**Tim Zhang (110746199)**

**CSE519 Final Report: Dating Documents**

**Project Overview**

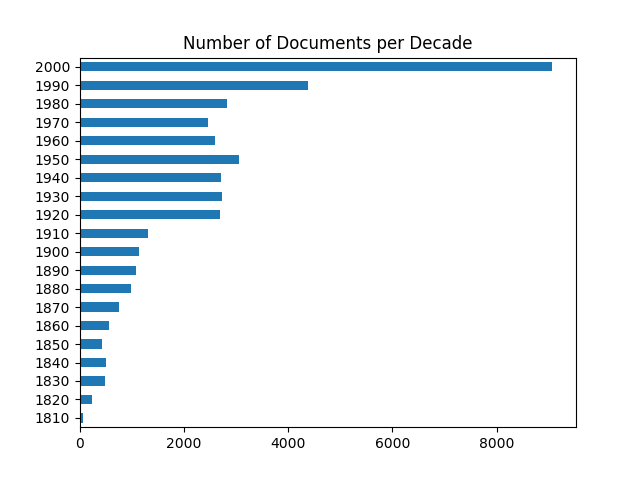
The problem of dating documents from their word distributions is an interesting and difficult task of natural language processing. This problem is an instance of k-classification, where *k*=20 in this case and spans the decade labels from 1810-2000. I believe that my final implementation serves as a good model for learning in this space and has relatively strong performance compared to prior attempts such as the RNN model implemented by Yingtao.

My final report will be partitioned into two parts. The first is a quick survey of the work I have done up to and including the Progress Report. The second part will encompass the new contributions which I implemented for my final project submission. All code is available at: <https://github.com/moduio/CSE519-2017-110746199/blob/master/Project/project.ipynb>

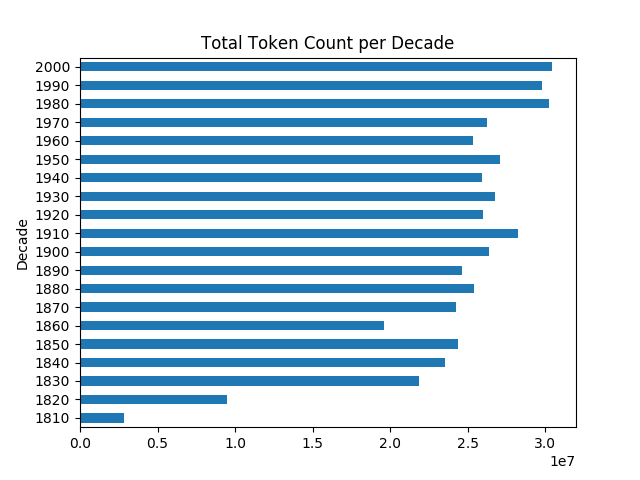
The sections preceding **“Refined Models”** contain work from the Preliminary and Progress Reports with the remaining code containing my final contributions.

**Corpora**

There are few datasets which have information regarding the publication dates of documents while also spanning the temporal range required for my project. Yingtao graciously provided the Corpus of Historical American English (COHA) which was otherwise behind a paywall. While COHA is the largest corpus of its kind, it is not necessarily clean. HTML tags and other Google meta-data pollute the data and needed to be sanitized prior to analysis. During the data exploration phase, I was most interested in learning the distributions of the corpus.



From the above figure, we can see that COHA is not balanced with respect document distributions over time. This aspect of the dataset is concerning.



By plotting the token frequency distribution we see a different story. COHA gives us a relatively strong guarantee of balanced token frequency between classes. This gives our algorithms a balanced empirical distribution that can be learned. In fact, the creator of COHA explicitly designed the corpus to be balanced by token frequency.

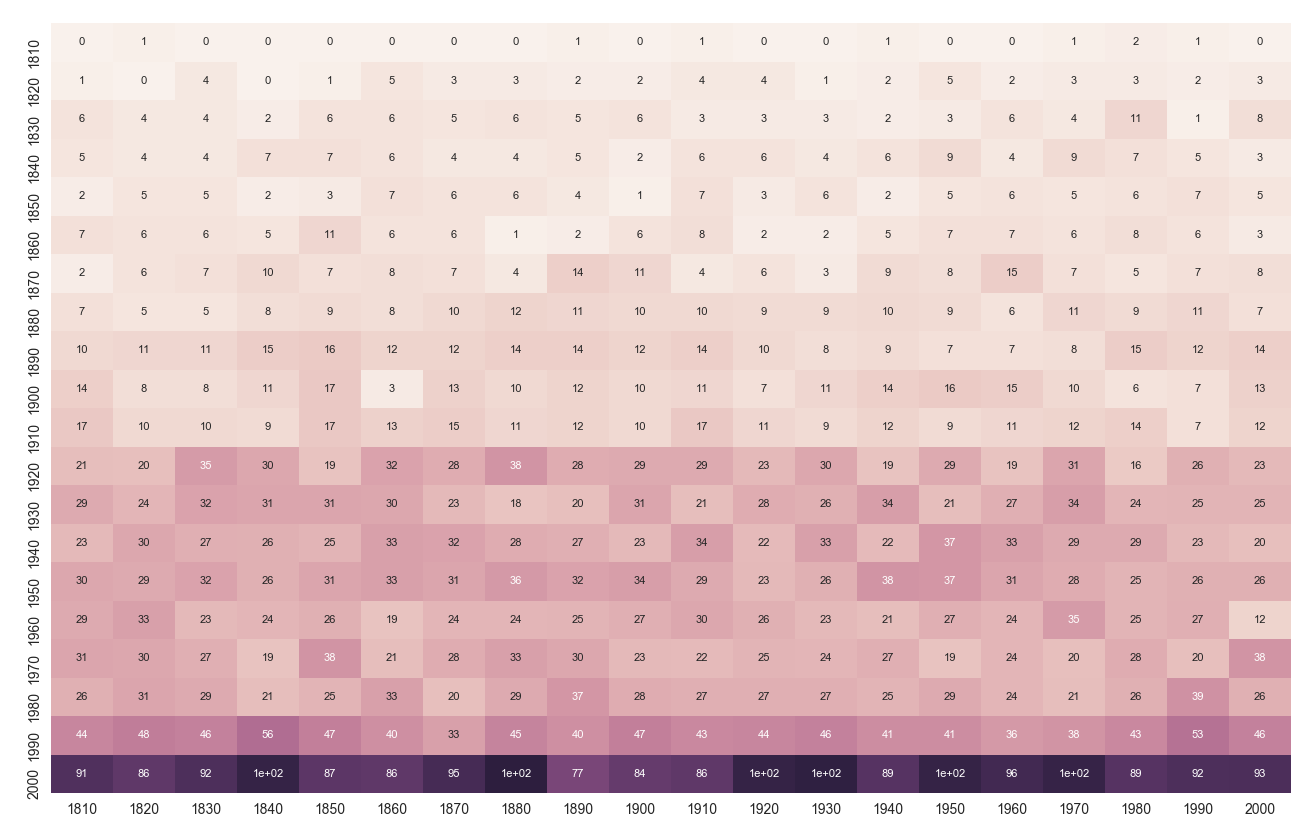
Acquiring COHA was a huge boon for my project, however I did not want to simply run analysis on a provided dataset. In search of more data I implemented programs to scrape the New York Times using their public API, along with a program which interfaced with Project Gutenberg documents and allowed for manual class assignment. I successfully gathered tens of thousands of documents from the New York Times. I will discuss the acquired NYT dataset in more detail during the second part of my report. Due to time constraints, I was not able to incorporate Gutenberg data into my models, however the labeling interface is implemented and almost 700 documents have been classified by hand.

**Modeling COHA**

My first attempts at document dating only considered the COHA dataset. I implemented an evaluation environment and performed many experiments with different learning models. Specifically, I trained Naïve Bayes, Random Forests, and SVM on the COHA dataset using unigram TF-IDF bag of words feature representations. I also tried Gradient Boosted Trees but was unable to train a model due to computational limitations. Additionally, I considered bigram and trigram TF-IDF BoW features but abandoned this approach as I found my algorithms enjoyed relatively small improvements compared to increased amount of training time required to fit the parameters of the language model.

Next I will briefly discuss the performance of SVM, which was my strongest model, along with the uniform random guessing algorithm which does a good job of highlighting certain aspects of the test set distribution. It should be noted that during this point in my project I considered full length test documents as samples. This is in contrast with my later results which were computed on subsets of COHA (COHA\_100, COHA\_500, COHA\_1000, COHA\_2000) as in the RNN implementation.

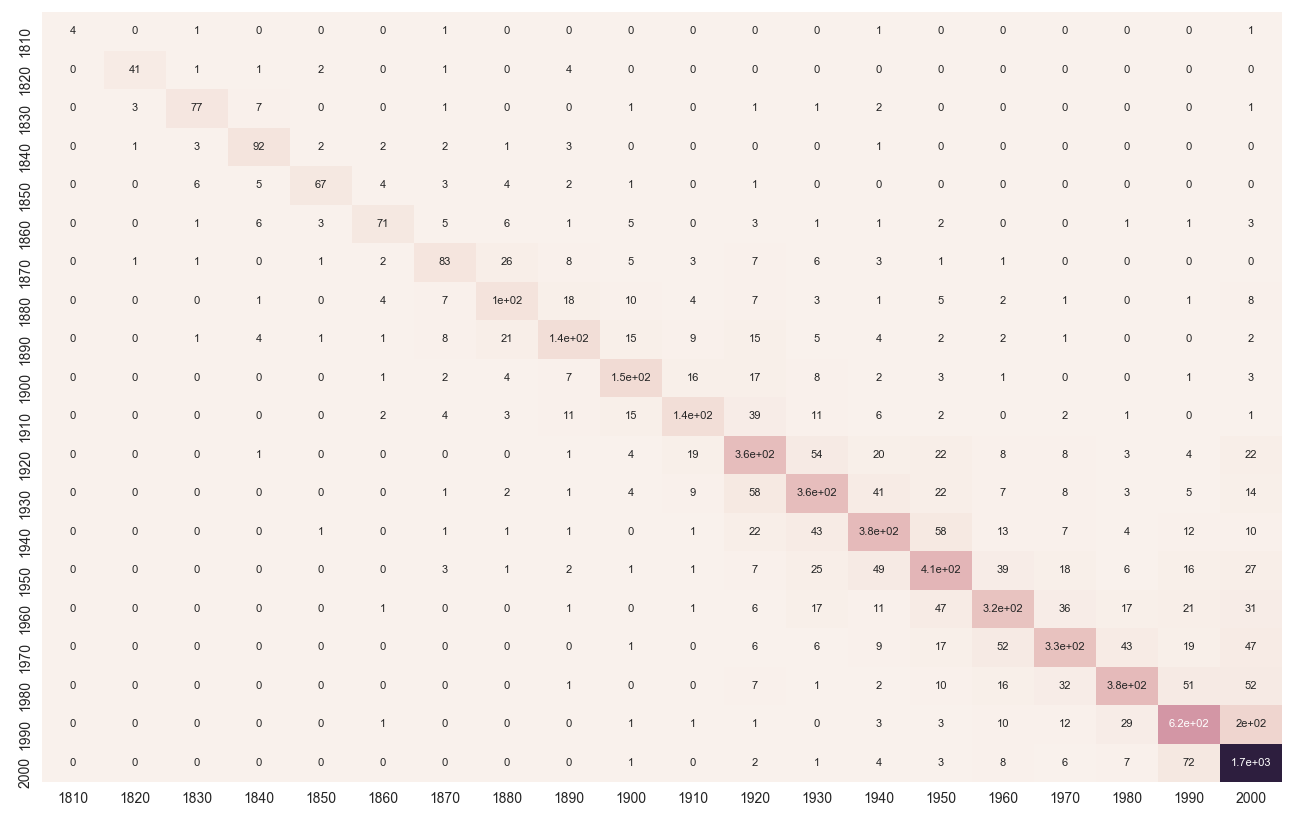
**The Monkey:**

****

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **MAE** | **Accuracy** | **Precision** | **Recall** | **F1** | **Support** |
| 71.4225 | 0.0515 | 0.10 | 0.05 | 0.06 | 8000 |

Analysis on the behavior of the monkey shows features of the test set which align with our prior beliefs about the problem space. We notice that our test set is mostly comprised with documents from the 2000s, which is reflected in the underlying COHA distribution. Additionally, most of the test data comes from the decade range 1920-2000. Another observation is that the accuracy is roughly 5% which is to be expected given that *k*=20.

**Linear SVM:**



|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **MAE** | **Accuracy** | **Precision** | **Recall** | **F1** | **Support** |
| 6.785 | 0.732125 | 0.74 | 0.73 | 0.73 | 8000 |

Immediately we see that SVM using the default settings from Scikit-learn is performing exceptionally well at classifying the COHA documents. It is clear from the confusion matrix that the SVM model is capturing some real aspect of the problem domain. We can see this in the beautifully presented tightly distributed prediction mass on the main diagonal. I am perhaps overly enthusiastic about this result since my later results are unfortunately not as strong.

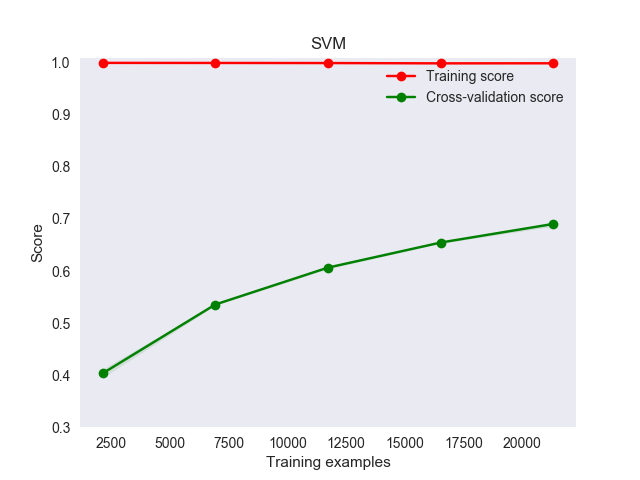
**Final Progress Overview**

For my final submission, I have accomplished most of the goals which I previously introduced in my Progress Report. This includes extending my corpus, visualizing the model learning behavior, improving my models using cross validated hyperparameter tuning, and performing an “apples to apples” comparison against the RNN implementation of Yingtao. In the direct comparison with the RNN model, I found my SVM implementation to perform favorably in each of the text domains (New York Times, COHA\_100, etc) with almost half the MAE scores in 4/5 domains. Unfortunately, due to computation constraints I was unable to experiment with advanced models such as CNN or Gradient Boosted Trees on this domain.

The remainder of this report will detail my process and implementations for improving the model on COHA, extending the corpus, training a model for the extended corpus, and finally a comparison with the RNN implementation.

**Improving SVM**

Having found very positive results on COHA using SVM I was interested in visualizing how well the model was generalizing to the test set. Additionally, I wanted to know if my model would improve from incorporating more sources of data. To this end I implemented learning curves for SVM on COHA, which show the test and cross validation scores as a function of training set size.



We can interpret this plot as indicating that using more training data would lead to better model performance, although the gains may be less pronounced since the slope is decreasing.

I was also interested in seeing how much more predictive power I could squeeze out of SVM. For this I implemented cross validated grid search for hyper parameters. I found that the optimal value of *C* for Linear SVM trained on COHA resulted in the following scores:

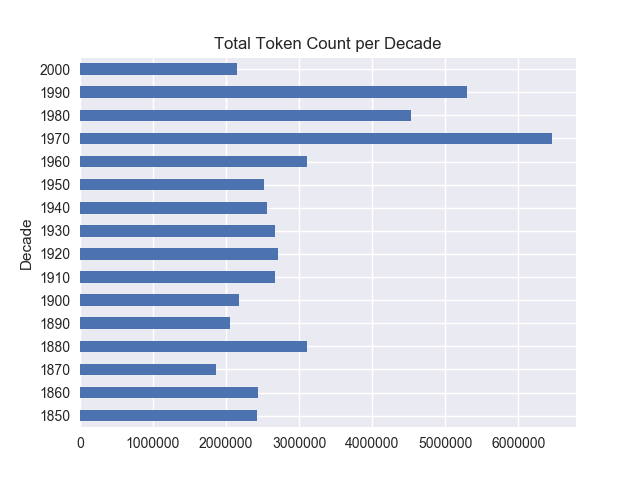
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **MAE** | **Accuracy** | **Precision** | **Recall** | **F1** | **Support** |
| 6.09 | 0.733 | 0.73 | 0.73 | 0.73 | 8000 |

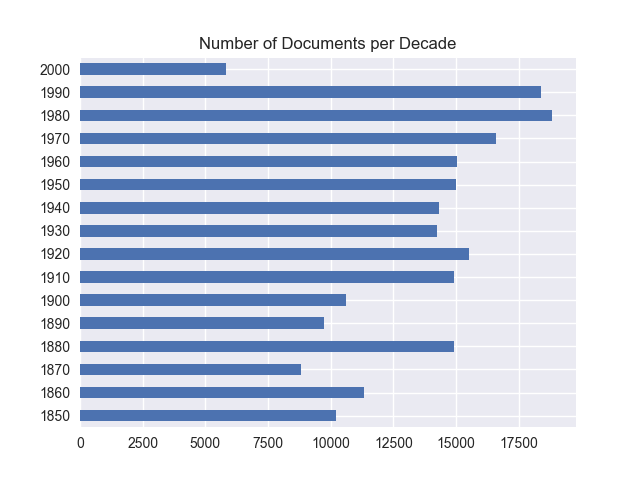
These scores (optimized with respect to MAE) show a roughly 10% improvement in MAE. This is not that much better than the performance of SVM using the default settings. While slightly disappointing from my perspective, it does go to show how difficult this problem domain is.

It should be noted that although my Jupyter Notebook file only has a hyperparameter search range of [.001, .01, …, 1.5], I tried many other combinations which I omitted for cleanliness of the notebook. I also attempted cross validated grid search for Random Forest while tuning the number of trees, but this approach did not perform better than SVM so I omitted these experiments from my final notebook.

**Extending the Corpus**

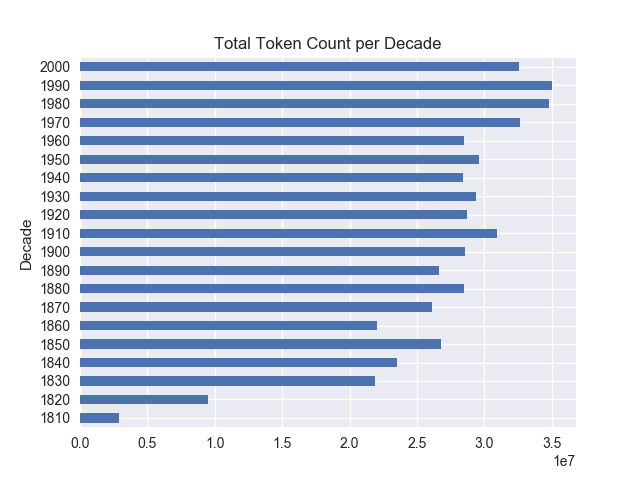
Having seen the limits of my previous model along with the promise of more data, I felt confident in incorporating the New York Times dataset which I had painfully acquired via custom API scraping. This data included article snippets and summaries from each year available in their database from the 1850s – 2000s. To avoid some issues of noise, I only considered those “documents” which were longer than 10 words.

From the above plot we can see that the token frequency distribution is relatively balanced

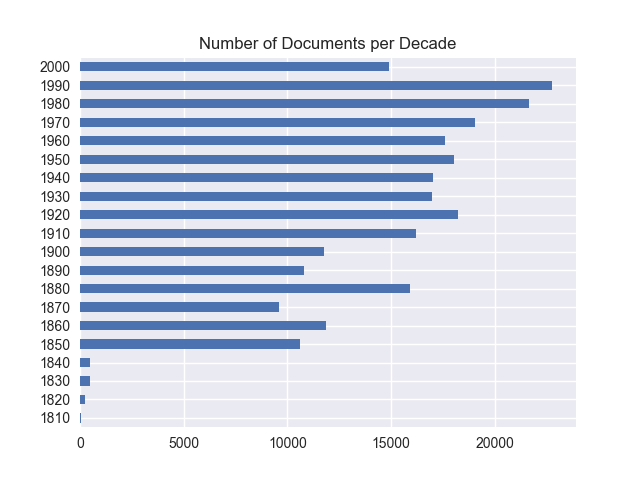
except in the 1970-1990 range which has almost twice the representation of tokens.

We see that the document frequency is relatively balanced in the NYT data.

We would like to verify that the token frequency of the merged dataset is relatively balanced, as COHA was only balanced via tokens. This distribution is shown below:



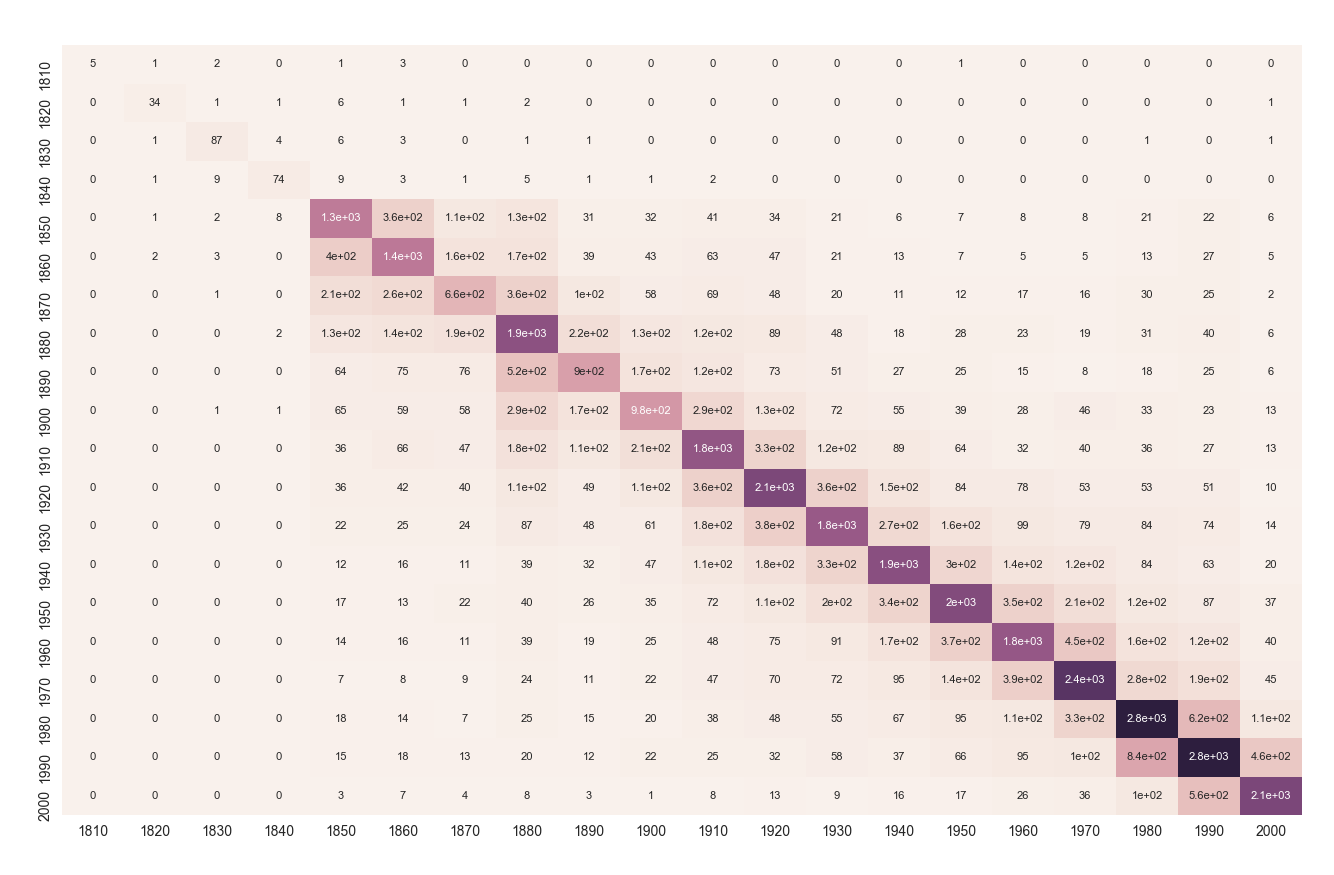
We see that the token frequency of COHA/NYT is very similar in shape as that of COHA alone. This indicates that our merging has maintained a distribution which has the same balance assumptions as the original COHA dataset. We can thereby feel confident that merging the two datasets will give a similar environment as with COHA alone.



Plotting the document distribution of COHA/NYT we see that the labels are still imbalanced.

**Modeling COHA/NYT**

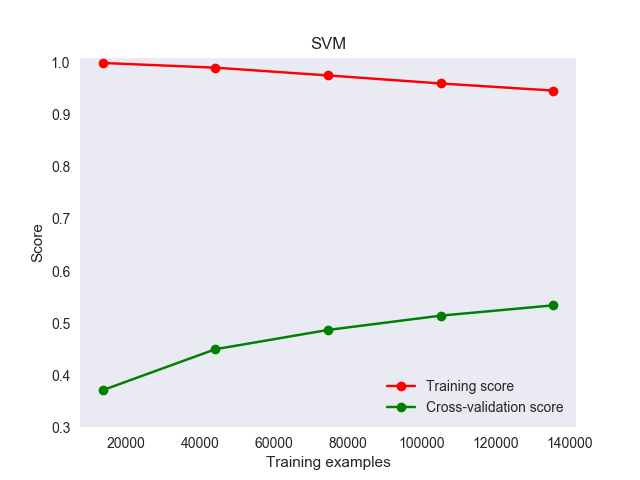
Having implemented cross validation, learning curves, etc previously, I trained my SVM model on the merged COHA/NYT dataset without the need for additional domain specific implementations. First I found optimal hyper parameters via grid search using trial and error to determine reasonable hyper parameter ranges. The model test results are as follows:



|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **MAE** | **Accuracy** | **Precision** | **Recall** | **F1** | **Support** |
| 11.075 | 0.566 | 0.57 | 0.57 | 0.56 | 50844 |

We can see that the accuracy score of the SVM is almost 20% worse than on vanilla COHA. Additionally, the MAE score increased to almost double the value of 6.09 achieved previously.

When I visualized the learning curve for COHA/NYT I immediately realized that I had underestimated how much more difficult the NYT corpus made the document dating problem. We see that the accuracy seems to be converging to approximately 0.5 as the data size grows.



**Apples to Apples Comparison**

Although my model performance had dramatically decreased on the extended dataset, there was no way to determine how to interpret the meaning of this degradation without some baseline comparison. To achieve this I would compare my model performance against the RNN implementation by Yingtao. During my extensive discussion with Yingtao on this subject we came to the agreement that the best way to directly compare our models was by running the trained RNN on the same test set which I used to evaluate my models. Due to the differences in feature representation (embedded Ngrams vs TF-IDF BoW fit with COHA/NYT), there was no meaningful way to “train” either model on the other models training set.

To run the RNN on my test data I needed to extract the documents and labels from my DataFrame, and then save the result as a new file in the same format that was parsed in the RNN implementation. I then made some modifications to the RNN code which would allow parsing and forward inference which produced the following results:

|  |  |  |
| --- | --- | --- |
| **Dataset** | **MAE** | **Accuracy** |
| NYTimes | 25.115 | 0.018 |
| COHA\_100 | 25.862 | 0.026 |
| COHA\_500 | 21.985 | 0.031 |
| COHA\_1000 | 21.377 | 0.032 |
| COHA\_2000 | 20.979 | 0.034 |

Next I would need to implement the same subdocument parsing logic to test my model on subdocument splits of COHA. My tuned SVM trained on COHA/NYT results were:

|  |  |  |
| --- | --- | --- |
| **Dataset** | **MAE** | **Accuracy** |
| NYTimes | 11.918 | 0.540 |
| COHA\_100 | 25.179 | 0.335 |
| COHA\_500 | 14.190 | 0.526 |
| COHA\_1000 | 11.865 | 0.575 |
| COHA\_2000 | 10.500 | 0.610 |

As we can see my results are almost twice as good with respect to MAE. Concerning accuracy, there may be some bug in the RNN implementation since my accuracy scores are much better.

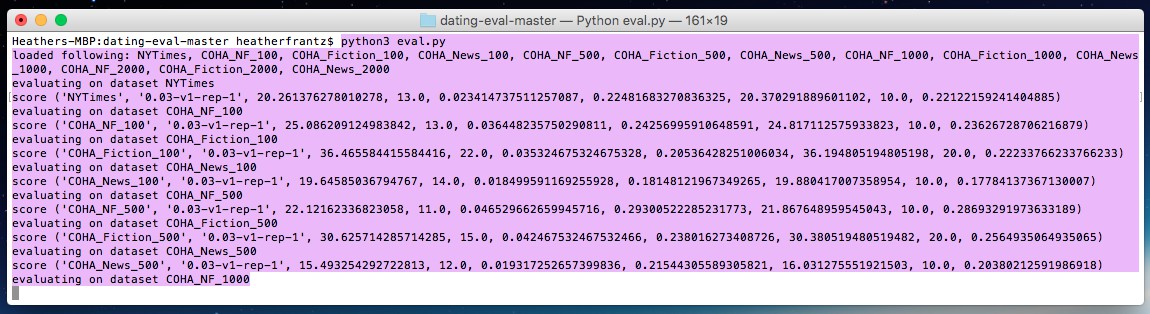
**Interpretation**

From the array of experiments conducted on the merged corpus it seems that the New York Times data increased the difficulty of the dating problem substantially. It is unclear how much this is a function of the structure of the data itself and how much is inherent in the problem space when considering news articles.

News headings and snippets may follow some regular structure which is not present in other documents, such as declarations and specific grammatical structure or phrases. This regular structure may be uniform over certain timeframes which makes it difficult for the algorithms to discern the class labels. It is also possible that news articles as a general class of documents follow certain structural forms which include many of the same words regardless of timeframe (perhaps due to tradition or protocol).

My thoughts regarding the experiments using my models vs the previous work are slightly conflicted. There is some merit to the idea that by training a model on the corpus itself there should be some bias or “overfitting” towards that area of document space. However, given that Google Ngrams should be representative of a much larger space of text I find it hard to believe that a model trained using Ngrams should be so much worse than mine. In fact, if the documents of COHA are representative of documents from the time periods of interest, then a model trained on Ngrams should be able to model this distribution well.

I think it may be the case that there is some bug in the implementation of the RNN program. This may be at the corpus level where there were some interesting design decisions. A corpus representation problem would also explain why the Naïve Bayes model performed poorly as well. Another possibility is that the RNN was not tuned sufficiently for whatever reason. It is surprising how large the disparity between MAE and accuracy is in the RNN. I have included a screenshot of the original program running in the original test environment:



Accuracy is the 5th value in the score object, and assuming it is legible, these scores are clearly also extremely low. In fact, the accuracy is worse than that of the monkey.

I should make it clear that I am in no way denigrating the performance of my implementations using SVM. It could also simply be the case that an SVM is better suited to learn from this imbalanced dataset. The possibility of an inherent advantage of the maximum margin objective function along with the choice of a TF-IDF BoW language model fit using the full corpus may simply be the better model design for the investigated corpora.

**Conclusion**

The document dating problem has turned out to be a very difficult task. It seems unlikely that any method could break 90% or even 70% accuracy. This entire process has proved to be enlightening in many aspects for me personally. I learned to temper my expectations with respect to my model performance and parameter tuning. My project provided an open-ended problem with dirty and insufficient data sources. By cleaning my data and visualizing the entire problem domain in a principled manner, I gained valuable experience in approaching problems as a data scientist, and not only as a machine learning API monkey.