**Part A**

A1) What are the top 5 parts of speech in this corpus of job descriptions? How frequently do they appear?

As you can see from the bar graph below, the top five parts of speech are (frequencies in parenthesis): NN(~37000), IN(~33000), DT(~22000), NNS(~15000), JJ(~14000).

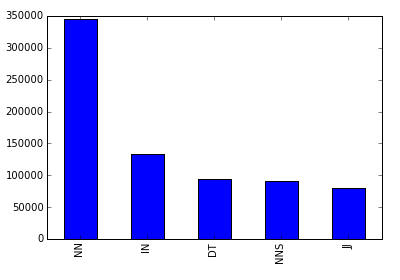


Figure 1: Top 5 Parts of Speech Frequencies

A2) Does this corpus support Zipf’s law? Plot the most common 100 words in the corpus against the theoretical prediction of the law. For this question, do not remove stop words. Also do not perform stemming or lemmatization.

Yes, this corpus does support Zipf’s Law in the fact that as rank increases, frequency decreases somewhat linearly. Figure 2 below shows the top 100 words support this.

In Figure 3 below I included the top 10,000 words to see when Zipf’s law would break down. You can see at the end between 103 and 104 the harmonic series diverges, which is mentioned in Zipf’s Law.

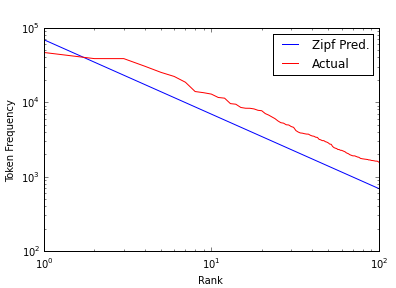


Figure 2 : Top 100 Words

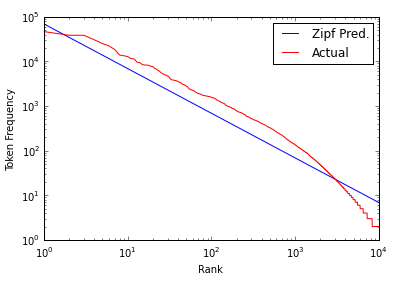


Figure 3: Top 10,000 Words

A3) If we remove stopwords and lemmatize the corpus, what are the 10 most common words? What is their frequency?

Part B

Preface) Here we will describe our data munging methodology.

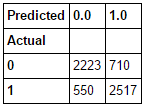
We used 15,000 descriptions from the total data set. We chose the 15000 randomly but ensured that our sample set was 50% of the descriptions with normalized salaries above the 75th percentile and 50% below the 75th percentile, as was recommended in class. From there we split our dataset so that 60% of the data would be in our training set and 40% would be in our test set.

For the data itself we used the TfidfVectorizer to create a sparse matrix for our x variables. In this function we specified that we did not want any material with less than 2 counts total (min\_df) and to use TF-IDF smoothing (smooth\_idf = True). For our model we used the Multinomial Naïve Bayes approach.

B1) Create a classification model with *all* words and the bag-of-words approach. How accurate is the model (show the confusion matrix)?

The Multinomial Naïve Bayes approach with the all of the words included in our bag of words gave us an accuracy of 79.0%. Table 1 below shows our confusion matrix.

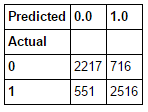
Table 1: Confusion Matrix Bag of Words Model



B2) Speculate before running the following analysis whether lemmatization would help improve the accuracy of classification. Now create a classification model after lemmatization. Did the classification accuracy increase relative to B1? Comment on your speculation versus the actual results you obtained.

We believed that lemmatization would in fact help improve the accuracy of our model. Below is the confusion matrix for our Multinomial Naïve Bayes approach on a bag of words with lemmatization. Our accuracy came to be 78.8% which is a bit lower than expected but nearly negligible compared to the approach from B1. Table 2 below shows us the lemmatized confusion matrix.

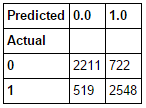
Table 2: Confusion Matrix Lemmatized Bag of Words Model



B3) If you got better results with lemmatization, retain the lemmatized data, else use the original one. Now speculate whether stopwords removal would help increase the accuracy of the model. Take out the stopwords, build a classification model and check the accuracy, and compare with that in B1 & B2.

We used the original dataset from B1 since Lemmatization did not do much for us. We speculated that stop words should help us improve accuracy because we are essentially removing noise. Our model gave us an accuracy of 79.3% which is marginally better than our previous best (B1 at 79.0). Table 3 below shows us the confusion matrix for the stop words removed.

Table 3: Confusion Matrix Stop Words Removed



B4) Use the job descriptions without lemmatiztion and stopword removal. Add parts-of-speech bigrams to the bag-of-words, and run a new classification model. Does the accuracy increase over the results in B1?

The accuracy actually dropped significantly with POS bi-grams. We ended up getting an accuracy of 63.6%

Table 4: Confusion Matrix POS Bi-Grams

