**Machine Learning Engineer Nanodegree**

## Capstone Proposal

Chester Fung
January 21st, 2016

## Proposal

### Domain Background

The project of this capstone is on the entertainment film industry. The global film industry shows healthy projections for the coming years, as the global box office revenue is forecast to increase from about 38 billion U.S. dollars in 2016 to nearly 50 billion U.S. dollar in 2020. The U.S. is the third largest film market in the world in terms of tickets sold per year, only behind China and India. More than 1.2 billion movie tickets were sold in the U.S. in 2015. Many websites offer portals for users to give them feedback or reviews of the movies they watch. These reviews include both positive and negative. Being a big movie fan, I visit these review sites often to look for which movies I should watch.

Several papers have been published before on this topic, including
Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. (2011). "Learning Word Vectors for Sentiment Analysis." *The 49th Annual Meeting of the Association for Computational Linguistics (ACL 2011)*. (link)

The paper and also this project is based on a Kaggle Competition.

### Problem Statement

This capstone project is to classify the user reviews from IMDB.

Specifically, it is to classify the sentiment of sentences from the IMDB dataset. Label sentiment of the review; 1 for positive reviews and 0 for negative reviews

We'll first preprocess /clean the data by doing the following steps:

- make sure no NULL values are present.
- remove any html tags
- remove non-letters
- remove punctuation
- convert all letters to lower case and separate the phrases into individual words.

We'll also remove "stop" words. These are words that do not have much meaning, i.e. 'and', 'is', 'a'.

We'll then use feature_extraction module from scikit-learn to create bag-of-words features.

### Datasets and Inputs

Data set is from IMDB

Size of the data set has 50,000 movie reviews that are labeled. And another 50,000 that are unlabeled.

The labels in the dataset include id, sentiment and review. ID is a numeric value assigned to the review. Sentiment (score of 0 to 10) is an integer representing the sentiment score. Review is a string written by users.

25000 reviews from the labeled test data set will be used for training. Another 25000 will be used for testing.

**Solution Statement**

Sentiment of the review should be labeled as 1 for positive and 0 for negative. IMDB rating < 5 results in a sentiment score of 0, and rating >=7 have a sentiment score of 1

We'll use an approach called "Bag of Words". The Bag of Words model learns a vocabulary from all of the documents, then models each document by counting the number of times each word appears. Scikit-learn will then be used to create features from "Bag of Words" Support Vector Machines and Random Forest will then be used to tackle this classification problem. We'll then try different options in CountVectorizer to tune and optimize the model.

The following algorithms will be used and results will then be compared:

Logistic regression - advantages: Good if problem (target variable) is linearly separable. Robust to noise. No distribution requirement

Naive Bayes classifier - Advantages: fast to train, not sensitive to irrelevant features

weaknesses: assumes independence of features

Reason for choosing: Naive Bayes works well for small training set sizes and it's fast, and our dataset is small

SVM (Support Vector Machine) - SVM works great for linear problems

Disadvantage: Inefficient to train. Therefore, it's not good for problems with many training points

Advantages: high accuracy

Reason for choosing: high accuracy

Random Forest Boosting - Advantage (Random Forest): fits well with uneven data sets with missing variables. Lower classification error rate compared to decision tree. Faster training time compared to SVM

**Benchmark Model**

This is a Kaggle competition. Winner of the competition achieved an ROC curve score of **0.99259**. The goal of this capstone project is to obtain **0.9 or above**

**Evaluation Metrics**

Evaluation metric that will be used in this capstone project is accuracy score.

**Project Design**

*(approx. 1 page)*

In this final section, summarize a theoretical workflow for approaching a solution given the problem. Provide thorough discussion for what strategies you may consider employing, what analysis of the data might be required before being used, or which algorithms will be considered for your implementation. The workflow and discussion that you provide should align with the qualities of the previous sections. Additionally, you are encouraged to include small visualizations, pseudocode, or diagrams to aid in describing the project design, but it is not required. The discussion should clearly outline your intended workflow of the capstone project.

The following workflow will be used:

- download the dataset from the kaggle website.
- preprocessing of the data
    - make sure no NULL values are present.
    - remove any html tags
    - remove non-letters
    - remove punctuation

- extract the features (by using feature_extraction module from scikit-learn)
- try different machine learning algorithms
    - Logistic regression
    - Naive Bayes
    - Support Vector Machine
    - Random Forest Boosting

- do hyper parameter optimization.
- compare the result.
- chose a final model.
- check the result against benchmark