

Machine Learning Engineer Nanodegree

Capstone Proposal

Challa Chaitra

February 15th, 2019

Proposal

Tweets-Sentiment Analysis

Domain background

Sentiment analysis is the process of determining the emotional tone behind a series of words which is used to gain an understanding of the attitudes, opinions and emotions expressed within an online mention. It is extremely useful in social media monitoring as it allows us to gain an overview of the wider public opinion behind certain topics.

The applications of sentiment analysis are broad and powerful. The ability to extract insights from social data is a practice that is being widely adopted by organisations across the world.

Twitter Sentiment Analysis may, therefore, be described as a text mining technique for analyzing the underlying sentiment of a text message, i.e., a tweet. Twitter sentiment or opinion expressed through it may be positive, negative or neutral.

It has a number of applications: In business, Companies use Twitter Sentiment Analysis to develop their business strategies, to assess customers' feelings towards products or brand. In politics, used to keep track of political views, to detect consistency and inconsistency between statements and actions at the government level etc.

Sources:

<https://www.brandwatch.com/blog/understanding-sentiment-analysis/>

<https://www.digitalvidya.com/blog/twitter-sentiment-analysis-introduction-and-techniques/>

Problem Statement

The aim of this project is to correctly classify tweets from the dataset as either positive or negative.

I am going to use various machine learning classifiers for this.

Datasets and Inputs

The dataset I am working with has been downloaded from

<https://datahack.analyticsvidhya.com/contest/practice-problem-twitter-sentiment-analysis/>

It consists of around 31k rows and 3 columns. And the features are

Id: It is of integer type and starts from 1.

Label: It has only two values viz., 0(negative tweet) and 1(positive tweet)

Tweet: It is the text or message which is a mixture of alphabets, numbers and special characters.

Here, label is the target variable and the rest are the features.

This dataset is open-sourced and can be used without any citations.

Solution Statement

Here, I am going to build different classifier models using various machine learning techniques, train them individually on the dataset and finally choose the one with better performance (of all these models) as the best model.

Benchmark Model

In this project, I want to use K-nearest neighbours as my worst case benchmark model. Any machine learning model that performs better than this model is treated as good one for the sentiment analysis of tweets.

Evaluation Metrics

There are various metrics available for evaluating the performance of the classifiers like accuracy, precision, recall etc.

Accuracy can be simply defined as the ratio of correctly predicted observations to the total observations.

$$\text{Accuracy} = \frac{TP+TN}{TP+FP+FN+TN}$$

TP-True Positives

TN-True Negatives

FP-False Positives

FN-False Negatives

Precision is the ratio of correctly predicted positive observations to the total predicted positive observations.

$$\text{Precision} = \frac{TP}{TP+FP}$$

Recall is the ratio of correctly predicted positive observations to the all positive observations in actual class.

$$\text{Recall} = \frac{TP}{TP+FN}$$

F1 Score is the weighted average of Precision and Recall. Therefore, this score takes both false positives and false negatives into account.







$$\text{F1 Score} = \frac{2 * (\text{Recall} * \text{Precision})}{(\text{Recall} + \text{Precision})}$$

Accuracy works well when the dataset is balanced. But when it comes to imbalanced dataset, F1 score evaluates better. So, in this project, I use these two metrics for evaluation.

Project Design

The workflow to get to the solution for the above stated problem can be as follows:

- First, we should read the dataset and inspect it to get to know about things like
 - how big the dataset is

-  whether the dataset is imbalanced
- And then data pre-processing which includes
 -  Removing unwanted data like short words and stop-words
 -  Normalizing the text
- Now, to analyse a pre-processed data, it needs to be converted into features. Depending upon the usage, text features can be constructed using different techniques like
 -  Bag of Words
 -  TF-IDF
 -  Word Embeddings.
- Now, we will split the data into training and testing sets for classification
- We train various models using machine learning classifiers like knn, naive bayes, decision tree, logistic regression etc and evaluate their performance.
- And the model which performs the best will be chosen as the final one.