Sentiment Analysis of tweets on Self-Driving Cars

Abhishek Naik
School of Informatics, Computing and Engineering
Indiana University, Bloomington
IN, 47408, U.S.A.
ahnaik@indiana.edu

Akshada More
School of Informatics, Computing and Engineering
Indiana University, Bloomington
IN, 47408, U.S.A.
akmore@indiana.edu

ABSTRACT

There is lot of research going on in the field of self-driving cars with many big companies such as Tesla, Uber, Google, BMW, etc. investing a lot of money and resources to be at the top of the race. But the main question still remains, are we prepared for self-driving cars? [16]. Are we ready to take a road trip in which we pay more attention to a new book or movie than we do to the driving? [16] Or are we still sceptical to trust the vehicle and hand our life to it? In our project we aim to determine the views of people towards self-driving cars. We intend to perform sentiment analysis on the tweets [**?**], train a model to classify tweets as positive, negative or neutral and then use the trained model to determine the sentiment of the test tweets.

With this classification, we will come to know the attitude of the majority towards self-driving cars. Besides, if we are somehow able to find out the reasons for the negative attitudes, then it will enable the companies involved in this research to take actions so as to overcome the concerns. In the worst scenario, if majority of the people are sceptical towards self-driving cars, then it might be a good indication that the research is headed in the wrong direction and that the energies must be channelized elsewhere.

As a part of this research, we determined the best classifiers to use for this task, trained them on some extracted tweets, tested them on the test data set and then inferred the observations. These results and observations will help us determine the attitude of the people towards self driving cars. Since we also emphasize on the analysis of the negative tweets, we will also come to know the reason for the rejection (if any) of self-driving cars. If the companies working and investing in self-driving cars work upon those points, then there would be a high probability for the negative views of such users to change.

1. RELATED WORK

There has been substantial research in closely related areas. We could find several of them pretty useful that would help us in doing a sentiment analysis on the Twitter data. Out of the many papers we referred, we found 3 of them typically useful.

In 'Techniques for Sentiment Analysis of Twitter Data: A comprehensive study' [6] the authors Mitali Desai and Mayuri Mehta outline the different techniques that can typically be employed for carrying out sentiment analysis. They have used a Twitter data set to describe their analysis. Their comparative analysis helped us understand the pros and cons of using a particular model. We were thus able to make informed decisions about the model that we needed to use for a detailed and precise analysis.

Akshi Kumar and Teeja Mary Sebastian published a paper 'Sentiment Analysis on Twitter' [9] again highlighting the different techniques like corpus based and dictionary based methods that can be employed for such analysis of the Twitter data set. The corpus-based technique was utilized for the analysis of adjectives, while the dictionary-based technique was used for that of verbs and adverbs. This paper provided us with some guidance about how we could utilize similar methods for carrying out an analysis of the Twitter data set to mine public opinion about self-driving cars.

M. Vadivukarassi, N. Puviarasan and P. Aruna in their paper 'Sentimental Analysis of Tweets Using Naive Bayes Algorithm' [**?**] have outlined the general methodology that they had employed for sentiment analysis using the Naïve Bayes (NB) Classifier. They have mentioned a detailed algorithm as well as the related flowchart for carrying out such sentiment analysis. We used this for understanding the methodology that we ourselves had to follow and the typical way in which we could do this. Although we also decided to use the Support Vector Machine (SVM) classifier, it was not difficult for us to just build upon the approach that the authors have mentioned in this paper for the Naïve Bayes classifier.

Also, there are many other research papers discussing sentiment analysis. Turney applied simple unsupervised learning algorithm for classifying reviews as recommended or not [8]. The classification of a review is predicted by the average semantic orientation of the phrases in the review that contain adjectives or adverbs. There are many other papers, which describe different classification techniques for sentiment analysis. Sentiment classification can be thought of as a supervised problem with two class labels (positive and negative). In 'Thumbs Up? Sentiment Classification using Machine Learning Techniques' the authors Bo Pang, Lillian Lee and Shivakumar Vaithyanathan apply supervised learning methods such as Naïve Bayesian and Support Vector

ACM ISBN 978-1-4503-2138-9.
DOI: 10.1145/1235

Machines (SVM) to classify movie reviews into two classes [3].

As far as self-driving cars are concerned, we have seen a tremendous growth in the technology in recent past. Some tweets from people displayed a great excitement about the self-driving car technology [**?**]. At the same time, however, given the long duration this development has been taking, we wanted to know the latest opinion of the people. For e.g., a user Kohei Kurihara recently tweeted "the willingness of UK consumers to pay for vehicle automation is declining" [10]. What do majority of the people think? Has the end user become frustrated over a period of time and given up his hopes? Given the fact that it is the end users who are going to pay, is it still sensible for the companies to heavily invest in this technology? Or, do they need to channelize their energies elsewhere? Finding out what the thoughts of the people were, by performing a sentiment analysis on their tweets on this topic is one of the main aims of our research.

Besides this, for those tweets that are found to be negative, we try to find the reason for their unhappiness. We believe that this information would be helpful to the companies investing in autonomous cars. They would be able to understand the reasons for the negative attitudes and take steps in order to overcome those concerns.

2. DATA EXTRACTION AND PROCESSING

2.1 Data Collection

We used a python library, 'Tweepy' for accessing the Twitter API [13]. 'Tweepy' is an open source python library that can be used to access the Twitter API [**?**]. We then created an application on Twitter. This generated a custom set of secret access keys and token which we use to access Twitter via our python code. We extracted the tweets which contain the hashtags 'self-driving cars' or 'autonomous driving' or both. Twitter API does not allow the retrieval of tweets which are older than 7 days [11]. Since 'self-driving cars' is not a frequently tweeted topic, we were not able to gather enough tweets (only about 800) for the project using the Tweepy API. Sentiment analysis using such a small data set would have resulted in incorrect outcomes. So, we decided to use the 'got3' [7] module developed by Jefferson Henrique for the extraction of the older tweets.

The 'got3' module helps bypass some of the restrictions imposed by the Twitter API. Twitter loads older tweets on its main website when we scroll to the bottom, by making repeated calls to a JSON provider [7]. Jefferson Henrique's 'got3' module automates these calls to the JSON provider.

It provides some additional functionality as follows:

- Getting tweets by usernames;
- Getting tweets by query search;
- Getting tweets by username and bound dates;
- Getting the last 10 top tweets by username;

Using this module, we successfully extracted tweets from 2015, 2016 and 2017. We also ensured that we extracted the tweets not just from the first 6 but also from the remaining 6 months of the years. Using the data varied across a span of three years gave us a better understanding about the views of the people towards self-driving cars/autonomous driving. All in all, we used a total of 6967 tweets for our sentiment analysis. Out of these 6967 tweets, 6167 were extracted using the 'got3' module.

2.2 Data Cleaning

Cleaning (or pre-processing) of the data is one of the most important steps that we had to carry out. The data extracted from twitter was not clean at all and we needed to clean it. It had links to images, 'RT' symbols denoting re-tweets, etc. This cleaning was of utmost importance since using unclean/unprocessed data would have severly impacted the accuracy of the classifier. In case of the NB classifier, it would have been unable to correctly generate the bag of words during the training phase, resulting in incorrect predictions during the testing phase. Similarly in case of the SVM classifier, it would have been unable to correctly generate the hyperplanes during the training phase, consequently resulting in incorrect predictions during the testing phase.

The following things were cleaned from the tweets:

- Clean the start of the tweets (b')
- Clean URLs from the tweets
- Clean usernames from the tweets
- Clean numeric characters from the tweets
- Clean punctuations from the tweets
- Clean encoded strings from the tweets
- Clean '&amp' string from the tweets
- Clean symbols from the tweets

We used regex expressions to clean the tweet text in the data set. We ensured that the tweets are preprocessed correctly and that all the irrelevant tweets (if any) are removed.

3. METHODOLOGY USED

3.1 Labelling the data

Since we wanted to capture the emotion of the user towards self-driving cars, our next aim was to have labelled training data. We typically wanted three classes:

- Positive - denoting a set of tweets from users who were excited and eager about the technology of self-driving or autonomous driving cars;
- Negative - denoting a set of tweets from users who were not excited and had apprehensions about the self-driving car technology.
- Neutral - denoting a set of tweets from users who had no opinion whatsoever. They either tweeted just some facts or some simple questions related to those facts.

Since we have manually extracted the tweets ourselves (as against using ready data sets from Kaggle, etc. wherein it is already labelled), we didn't have already labelled data.

One of the options for this labeling was using emoticons (emotional icons) as an indicator of the sentiments. So, smileys like ":)", ":D" would denote a positive sentiment; ":(",

Table 1: Inter-rater Reliability

Match	13
Total	20
IRR	65%

":'(" would denote a negative sentiment; ":|" would likely denote a neutral sentiment and so on. We decided to manually test this option first and then automate it (using regex) if it works. However after a few manual tests, we concluded that this would not be a good way to classify the tweets, since many tweets related to self-driving or autonomous-driving didn't have any smileys at all.

Other option that we then considered was manual labeling of the data. In this approach, we would manually go through each tweet in the training data set and label it as either positive, negative or neutral as per our human intuition. We didn't consider any syntactical aspects like emoticons, etc. which meant that we couldn't automate it. Besides, since we were labeling the data manually depending upon the considerations of just two people, we wanted to know the actual class-conformance of a particular tweet. For example, if one individual considers a tweet to be positive, then would someone else also think that to be positive? Or is there a strong possibility that he/she might consider that to be negative? This was essential in order to know if the data could actually be labelled by just two people. If this class-conformance due to two individuals was found to be low, then we might have to carry out this labelling of the tweets on a broader scale and then go with the majority outcome. Taking all these points into consideration, we decided to calculate the inter-rater reliability. Inter-rater reliability (IRR) is defined as the degree of agreement among different raters [19]. We wanted to do this to ensure that our intuition about this classification was correct. If the IRR would be high enough, then we would go ahead with this approach; else we might have to do some research on other options. So, both the team members carried out the labelling of 20 tweets independently. The matches and the non-matches were used to calculate the IRR percentage as follows:

This percentage of IRR meant that it was obvious enough (for humans) to classify the tweets manually. So, we decided to go ahead with the manual classification of the training data set.

4. CLASSIFICATION

Next, for classification purposes, we used the Naïve Bayes (NB) and Support Vector Machine (SVM) classifiers. Out of many options like Naïve Bayes, Support Vector Machine, Logistic Regression, etc. we chose Naïve Bayes and SVM for the following reasons:

4.1 Reasons for choosing Naïve Bayes [4]:

- We had comparatively less volume of data for training purposes (just 875 records).
- Our classification task at hand was simple, given that we had to count the number of records.
- Its disadvantage is that it cannot learn the dependencies among the features; but we didn't have any such dependencies among the elements in the data set.

4.2 Reasons for choosing SVM [4]:

- It has pretty good accuracy.
- It is robust in the sense that it can work well even if the data is inseparable.
- One of its drawbacks is that it is memory intensive, but given the fact that we had comparatively less records, memory requirement was not a concern.

Given the above advantages of the two classifiers, we chose them for the classification purposes. Their usage in the project can be described as below:

4.3 Naïve Bayes

We are using the Naïve Bayes classifier [5] [15] [21] which is a classifier based on Bayes theorem [18]. Bayes Theorem can be given as follows [18]:

$$P(\frac{A}{B}) = \frac{P(\frac{B}{A})P(A)}{P(B)}$$

where, A and B denote events and $P(B) \neq 0$. $P(A|B)$ is a conditional probability, denoting the probability that A occurs, given that B has already occurred. Similarly, $P(B|A)$ is again a conditional probability, denoting the probability that B occurs, given that A has already occurred. $P(A)$ and $P(B)$ are the probabilities of both A and B occurring independently.

We use the Multinomial Bayes theorem [14] since multiple occurrences of the words matter while performing sentiment analysis. Naïve Bayes is basically a bag of words model. In such a model, we check to see if a given word appears in the positive-word list, negative-word list or the neutral-word list. The score of a particular tweet is calculated based on the occurrence of the words in the respective word lists. In the end, the total score of the tweet is calculated. If the score is positive, then the tweet is classified as a positive tweet; if the score is negative, then the tweet is classified as a negative tweet; if the score is zero, then the tweet is classified as a neutral tweet. Also, it is important to note that in this classifier, we do not use a small subset of words — we use all the words from the training data set. During testing, if some word appears that has not appeared before in the training data set, then the classifier has no data available to classify it and thus, uses Laplacian Smoothing [2] [20].

We used the Multinomial Bayes classifier from scikit learn. We first trained the classifier in order to enable it to identify and classify the unforeseen tweets. Our training data set consists of 875 tweets and our testing data set consists of 200 tweets. The accuracy of Naïve Bayes model on training data set is around 85.48% and the accuracy of Naïve Bayes model on testing data set is around 54.72%.

4.4 Support Vector Machine

The other classifier that we used is the Support Vector Machine [17]. A SVM is a supervised learning model that can be used for classification purposes. It is a non-probabilistic binary linear classifier [23]. A SVM tries to determine a 'hyperplane' that would segregate the classes (in our case, positive, negative and neutral) in the best possible way. In order to determine the best hyperplane, the SVM classifier needs to do two things — determine the set of hyperplanes

that carry out a better task of segregation; and then determine the hyperplane from this set, that has the highest margin. This final hyperplane is the hyperplane that would be used later for classification of the test data [1].

Just as in the Naive Bayes approach, we first trained the classifier in order to enable it to identify and classify the unforeseen tweets. Our training data set consists of 875 tweets and our testing data set consists of 200 tweets. The accuracy of SVM on training data set is around 93.02% while its accuracy on the testing data is around 52.74%.

At the same time, however, it is important to note that our classifier is not able to classify negative tweets. One of the reasons for this is that the words in positive, negative and neutral tweets is the same. Hence it becomes difficult for the classifier to predict the sentiment of negative tweets. This has resulted in low accuracy.

Another reason for the low accuracy is that the training data set is not balanced among all the three sentiment classes. The number of negative tweets are pretty less compared to the positive and neutral tweets. Hence, we are not able to train the model efficiently to classify the negative tweets.

In addition to all this, since we have an IRR of around 65%, there is some difference in how even we humans could interpret the tweets. As such, this difference was propagated in the training imparted to the two classifiers, which was in turn propagated eventually to the test data set as well. This again reduced the accuracy.

Algorithmically, we can represent the steps carried out (after tweet extraction) as below:

1. Preprocessing:
   (a) Reading the input file in a Pandas data frame. Removing the NaN (not a number) rows from this data frame (considering only the even numbered rows).
   (b) Cleaning the tweet text by removing URLs, usernames, numeric characters, punctuations, encoded strings, '&' symbol from the tweets.
2. Training:
   (a) Used a vectorizer to convert the tweet text into a bag of words representataion.
   (b) Removed the stop words.
   (c) Words occurring in less than 2 tweets were treated as noise and hence removed.
   (d) Similarly, words occurring in more than 90% of the tweets were also treated as noise and were removed.
   (e) Training of the Naïve Bayes and SVM models was carried out onn the set of first 875 tweets.
3. Testing (or prediction) of tweets:
   (a) Predicted the sentiments of all the 6167 tweets in the data set using the trained Naïves Bayes and SVM models.
   (b) Predicted the accuracy score on the train data set.
   (c) Predicted the accuracy score on the test data set (200 manually labelled tweets).
4. Negative tweet analysis:
   (a) Collected all the negative tweets for a detailed analysis.
   (b) Tokenized the negative tweets using word_tokenize from nltk.tokenize [12]
   (c) Removed stop words from the tokenized text.
   (d) Created a dictionary of the remaining words and their respective count. This dictionary contains the major words that denote fear or negative emotions of the users towards self-driving cars.

5. VISUALIZATIONS AND INFERENCES

The 4 images show the accuracy of the Naïve Bayes Model on the training and testing data as well as that of the SVM on the training and testing data.

From the overall visualizations, we can infer that accuracy of the classifiers is low. One of the main reason for this is that the positive and negative tweets make use of similar words. This reason is responsible for the negative tweets going undetected in the training as well as the testing data sets. Other reason, as far as the SVM is concerned, is that because of the similar words in the tweets, the SVM classifier is unable to find hyperplanes that could precisely classify the test data sets. Due to this, the predictions are sometimes incorrect.

At the same time, however, we can also observe that the number of positive tweets are significantly higher compared to the negative and neutral tweets. This means that the majority of the people are positive and look forward to the evolution of the self-driving cars technology.

5.1 Observations of NB:

We can observe that the NB classifier had good accuracy on the training data. However, as seen in the graphs, the training data was not evenly distributed - the number of positive tweets were significantly more than the neutral and negative tweets. Due to this, the classifier could not be trained well in the first place.

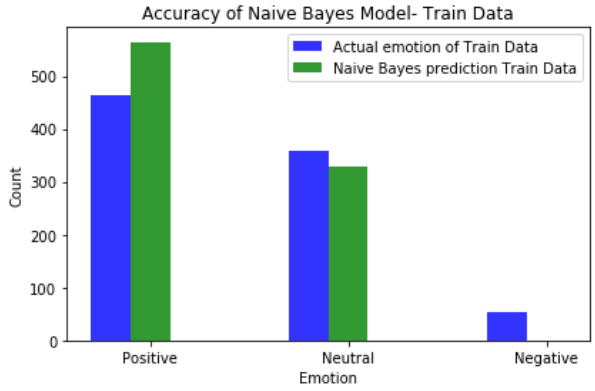
Although the classifier could detect and classify the positive and neutral tweets, it couldn't identify any negative tweets during the training phase. This was because the positive and negative tweets were pretty similar to each other. The bag of words representing the positive and the negative tweets were similar to each other and so the classifier was unable to determine the class for many tweets. This incorrect classification error manifested itself when we tested the classifier on the test data set.

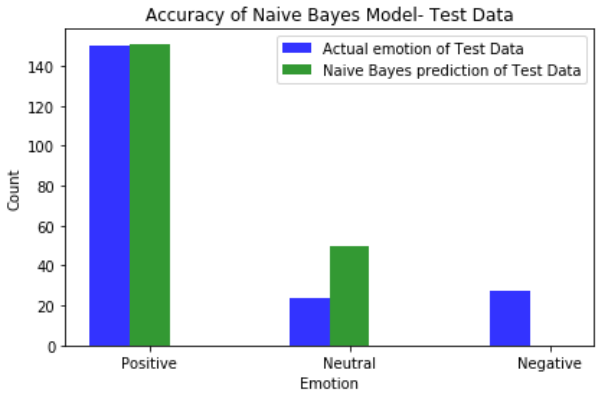
This resulted in low accuracy of the Naïve Bayes classifier.

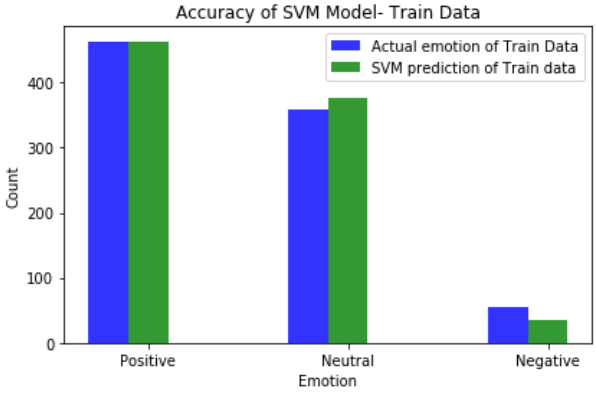
5.2 Observations of SVM:

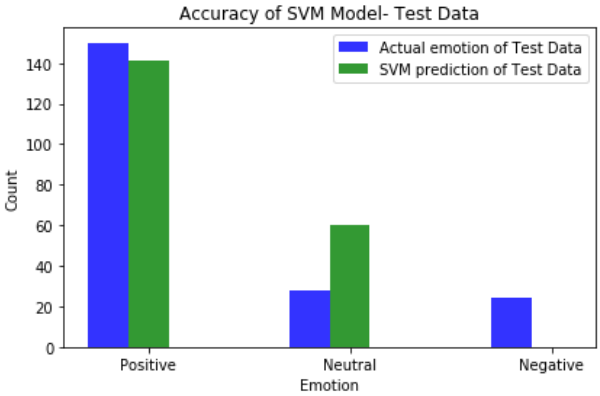
We can observe that, just like in case of NB, SVM too had good accuracy on the training data. However, this training data was again not evenly distributed into positive, negative and neutral tweets - the number of positive and neutral tweets was more than that of the negative tweets.

Although in case of SVM, the classifier could classify the tweets as positive, negative and neutral during the training phase, lack of negative tweets resulted in poor training for the classifier. It was unable to determine robust hyperplanes that could correctly classify the tweets into the three classes. Since it was poorly trained, it got confused when it saw new tweets in the test data set and performed poorly.


Accuracy of Naive Bayes Model- Train Data
Actual emotion of Train Data
Naive Bayes prediction Train Data
Count
0
100
200
300
400
500
Positive
Neutral
Negative
Emotion


Accuracy of Naive Bayes Model- Test Data
Actual emotion of Test Data
Naive Bayes prediction of Test Data
Count
0
20
40
60
80
100
120
140
Positive
Neutral
Negative
Emotion


Accuracy of SVM Model- Train Data
Actual emotion of Train Data
SVM prediction of Train data
Count
0
100
200
300
400
Positive
Neutral
Negative
Emotion


Accuracy of SVM Model- Test Data
Actual emotion of Test Data
SVM prediction of Test Data
Count
0
20
40
60
80
100
120
140
Positive
Neutral
Negative
Emotion

This resulted in low accuracy for the Support Vector Machine (SVM) classifier.

5.3 Observations about negative tweets:

As mentioned before, one of the main aims of our research is to find out the reasons for the negative tweets. We do this at the end, by displaying the main words in the negative tweets along with the frequency with which they appear.

For e.g., the word 'vulnerable' appears with a frequency of 2. Simiarly, words like 'park' appears with a frequency of 4, while 'lives' appears with a frequency of 2. The companies can use this information to infer things like quite a few people are still concerned about their lives and have parking-related concerns. These companies can then do targeted blogs or advertisements about how they address these issues in their cars.

6. STATISTICAL RIGOROSITY

In order to ensure that the data we collected is relevant to our research, we extracted only those tweets that had either the hashtags 'SelfDrivingCar', 'AutonomousCar' or both. This ensured that irrelevant tweets were not present in the data set in the first place.

Besides, we included the tweets from people belonging to all regions and of different ages, sex and economic background. This ensured that the data set was not biased in any form. We also carried out manual labeling of the training data set, after calculating the IRR value.

In order to ensure that the number of tweets we extracted were statistically rigorous, we extracted all the tweets with the above hashtags from 2015 to present. They were cleaned in order to ensure that the classifiers could correctly carry out the classification. The training of the classifiers was carried out on a set of 875 tweets, while 200 tweets were used for testing.

That being said, our classifiers were not smart enough to detect sarcasm or humor. This is because, our training data did not have sarcastic or humorous tweets. As such, if they encountered any tweets that were sarcastic or humorous in nature during testing, then they were bound to fail.

7. CONCLUSIONS

To summarize, we carried out twitter sentiment analysis of the tweets related to Self-driving cars. We found out that majority of the people are positive about self-driving cars and that they are looking forward to it. Given this observation that we made, it might be well worth the effort to continue with the research in this domain.

The companies would also be able to find out the reasons for the negative attitude of some people. They would then be able to take corrective steps in order to mitigate those negative opinions.

At the same time however, it should be noted that our classifiers do not perform well since the positive and negative tweets contain similar words. It does not even consider the order of the words in the tweets. Future research should focus on using models that will work better in such conditions and also take into account the "connectedness" factor.

8. FUTURE RESEARCH

As mentioned above, since the positive and negative tweets contain similar words, our classifiers perform poorly. Recurrent Neural Networks will take into account the relationship among different words and help in better classification. So, in future, we can consider to carry out this classification task using Recurrent Neutral Networks.

Another area of future research would be related to more detailed analysis of the negative tweets. Actual testing of self-driving cars on the roads has already been started in some parts of the United States, like Arizona [24]. Studying the tweets (especially the negative ones) from users in such parts of the United States like Arizona, might help discover new concerns or issues that the users face on the roads. It will also help us understand their attitudes towards these experiments. If the results are found to be positive, then it will be a good indication to the companies to start such testing in other states as well. If the results are found to be negative, then it will be a strong indicator about the things that make the people apprehensive. Their feedback can be used for better targeted blogs and marketing. Besides, it will also help unearth some different aspects that are related to self-driving cars, although not directly. For e.g., the way the roads should be constructed, the concerns that might arise if there are manual drivers in a lane full of autonomous cars, etc. Thus, such analysis of tweets especially from the users in states wherein testing is actually going on will help get insightful feedback for better development of not just the self-driving cars, but also other aspects that are related to self-driving cars in some way or the other.

To conclude, we believe that our research would provide a strong motivation for future research.

9. CONTRIBUTIONS

We carried out the project planning and the problem solving tasks together. As far as planning is concerned, we had to decide upon the classifiers to be used as well as the manually label the training data set. Once the tasks were planned, we carried out the activities individually, although still complementing each other as and when required. The task distribution was carried out as follows:

Akshada - Carried out extraction and pre-processing of the data and coded the Naïve Bayes classifier. Also helped in documenting the report in LaTeX.

Abhishek - Coded the Support Vector Machine classifier, and visualizations of the training and testing data. Also helped in documenting the report in LaTeX.

10. REFERENCES

[1] AnalyticsVidhya. https://www.analyticsvidhya.com/blog/2017/09/understaing-support-vector-machine-example-code/. Webpage.

[2] Ataspinar. http://ataspinar.com/2016/02/15/sentiment-analysis-with-the-naive-bayes-classifier/. Webpage.

[3] S. V. Bo Pang, Lillian Lee. In *Thumbs up? Sentiment Classification using Machine Learning Techniques.*

[4] E. Chen. http://blog.echen.me/2011/04/27/choosing-a-machine-learning-classifier/. Webpage.

[5] Datumbox. http://blog.datumbox.com/machine-learning-tutorial-the-naive-bayes-text-classifier/. Webpage.

[6] M. Desai and M. A. Mehta. Techniques for sentiment analysis of twitter data: A comprehensive survey. In *2016 International Conference on Computing, Communication and Automation (ICCCA)*, pages 149–154, April 2016.

[7] J. Henrique. https://github.com/jefferson-henrique/getoldtweets-python/tree/master/got3. Webpage.

[8] A. Kumar and T. M. Sebastian. In *Thumbs Up or Thumbs Down? Semantic Orientation Applied to Unsupervised Classification of Reviews.*

[9] A. Kumar and T. M. Sebastian. Sentiment analysis: A perspective on its past, present and future. In *I.J. Intelligent Systems and Applications*, 2012.

[10] K. Kurihara. https://twitter.com/kuriharan/status/930080053240537088. Webpage.

[11] Luigi. https://stackoverflow.com/a/24246840/2172854. Webpage.

[12] NLTK. http://www.nltk.org/api/nltk.tokenize.html. Webpage.

[13] J. Roesslein. http://docs.tweepy.org/en/v3.5.0/. Webpage.

[14] scikit-learn developers. http://scikit-learn.org/stable/modules/generated/sklearn.naive$_b$*ayes.multinomialnb.html.Webpage.*

[15] scikit-learn developers. http://scikit-learn.org/stable/modules/naive$_b$*ayes.html.Webpage.*

[16] I. The Conversation US. https://theconversation.com/self-driving-cars-are-coming-but-are-we-ready-81538. Webpage.

[17] Wikipedia. http://scikit-learn.org/stable/modules/svm.html. Webpage.

[18] Wikipedia. https://en.wikipedia.org/wiki/bayes%27$_t$*heorem.Webpage.*

[19] Wikipedia. https://en.wikipedia.org/wiki/inter-rater$_r$*eliability.Webpage.*

[20] Wikipedia. https://en.wikipedia.org/wiki/laplacian$_s$*moothing.Webpage.*

[21] Wikipedia. https://en.wikipedia.org/wiki/naive$_b$*ayes$_c$lassifier.Webpage.*

[22] Wikipedia. https://en.wikipedia.org/wiki/recurrent$_n$*eural$_n$etwork.Webpage.*

[23] Wikipedia. https://en.wikipedia.org/wiki/support$_v$*ector$_m$achine.Webpage.*

[24] Wired. https://www.wired.com/story/mobileye-self-driving-cars-arizona/. Webpage.