1	Acc = 62 F1 = 44	Acc = 61 F1 = 45	Acc = 61 F1 = 45
C =10	Acc = 61 F1 = 46	Acc = 61 F1 = 47	Acc = 61 F1 = 47

K-Fold Accuracies



Confusion Matrix

Final Solutions & Recommendations

After running the tuning, we saw a final F1-Score of **0.466**, which still leaves room for improvement. We would like to make some **recommendations to improve** what the model can handle and predict.

Stem & Lem

Trim the datasets to be more accurate

Iterative

Adjust Logistic Regression to predict for 3 classifications

Partly = False

Treat partly true data as false, to avoid user views

Thank you!

Any Questions?

“The fake news media is going crazy with their conspiracy theories and blind hatred”
Philosopher, Bankrupt Businessman & the 45th President of America, Donald Trump

Appendix: Data Review

Dataset: Claims

Related Articles	41099	89899	72543	82644
Claims	When it comes to fighting terrorism...			
Claimant	Hillary Clinton			
Date	2016-03-22			
ID	6			
Label	2 - True			

Dataset: Articles

Related Articles	Articles
82644	No one like umpa lumpas...
89899	Orange isn't the new black...
72543	No Trump card for America...
41099	I spy a mole in the FBI...

Appendix: TF-IDF Implementation

Since we have understood the output to be whether or not the statement is true, we need supporting data for the prediction. Although we can use the claim as the only data point, we would like to supplement that information with the articles for more data to generate predictions.

Words	I	fake	some	news	is	best	scream	Sentence
Doc 1	0.48	0.18	0.48			0.18		I fake some news
Doc 2		0.18		0.36	0.48	0.18		fake news is best news
Doc 3				0.54			0.48	scream news news news

$$TF - IDF = Count * \log\left(\frac{\#\ documents}{\#\ documents\ containing\ word}\right)$$

Appendix: Decision Tree Visuals