Sentiment Analysis on Tweets for Airlines

Babaei Elham

Ruggeri Gabriele

STATISTICAL LEARNING FOR DATA SCIENCE

DSSC

Abstract

This project aims at analyzing the *Twitter US Airline Sentiment* dataset, provided by kaggle, by performing an explanatory analysis on the data and by fitting a machine learning based tool that is able to predict whether a tweet, about one of the airline companies, is either positive, neutral or negative.

Dataset and Problem Statement

The dataset is made of 6000 observations and 15 columns, namely:

tweet_id	airline_sentiment	airline_sentiment_confidence
negativereason	negativereason_confidence	airline
airline_sentiment_gold	name	negativereason_gold
retweet_count	text	tweet_coord
tweet_created	tweet_location	user_timezone

.

Data show many redundant information that are beyond the goal of this project, so we will discard them, here a list: tweet_id, airline_sentiment_confidence, negativereason_confidence, airline_sentiment_gold, name, negativereason_gold, retweet_count, tweet_coord, tweet_created, user_timezone. All these predictors are considered not informative (like user_timezone, retweet_count,...) or simply not clear since kaggle webpage doesn't provide any meaningfull description (like airline_sentiment_gold, airline_sentiment_confidence, ...), but in any case after removing the correspondent columns we are left with just 4 predictors renamed as: *Sentiment*, *Negative_Reason*, *Airline* and *Text*.
The *Sentiment* variable labels every tweet as one of 'positive', 'neutral' or 'negative'; *Negative_Reason* is a variable that is not empty only for 'negative'

tweets and gives a brief motivation for the customer's disappointment; *Airline* tells us to which airline company the specific tweet refers to, and finally, *Text* that contains all the tweets observed. Here we must notice that the *Airline* features makes sense for every observation since, given how Twitter works, all the texts that are taken under analysis contain a string with "@" followed by the name of one of the airlines and so it will be always clear to which company the tweet is about. After these consideration we can formally state the problem that we want to solve:

Given a text, representing a tweet, label it as one of 'positive', 'neutral' or 'negative'.

Data Preprocessing and Visualization

Data Preprocessing

After removing the above mentioned predictors we had a look to how the single features were represented in the dataset and saw that all of them were of type 'chr'. The first step was to convert *Sentiment* and *Airline* to 'factor' since both are categorical variables, respectively with 3 and 6 levels. After that we moved to the actual processing on the *Text* variable, in order to tranform the raw tweets into a set of tokens that can be later numerically interpreted and most importantly used as set of features for the observations of the dataset. We want to apply some tranformations on the textual data so to have a new representation of the tweets, whose features are semantically relevant and able to characterize each tweets with respect to the *Sentiment* dependent variable. The first step was to remove from all the tweets any expression that started with '@' because all of these were either a tag to the airline (redundant since already known by *Airline*) or indicative that the tweet is actually a re-tweet (not informative because of the fact we removed any information about the account of the users). Then we performed a 70% / 30% split into training and test set so to be able to evaluate the future model performances on unseen data. At this point we focused on the training set and explore many possible processing steps and representations of such data; in particular we followed a de-facto standard pipeline:

1. Tokenization
2. Lower Casing
3. Remove Stopwords
4. Stemming

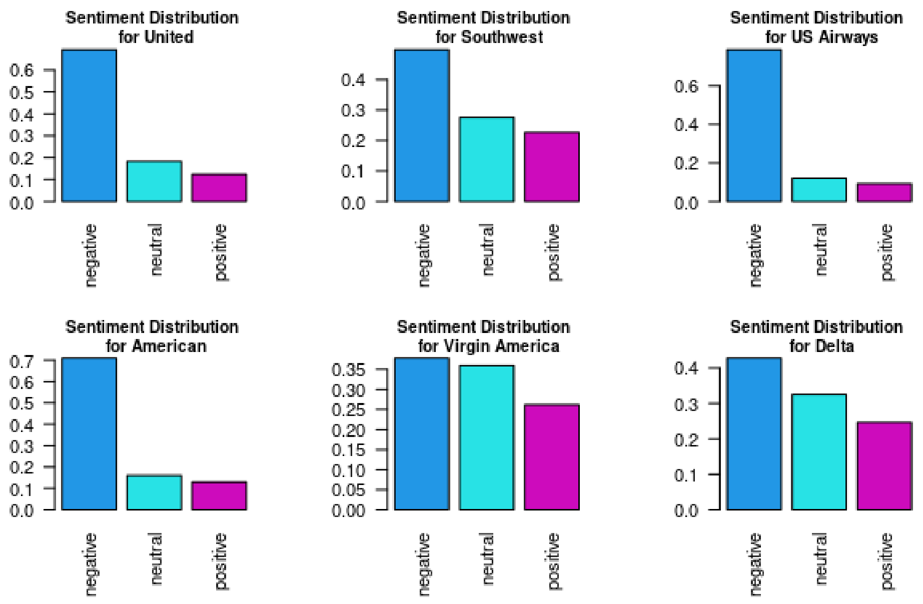
In this process the most meaningfull step was probably the tokenization since we had to make some choices and moreover it was the starting point of the whole pipeline. We considered Bag of Words (unigram) representation and we removed: punctuation, symbols (emojis), separators, numbers and urls. Altought the choice of what to remove is not universal and mostly domain dependent we thought that for the purposes of the analysis the words/expressions that will be more relevant are going to be characterized by the presence/lack of specific adjectives or combinations of adjective-noun, so that in any case such word types will be preserved. Both the removal of stopwords and the stemming were performed by the R package *quentada* (check the *stopwords()* function documentation for the entire list of english stopwords removed).

Data Visualization

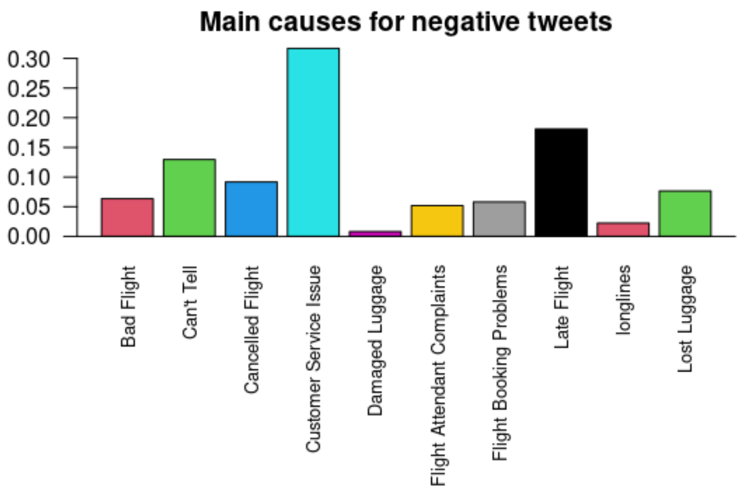
In this section we will present a detailed data driven process of question-answer that aims at understanding the people behind the tweets and the people's attitude towards the airlines with the goal of producing usefull insights for the companies that hopefully can lead to an improvement of the service and so of the brand popularity.

What is the satisfaction rate of the airlines?

Since for every tweet we both know the target airline and a sentiment evaluation we can ask what are the most popular airlines and hopefully why the others are not appreciated in the same way.

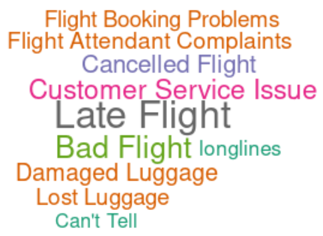


As we can see it looks like **Virgin America** and **Delta** are the companies with the most balanced response while the percentage of negative tweets for other airlines is clearly predominant. It is also clear that the main sentiment is the negative one. How can we explain it? Our hyphotesis is that people are most likely to post and evaluate their experience in case of a negative one, probably to point out their disappointment toward the company or a specific problem they faced.



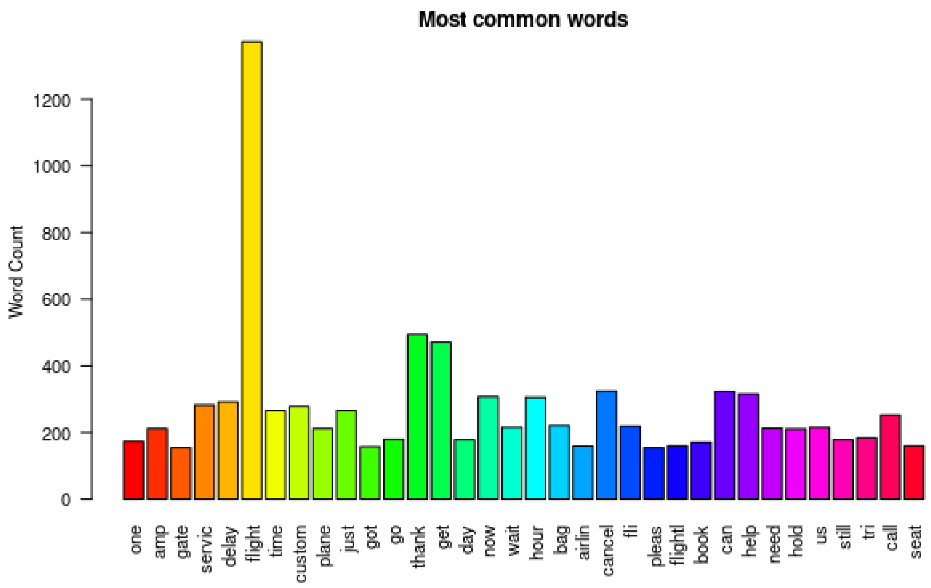
What are the most common causes for disappointment?

By looking at *Negative_Reason* we could infer a possible explanation for consumer's disappointment:

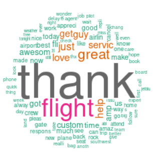


What are the most common words and how words characterise Sentiments?

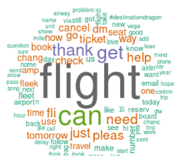
We can look at the words that appeared at least 150 times in the tweets and moreover, given the sentiment label, at the most frequent words. Our goal is to see if words can make clear on their own the topic of the tweets and moreover the associated sentiment behind them.
As expected the most common word is "flight", followed by "thank" and "get"; moreover we see words like "seat", "fli", "bag" and "plane" that clarify the domain and the semantic area.



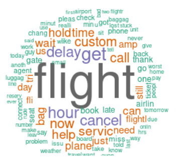
Now we can look at the most common words for every type of tweet We have a clear match especially for positive and negative tweets, while for



positive words



neutral words



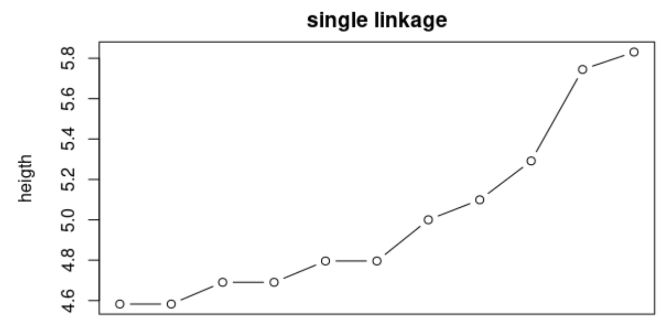
negative words

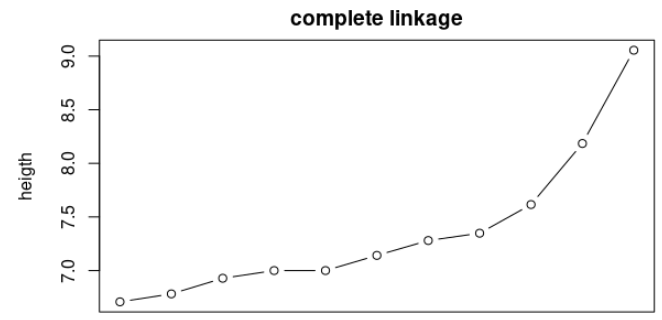
neutral ones it is less clear because we don't know exactly what to expect. What is a neutral tweet? The problem is that the labeling of the tweets is an arbitrary task and it was performed according to a subjective point of view.

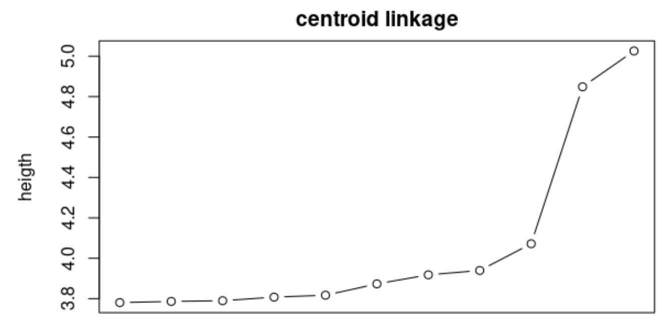
Do we really have neutral tweets?

At this point, recall that the texts are represented by means of BOW, we want to investigate whether the tweets suggest the presence of the "neutral" class. By means of clustering algorithms we tried to confirm that the choice of having these 3 categories makes sense, against the easier positive-negative approach. In particular we ran Hierarchical Clustering with: single linkage, complete linkage and centroid distance; then we looked at the biggest step between separation's height among the various number of clusters:

Although results showed some little instability with respect to the sampling of initial data we always got that the algorithm suggested, with equal probability, the presence of at least 2 groups and no more than 3. So in the end we can accept the initial labeling of tweets as data confirmed. A possible further step could be to exclude neutral tweets from the training data; fit a binary classifier and try to relabel them as one of positive or negative.







Data Representation

TF-IDF Matrix

After going through all the preprocessing steps, what we obtain is a Bag of Words in which the frequency of each term (token) is included. In other words, we have the document-term matrix (document-feature matrix) which shows the frequency of each term in each document. But as we know the amount of frequency by itself is not an adequate measure to determine the importance of each term in the corpus; because it certainly depends on the size of documents. For instance, the frequency of 100 for a term within a document of length 1000 is not comparable to the frequency of 5 for a term within a document of length 10. Therefore, a solution to have a more proper measurement is using the product of Term Frequency (TF) and Inverse Document Frequency (IDF) which are defined as bellow:

$$TF(t,d) = \frac{\sum_{i:t_i \in d} I(t = t_i)}{|d|}$$

$$IDF(t,m) = log\left(\frac{m}{m_t}\right)$$

Where m is the corpus size and m_t the number of documents in which t appears. In the end we can define the TF-IDF metric:

$$TF - IDF(t,d) = TF(t,d)IDF(t,m)$$

Now we have transformed our original text dataset to a numerical dataset which is ready to be analyzed and be used for classification.

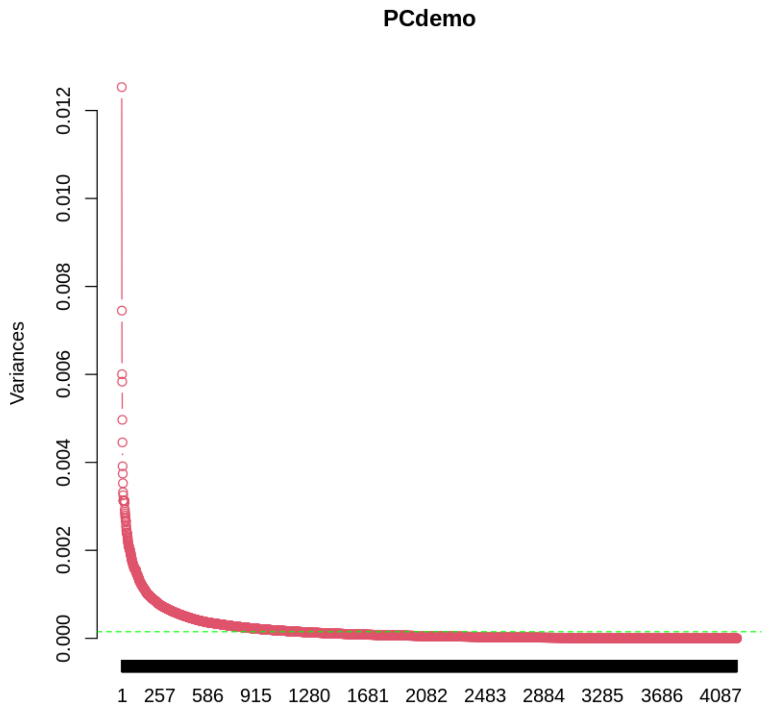
Dimensionality Reduction

PCA

Well known problems in Textual Data Analysis are the curse of dimensionality and the sparsity of the document-term matrix. This is also true in our case; the size of this matrix is 4201*5018 and many cells are equal to zero due to terms with zero values for TF. A good solution to handle this issue, could be Principal Component Analysis (PCA) which is one of the approaches to decrease the dimension of the problem by projecting the variables onto a new subspace in which the new variables are orthogonal and therefore uncorrelated.

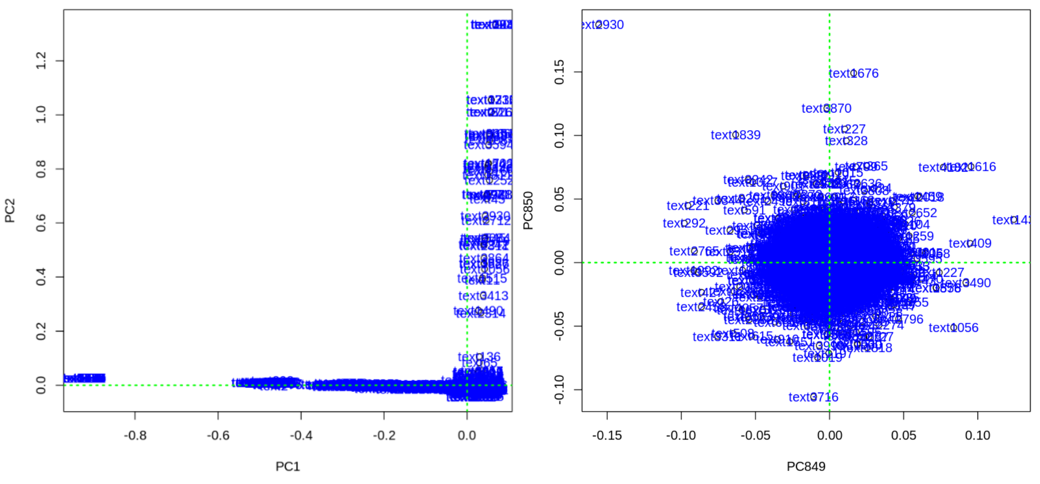
Implementation

To implement PCA on our document-term matrix, we first look at the covariance and correlation matrices of our terms (features) to see if the variances between pairs of terms are significantly different or not. It turned out that the variances are not so different; this makes sense as the values for the terms are the same type TF.IDF which are usually small continuous numbers; hence we do not need to standardized the features for PCA.
After running PCA on our dataset, we found out the minimum number of Principle Components (PCs) that are able to capture almost 80% of the variability of the original features is 850. This result is great and acceptable since we managed to decrease the number of features from 5018 to only 850 and moreover the new ones preserve a large proportion of information of the original variables. The following plot shows the best number of PCs at its elbow specified by the green dashed line.



Results

Then we look at the plot of the first two PCs. We can see that tweets tend to dispose along the axes; so that the proportion of variance contained in the first two PCs is able distinguish tweets well.
On the contrary, the plot of the last two PCs does not show anything special because they describe a small portion of variance of the data.



Both plots show that tweet 930 is different from others. By looking at the dataset, we saw that the content of that tweet is somehow unrelated to the topic Airline Sentiment; it is:

"just touched down in #NewOrleans for the annual national distributor conference! #livethelegend".

That is why it has fallen far from other tweets.

Biplot

Biplot either in case alpha is 1 and 0 illustrates the terms "fleek" and "fleet" are the most important terms in the dataset. By looking at the dataset we noticed the expression *"Our fleet's on fleek"* has been tweeted by users many times. However, since the dataset is really large these plots don't show the relationships clearly, so it is difficult to interpret them.

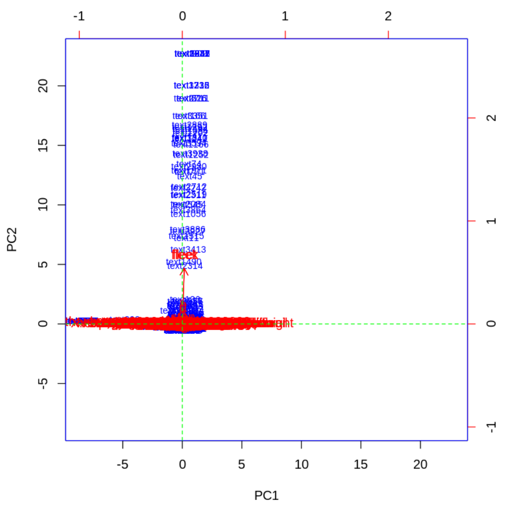


Figure 1: $alpha = 1$

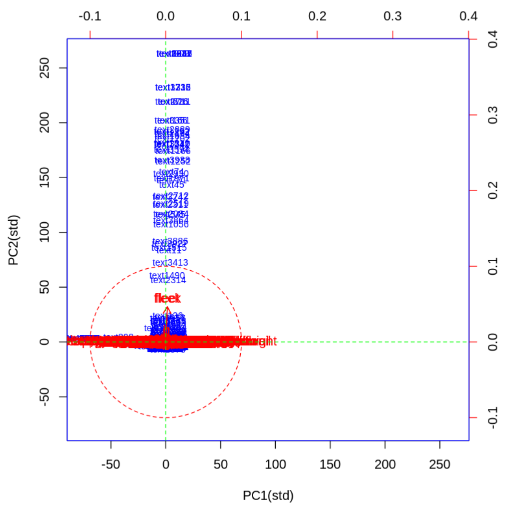


Figure 2: $alpha = 0$

Classification

In this phase we are ready to work on some classification algorithms to extract the sentiment of people from their tweets about their flight. Before that, it is worth mentioning that the test dataset for model validation purpose must be transformed to the same structure as the training dataset. To do this we repeated all the same refining and preprocessing steps of training data on the test data. The point is that in order to transform TF-IDF matrix to a PCs matrix, we need to use the rotation matrix (V) of PCA on training data to project TF-IDF values to PCs:

$$Z = XV$$

We will have the same number of PCs by taking the first 850 of them,like in the training.

Algorithms used for classification

The algorithms that we want to discuss about are the following ones:

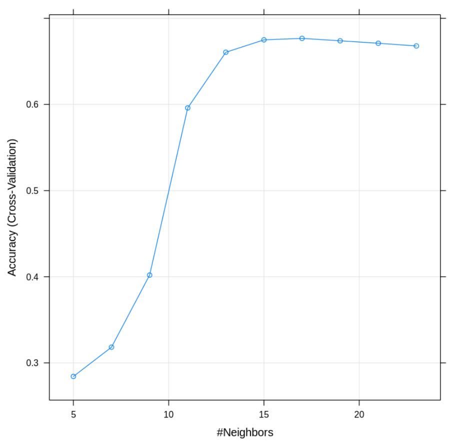
- Linear Discriminant Analysis
- KNN
- Random Forest
- SVM (linear kernel)

Hyperparameters

Notice that both KNN and Random Forest have some hyperparameters:

- KNN: k, number of neighbors
- RF: T, number of trees; M, number of variables considered at each split

For RF we used the default values that are $T = 500$ and $M = \sqrt{p} = 29$.
For KNN we used a 10-CV testing $k \in \{5, 7, 9, 11, 13, 15, 17, 19, 21, 23\}$ and in the end the best value for k was 17:



Performance Evaluation

A summary of test errors for each model:

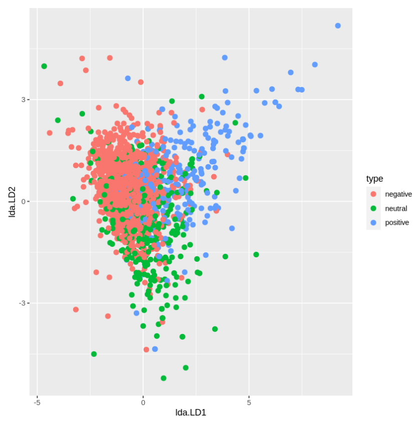
Model	Test Error
LDA	25%
KNN	31%
RF	30%
SVM	29%

LDA

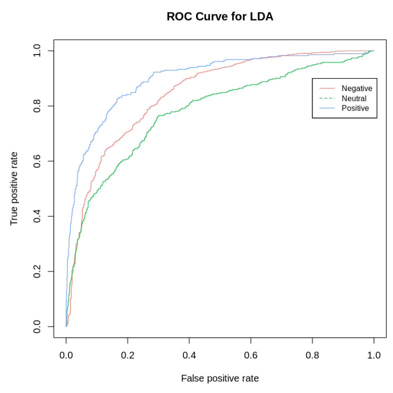
By looking at the confusion matrix (Actual vs Predicted) of LDA predictions:

	negative	neutral	positive
negative	1049	211	103
neutral	53	145	26
positive	28	28	156

We can clearly see that negative tweets are the ones the algorithm better detects, with a Sensitivity of 93%; which is what airlines could be more interested in. We can also visualize units in the plot of linear discriminants:



The plot shows that the classes are separated, however neutral tweets are more spread than the others. By looking at the results of the model from a "onevsall" perspective we computed the ROC curve that enforces our prior idea about the hardness of classifying neutral tweets

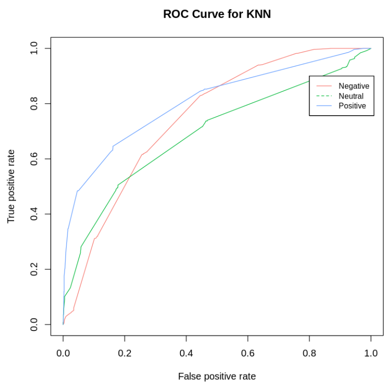


KNN

The confusion matrix:

	negative	neutral	positive
negative	1093	304	192
neutral	34	70	20
positive	3	10	73

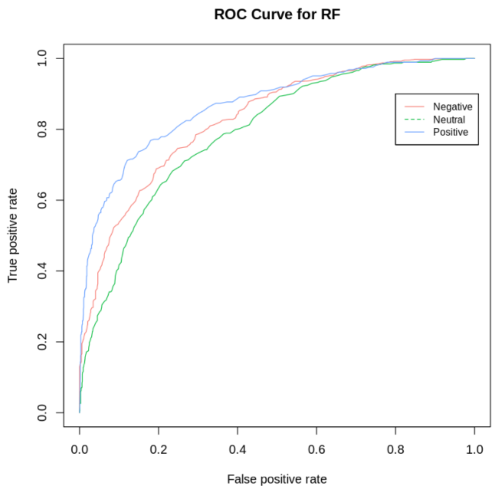
Again we got a very good Sensitivity (96.7%) for negative tweets. Notice that the consufion matrix suggests that the algorithm is very likely to label a tweet as negative; this could be the consequence of the fact that the negative tweets are the most common in the data and so the prior probability relative to the class is the highest one, which could translate in a form of bias. The ROC curve suggests again a poor preditictive power for neutral tweets

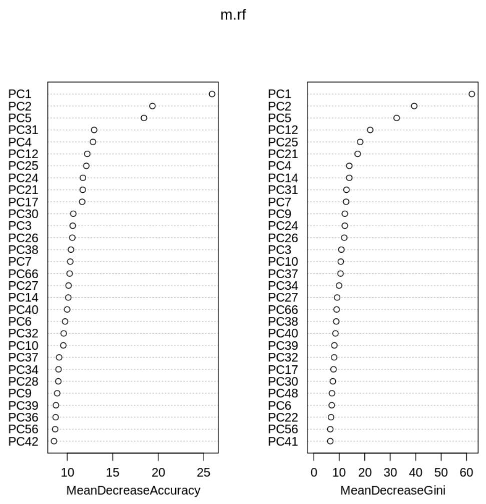


RF

	negative	neutral	positive
negative	1112	324	162
neutral	7	41	8
positive	11	19	115

RF shows the best performances in term of negative Sensitivity against the other algorithms, scoring 98%. Here we see the ROC curve plot and the variable importance plot, that the algorithm measured during the fit, from which it's clear that the first 5 PCs are the most important.





SVM

	negative	neutral	positive
negative	988	185	98
neutral	93	150	44
positive	49	49	143

SVM scored a Sensitivity of 87%.

Model comparison

The first candidate model to classify tweets is LDA with 25% test error. The basic assumption in LDA is that the features are normally distributed. In our dataset, the features are PCs derived from TF-IDF values which are the result of transforming a test dataset to a numerical matrix. This could be one of the potential reasons of low accuracy of LDA. The scatterplot of classified labels with respect to LD1 and LD2 and the Receiver Operating Characteristic (ROC) curve show that LDA is a good choice to classify the tweets. Area Under the Curve (AUC) for Negative, Neutral and Positive sentiments is 85%, 78%, and 90% respectively. The second-best algorithm is SVM with 29% test error. It is a supervised classifier which usually obtains good results since it can efficiently perform a non-linear classification using what is called the kernel trick. The third best algorithm is RF, as an ensemble technique, with 30% test error. The Importance plot of RF turns out that the first 5 PCs are the most important ones to explain the variance of the original variables. The ROC curve of RF is a bit worse than LDA and AUC for Negative, Neutral and Positive sentiments is 83%, 79%, and 87% respectively Finally, the last algorithm is KNN with 31% test error. Since this algorithm is non-parametric and works good on large datasets, probably the already mentioned reasons; the fact that features are PCs and other than

Negative class there are not large datasets for the other two classes; affect the result of KNN. AUC for Negative, Neutral and Positive sentiments is 75%, 69%, and 81% respectively.

Conclusion

In this project we analyzed the text dataset of tweets related to 6 airlines in United States in February 2015. The first aim of the project was to find out what is the common opinion of people towards airlines and what are the reasons for that comment. We realized the majority of tweets were negative feedbacks for all airlines with some reasons mainly corresponding to Customer Services and Flight Delays. Although this is something expected as people usually do not share their positive and neutral view in social media! Among the airlines, Virgin America and Delta were the two airlines with less difference between the proportion of three feedbacks from their customers. In the second phase, we tried some classification algorithms to make some models to predict the sentiment of new tweets towards airlines that users have traveled with. We implemented four algorithms LDA, KNN, RF, and SVM. For this purpose, we needed a numerical matrix of data to be able to apply the mathematical aspect of the algorithms on. Therefore, we transformed our text dataset to TF.IDF matrix and then by using PCA we relaxed the curse of dimensionality and sparsity issues. The predictions of the algorithms showed that LDA, SVM, RF and KNN are all good to classify sentiments with average accuracy of 70%-75%. However, RF (as an ensemble technique) and SVM are computationally intensive methods, so our conclusion is that the best algorithms for the problem are LDA and KNN.