Project Report:
Machine Learning ECE 417
Sentiment Analysis and Predictions on Twitter Reviews

Sevi Yfanti SYFANTI@INF.UTH.GR
Panagiotis Grigoriadis PGRIGORIADIS@INF.UTH.GR
Vasilis Zaridis VZARIDIS@INF.UTH.GR

Github:

Github:https://github.com/gitglob/ECE-417

Abstract

Airline companies dont have a lot of feedback considering the credibility of their services. Twitter is a major source of information, which they want to make good use of. We aim to extract this information, process it and use it to make useful predictions that they can use to improve their services.

1. Introduction

Air travel nowadays is an everyday thing. People constantly use it for vacations or/and job requirements.

There is a huge competition among airlines companies on which will dominate the market. In order to do that, however, they need to make constant improvements on the services they provide. That being said, they need to get constant feedback from their client, so that they can identify their issues, realize their extend and face them.

Twitter is a constant source of information. People share almost everything that happens to them there. Airline companies, among almost every other company, seek to exploit this to solve the fore-mentioned problem.

Our goal is to make use of twitter's data, and after processing it, use Machine Learning techniques to extract some conclusions on the sentiments that people express regarding some US airlines.

2. Background

2.1. Input Data

The data that we use are from a Kaggle 2015 competition. It is a csv file with 14485 rows and 15 columns. The most important, for this project's incentives, column is the column 'text' which includes the tweet made.With that column, predictions are made. The rest of the columns are extra information about the tweet's location, user and reason. Having these columns contributes in furthermore analysis helping us understand the data better and providing us with useful feedback for the airline companies. More specifically, columns such as negative reason, airline and tweet coord helps us visualize the data.

2.2. Methods

2.2.1. Preprocessing

The first step to a good prediction is to prepare the dataset. firstly , we deleted columns that are not important to our research or that have high correlation, like tweet id. Secondly, we handled the missing values (Nan) by replacing them with the columns mean value and finally, we did one-hot encoding to transform the categorical values.

Because our data is text, an NLP (Natural Language Process) had to be done. More specifically, from the tweets we removed twitter handles (@) , punctuation, numbers, and special characters. Also, stop words, short words and rare words. Finally, we conducted a lower case transformation.

University of Thessaly, ECE 417 Course Project

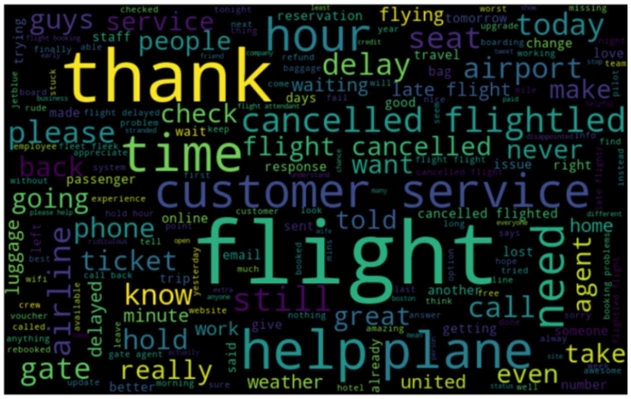

Figure 1. Wordcloud of the tweets

In order for the data to be applied in our machine learning algorithms and neural networks, it needs to be transformed into vectors and be computational. For that reason, a new table is produced using Word Embedding like Bag of Words, TF-IDF and word2Vec. All produce sparse matrices. There is a second approach, with textBlob. Instead of producing a vector with many coordinates, textBlob produces a single number in the range of [-1,1]. That number indicates how negative or positive the sentiment is.

2.2.2. Apply classic ML algorithms

We applied a long series of simple ML algorithms so that we would be able to compare the results and keep the most accurate. The applied algorithms are Logistic regression, SVM, KNN, Random Forest, Decision Tree, Naive Bayes and Gradient Boosting and all were examined with all 3 word embedding methods and with the created sentiment from textBlob.

The best result was found with the combination of Gradient Boosting and textblob with accuracy of 89.8%.

2.2.3. Apply Neural Networks

We also applied some Neural Networks to solve our problem. The NNs used were: Simple Keras Classifier, Convolutional Neural Networks, Simple Neural Network with embedding layer, Simple Neural Network with 2 hidden layers and LSTM network.

The simple Keras Classifier and the Simple Neural Network with 2 hidden layers were applied in combination with textBlob sentiment analysis. We applied 10-folds evaluation, train/test split evaluation and GRID search on learning rate, regularization term and optimizer.

The Convolutional Neural Network, Simple Neural Network with embedding layer and LSTM network were applied in combination with TF-IDF word embedding.

Also worth mentiong, that in the LSTM network, we only examined positive and negative emotions.

The best accuracy was found with LSTM, simple Keras Classifier with 10-folds and with some combination of n (learning rate) and (regularization term) in the Grid search on the 2-layers simple Neural Network.

3. Results and Analysis

Table 1. Simple Methods with Bag of Words

Algorithms	Accuracy	Time
Logistic Regression	76.3	0.73
SVM	76.8	6.5
KNN	60.8	0.003
Random Forest	73	7.5
Decision Tree	66.8	0.7
Nave Bayes	39.9	0.28
Gradient Boosting	71.2	2.8

Table 2. Simple Methods with TF - IDF

Algorithms	Accuracy	Time
Logistic Regression	76.5	0.46
SVM	76.8	5.72
KNN	51.9	0.001
Random Forest	73.1	6.94
Decision Tree	69.2	1.18
Nave Bayes	37.6	0.26
Gradient Boosting	73.6	8.08

Table 3. Simple Methods with Word2Vec

Algorithms	Accuracy	Time
Logistic Regression	76.5	0.46
SVM	78	2.24
KNN	78.2	27.21
Random Forest	73.4	0.17
Decision Tree	63.4	6.53
Nave Bayes	70	0.064
Gradient Boosting	75.9	122.84

Table 4. Simple Methods with TextBlob

ALGORITHMS	ACCURACY	TIME
Logistic Regression	84.2	0.73
SVM	88.8	37.8
KNN	85.7	0.12
Random Forest	88.9	0.42
Decision Tree	88.9	0.01
Nave Bayes	88	0.008
Gradient Boosting	88.9	2.9

Table 5. NN's

ALGORITHMS	ACCURACY	TIME
Simple Keras Clasifier	90.2	363.12
Simple NN (2 HL)	90.1	5.2
KNN	85.7	0.12
Simple NN with emedding layer	85.13	155.71
LSTM	90	67.4

4. Conclusion

Nowadays, the information is limitless and the possibilities are infinite. Being able to extract what is important can make serious difference in the prediction outcomes. NLP is a crucial process when trying to extract information from text because it limits the words, holding what's important and dropping things that not only are useless but also can cause problems and mess up the algorithm's training. There are many possible techniques and finding the right one for your problem can increase your accuracy by great amount.

An important thing to have in mind when we deal with neural networks is avoid overfitting and also choose the correct parameters for each neural net. It is crucial to understand that a bigger or wider network will not necessarily bring the best results possible.

Another thing to consider is the computational power and time that is demanded from our network. We need to evaluate if it is worth it considering the accuracy it provides.

Adding to that we must point out that preprocessing can be as demanding, or even more demanding and important than training our network itself. It provides crucial insight and helps us understand the nature of our problem better.

Companies that tackle such problems should first and foremost identify their exact goals and define the problem with details before proceeding to any process. Then, they can chose the proper techniques to to be applied. Many times general accuracy is not the only thing to consider. For example, in our problem, a company might be much more interested at classifying negative sentiments with accuracy than positive. In that case, if we had much more positive tweets than negative, general accuracy would not be representative of the success of our results.

In a future study , we visualize on getting real time information (tweets) and being able to identify it's class through unsupervised learning (clustering).