Domain Information:

Sentiment analysis (also known as Opinion Mining or Emotion AI) is an interesting application of Natural Language Processing (NLP). I have chosen this subject for my capstone project as it is a powerful tool that can be employed to gauge customer opinion about pretty much everything: a new product, public policies, movies, etc. This can then be used to shape company strategy. Not only is this a very interesting technical challenge, its application is wide open.

Early works in sentiment analysis include Pang [1] and Turney [2]. Historically, topic-based text categorization is a precursor to the sentiment analysis [1]. Sentiment analysis can be imagined as a special case of topic-based categorization [1], with the two topics being 'positive sentiment' and 'negative sentiment'. However, special techniques need to be developed for sentiment analysis because of the linguistic nuances involved i.e. negations, sarcasm, etc.

Problem Statement:

Problem consists of classifying the sentiment polarity (positive or negative) of customer reviews. A successful algorithm will classify the customer reviews with high accuracy.

Datasets and Inputs:

Datasets for this project have been taken from UCI dataset repository:
https://archive.ics.uci.edu/ml/datasets/Sentiment+Labelled+Sentences#

This dataset was created for the Paper [3]. It contains reviews labelled with positive or negative sentiment.
Score is either 1 (for positive) or 0 (for negative).
The reviews come from three different websites/fields:

imdb.com
amazon.com
yelp.com

For each website, there exist 500 positive and 500 negative reviews. Those were selected randomly for larger datasets of reviews. The authors attempted to select reviews that have a clearly positive or negative connotation, the goal was for no neutral reviews to be selected.

Solution Statement:

Data will preprocessed by converting each review into a list of lower case words. Next preprocessing step will apply stemming on this data. Final preprocessing step will convert the lists of words into feature vectors using TF-IDF (Term Frequency-Inverse Document Frequency) [4]. This generates feature vectors of words that are proportional to their frequency in a given review and inversely proportional to their total frequency in the corpus. This step also removes stop words from reviews, as implemented in

`TfidfVectorizer` function in sklearn. After partitioning the data into training set and test set, we will train multiple supervised machine learning models using the training set. We will then document and compare the performance of three algorithms (or models) on the test set and recommend the one with the best performance.

Benchmark Model:

The benchmark model that we are going to use for this project is the one that always returns a 1, meaning the review is positive. This benchmark model achieves an accuracy of 50% as only 50% of reviews are positive. However, in literature other baseline models have been used. [1] For instance, uses a model that manually captures the features (words, punctuations) from reviews and uses these to classify the reviews. This model achieves an accuracy of 69%.

Evaluation Metrics:

As the dataset is balanced, accuracy will be used as evaluation metric for this project, as opposed to F score.

Project Design:

We will combine all three data sources (imdb.com, amazon.com and yelp.com) into one data set. Data will preprocessed by converting each review into a list of lower case words. Next preprocessing step will apply stemming on this data. Final preprocessing step will convert the lists of words into feature vectors using TF-IDF. This generates feature vectors of words that are proportional to their frequency in a given review and inversely proportional to their total frequency in the corpus. This step also removes stop words from reviews, as implemented in `TfidfVectorizer` function in sklearn.

Then, we will partition the data into training set (80%) and test set (20%) while using ‘stratify’ parameter to maintain the balance of positive and negative classes in both training set and test set.

Next, we will apply four supervised machine learning algorithms, which are consistent with size of data (2400 training examples, not too large). The three algorithms are:

1.) Naive Bayes, natural choice as we are dealing with NLP
2.) Logistic Regression

3.) Random Forest (not sure if this works as the data represents sparse matrix [4])
4.) SVM

To fine tune the models, we will use multi-fold cross validation (GridSearchCV), i.e. taking an average of accuracy, while holding out a fold as validation set every time. Finally, the fine-tuned models will be used to find testing accuracy. Final model selection will be done based on test accuracy.

[1] Pang, Bo; Lee, Lillian; Vaithyanathan, Shivakumar (2002). "Thumbs up? Sentiment Classification using Machine Learning Techniques". *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*. pp. 79–86.

[2] Turney, Peter (2002). "Thumbs Up or Thumbs Down? Semantic Orientation Applied to Unsupervised Classification of Reviews". *Proceedings of the Association for Computational Linguistics*. pp. 417–424. arXiv:cs.LG/0212032

[3] 'From Group to Individual Labels using Deep Features', Kotzias et. al,. KDD 2015

[4] https://jessesw.com/NLP-Movie-Reviews/