Machine Learning Engineer Nanodegree

Avnish Srivastava March 28th, 2019

1.1 Capstone Proposal

1.1.1 Domain Background

Natural language processing (NLP) is a field of computer science, artificial intelligence and computational linguistics concerned with the interactions between computers and human (natural) languages and concerned with programming computers to fruitfully process large natural language corpora.

The project predicts whether reviews are positive or negative also considered as "Sentiment Analysis". Sentiment Analysis is the process of computationally identifying and categorizing opinions expressed in a piece of text, especially to determine whether the writer's attitude towards a topic, product, etc. is positive, negative, or neutral through ML-based and Lexicon-based. In ML-based sentiment analysis algorithms, it requires you to create a model by training the classifier with a set of examples. This ideally means that you must gather a dataset with relevant examples for positive and negative classes, extract these features from the examples and then train your algorithm based on these examples. These algorithms are essentially used for computing the polarity of a document.

1.1.2 Problem Statement

It is important for companies to understand the sentiments of their customers or clients to remain competitive and grow their businesses. Traditionally this depended upon feedback surveys and informal, small-scale and often inefficient methods. Today, however, there is great deal of information on social media and e-commerce platforms related to customer's experiences with products and services. This is available for companies to analyse and better understand their customer's experience and satisfaction.

Accurately identifying the sentiment in a body of text requires accuracy (to be useful) and automation (to deal with the scale and nature of data available today). Machine learning models provide one solution to this problem.

Build a machine learning model using reviews about companies' products and services that will predict whether reviews are positive or negative.

1.1.3 Datasets and Inputs

Here we have collected reviews for various products and services from different sources written between December 2016 and March 2017.

The training dataset is divided into positive and negative reviews:

```
Positive: 32470 reviews

Negative: 7256 reviews
```

Data Dictionary:

Each review is in JSON format and contains the following fields:

- 'author': author of this review
- 'crawled': site from where it is crawled
- 'entities': if any
- 'external_links': if any links
- 'highlightText': if any
- 'highlightTitle': if any
- 'language': language of the review
- 'locations': if tracked of the author
- 'ord_in_thread': if any
- 'organizations': organization of author
- 'persons': if any
- 'published': if any
- 'text': review written by author
- 'thread': thread of review
- 'title': title of review
- 'url': link of review
- 'uuid': id of review

1.1.4 Solution Statement

The model will be either a logistic regression or weighted aggregation of multiple models which may or may not include SVM, Random Forest, XGBoost after TF-IDF vectorizer. Depending upon the MCC obtained we will decide whether a single model does better performance, or an ensemble performs better.

The approach will be to leverage the power of machine learning model to outperform the basic model which will not have cleaning datasets

1.1.5 Benchmark Model

The benchmark model will be a Multinomial Naive Bayes Bag of Words model which will be running on the datasets with the 'text' and 'reviews' column and we will compare this model to our developed model using MCC as metric.

The bag of words model will be run after pre-processing of text data which includes Count Vectorizer. With this we are trying to prove that cleaning and data preparing helps us better in the prediction of matches.

1.1.6 Evaluation Metrics

Project will be scored using **Matthews Correlation Coefficient (MCC)**

MCC is defines as follows:

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}}$$

MCC ranges from -1 (worst, complete disagreement between prediction and observation) to +1 (complete concordance between prediction and observation) with a value of zero indicating that the model is no better than random prediction. Read more about Mathews Correlation Coefficient **here**

Why MCC?

1. MCC is considered one of the best single number representations of the confusion matrix - the formula for MCC considers all four cells in the confusion matrix.

2. MCC is particularly well-suited to unbalanced data sets.

3. MCC does not depend on which class is defined as positive and which one as negative.

1.1.7 Project Design

The input data is in JSON format so JSON files will be first extracted and convert it into panda’s data frame with 'text' and reviews (1 - positive, 0 - negative) as columns. I have ignored other columns since there are multiple null values present in the dataset. Also, the main focussed variable is 'text' only where customers have their written their reviews. The customer's reviews are not cleaned, so they need to be cleaned with certain pre-processing steps like removing special characters and punctuations, removing HTML tags, removing contraction, removing English stop words with added custom stop words, lemmatization, etc.

Apart from above, we will also see the top common and rare word based on the frequency count and if they are not useful then we will remove those words. There will be certain feature engineering as well, like adding new features ex: Count of Words, count of sentence, etc and will see the performance of model with or without these features.

Only around 18% data have negative reviews which indicated that that data is highly imbalanced, so once the data is cleaned it will be divided into training and validation set (Training will have 70% of data, validation will have 30% of data). The data will be converted into document term matrix using TF-IDF vectorizer.

The probability of multiple model gets stored in the variable and then take the final out after averaging the result when all model gets trained. The MCC score then checked on the final model and the result will be compared with benchmark model.