CmpE492 Special Project Final Report

SENTIMENT ANALYSIS IN TURKISH BY STREAMING TWITTER

Furkan ŞENHARPUTLU
2013400171

Advisor: Ali Taylan CEMGİL

Coordinator: Lale AKARUN

SPRING 2018

1. Introduction and Motivation

My project is about analyzing current status of a person or institution or organization on society by using Twitter. What do people think about an institution? Is the general judiciary positive or negative? Or neutral? Twitter is a platform where people can instantly share their thoughts. Users around the world post around 350,000 new tweets every minute. This may create a big data about related person or institution or organization. When this data is analyzed emotionally, the result represents the general thought about that subject. Therefore, Twitter is now a hugely valuable resource from which you can extract insights by using sentiment analysis.

There are many libraries making sentiment analysis on English. However, it is not common for Turkish. My purpose is to create a sentiment analyzer for Turkish. Turkish has its own word structure so it is necessary to make stemming special to Turkish.

This is a rather applicable project. This technique can be used in different areas. Many institutions wonder how people think about them. Or, somebody in politics may wonder what people think about him/her. It is well-known that U.S.A President Donald Trump gives importance to Twitter data. For example, recently in Turkey, a politician may wonder how Afrin Military Operation affects him/her. It can be made by streaming Twitter data about Afrin and the politician. Or, a food company can wonder what people think about it and its competing company. Thus, the project is applicable on market and I think to improve this project.

2. State of the Art

There are many companies making sentiment analysis over Twitter. They provide support other companies. Also, there is a wide variety of tools that are used in sentiment analysis. These tools can be classified in several different ways. One way to classify the tools is through the different techniques that are used for sentiment analysis.

Some of the commercial tools and services used include:

Brandwatch: It provides tool or service based on machine learning. 500 classifiers are used across different languages to ensure accurate automation of the sentiment categorization process.

Laxalytics: It provides several different tools and services for sentiment analysis mostly focusing on NLP and text analysis tools.

Sentdex: It uses a bot to pull data or news from the internet through NLP. The data is analyzed and sentiment is derived from the text.

Clickworker: It uses a mix of human and machine learning for a variety of tasks related to sentiment analysis including; web research, keyword assignment, and categorization, product data management etc.

Although not for the start, as the dataset grows in the future, the stream operation will require a lot of workload. So the data will be very big. To stream data quickly and analyze it faster, the database system needs to be set up well. Hazelcast Company has a product called JET. It works very well when streaming because it works in Memory, it provides very fast access. Large data can be easily stored without loss because it is a distributed system. If this project is used and the amount of data increases, it would be very logical to use Hazelcast IMDG.

Although there are many tools and companies making sentiment analysis in English, there is no enough work in Turkish. My main purpose is to apply sentiment analysis on Turkish.

3. Methods

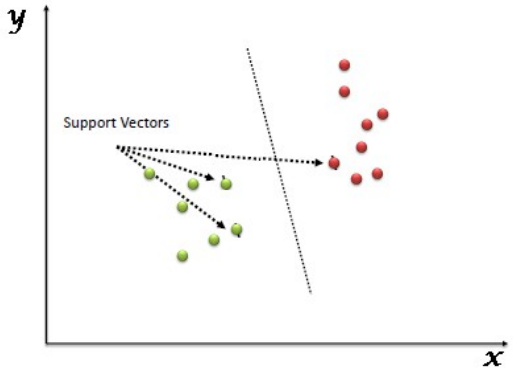
3.1. K Nearest Neighbors

The k-nearest neighbors algorithm is used around the simple idea of predicting unknown values by matching them with the most similar known values. Before we can predict KNN, we need to find some way to figure out which data rows are closest to row we are trying to predict on. A simple way to do this is Euclidean distance. The formula is:

$$\sqrt{(q_1 - p_1)^2 + (q_2 - p_2)^2 + \cdots + (q_n - p_n)^2}$$

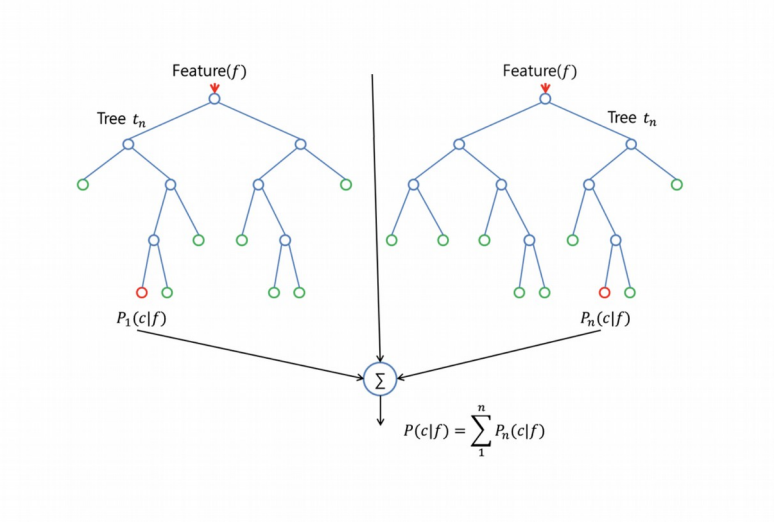
3.2. Linear SVM

“Support Vector Machine” (SVM) is a supervised machine learning algorithm which can be used for both classification or regression challenges. However, it is mostly used in classification problems. In this algorithm, we plot each data item as a point in n-dimensional space (where n is number of features you have) with the value of each feature being the value of a particular coordinate. Then, we perform classification by finding the hyper-plane that differentiate the two classes very well (look at the below snapshot).



3.3. Decision Tree

Decision Tree Learning uses a decision tree (as a predictive model) to go from observations about an item (represented in the branches) to conclusions about the item's target value (represented in the leaves). It is one of the predictive modelling approaches used in statistics, data mining and machine learning.

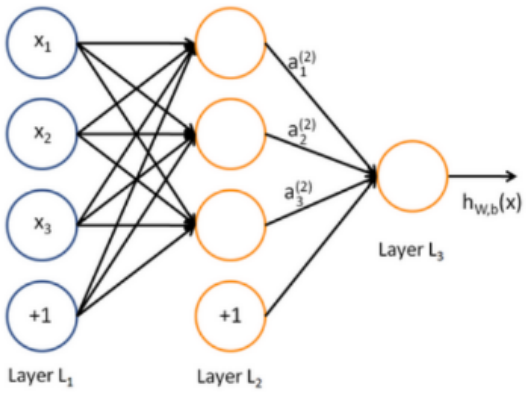


3.4. Random Forest

Random Forest is an ensemble learning method for classification, regression and other tasks, that operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees. Random decision forests correct for decision trees' habit of overfitting to their training.

3.5. Neural Networks

Neural networks are immensely powerful simply because they represent every word as a vector and an operator. It seems very intuitive. Thus the word "not" can be a rotation matrix that acts on the next word (for eg. good) and changes it's polarity by rotating the vector of good to now mean not good. This is a very powerful concept. However these networks require a lot of training data (parse trees are required to train these networks).



Neural Network

3.6. AdaBoost

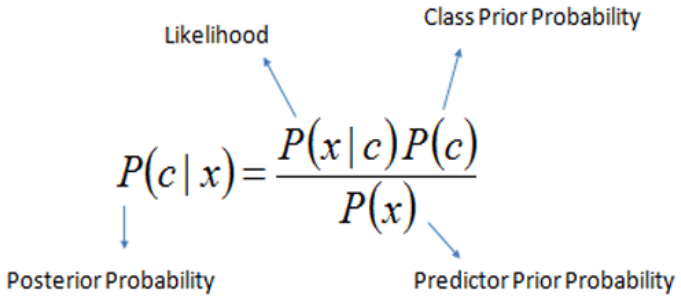
AdaBoost classifier is a meta-estimator that begins by fitting a classifier on the original dataset and then fits additional copies of the classifier on the same dataset but where the weights of incorrectly classified instances are adjusted such that subsequent classifiers focus more on difficult cases.

3.7. Naive Bayes

In simple terms, a Naive Bayes classifier assumes that the presence of a particular feature in a class is unrelated to the presence of any other feature. For example, a fruit may be considered to be an apple if it is red, round, and about 3 inches in diameter. Even if these features depend on each other or upon the existence of the other features, all of these features independently contribute to the probability that this fruit is an apple and that is why it is known as 'Naive'.

Naive Bayes model is easy to build and particularly useful for very large datasets. Along with simplicity, Naive Bayes is known to outperform even highly sophisticated classification methods. Bayes theorem provides a way of calculating posterior probability *P(c|x)* from *P(c),P(x)* and *P(x|c).*

Look at the equation below:



$$P(c \mid \mathrm{X}) = P(x_1 \mid c) \times P(x_2 \mid c) \times \cdots \times P(x_n \mid c) \times P(c)$$

Naive Bayes has Pros and Cons.

Pros:

- It is easy to predict class of test data set. It also performs well in multi-class prediction.
- When assumption of independence holds, a Naive Bayes classifier performs better compare to other models like logistic regression and you need less training data

Cons:

- If categorical variable has a category which was not observed in training dataset, then the model will assign a zero probability and will be unable to make a prediction. To solve this, we can use the smoothing technique. One of the simplest smoothing techniques is called Laplace estimation.
- **In, the assumption of independent predictors is almost impossible.**

3.8. My Model

I used an ensemble learning approach. I used a formula to classify the tweets. I found accuracies of each classifiers and used their weighted sum to determine labels. Here is the accuracies:

Nearest Neighbors	57.16388616290480
Linear SVM	52.79685966633954
RBF SVM	55.34838076545632
Decision Tree	58.48871442590775
Random Forest	52.79685966633954
Neural Network	64.27870461236506
AdaBoost	58.68498527968596
Naive Bayes	58.24337585868498
Overall	65.09619159795839

My model's accuracy is **65.1 %.**

For my experiments, I needed to use a train data. For train data, I used Twitter API. However, Twitter does not give old tweets than a week. One has to spend money to access older tweets. I was searching a tool that makes easy to access older tweets, eventually I found a Github Repository which makes this job. I entered the start date and end date then I got all tweets related with my search easily. My first search was *'torku çikolata'* and the second one was *'ülker çikolata'*.
I chose *Start Date 2015-03-27* and *End Date 2018-03-27.* Thousands of tweets were written. I printed all results on a txt file and put a tab space at the beginning of each line. Then, I have started to mark tweets as positive, negative and neutral. Still, I am making this marking and I will continue to it until final submission because the more I mark, the better I will get as result.

There is no problem in terms of ethics because Twitter shares its data with developers and people accept this while creating their accounts. Also, when people are asked, they like these experiments because their thoughts change something.

I have used different classifier for classification. However, before classification, I applied a preprocessing to data to increase the performance of classifiers. My preprocessing steps are:

- case-folding
- removing punctuation
- removing stop words
- stemming

Here, it is easy to make case-folding and removing stop words in Turkish. However, it is hard to make stemming easily because Turkish has a different language structure from English. I have searched a library to make a stemming in Turkish. After some research, I found an open source Github Repository making this job.

4. Results

Data coming from social media is as expected hard to work with. I tried using stemming and stopword removal. Neither made a huge effect, roughly speaking, each increased accuracy about 0.5 percentage point.

I have seen that result of different classifiers are quite parallel on the data so using multiple of them only ensured no unexpected failures occur.

What I could have done for better performance:

- I could use positive and negative words lists. I could tokenize the input text, remove stop words, normalize the text and then refer to the list of words, also take care of negations and positive words and then calculate the score accordingly.
- I could have chosen better coefficients for votes of classifiers. I now see that our coefficients were faulty.

5. Conclusion and Discussion

I believe that the results are not bad. There is certainly room for improvement but considering the fact that documents (tweets) are usually quite short, which reduces performance of all sorts of classifiers, the results are good.

When it comes to potential impact of my project, I want to mention about crisis management. Let’s say you have a company. Crisis management is important because it tells your customers that you care. More importantly, active crisis management indicates to your clients that you as a brand are committed to staying accountable for the problems they face. Companies that hide are deemed less trustworthy than companies that go out there and face the crisis head-on. Therefore, Brands that do not handle social media crises well usually pay a high price for it. On the flip side, brands that do handle these crises well, develop new loyal customers.

6. Future Work

- I will mark more tweets to improve the performance of classifiers.
- I will try Deep Learning.
- I will use Word2Vec approach.

7. References

- https://twitter.com/
- https://github.com/Jefferson-Henrique/GetOldTweets-python
- https://www.analyticsvidhya.com/blog/2017/09/naive-bayes-explained/
- https://github.com/otuncelli/turkish-stemmer-python
- http://blog.aylien.com/build-a-sentiment-analysis-tool-for-twitter-with-this-simple-python-script/
- https://towardsdatascience.com/machine-learning-nlp-text-classification-using-scikit-learn-python-and-nltk-c52b92a7c73a
- https://www.dataquest.io/blog/k-nearest-neighbors-in-python/
- https://www.marketmotive.com/blog/discipline-specific/social-media/sentiment-analysis-article
- https://machinelearningmastery.com/boosting-and-adaboost-for-machine-learning/
- https://medium.com/machine-learning-101/chapter-3-decision-trees-theory-e7398adac567
- http://mines.humanoriented.com/classes/2010/fall/csci568/portfolio_exports/lguo/decisionTree.html