Praxitelis-Nikolaos Kouroupetroglou

January 2019

Capstone Proposal

Review of Feature Extraction Techniques for Textual Sentiment Analysis

Domain Background

The project's domain background revolves around the area of Sentiment analysis. Sentiment analysis or Opinion Mining is a substantial task in Natural Language Processing, in Machine Learning and Data Science. It is used to understand the sentiment in social media, in survey responses, and healthcare for applications ranging from marketing to customer service to clinical medicine. In general Sentiment analysis main goal is to determine the attitude of a speaker or writer [1].

The history of Sentiment analysis dates to WW2, during that era the primary motivation is highly political in nature. The rise of modern sentiment analysis happened only in the mid-2000s, and it focused on the product reviews available on the Web. Before 2000, the use of sentiment analysis has reached numerous other areas such as the prediction of financial markets and reactions to terrorist attacks. Moreover, the use of Sentiment analysis was useful for many problems such as irony detection and multi-lingual identification. Furthermore, over the years more research efforts are advancing from simple polarity detection to more complex identification of emotions and differentiating negative emotions such as anger and grief. Nowadays The area of sentiment analysis has become so large that anyone can face many challenges and issues when you try to keep track of all the activities in the area and the information overload [1].

In general, to process textual data, there is a need to convert the text and words to tangible data suitable for use for Exploratory data analysis, unsupervised and supervised learning. Nowadays, there are numerous feature extraction techniques that are used for this task. Some of them are the following:

- Bag-of-words or one-hot encoding or Vector Space Feature Extraction Techniques which some of them are the following:
  - TF-IDF which stands for Term Frequency – Inverse Term Frequency, is used to examine the relevance of key-words to documents in corpus [2].
  - Counter vectorization, convert a collection of text documents to a matrix of token counts. This implementation produces a sparse representation of the counts of the words in a sentence [3].

Although the simplicity from these two feature extraction from text techniques there is a drawback, they lead to high dimensional spaces which from its part leads to the curse of dimensionality. However, recently more robust feature reduction methods have been developed which they contain the most related information from the textual data and reduce the textual information in a lower dimensionality space [4].

- Word Embedding Techniques, Word Embedding solve the problem of high dimensional space. Word embedding is a technique for language modelling and feature learning, which transforms words in a vocabulary to vectors of continuous real numbers. The technique normally involves a mathematic embedding from a high-dimensional sparse vector space to a lower-dimensional dense vector space. Each dimension of the embedding vector represents a latent feature of a word [5]. Two-word embedding techniques will be used for the project combined with Deep Learning models:

- Training word Embeddings
- Use of pre-trained Embeddings

Finally, this project was personally selected because when it comes to understanding online reviews, there is undoubtedly nothing more powerful than sentiment analysis. It is a powerful tool that is used to draw insights from social media to determine the crowd's emotions behind a series of words, this can be very beneficial for companies and for marketing purposes to understand people's reactions, opinions and needs.

Problem Statement

The project that the proposal infers to is called "Movie Review Sentiment Analysis" a past Kaggle Competition. The competition's main goal is to classify the sentiment of reviews from users from the Rotten Tomatoes dataset" and is located in Kaggle. The Rotten Tomatoes movie review dataset is a corpus of movie reviews used for sentiment analysis. This competition provides the chance to Kaggle users to implement sentiment-analysis on the Rotten Tomatoes dataset. The main task is to label phrases on a scale of five values: negative, somewhat negative, neutral, somewhat positive, positive. There are many obstacles such as sentence negation, sarcasm, terseness, language ambiguity, and many others make this task very challenging. In general, this particular Sentiment Analysis is a multiclass classification task to be faced [6].

Databases and Inputs

The dataset contains tab-separated files with phrases from the Rotten Tomatoes dataset. The train/test split has been preserved to benchmark, but the sentences have been shuffled from their original order. Each Sentence has been parsed into many phrases by the Stanford parser. Each phrase has a PhraseId. Each sentence has a SentenceId. Phrases that are repeated (such as short/common words) are only included once in the data [6].

The Train Set (source) has 4 columns and 156060 cases/rows. Its features are the following:

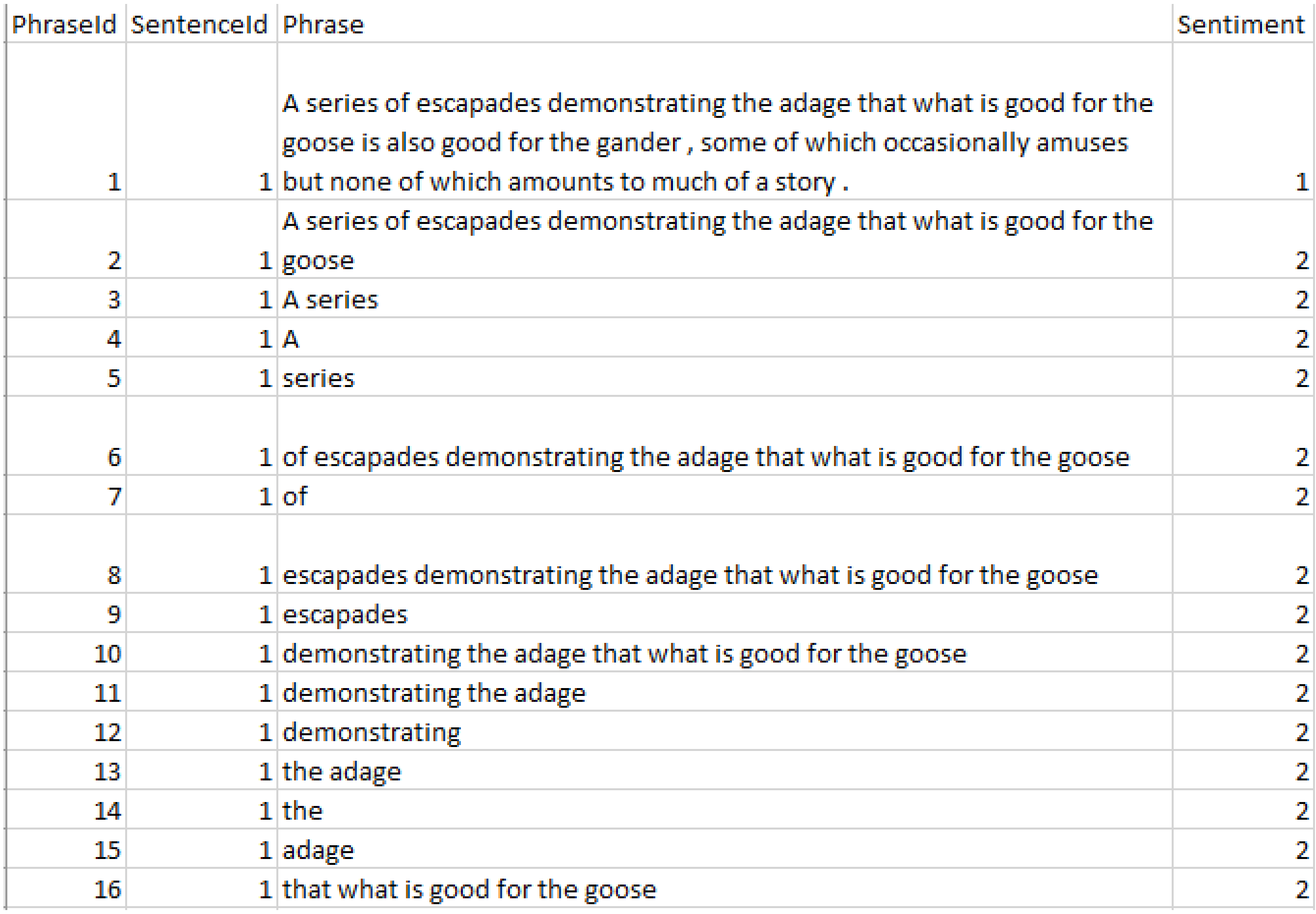
1. ***PhraseId***, is a unique Phrase identifier per phrase. Multiple phrases originate from the same Sentence/movie review and its type is "numeric". We have 156060 unique PhraseIds in the train set.
2. ***SentenceId***, is a unique Sentence / review identifier. In the trainset we have 8543 unique Sentences/reviews in the train set.
3. ***Phrase***, it is type of "string" and it stems from the Sentence that is referenced by SentenceId. In total they are 156060 unique Phrases and each phrase is the result from a unique split to the Sentence /review that belongs to.
4. ***Sentiment***: Is the Sentiment Labels and the target feature that must be predicted in the Test Set. Its labels are the following: 0 – negative, 1 - somewhat negative, 2 – neutral, 3 - somewhat positive, 4 – positive.

The Test Set (source) has 3 columns and they are the following:

1. ***PhraseId***, is a unique Phrase identifier per phrase. Multiple phrases originate from the same Sentence/movie review and its type is "numeric". We have 66292 unique PhraseIds in the test set.
2. ***SentenceId***, is a unique Sentence / review identifier. In the trainset we have 3310 unique Sentences/reviews in the test set.

3. ***Phrase***, it is type of “string” and it stems from the Sentence that is referenced by **SentenceId**. In total they are 156060 unique Phrases in the test set and each phrase is the result from a unique split to the Sentence /review that belongs to.

The following figure demonstrates how the cases look like from the train set:



PhraseId	SentenceId	Phrase	Sentiment
1	1	A series of escapades demonstrating the adage that what is good for the goose is also good for the gander , some of which occasionally amuses but none of which amounts to much of a story .	1
2	1	A series of escapades demonstrating the adage that what is good for the goose	2
3	1	A series	2
4	1	A	2
5	1	series	2
6	1	of escapades demonstrating the adage that what is good for the goose	2
7	1	of	2
8	1	escapades demonstrating the adage that what is good for the goose	2
9	1	escapades	2
10	1	demonstrating the adage that what is good for the goose	2
11	1	demonstrating the adage	2
12	1	demonstrating	2
13	1	the adage	2
14	1	the	2
15	1	adage	2
16	1	that what is good for the goose	2

Figure 1 - Example Cases from Train Set

During the training of the Machine and Deep Learning models the **PhraseId** and **SetenceId** will not be used since they do not provide any predictive advantage, they are just Id incremental numbers and they do not have any predictive ability during Machine Learning and Deep Learning training. However the Phrases will definitely be used during the project.

Furthermore, the dataset is unbalanced, which means that the train set does not provide almost equal number of cases for all the different types of sentiment that must be predicted. This is obvious at the following figure which depicts the distribution of the sentiment at the train set:

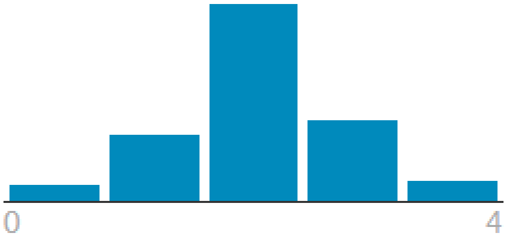


Figure 2 - Sentiment Distribution from Train Set

Sentiment Distribution

Sentiment	Count
0 - negative	7072
1 - somewhat negative	27273
2 - neutral	79582
3 - somewhat positive	32927
4 - positive	9206

It is obvious that the Sentiment “2 - Neutral” is the dominant one. Having an unbalanced dataset may lead us to classifiers/models that can not identify and classify cases that belong to positive or negative Sentiments and they may misclassify them.

A solution statement

The solution to this kaggle competition is to provide multiclass sentiment classification (0 – negative, 1 - somewhat negative, 2 – neutral, 3 - somewhat positive and 4 - positive) for each item in the testing data using the following feature extraction techniques:

- TF – IDF vector space
- Live training for Word Embeddings
- Download and use of pretrained Word Embeddings

The experiments will be exclusively run on Kaggle as the competition oblige to do so. Additionally, there will be used both Machine Learning and Deep Learning techniques. There will be tried several different approaches as discussed in the Project Design section below. Finally, the predictions over the test set will be evaluated from Kaggle to score and measure the models’ performance.

To be more specific I will do the following experiments:

1. Create Machine Learning models with
   - Feature Extraction using TF – IDF
   - Download and use of pretrained Word Embeddings for Feature Extraction
2. Create Deep Learning models with
   - Live on-premise training for Word Embeddings
   - Download and use of pretrained Word Embeddings for Feature Extraction

In every experiment I will evaluate my models with train – validation split with ratio of 80 / 20 to evaluate the models’ performance.

The Machine Learning models that will be used are the following

- Logistic Regression (LR)
- K-Nearest Neighbors (KNN)
- Classification Trees (CART)
- Naive Bayes models (NB)
- Support Vector Machines (SVM)
- Random Forests (RF)
- XGBoost (XGB)
- Ensemble after training and evaluation the top performed Machine Learning Models

The list above summarizes all the Machine Learning families. They will be used and evaluated each one of them and those with the best accuracy will be kept.

The Deep Learning architectures that will be used is the following:

- Long Short-term Memory Recurrent Networks (LSTM)
- Convolutional Neural Networks (CNN)
- Bidirectional Long Short-term Memory Recurrent Networks (BiLSTM)
- Long Short-term Memory Recurrent Networks - Convolutional Neural Networks (LSTM - CNN)
- Bidirectional Long Short-term Memory Recurrent Networks - Convolutional Neural Networks (BiLSTM - CNN)
- Ensemble after training and evaluation the top performed Deep Learning Models

The Deep Learning models were chosen based on the scientific literature and because the deeper the Deep Learning architecture the better fit to the data [5].

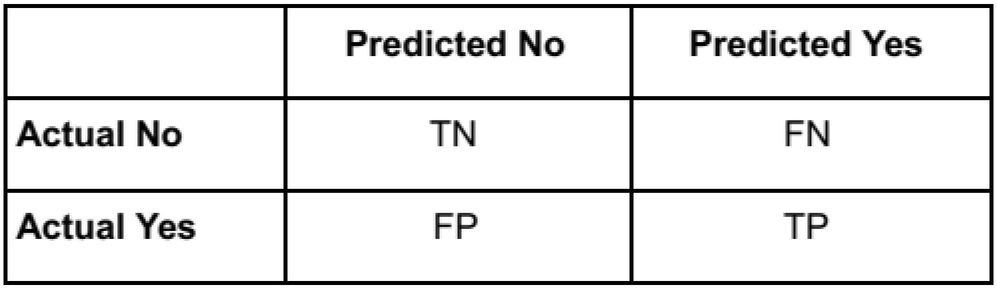
Benchmark Model

The given dataset is a typical supervised learning problem. In Machine Learning and in general in many Kaggle Competitions XGB - Extreme Gradient Boosting models perform better than others [7]. So Extreme Gradient Boosting (XGB) as a benchmark will be picked and it will be tried to try to be the benchmark with other machine learning models and even with hyperparameter tuning.

For the Deep Learning models, as a benchmark will be used the LSTM - Long Short-term Memory Recurrent Networks. The notion behind this pick is a philosophical principle which is called “Occam’s Razor” which says that between two explanations choose the one that is has the least speculations/assumptions [9]. In other words, sometimes follow the simplest ideas. Since LSTM is simpler to be implemented in code than the other 4 Deep Learning models as described above, then this model as a benchmark will be picked and I will try to be the benchmark with the rest Deep Learning models. Besides, LSTM models are widely used for Sentiment Analysis [8], so based on that I will try to find more effective Deep Learning models to increase their accuracy.

Evaluation Metrics

The performance of each classifier is evaluated using four metrics; classification accuracy, precision, recall and F1 score. It is using true positive (TP), true negative (TN), false positive (FP) and false negative (FN). True Positive (TP) stands for the number of correct predictions that a case is true which means that it is occurring when the positive prediction of the classifier agrees with a positive prediction of target variable. True Negative (TN) is the a number of correct predictions that a case is false, for example it occurs when both the classifier, and the target variable suggests the absence of a positive prediction. The False Positive (FP) is the number of incorrect predictions that a case is true. Finally, False Negative (FN) is the number of incorrect predictions that a case is false. The table below shows the confusion matrix for a two-class classifier.



	Predicted No	**Predicted Yes**
Actual No	TN	FN
Actual Yes	FP	TP

Rotten Tomatoes – Movie Review Sentiment Analysis requires all the submissions to be evaluated in their predictions' accuracy over the Test Set [10]. Classification accuracy is defined as the ratio of the number of correctly classified cases and its formula to the sum of TP and TN divided by the total number of cases.

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

Since the train set is unbalanced, F1 score as a secondary metric will be used which combines the other two metrics; precision and recall. Their formulas are the following:

Precision is defined as the number of true positives (TP) over the number of true positives plus the number of false positives (FP).

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

Recall is defined as the number of true positives (TP) over the number of true positives plus the number of false negatives (FN).

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

F1 score it considers both the precision and the recall.

$$\mathrm{F1} \; = 2 * \frac{\mathrm{precision} * \mathrm{recall}}{\mathrm{precision} + \mathrm{recall}}$$

Project Design

Bellow the figure presents the flow and procedures of the capstone project:

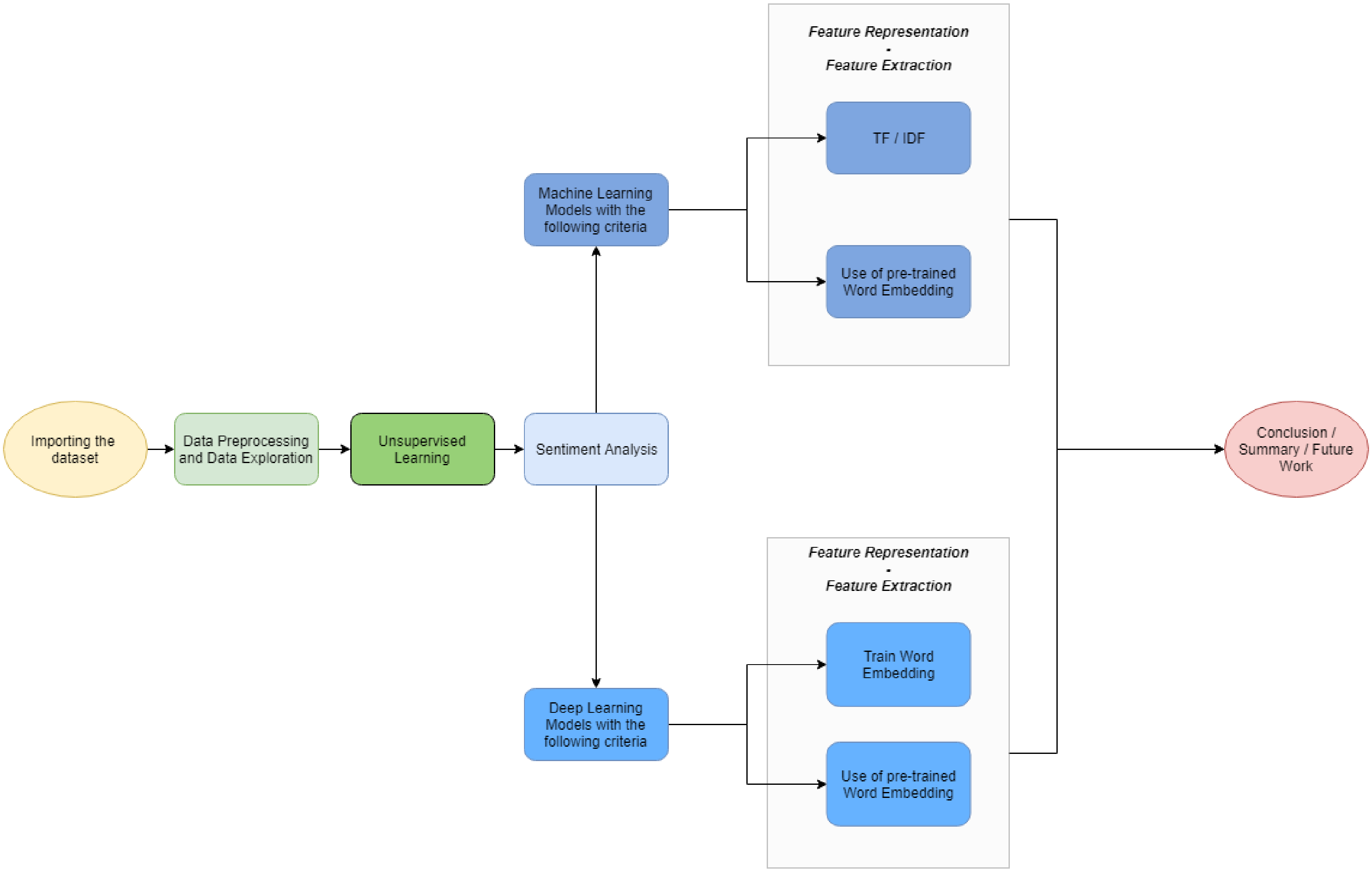

Importing the dataset
Data Preprocessing and Data Exploration
Unsupervised Learning
Sentiment Analysis
Machine Learning Models with the following criteria
Feature Representation - Feature Extraction
TF / IDF
Use of pre-trained Word Embedding
Deep Learning Models with the following criteria
Feature Representation - Feature Extraction
Train Word Embedding
Use of pre-trained Word Embedding
Conclusion / Summary / Future Work

Figure 3 - Capstone Project Flowchart

The workflow of solving this problem will be in the following order:

1. Loading the data
2. Data Preprocessing and Data Exploration.
    a. Cleaning the text data from noisy information.
    b. Observing anomalies in the Train Set
    c. Measure word frequencies (Unigrams, Bigrams and Trigrams).
    d. Recognize named entities.
    e. Create wordclouds.
    f. Discover most significant words.
3. Unsupervised Learning
    a. Train set reviews' visualization over the 2-axis using t-SNE.
    b. K-means clustering over the reviews from train set and visualize the clusters using t-SNE.
    c. Topic Detection over the reviews from train set using LDA (Latent Dirichlet Allocation algorithm) and visualize the topics using t-SNE.
    d. Word Embeddings over the train set and visualize their similarity using t-SNE.

   e. Dimensionality reduction techniques such as PCA – Principal Component Analysis and SVD – singular value Decomposition may be used during Unsupervised Learning.

4. Machine Learning
   a. Apply Machine Learning models and measure their accuracy using TF – IDF as feature extraction.
   b. Apply Machine Learning models and measure their accuracy using word embeddings as feature extraction.
5. Deep Learning
   a. Apply Deep Learning models and measure their accuracy using the training of word embeddings as feature extraction.
   b. Apply Deep Learning models and measure their accuracy using pretrained word embeddings as feature extraction.
6. Summarize, Conclusions, Future Work

References

[1] Mika V. Mäntylä Daniel Graziotin, Miikka Kuutila, The Evolution of Sentiment Analysis - A Review of Research Topics, Venues, and Top Cited Papers

[2] Qaiser, Shahzad & Ali, Ramsha. Text Mining: Use of TF-IDF to Examine the Relevance of Words to Documents. International Journal of Computer Applications.

[3] Sklearn feature extraction - CountVectorizer

[4] Dixa Saxena, S. K. Saritha, K. N. S. S. V., Survey Paper on Feature Extraction Methods in Text Categorization

[5] Lei Zhang, Shuai Wang, Bing Liu, Deep Learning for Sentiment Analysis: A Survey

[6] Movie review sentiment analysis

[7] Xgboost top machine learning method kaggle explained

[8] Sentiment analysis using rnns lstm

[9] Occam's razor

[10] Movie review sentiment analysis - Evaluation