**Capstone Project 2: Movie Review Sentiment Analysis**

**Milestone Report**

#### Dataset

Movie reviews from IMDb(Internet movie database).The dataset obtained from <http://ai.stanford.edu/~amaas/data/sentiment/>. The data set contains 50,000 movie reviews labeled whether they are positive or negative based on the content.

#### Approach

A classification analysis on reviews to predict the sentiment positive or negative. The task is to predict the sentiment of 15,000 labeled movie reviews and use the remaining 35,000 reviews for training the supervised models. The techniques used include text preprocessing, normalization and in-depth analysis of models using python's built in packages and custom modules like text normalizer and model\_evaluation\_utils (source credit: Practical Machine Learning with Python: A Problem-Solver's Guide to Building Real-World Intelligent SystemsBook by Dipanjan Sarkar, Raghav Bali, and Tushar Sharma. <https://github.com/dipanjanS/practical-machine-learning-with-python/tree/master/notebooks/Ch07_Analyzing_Movie_Reviews_Sentiment>)

#### Steps for supervised sentiment analysis

1. Prepare train and test datasets
2. Text pre-processing
3. Feature engineering
4. Model training
5. Model prediction and evaluation

#### Text pre-processing and normalization

Normalizing movie review data includes creating functions to remove HTML tags, accented characters, expanding contractions, removing special characters, lemmatization to get the root word and removing stopwords.Then using all these components and tie them together in the function called normalize corpus which can be used to take a document corpus as input and return the same corpus with cleaned and normalized text documents.

Reference - practical-machine-learning-with-python/notebooks/Ch07\_Analyzing\_Movie\_Reviews\_Sentiment/Text Normalization Demo.ipynb

#### Analyzing topic models

The first step in this analysis is to combine the normalized train and test reviews and separate out these reviews in to positive and negative reviews. Second step is extract features from positive and negative reviews using TF-IDF feature vectorizer.

### Extract features from positive and negative reviews

TF-IDF - “Term Frequency — Inverse Data Frequency”

Term Frequency (tf): gives us the frequency of the word in each document in the corpus. It is the ratio of number of times the word appears in a document compared to the total number of words in that document.

Inverse Data Frequency (idf): used to calculate the weight of rare words across all documents in the corpus. The words that occur rarely in the corpus have a high IDF score.

#### Topic Modeling

For topic modeling we use the NMF class from scikit-learn and pyLDAvis for building interactive visualizations of topic models. Also, some utility functions from topic-model-utils module to display the results in a clean format.

##### *NMF (Nonnegative Matrix Factorization)*

**NMF** decomposition of the term-document matrix would yield components that could be considered “topics” and decompose each document into a weighted sum of topics.

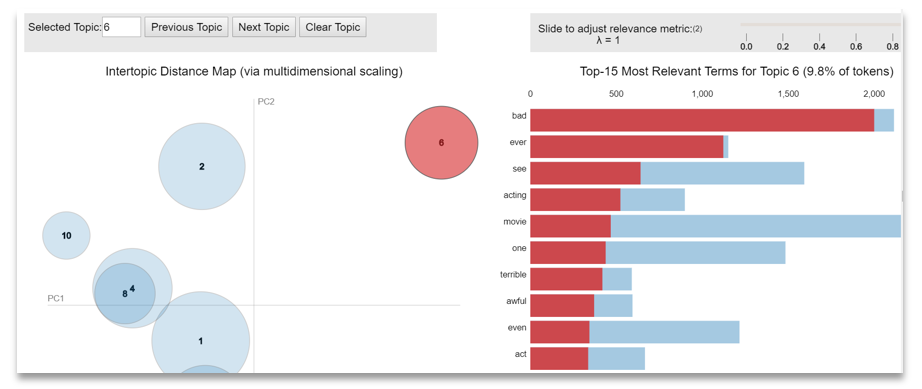
##### *pyLDAvis - Python library for interactive topic model visualization*

**pyLDAvis** is designed to help users interpret the topics in a topic model that has been fit to a corpus of text data. The package extracts information from a fitted LDA topic model to inform an interactive web-based visualization.

#### Visualize topics for positive reviews

### 

#### Visualize topics for negative reviews

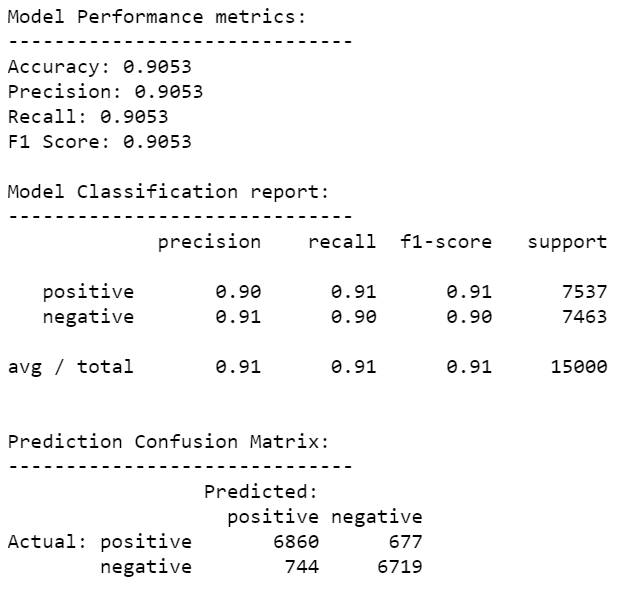


The visualizations shows 10 models from positive and negative reviews. The visualizations are interactive (if using jupyter notebook) and you can click on any of the bubbles representing topics in the Intertopic Distance Map on the left and see the most relevant terms in each of the topics in the right bar chart.

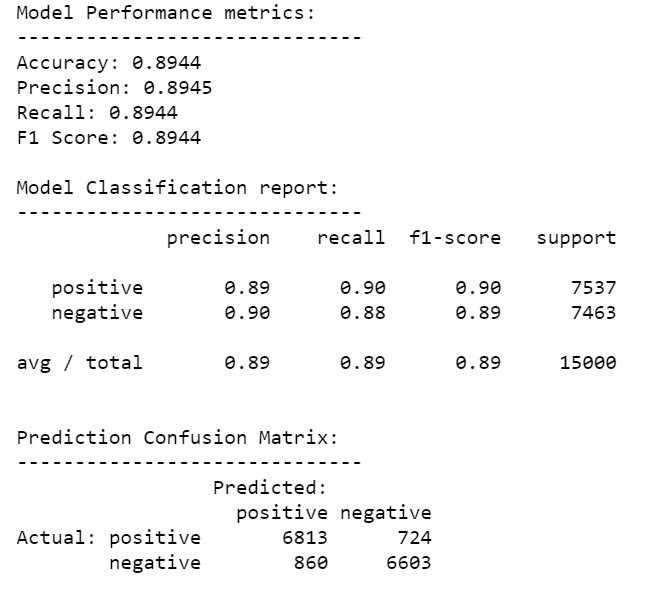
#### Machine learning models and evaluation

#### Feature engineering techniques is based on the bag of words and TF-IDF models. For classification we will be using Logistic regression.

For building logistic regression model on train features and evaluation on test features we use the utility function train\_predict\_model() from our custom model\_evaluation\_utils module.



##### *The logistic regression model got 90% accuracy And F1-score 91% on our BOW models.*



##### *The TF-IDF model got accuracy And F1-score 89%.*

#### 51% of reviews in our sample are positive and 49% is negative.

#### Deep learning models

#### Prediction class label encoding

For the deep learning models, we use the one-hot encoding to change the sentiment labels to numeric representations.

#### Feature Engineering with word embeddings

Here we are going to use 2 models for feature engineering. word2vec model and GloVe model.

##### *Word embeddings*

Word embeddings are vectors that represent words. For example, the word "dog" might be represented as [0.1, -2.1, 1.2] whilst "cat" might be represented as [0.2, 2.4, 1.1]. These vectors are important in neural networks because neural networks can only work with continuous numbers whereas words are discrete symbols.

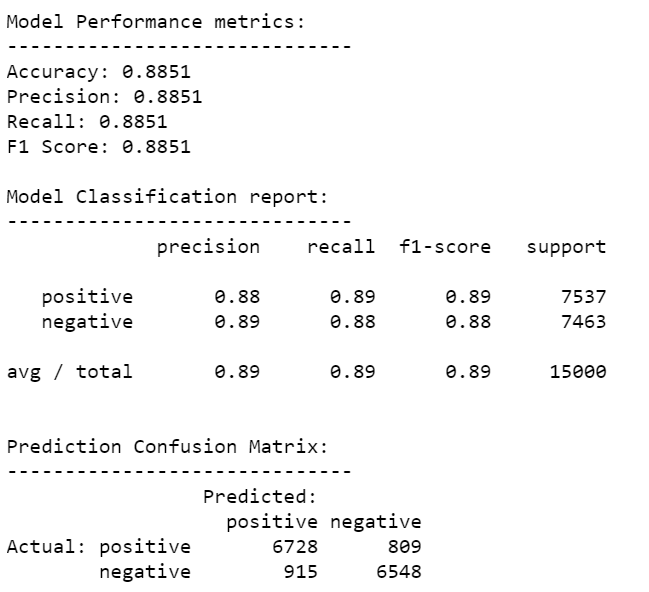
word2vec - The purpose and usefulness of Word2vec is to group the vectors of similar words together in vectorspace. That is, it detects similarities mathematically. Word2vec creates vectors that are distributed numerical representations of word features, features such as the context of individual words.

GloVe (Global Vectors) - GloVe takes a different approach. Instead of extracting the embeddings from a neural network that is designed to perform a surrogate task (predicting neighboring words), the embeddings are optimized directly so that the dot product of two-word vectors equals the log of the number of times the two words will occur near each other (within 5 words for example).

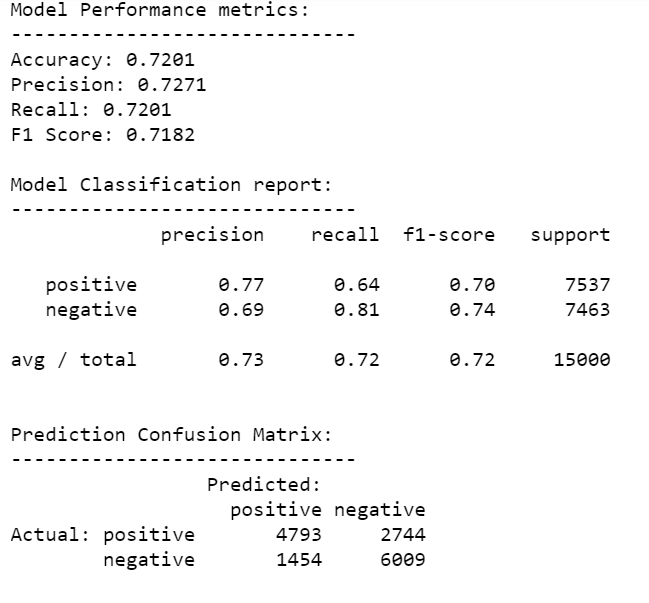
#### Building Deep neural network architecture

Here we build a deep neural network on the word2vec and GloVe model features.

#### Model Training, Prediction and Performance Evaluation



##### *The DNN model accuracy and F1-score on word2vec features is 89%*



##### *The DNN model accuracy and F1-score on GloVe features is 72%*

#### 41% positive reviews and 58% negative reviews

#### Summary

For predicting the sentiment from movie reviews, we use machine learning approaches like Logistic regression and deep neural networks. The Linear regression on BOW features got the highest accuracy and F1-score 91% and DNN model on GloVe features got the lowest accuracy and F1-score 72%.