Group 10 : TUIRAN Daniela, PALIWAL Harshit , MIRZAEE Saeed

Sentiment Analysis for Tesla using Twitter

Sentiment analysis has emerged as an important topic with the increase of social media interactions, use of forums and blogs, sales comments and ratings through e-commerce websites. Sentiment analysis is a field of Natural Language Processing (NLP) that offers a way to determine the writer's opinions polarity as positive or negative in a piece of text, about a particular topic, and product etc. The field applications contributed to many tasks on the evaluation of consumer products, understanding the impacts of some social events, and evaluating movie reviews.

Our purpose is to create a model capable to predict the inherent sentiment in a tweet related to electric car's company Tesla. In this work we train two different machine learning models with best parameters possible to obtain the highest performance for each one of them. Afterwards, we compare results in both models to choose the one in which we are going to generalize the results.

We are going to explain each one of the steps followed in 5 steps: first import and pre processing of the data, structure and inspection of the extracted text, model building, model results and comparison and finally results generalization.Some difficulties faced during text mining and model model building process and propose some ways of improvement we identify from them are presented.

1.Import and Pre process data

The data set used for sentiment analysis was obtained from Twitter API, using the keyword TESLA we obtained 17.900 tweets excluding re tweets.First of all, we needed to ensure language standardization, that's why we remove all tweets in languages different to English, so our tweets data base was reduced to 10.471 observations.In order to eliminate other undesired elements from the tweets and put all of them in a clean format we performed some pre processing operations as follows:

1. **Converto to lower case:** In order to standardize the tweets we convert all of them lowercase the text to ensure equal treatment of all words.

2. **Remove stopwords**: Stop words are a set of commonly used words. The reason why we remove stop words, if we remove the words that are very commonly used in a given language, we can focus on the important words instead. In this step we also include some words that are very used when talking about Tesla including the name of Tesla itself and all the possible variations like Tesla and tsla; Also its CEO's name and first name Elon Musk.

3. **Delete users name y @ sign:** Specifically related to Twitter, we delete users and @ sign, this information give no insights about the sentiments related to the tweets.

```
gsubtransfo <- content_transformer(function(x,from, to) gsub(from, to, x))
tweets_corpus <- tm_map(tweets_corpus, gsubtransfo, "@\\w+",  "")
```

4. **Delete URL's:** Using regular expressions, we create a function for the tm package so we can delete all urls attached to each tweet.

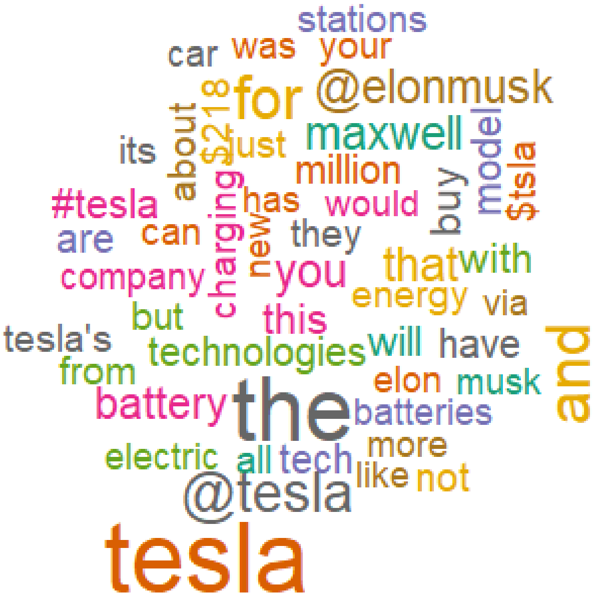
```
gsubtransfoURL <- content_transformer(function(x,from, to) gsub(from, to, x))
tweets_corpus <- tm_map(tweets_corpus, gsubtransfoURL,
"\\s?(f|ht)(tp)(s?)(://)([^\\.]*)[\\.|/](\\S*)",  "")
```

5. **Remove numbers, puntuation and white spaces:** Finally we remove all the mentioned elements to finish with the cleaning and pre processing steps.

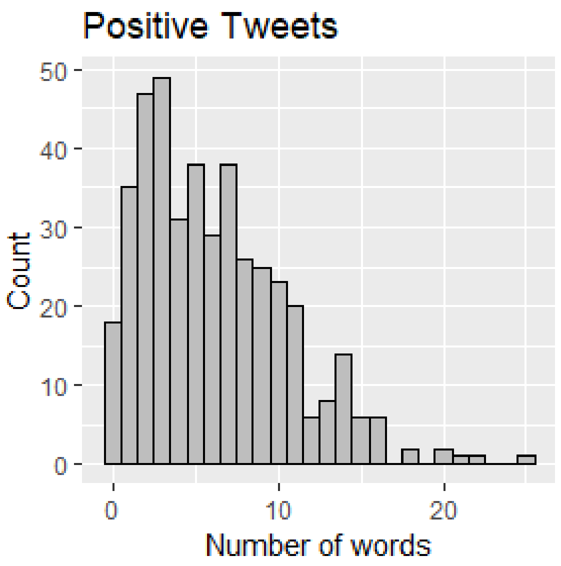
2. Apply a structure to the text and inspect the text

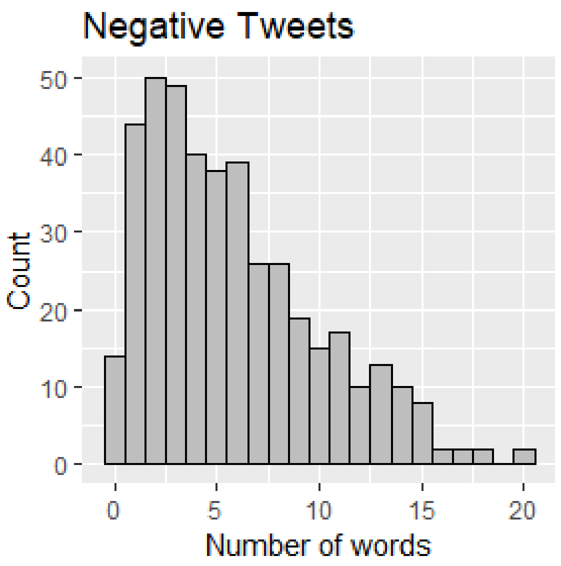
A term-document matrix represents the processed text from a text analysis as a table or matrix where the rows represent the text responses, or documents, and the columns represent the words or phrases (the terms). In our case with the processed labeled and unlabeled tweets we apply this kind of structure. In this way, each term represent one predictive variable or independent variable and the sentiment represents the target we want to predicts.

In order to understand better the general content of the tweets we performed some descriptive analytics presented below. First we explore the most used words in tweets with out any processing steps :

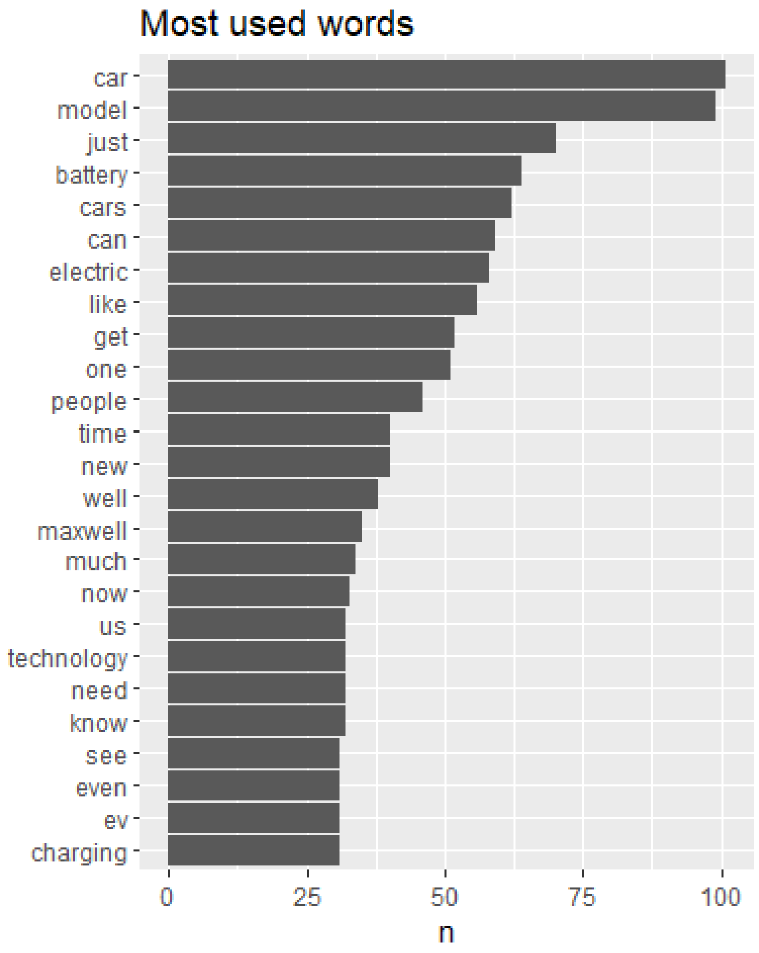


Most of the positive and negative tweets contains from zero to fifteen words as we can appreciate in the following distributions:

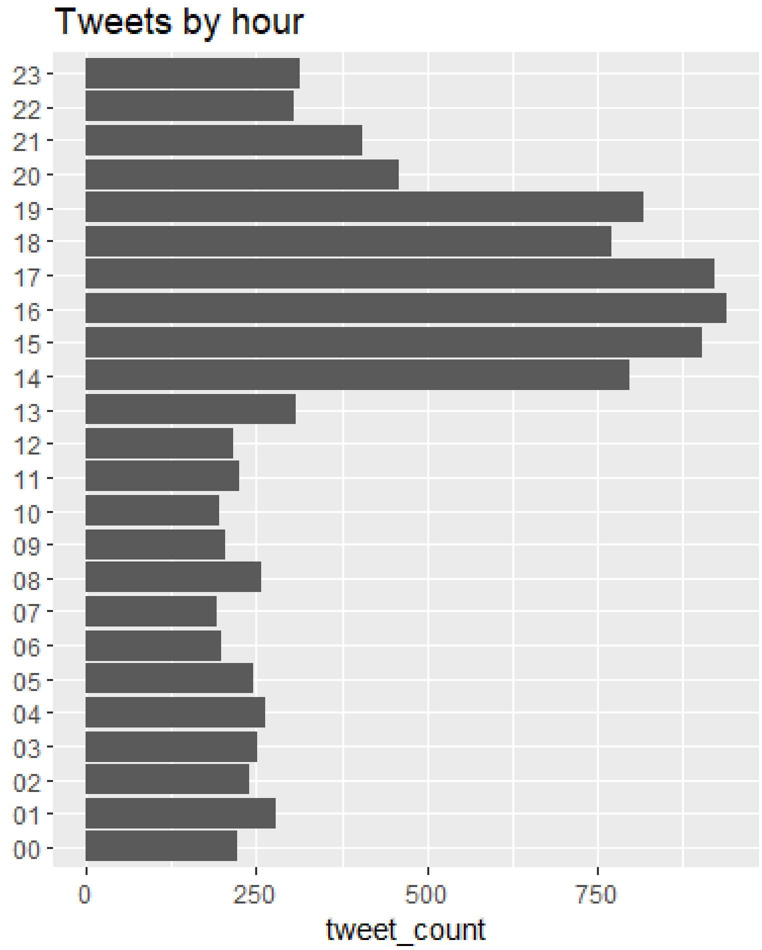




Now, we can see the frequency of the most used words in the related tweets. Car, model, and battery appears as the dominant most common words. It is important to mention that we filter to show just words with a frequency higher than 30.

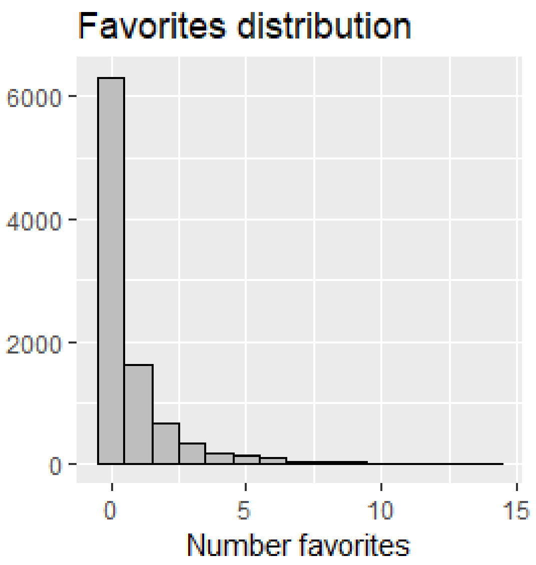


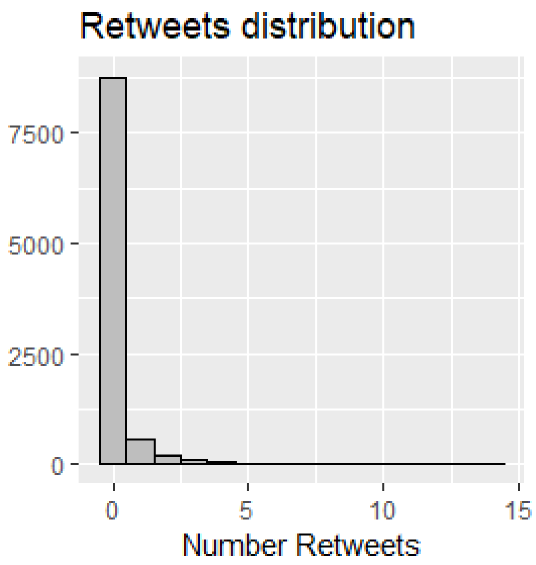
From 16:00 h to 19:00 h , people appear to be talking more about Tesla and generating more content about it. 16:00 h represent the highest frequency for this point of analysis.

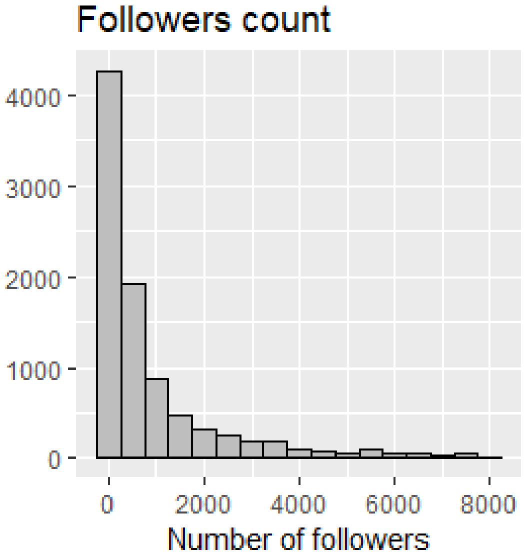


Reviewing the most important metrics related to tweets we found:

- **Favorites:**Most of the tweets have from zero to six likes or favorite mark.
- **Re tweets:** As with favorites, re tweets are not that predominant in tweets about Tesla. Concentration around zero to two re tweets per tweet.
- **Followers:** Accounts participating in discussions about Tesla have

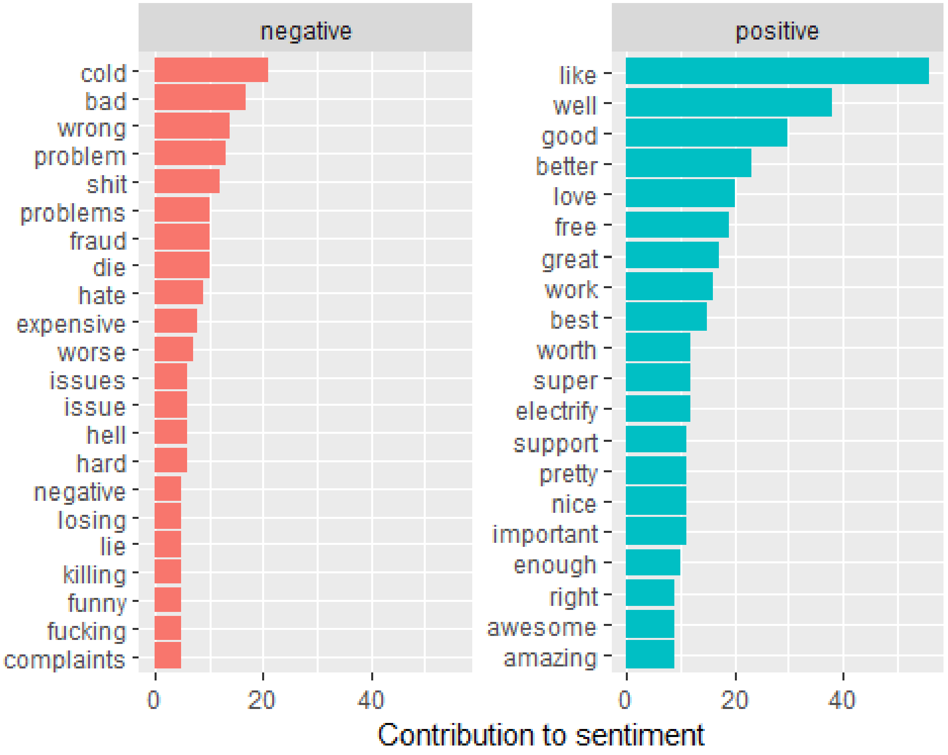






Positives and negatives review

Going deeper into the sentiment separation now we are presenting the most used words for each type of sentiment as well as the frequency of the most used words with the same separation. For negative sentiment, we see wrong, problem, cold, shit and fraud as the most predominant words. In the other side, Like, better, great, super, electrify, among other appear for positive.



3. Model building

To generate our main dependent variable, it means, consumer sentiment,we employed supervised sentiment analysis to determine the positivism of the statement in each tweet.Whereas unsupervised approaches use external input such as lexicons dictionaries to classify text messages,supervised approaches use machine-learning techniques that require training data as input.

As a first step we generated training data set for our tweets sample by randomly selecting tweets from the data extracted and manually classifying these posts as positive and negative. Only unambiguous posts were included in the training data set, and training data set consisted of the same number of positive and negative tweets.This process was made under supervision of the team members, so it is influenced by the individual perception of each one of them. At the end we result with 852 tweets equally classified as positive and negative. Based on these sub sample we are building our two machine learning models to try to predict the sentiment on tweets related to Tesla as we previously explained.

text	cat
everywhere	1
adds expertise capacitors speed electriccar charging	1
weve said solar roof product needs last decades therefore long development cycle weve thoughtful deliberate gradually ramped production spokeswoman said announce	-1
first never seen solar roof actually want weird	-1

something important enough believe something important enough even scared keep going ceo motors spacex entrepreneurship mobileentrepreneur entrepreneur smallbusiness startup startupbusiness	1
blog post colorful billionaire founder promised company initiate patent lawsuits anyone good faith wants use technology	1

In our analysis, we used a support vector machine (SVM) algorithm and a random forest algorithm. We are going to explain the process followed in each one of them, evaluate performance and finally compare results and select the best model for generalization of the results.

For train, the duplicate and irrelevant terms were dropped and finally consisted 433 variables and 595 observations. As test/validation consist 30% of total labelled data set, final test/validation contained 257 observation and 433 variables.

3.1 Random Forest Approach

In order to train the model, the Random Forest algorithm was selected as a counterpart of SVM, in order to compare the accuracy and final sentimental variations in the results. While we tried with random number of trees varying from 201 to 2051, we found a pretty good accuracy for 1699 trees.

```
##
## Call:
##  randomForest(formula = cat ~ ., data = train, ntree = 1699, maxnodes =
100)
##                Type of random forest: classification
##                      Number of trees: 1699
## No. of variables tried at each split: 20
##
##         OOB estimate of  error rate: 37.54%
## Confusion matrix:
##     -1   1 class.error
## -1 172 121   0.4129693
## 1  102 199   0.3388704
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction -1  1
##         -1 80 35
##         1  53 90
##
##                Accuracy : 0.6589
##                  95% CI : (0.5976, 0.7166)
##     No Information Rate : 0.5155
##     P-Value [Acc > NIR] : 2.215e-06
##
##                   Kappa : 0.3201
##  Mcnemar's Test P-Value : 0.06995
##
##             Sensitivity : 0.7200
##             Specificity : 0.6015
##          Pos Pred Value : 0.6294
##          Neg Pred Value : 0.6957
##              Prevalence : 0.4845
##          Detection Rate : 0.3488
##    Detection Prevalence : 0.5543
##       Balanced Accuracy : 0.6608
##
##        'Positive' Class : 1
##
```

The random forest model presents a healthy level of accuracy 65,89 % , leaving an error rate of 31.11%. From the total of positives around 72% of them were calculated as such, in the other side a 60% of the actual negatives where corrected predicted by the model.

Cohen’s Kappa is a measure of how well the classifier performed as compared to how well it would have performed simply by chance. In other words, a model will have a high Kappa score if there is a big difference between the accuracy and the null error rate.The model presents a 32% Kappa, which indicates a good performance to be differentiate from a random model.

3.2 Support vector machine

We built the SVM classifier models using stratified tenfold cross-validation for the variable cost.Cost (C) refers to how much we penalize the SVM for miss classified data points. Large C means penalize a lot. If your C is large, the SVM will try to find a hyper plane and margin so that there are few very points within the margin, which could mean an overly complex model with a small margin if the points aren’t easily separable. A lower C gives higher error on the training set, but finds a larger margin that might be more robust.

```
SVM_model_train <- svm(cat~., data=trainDenseL,
                       type="C-classification",
                       kernel="radial")



tune.out=tune(svm,cat~.,data=train,
              ranges=list(cost=c(0.01,0.1,1.5,10,100)))

##
## Parameter tuning of 'svm':
##
## - sampling method: 10-fold cross validation
##
## - best parameters:
##  cost
##    10
```

```
##
## - best performance: 0.3837288
```

We finally build the model using the resulting best parameters,made predictions with this model using the test data set.

```
SVM_model_train <- svm(cat~., data=train,
                       type="C-classification",
                       kernel="radial",
                       cost= 10)
```

4. Models evaluation and intermediate results

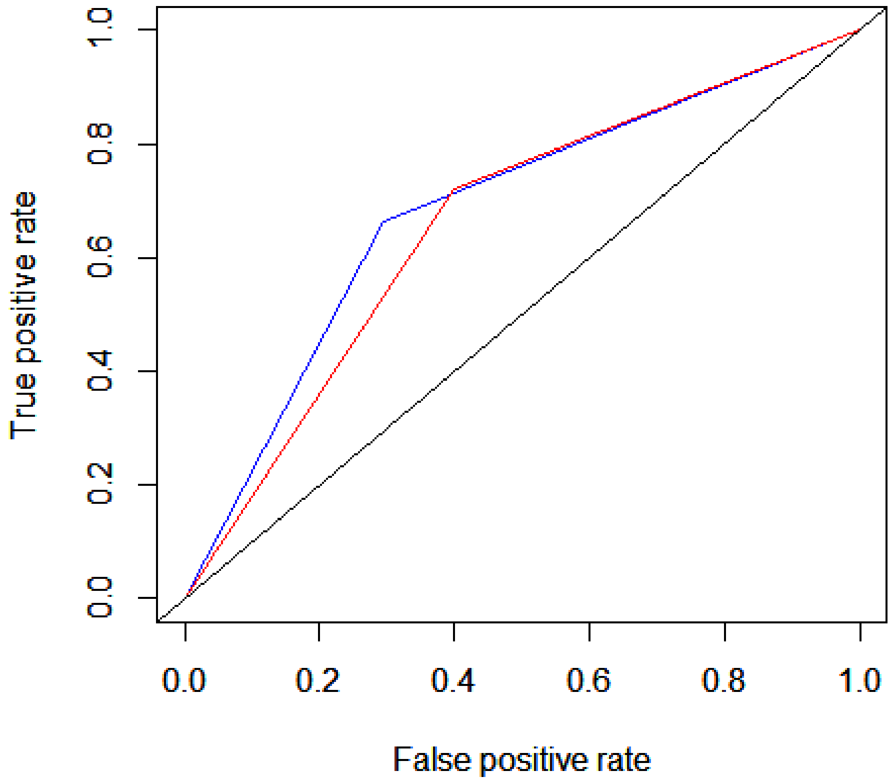
To evaluate the performance of the model we calculate confusion matrix and AUC curve, which will allow us make comparisons within models.Confusion Matrix shows classified positive and negative tweets. Accuracy is calculated based on confusion matrix and ROC curve.

```
#Create a confusion matrix
confMatrix1 <- confusionMatrix(SVM_predict, testDenseL$cat, positive="1")
confMatrix1

## Confusion Matrix and Statistics
##
##            Reference
## Prediction -1  1
##         -1 94 42
##         1  39 83
##
##                Accuracy : 0.686
##                  95% CI : (0.6256, 0.7422)
##     No Information Rate : 0.5155
##     P-Value [Acc > NIR] : 2.005e-08
##
##                   Kappa : 0.371
##  Mcnemar's Test P-Value : 0.8241
##
##             Sensitivity : 0.6640
##             Specificity : 0.7068
##          Pos Pred Value : 0.6803
##          Neg Pred Value : 0.6912
##              Prevalence : 0.4845
##          Detection Rate : 0.3217
##    Detection Prevalence : 0.4729
##       Balanced Accuracy : 0.6854
##
##        'Positive' Class : 1
##
```

The SVM model presents a healthy level of accuracy 68,6 % , leaving an error rate of 31.4%. From the total of positives around 66% of them were calculated as such, in the other side a 70% of the actual negatives where corrected predicted by the model.

As mentioned before; Cohen's Kappa is essentially a measure of how well the classifier performed as compared to how well it would have performed simply by chance. In other words, a model will have a high Kappa score if there is a big difference between the accuracy and the null error rate.The model presents a 37% Kappa, which indicates a good performance to be differentiate from a random model.



ROC Curve summarizes the performance of a classifier over all possible thresholds. It is generated by plotting the True Positive Rate (y-axis) against the False Positive Rate (x-axis) as you vary the threshold for assigning observations to a given class. The SVM model.SVM models return an AUC of 0.69 vs 0.66 for the random forest model. We will conclude later on about the

5.Results generalization.

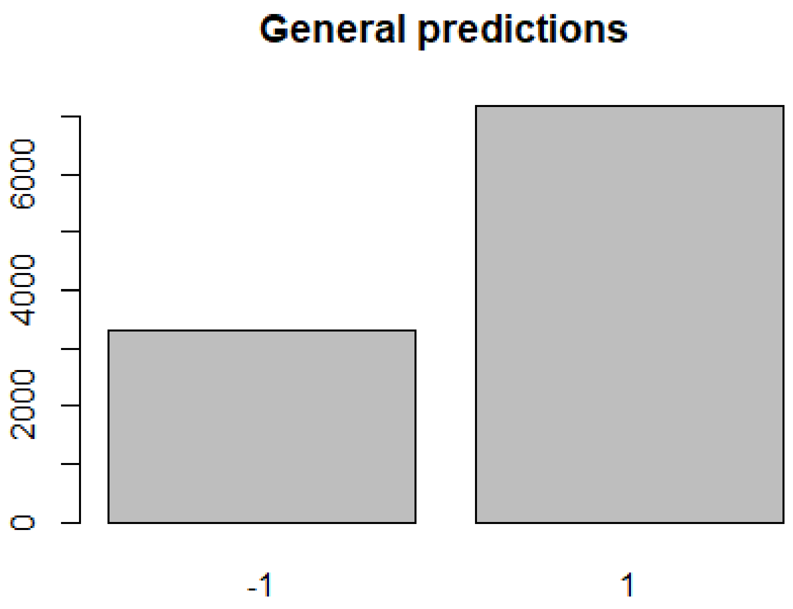
Both models have similar performances and levels of accuracy, svm does better when predicting positives and random forest does it for negatives;After analyzing the SVM and

Random forest model, high test accuracy were achieved for SVM algorithm, having 68% vs 65% for random forest.

Whit a level of confidence of 95%, we can say that this model will be reaching accuracies between 62% and 74%. On the final tweet data set of 10471 tweets, we got 3305 negative sentiment while 7166 positive sentiments.

The results clearly depicts the popularity of Tesla in the social sphere with around 68.5 % people apparently talking positive about Tesla. This gives a good overview of the social buzz for Tesla after the launch of the much anticipated model 3. Knowing the limitations of the model, that it is trained on a limited labelled data set and only been trained on English tweets, the results for final positive and negative sentiments may well vary, but the base model gives a fine picture of the trend.

The negative sentiments can be related to the fact that some users have been experiencing problems with their batteries during the winter season, and no good experiences while trying to reach customer service.



Difficulties and future work

- As we were working with two supervised machine learning model, we needed to perform manual labeling to the tweets in order to train the model and make it learn about the related words. This process is very time consuming and also the sample labeled is not very representative of the tweets data set extracted from the twitter API, it represents around 5% of it, so the model still would have space for more learning and improvements in terms of words recognition.

- The generalization of the model to predict the sentiments of unlabeled tweets is also very limited.Due to the supervised approach, we need to ensure that all the new tweets to evaluate contain the same features for which the model was built, so a lot of words are ignored to fit the model and make the prediction just with the fitted terms. As the modeling approach remained a crucial step to determine the further well-functioning of the entire ML algorithm, we decided to remove the errors by tweaking the problematic variables and reloaded the table into the environment. Major error that we were getting were with 3 words/variables in the final test data set were 'accelerate', 'Tilsburg' and 'next.'. After manipulating the variables, we were able to remove the error and continue to test the already trained model on the final test set.

- In every text mining technique we faced the difficulty related to language ambiguities, specially in social networks when people express themselves as they want and sometimes with some kind of sarcasm especially when talking about controversial topics.

- We were dealing with a big data set, so Regarding processing times it was difficult to perform preprocessing task such as spell check and lemmanization, which could have improved our model inputs and results.

- As a way of improvement for the work, we can think in the analysis of the sentiment over hours and relate the resulting sentiment with the news surrounding Tesla, to better understand the triggers of these sentiments.

- The predictions and generalizations of the model can be improved by adding more terms to the model, and it learn even more about the words and terms used to talk about Tesla in Twitter.

- Although these involves some limitation from Twitter API side, we could think in a way of track the sentiment around Tesla in twitter in real time.

References

Homburg, C. (2015, October). Sentiment in an Online Community Environment.

N. P., & H. Y. (2017, January). SENTIMENT ANALYSIS USING A RANDOM FOREST CLASSIFIER ON TURKISH WEB COMMENTS. Retrieved from https://www.researchgate.net/publication/322374160_Sentiment_analysis_using_a_random_forest_classifier_on_turkish_web_comments

Zainuddin, N., & Selamat, A. (2014, September). Sentiment Analysis Using Support Vector Machine.

Jadav, B., & Vaghela, V. (2016, July). Sentiment Analysis using Support Vector Machine based on Feature Selection and Semantic Analysis. Retrieved from https://pdfs.semanticscholar.org/7746/93175a159160697b5748561446313186b846.pdf

AMRANI, Y., LAZAAR, M., & EL KADIRI, K. (2018, January). Random Forest and Support Vector Machine based HybridApproach to Sentiment Analysis. Retrieved from

Group 10 : TUIRAN Daniela, PALIWAL Harshit , MIRZAEE Saeed

https://reader.elsevier.com/reader/sd/pii/S1877050918301625?token=B6D158452FA68069ADB18DE2A8188A1087522435DC59C3034013312DC7E04A064F3381B9725AD5D34CAEE31C1F0E164E