Machine Learning Engineer Nanodegree

Capstone Proposal

Tanay Mehta

21st September, 2019

Mental Depression Sentiment Prediction

Domain Background

Mental Health, from times unknown still continues to be a plague for the Human Race. It affects about 300 Million people worldwide regardless of their Race, Religion, Age, gender or culture [1]. Depression is one of the most debilitating conditions in the world, with severe depression rated depression rated in the same disability category as terminal stage cancer [2].

Possible causes of Mental Depression include a combination of Biological, Psychological and social sources of distress.

Problem Statement

The Problem is a Binary Classification one and the goal is to predict, based on given Input (which is just a sentence); whether the Input Sentence has Depressive Sentiments or not. In a general sense, this is a basic Natural Language Processing Problem.

After a substantial research, I found out that most of the tweets posted on twitter do convey either a positive or a negative sentiment. Using an already-available dataset and training a model on it will be quite effective in finding out the probability of a person's tweets and in-term, a view into a person's mental health.

Datasets and Inputs

The Dataset used in the project is available free of charge on Kaggle known as "*Sentiment140 dataset with 1.6 million tweets*" [3]. The project data consists of only 1 file named, "*training.1600000.processed.noemoticon.csv".*

The file contains 1,600,000 tweets extracted using the Twitter API.

Input Data fields:

- **target** – Polarity of Tweet, either 4 (Positive) or 0 (Negative).
- **id** – The Id of Tweet
- **Date** – The Date of Tweet
- **flag** – Query, If there is no query,"NO_QUERY" is set
- **user** – The Id of the User that tweeted
- **text** – The main Text of the tweet

Solution Statement

The Solution to this problem will be a Percentage of a Certain Textual Statement being Depressive. To achieve these results, I will apply certain Exploratory Data Analysis and Data Preprocessing Techniques to make sure that the Input Data is in the Best Possible form.

After that, I will be Splitting the Input Data into Training and Testing sets and then finally will be testing out different models with different hyperparameters and fine-tuning the best of the them.

Benchmark Model

The Benchmark model I will be using for this problem will be the Random Forest Model, which has been observed to work excellently well for Tasks in the domain of Sentiment Prediction only to be left behind by LSTMs and DNNs, which I won't be using in this project [4].

Alternatively, I will be trying to achieve a test-set ROC-AUC value of greater than or equal to **0.7** with whichever model that performs best.

Evaluation Metrics

Results of the model's prediction on the Test Set will be evaluated based on 2 evaluation metrics;

1. ROC-AUC Score
2. Confusion Metric

The reason for using 2 separate Metrics for performance evaluation is so that I can get a score on how well the Model is performing on Test data (using ROC-AUC score) and to get the total number of **false positives** and **false negatives**.

Project Design

1. **Data Importing and Cleaning**

    a. The project will be started with getting the data and importing it with proper formatting. Since the Dataset contains raw text, I will be first exploring a few encoding options to import the dataset in the most-efficient manner.

    b. I will then apply crucial Data Cleaning steps such as *Renaming Columns (wherever so required), Dropping missing 'nan' values, etc.*

2. **EDA, Data Preprocessing and Feature Extraction**

    a. In this step, I will be doing *mild* [5] Exploratory Data Analysis, like getting the list of all Negative and Positive Words, Making Wordclouds, and then Splitting the data into Training and Testing set.

    b. In this step, I shall Pre-process the Data and extract important features from it using certain techniques such as Tf-Idf Vectorizer, Porter Stemmer and using General Stopwords removal.

3. **Model Training and Optimal Model & Hyperparamter Search**

    a. In this step I will be Training some very well-known models for Sentiment Analysis related tasks such as Multinomial Naïve Bayes, Logistic Regression, Random Forest Classifier, K-Nearest Neighbours, etc.

    b. In this final step, I will be selecting the Best-performing models of all and then will be performing Optimal Hyperparameter Search on the selected model to improve upon the Test Set Scores.

References & Extra Points

[1] World Health Organization, Media Centre, Depression Fact Sheet, Updated February 2017.

[2] World Health Organization. The Global Burden of Disease: 2004 update. Available at http://www.who.int/healthinfo/global_burden_disease/2004_report_update/en/

[3] Sentiment140 Dataset: https://www.kaggle.com/kazanova/sentiment140/

[4] Looking at the size of the Dataset and the size of possible word-embeddings, I have decided to not use Deep Learning Techniques such as RNNs and LSTMs for this task.

[5] Since there are only 2 Useful Columns in the Data (One being the Text itself and the other being its corresponding ground-truth label), There is not much of a scope when it comes to EDA. At the very most I could perform Word Frequency Count using Wordcloud and other basic similar operations.