Machine Learning Engineer Nanodegree

Capstone Proposal

Ajay Shewale
August 31, 2018

Sentiment Analysis of Text Data

Domain Background

I have started my startup right after the college, Blubyn. Blubyn is voice-based travel assistant which helps the user to book flights, hotels, and events. We are providing personalized results using Machine Learning. I have built my own Recommendation Algorithm for the same. Now while dealing with this issue, we came to the problem where user feedbacks are very crucial. It helps to recommend a user what they like and what they apt to chose.

The ability to understand the public sentiment in social media is increasingly considered as an important tool for market understanding and customer segmentation.

So, from this project, I receive an opportunity to work in sentiment analysis field. Also, it will definitely be beneficial for my startup. Because while dealing with the reviews of customers, we want to interpret what user tends to portray so that we can give him best recommended results.

Apart from this, Sentiment analysis has been an interesting field of study. This is still an evolving subject, it has functions that are too complicated to understand by the machines such as sarcasm, negative emotions, hyperbole etc.

Because I am part of the industry, I know the potential in sentiment analysis. It adds a lot of value to the industry. Sentiment analysis bases its results on factors that are so inherently humane, it is bound to become one the major drivers of many business decisions in future.

Related Work

There are many papers written on sentiment analysis for the domain.

(Pang and Lee 2008) gives a survey of sentiment analysis. In a paper, Jansen has analyzed the commercial impact of social mediating technology, microblogging.

Overall, text classification using machine learning is a well-studied field (Manning and Schuetze 1999). There is an excellent work on the effects of various machine learning techniques such as Naive Bayes, Maximum Entropy, SVM in the movie reviews domain.(Pang and Lee 2002). They were able to achieve an accuracy of 82.9% using SVM and a unigram model.

Work (Read, 2005) has been done in using emoticons as labels for positive and sentiment. This is very connected to Twitter because many users have emoticons in their tweets. Researchers have also worked on detecting sentiment in text. (Turney 2002) presents a simple algorithm, called semantic orientation, for detecting sentiment.

Problem Statement

The goal of this project is to predict the sentiment analysis of users data. Sentiment analysis can predict many different emotions attached to the text, but in this project, only 3 major were considered: positive, negative and neutral.

For this purpose, I have chosen Twitter data because it is easily available.

I will run this model on my company user data although I can't share that data so have to go with the Twitter data. It will help to identify the users' preferences and accordingly to give them the best recommendation results.

Datasets and Inputs

The input data consisted of two CSV files:

- train.csv (5971 tweets)

- test.csv (4000 tweets)

one for training and one for testing. The format of the data was the following (test data didn't contain Category column)

The dataset contains the following attributes:

Train data:

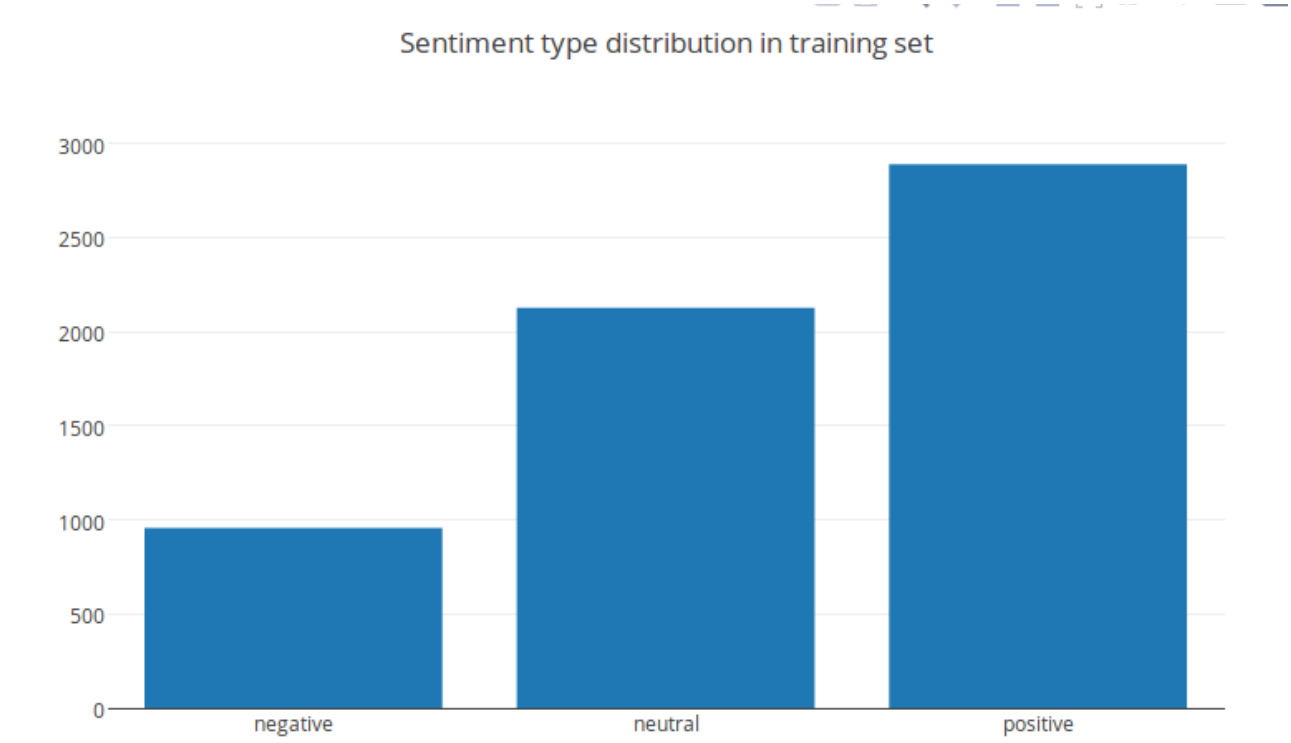
RangeIndex: 5970 entries, 0 to 5969
Data columns (total 3 columns):
Id 5970 non-null object
Category 5970 non-null object
Tweet 5970 non-null object
dtypes: object(3)
memory usage: 140.0+ KB

	Id	Category	Tweet
0	635769805279248384	negative	Not Available
1	635930169241374720	neutral	IOS 9 App Transport Security. Mm need to check...
2	635950258682523648	neutral	Mar if you have an iOS device, you should down...
3	636030803433009153	negative	@jimmie_vanagon my phone does not run on lates...
4	636100906224848896	positive	Not sure how to start your publication on iOS?...

Here, the' ***Category***' is the target class, given the '***Tweet***' column, '***Category***' defines whether the given user tweet is *positive, negative* or *neutral*.

Distibution of Target Class:

The dataset is skewed. Also, we can see that the training dataset contains more positive class and less negative class.



Text Data:

RangeIndex: 9968 entries, 0 to 9967
Data columns (total 2 columns):
Id 4000 non-null float64
Category 4000 non-null object
dtypes: float64(1), object(1)
memory usage: 155.8+ KB

As the test.csv file was full of empty entries, they will be removed.

	Id	Category
0	6.289494e+17	dear @Microsoft the newOoffice for Mac is grea...
1	6.289766e+17	@Microsoft how about you make a system that do...
2	6.290232e+17	Not Available
3	6.291792e+17	Not Available
4	6.291863e+17	If I make a game as a #windows10 Universal App...

Here, the ***Category*** is tweet data, according to the tweet data I will estimate it is a *neutral, positive* or *negative* tweet. For the future reference, I will name the ***Category*** column to the ***Tweet*** column to avoid confusion.

Data Source: Data is publicly available and taken from a Kaggle competition: https://www.kaggle.com/c/angry-tweets/data

Solution Statement:

The main purpose is to find the semantic of the user tweets. Semantics in terms of is it a positive, negative or neutral tweet.

Because it is a raw data, I need to clean it up before using any model. Data cleaning is one of the crucial parts of the Machine Learning.

Followed, I will prepare the data for the model, here preparation as in adding new features into the data, counting the various expressions, words in tweets etc. This all comes into Data preprocessing.

After preprocessing the data, I will use the classic approach to train the model. I will do the experiments with Naive based, random forest, XGBoost. And finally will train the model for the best result I have got from doing the above experiments.

Benchmark Model:

I will use the traditional models such as Naive Bayes after preprocessing the data. I will conduct experiments and check the accuracy of each model.

Experiments will be with model + Naive Bayes, Model + Random forest, extra features + random forest and with xgboost.

I will use sklearn.metric library to compute f1_score, precision_score, recall_score, accuracy_score.

Evaluation Metrics:

As discussed above I will use traditional techniques as well as experiments to evaluate this model.

I will use sklearn's metric library to do so.

The sklearn.metrics module implements several loss, score, and utility functions to measure classification performance. Some metrics might require probability estimates of the positive class, confidence values, or binary decisions values.

The accuracy_score function computes the accuracy, either the fraction (default) or the count (normalize=False) of correct predictions.

$$\texttt{accuracy}(y, \hat{y}) = \frac{1}{n_{\text{samples}}} \sum_{i=0}^{n_{\text{samples}}-1} 1(\hat{y}_i = y_i)$$

Precision(P) is the ability of the classifier not to label as positive a sample that is negative. Precision is defined as the number of true positives (T_p) over the number of true positives plus the number of false positives (F_p).

P = T_p \ (T_p+F_p)

Recall(R) is the ability of the classifier to find all the positive samples. Recall is defined as the number of true positives (T_p) over the number of true positives plus the number of false negatives (F_n).

R = T_p \ (T_p + F_n)

These quantities are also related to the (F_1) score, which is defined as the harmonic mean of precision and recall.

F1 = 2 * (P \times R) \ (P+R)

In sklearn.metric we have,

sklearn.metrics.precision_recall_fscore_support(*y_true*, *y_pred*, *beta=1.0*, *labels=None*, *pos_label=1*, *average=None*, *warn_for=('precision', 'recall', 'f-score')*, *sample_weight=None*)

Here, I am using accuracy feature even though class is imbalnced because this is actually the part of the proejct - to deal with such imbalance problem in the data. I will add extra features, preprocess the data, follow the necessary steps to overcome this issue.

Real world datasets are not balanced too. Along with this I will be looking for Recall, Precision and F1 score as well.

So consider my evaluation metric as F1, Recall, Precision and Accuracy.

Project Design:

Following are the major steps I will implement:

Get Familiar with Dataset:

- Load the data and get familiar with it
- Check how the data is distributed

Data Preprocessing:

To apply the model to the dataset, it has to be clean and suitable for modeling.

- Remove Urls
- Remove usernames (mentions)
- Remove tweets with Not Available text
- Remove special characters
- Remove numbers

Add extra Features

Text Processing:

- Tokenizing and stemming
- Transforming into Bag of words

Modeling/Training:

I will use following models to train my dataset and record the accuracy.

- Experiment 1: BOW + Naive Bayes with addition features
- Experiment 2: BOW + Random FOrest
- Experiment 3: BOW + Xgboost

Testing: Will check the accuracy from the above and apply the model on the test dataset.

References:

http://scikit-learn.org/stable/modules/model_evaluation.html

B. Jansen, M. Zhang, K. Sobel, A. Chowdury. The Commerical Impact of Social Mediating Technologies: Micro-blogging as Online Word-of-Mouth Branding, 2009.

C. Manning and H. Schuetze. Foundations of Statistical Natural Language Processing. 1999.

B. Pang, L. Lee, S. Vaithyanathan.Thumbs up?Sentiment Classification using Machine Learning Techniques, 2002.

B. Pang and L. Lee. "Opinion Mining and Sentiment Analysis" in Foundations and Trends in Information Retrieval, 2008.

B. Pang and L. Lee. "A Sentimental Education: Sentiment Analysis Using Subjectivity Summarization Based on Minimum Cuts" in Proceedings of ACL, 2004.

J. Read. Using Emotions to Reduce Dependency in Machine Learning Techniques for Sentiment Classification, 2005.

P. Turney. "Thumbs Up or Thumbs Down? Semantic Orientation Applied to Unsupervised Classification of Reviews" in Proceedings of the 40th Annual Meeting of the Association for Computatoinal Linguistics (ACL), 2002.