Twitter Hashtag Prediction

Ramaneek Gill

Supervised By Michael Guerzhoy

1 Introduction

We plan to build a a short-text classifier and to work on discovering features useful for short-text classification. We plan to use this short-text classifier to classify tweets into a set of hashtags. The short text classifier will essentially predict a set of hashtags for a tweet. As a first step, we will reimplement the algorithm in [2], specifically the Naive Bayes classifier. We will then try to improve on these results by training a logistic regression model. Then we will then train a Deep Boltzmann Machine on a large dataset of tweets in order to discover new features that are useful for tweet classification for hashtag prediction.

2 The Dataset

We plan to work with the dataset of [3]. The tweet corpus consists of 1,600,000 tweets. We plan to extract the topics for the training set in the same method described in [2] from this tweet corpus.

3 Prior Work

Mazzia et al [2] demonstrates the use of a Naive Bayes classifier that predicts hashtags for the tweets. The recommendation algorithm was evaluated through hold-out cross validation. To determine the relevance of hashtags to a specific tweet Mazzia et all [2] uses Bayes' rule to determine the posterior probability of C_i given the features of a tweet $x_1, ..., x_n$:

$$p(C_i|x_1, ..., x_n) = p(C_i)p(x_1|C_i)...p(x_n|C_i)/p(x_1...x_n)$$

Where x_i is the presence or absence of the word i$_{th}$ in the dictionary of words in our dataset and C_i is the i$_{th}$ hashtag in our dataset.

Mazzia et al evaluated the results of their Naive Bayes classifier by having their algorithm predict the top 20 hashtags for each tweet, this includes a relevance rank for each hashtag. The best possible rank is 0 and if the hashtag is not present in the list of top 20 recommended hashtags, it is given rank 20. Four different metrics were tracked with respect to the rank of the original hashtag in the tweet: the standard deviation, median, mean, and the presence of the hashtag.

4 Recreating the Prior Work of Mazzia et all

Note: A validation set was retained from 10% from the original dataset, the remaining 90% make up 33130 predictable tweets. When predicting hashtags we only chose the most popular n hashtags. n in this case is 56 for figures 2-5. When presenting predicted hashtags we present m hashtags, where m in this case is set to 20 for figures 1, 3-5.

Figures 1 and 2 shows the recreation results of the prior work of Mazzia et all. We achieved slightly better results by performing cross validation to find ideal values for our parameters: $\alpha = 0.92$ and $\epsilon = 0.01$, figures 3 and 4 highlight this.

Figure 5 shows how the size of the test set scales with the accuracy of the Naive Bayes model. This is because the size of the training set is inversely related to the size of the test set. Figure 5 also shows that a marginal increase in accuracy is achieved with a larger training set (or smaller test set). This shows that our model is not over-fitting the training data.

Figure 1: Accuracy of the Naive Bayes model when varying the number of hashtags that can be predicted; n

varying num hashtags to predict.png

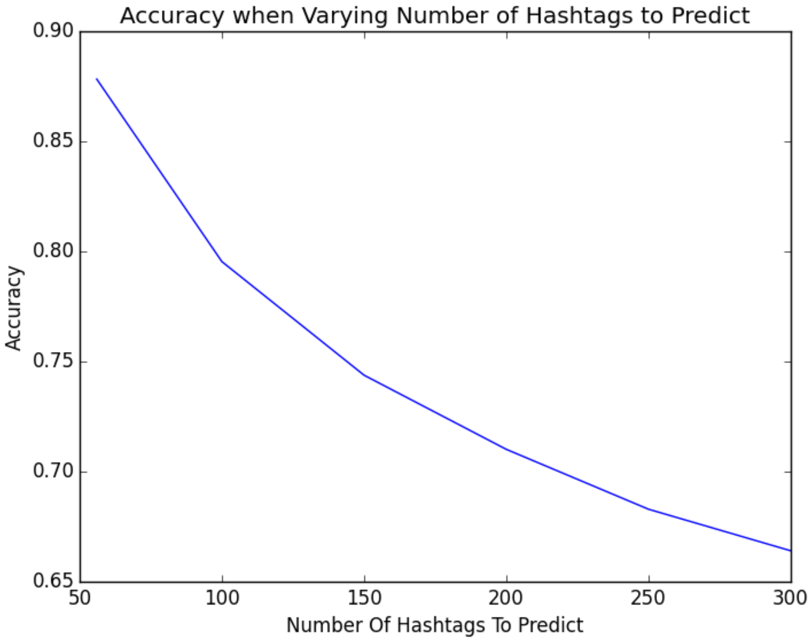


Figure 2: Accuracy when varying the number of hashtags predicted to present; m

hashtag prediction range.png

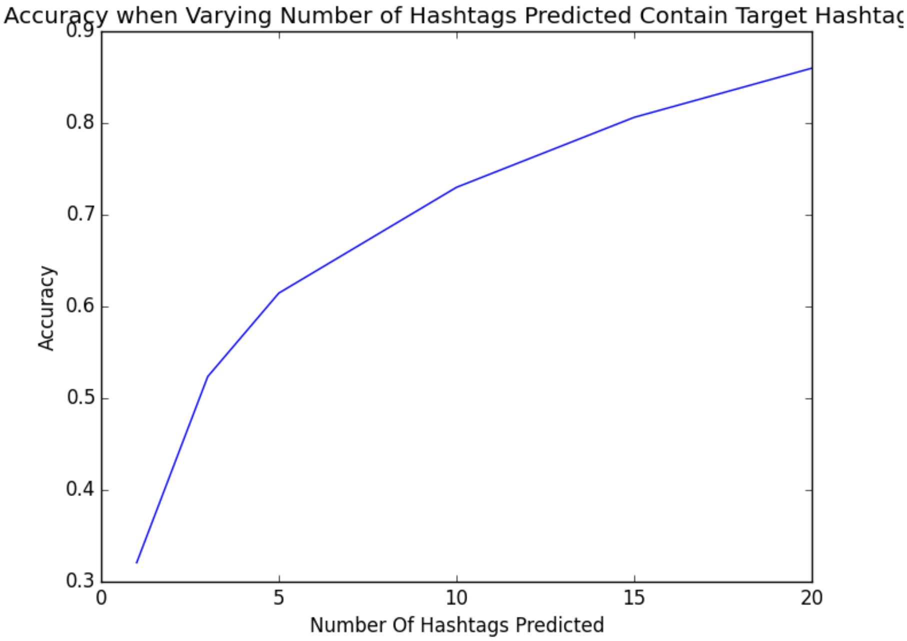


Figure 3: Accuracy when varying α
varying graph.png

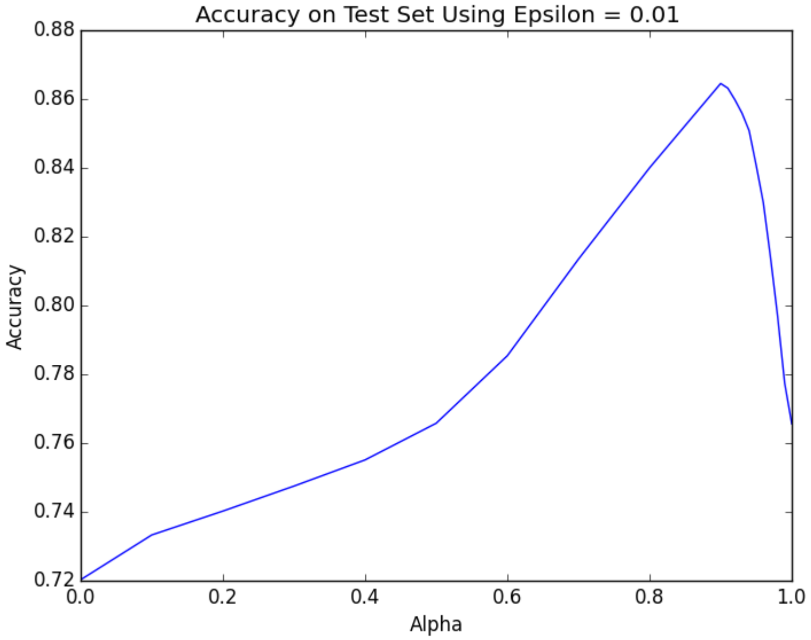


Figure 4: Accuracy when varying ϵ
varying graph.png

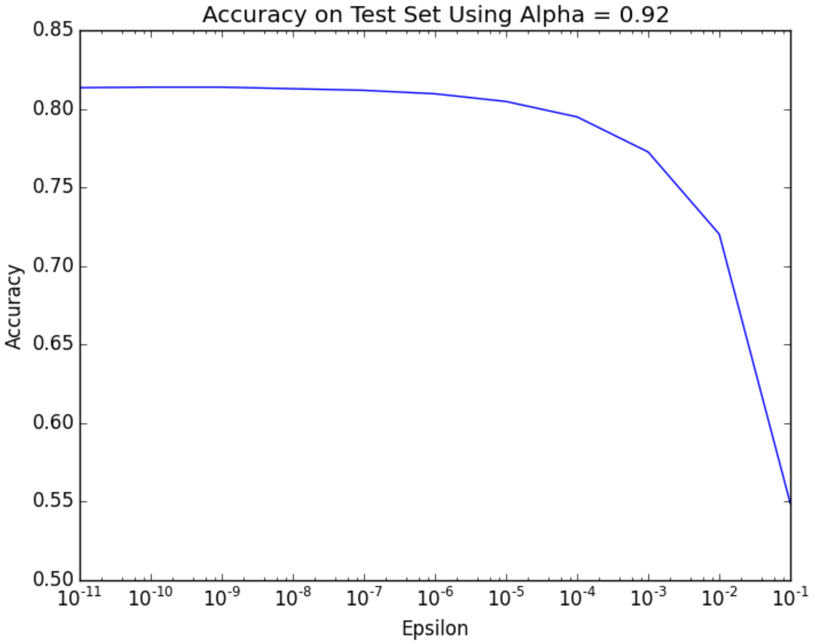
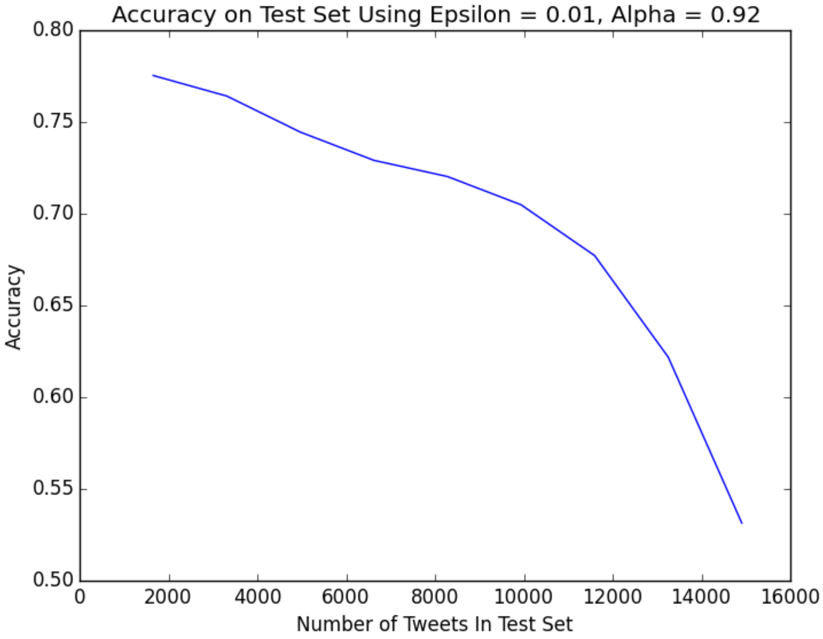


Figure 5: Accuracy when varying the size of the test set.
size ratio varying graph.png



5 Creating a Baseline for Different Models

We determined the predicting out a top 20 out of 56 hashtags was a very unreasonably easy task for it to be meaningful since a random classification algorithm would achieve close to 50% accuracy. So in order to develop a baseline for different short text classification models to implement we changed these metrics to 500 predictable hashtags with a range of top 5 to be presented. This is a closer metric to what a user would expect in a product.

Note: A validation set was retained from 10% from the original dataset, the remaining 90% make up 33130 predictable tweets. When predicting hashtags we only chose the most popular n hashtags. n in this case is 500 for figures 7-9. When presenting predicted hashtags we present m hashtags, where m in this case is set to 5 for figures 6, 8-10.

Figures 6 and 7 are reruns of the recreation results of the prior work of Mazzia et all with our new baselines. We achieved slightly better results by performing cross validation to find ideal values for our parameters: $\alpha = 0.9$ and $\epsilon = 1e^{-}9$, figures 8 and 9 highlight this.

Figure 10 shows how the size of the test set scales with the accuracy of the Naive Bayes model. This is because the size of the training set is inversely related to the size of the test set. Figure 10 also shows that a marginal increase in accuracy is achieved with a larger training set (or smaller test set). This shows that our model is not over-fitting the training data.

Figure 6: Accuracy of the Naive Bayes model when varying the number of hashtags that can be predicted; n

varying num hashtags to predict.png

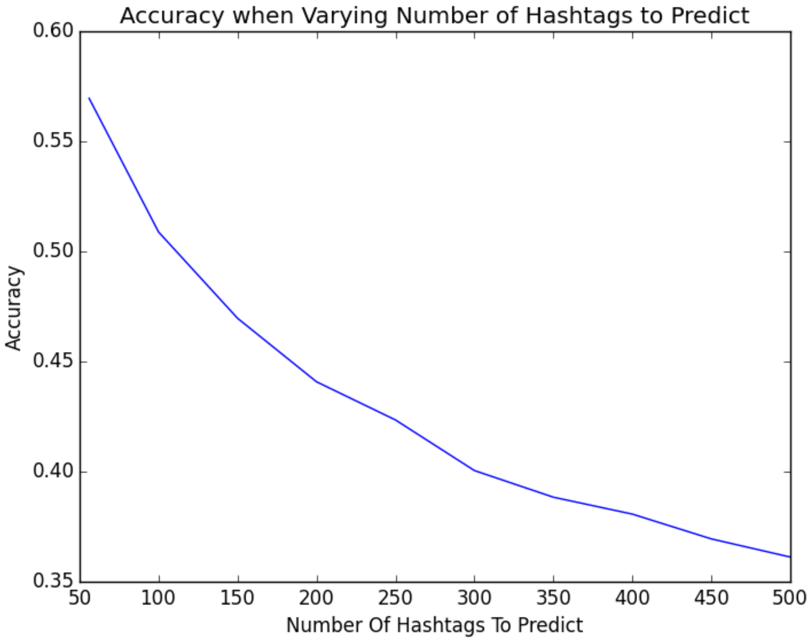


Figure 7: Accuracy when varying the number of hashtags predicted to present; m

hashtag prediction range.png

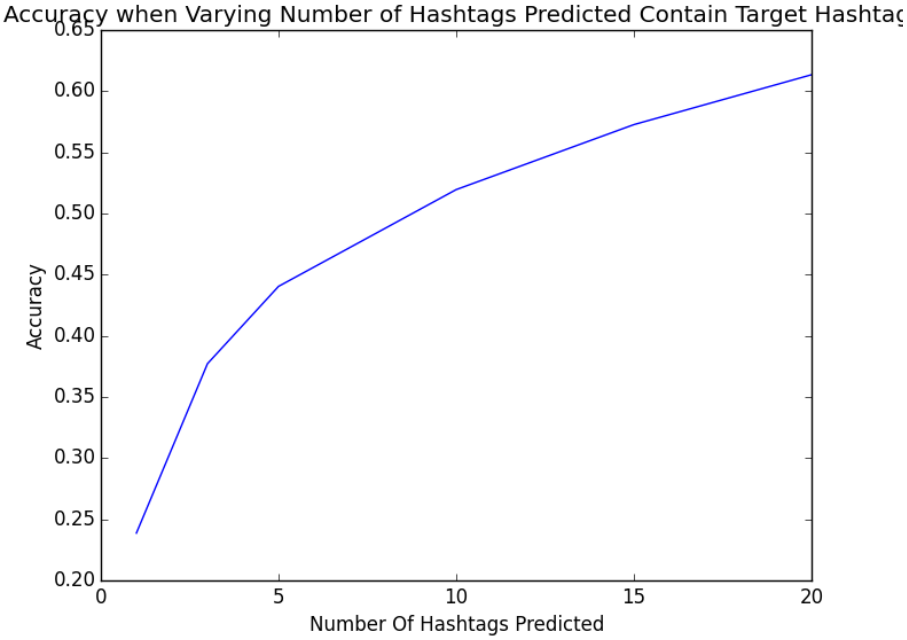


Figure 8: Accuracy when varying α
varying graph.png

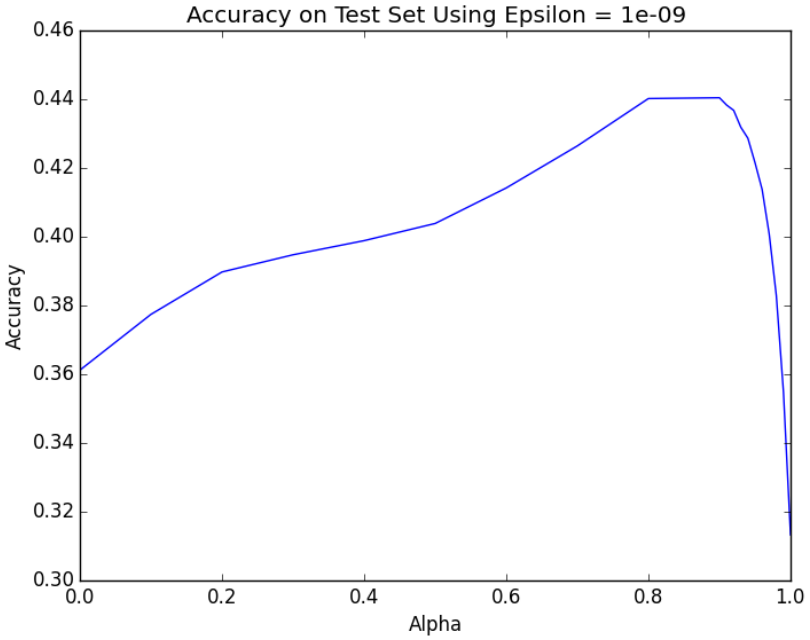


Figure 9: Accuracy when varying ϵ
varying graph.png

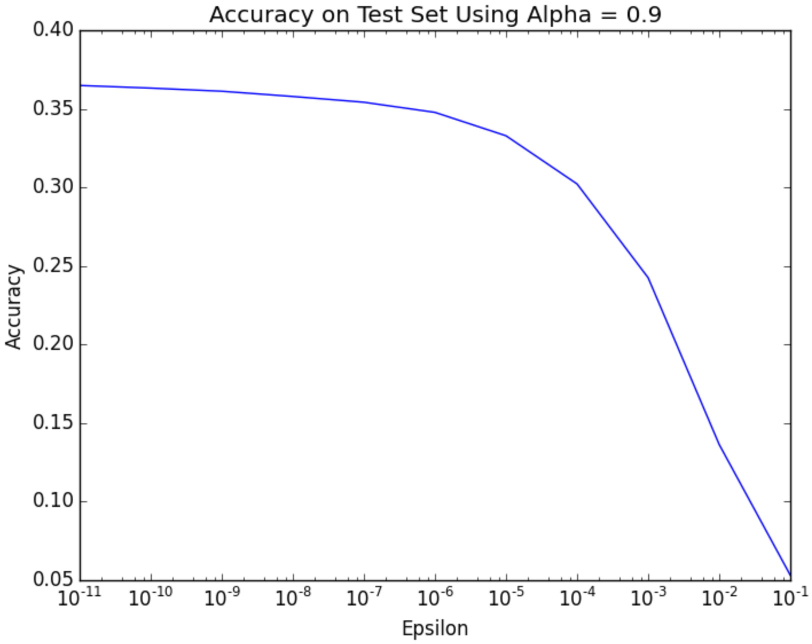
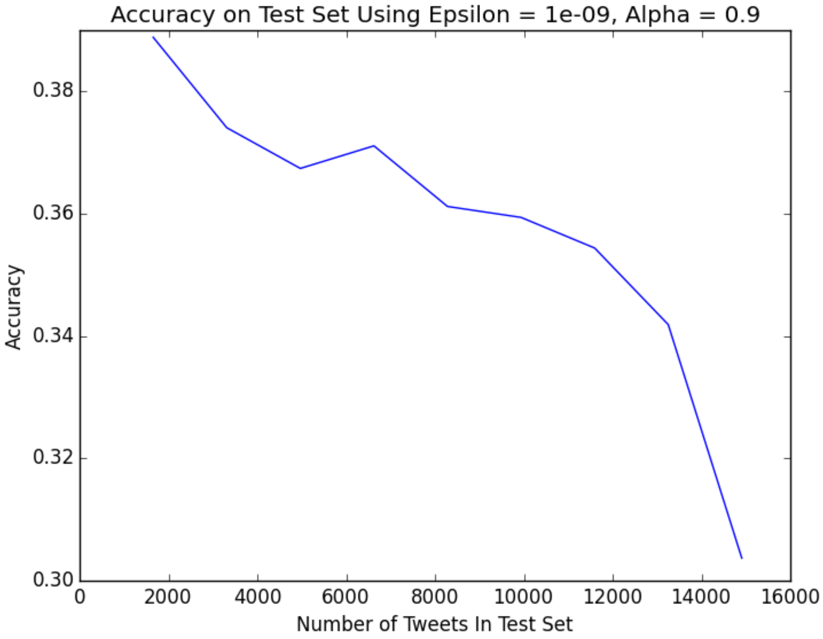


Figure 10: Accuracy when varying the size of the test set.
size ratio varying graph.png



6 Analyzing the Prediction Results from the Naive Bayes Classifier

The naive Bayes classifier showed that it was able to make some rough generalizations, an accuracy of %37 was achieved when predicting 5 hashtags out of 500. What is interesting to note is that the model was suspect to likely predict common hashtags for many tweets even if there were more relevant hashtags available for predictions.

An example that was noticed numerous times were tweets with the words 'thinking', 'great', 'race', 'rip', and 'silverstone' had the hashtag 'f1' associated with it. The model was very likely to predict 'f1' in the top 5, but it was consistently predicted with a lower probability than 'fb', 'ff', and 'followfriday'.

The hashtags 'fb', 'ff', and 'followfriday' were very commonly used throughout the dataset. Investigating the dataset showed that those three hashtags (or any other hashtag other than 'f1' and 'rip') occurred rarely with the words 'thinking', 'great', 'race', 'rip', and 'silverstone' yet the naive Bayes model still predicted them.

Further investigations showed that the model was consistently predicting the three most common hashtags as they were usually going to be correct with the dataset used. There were 0 cases where one of top 3 most common hashtags were not in a top 5 prediction.

7 Creating a Logistic Regression Classifier

A simple logistic regression model was created in Python using Tensorflow which is Google's opensource machine learning library. This model was not optimized in any way using cross validation, decays, early stopping, or regularization. It was trained and tested on a validation set to ensure over fitting did not occur. Figure 11 shows the results of the results of the training and testing on validation and test sets.

Figure 11 - Accuracies while training a Logistic Regression Model

```
TRAIN   VALID     TEST     COST       EPOCH
0.3217   0.3255   0.3172   5586.6922   0
0.4196   0.3891   0.3916   5048.2141   5
0.4798   0.4158   0.4187   4683.3321   10
0.5258   0.4284   0.4400   4370.3909   15
0.5584   0.4397   0.4601   4105.0878   20
0.5931   0.4509   0.4687   3880.3824   25
0.6237   0.4546   0.4729   3671.5684   30
0.6493   0.4598   0.4809   3483.7293   35
0.6750   0.4635   0.4835   3316.3807   40
0.6991   0.4701   0.4862   3155.8266   45
0.7232   0.4771   0.4857   3008.3967   50
0.7463   0.4799   0.4931   2877.5536   55
0.7678   0.4832   0.4920   2752.4329   60
0.7868   0.4832   0.4942   2636.9228   65
0.8037   0.4846   0.4947   2533.6726   70
0.8171   0.4878   0.4963   2430.9920   75
0.8324   0.4892   0.4973   2328.5888   80
0.8446   0.4935   0.5000   2244.2939   85
0.8578   0.4939   0.4995   2163.7039   90
0.8673   0.4977   0.5043   2087.4163   95
0.8775   0.5005   0.5043   2011.8925   100
0.8847   0.5014   0.5064   1943.6308   105
0.8926   0.5033   0.5085   1876.5630   110
0.8994   0.5061   0.5101   1819.0463   115
0.9048   0.5075   0.5074   1762.4684   120
0.9110   0.5065   0.5069   1709.9104   125
0.9159   0.5070   0.5074   1654.7716   130
0.9205   0.5075   0.5106   1614.5869   135
0.9241   0.5089   0.5101   1564.3276   140
0.9290   0.5089   0.5122   1518.7116   145
0.9317   0.5094   0.5133   1477.5996   150
```

8 Analyzing the Prediction Results from the Logistic Regression Classifier

The logisitic regression model out performed the naive Bayes model by %13 accuracy. It was much more consistent in its predictions without even being fine tuned with common techniques such as early stopping, regularization, and cross validation.

What's interesting is that logistic regression was not as prone as naive Bayes top consistently predict very common hashtags repeatedly. This shows that current model and more advanced model will be better for generalizing on new tweets.

Another strength to note in this model was that the correct hashtag predicted would quite frequently be the model's first and best guess. %28 of the time the model would guess the correct hashtag as the first result, %23 of the time it would be within the top 5 predictions, the remaining probability is a completely false prediction.

Some interesting examples to note were tweets with the word 'nintendo' were getting predictions such as 'e3' (a game news conference), 'jtv' (japanese tv station), and 'tcot' (an abbreviation for the game Mario Luigi : The Chronicles of Three) and 'sucks'. Another example that show cased the model's flexibility were tweets that contained the words 'book', 'read', 'can', 'join' were heavily associated with the hashtag 'wossybookclub' for obvious reasons.

9 Conclusion

The goal for this project was to research different machine learning methods for short text classification by predicting potential topics (hashtags) for short texts (tweets). The ambitious goal for creating a deep Boltzmann machine was not attempted because of time constraints.

The logistic regression model demonstrated considerable accuracy and generalization improvements for short text classification when compared to the naive Bayes model - a model which is famous for such a task. This research opens up the realizable possibility of more sophisticated algorithms making a strong case for a short text classification product.

References

[1] Srivastava, Nitish, Salakhutdinov, Ruslan, and Hinton, Geoffrey. Modeling documents with deep Boltzmann machines. In Uncertainty in Artificial Intelligence (2013).

[2] Mazzia, Allie, and James Juett. ”Suggesting hashtags on twitter.” EECS 545m, Machine Learning, Computer Science and Engineering, University of Michigan (2009).

[3] Go, Alec, Richa Bhayani, and Lei Huang. ”Twitter sentiment classification using distant supervision.” CS224N Project Report, Stanford 1 (2009): 12.