COMP135 Project 3 Writeup

May 2019

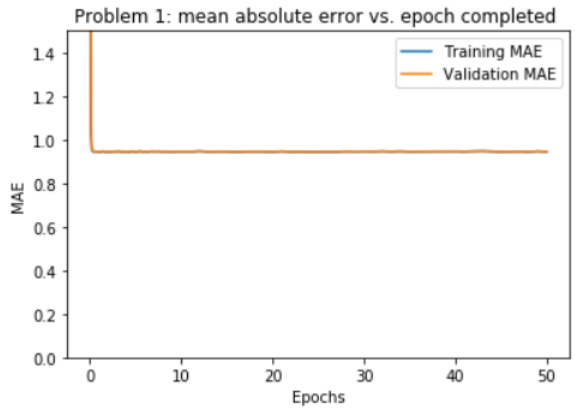
Darren Ting

1 Report for Question 1

1.1 Part A

M1

MAE vs Epochs



The recommendation system basically takes the mean of all data given and returns that as a predicted rating. Therefore, the results will always be the same because it is the mean of all data given. Thus, the training and validation MAE will be identical because the loss is measured by its residuals.

1.2 Part B

In the case of this specific dataset, the training and validation set are identical, so regularization would not really improve it, but at the same time it is good to have one. What is special about this task that makes a regularization term not as necessary is that it mainly looks at the mean value, and doesn't try to overfit based on the values given. In addition, what is special about this task is that we may just care about precision than overall MAE because it may just be important to have the first few recommendations very accurate

1.3 Part C

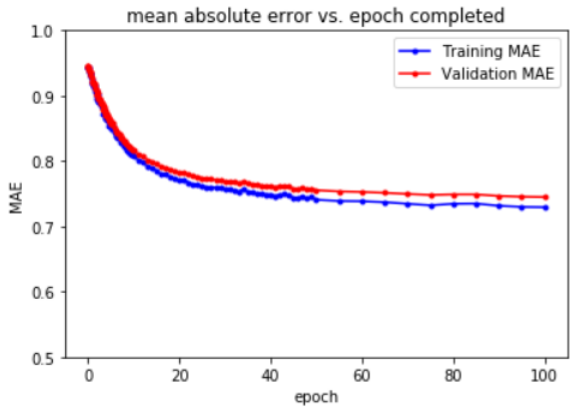
The operation is to calculate the mean of all of the ratings with numpy's mean function. The calculated mean is 3.53239. This is fairly close to my SGD solution, which was 3.5367.

2 Report for Question 2

2.1 Part A

M2

Training and Validation

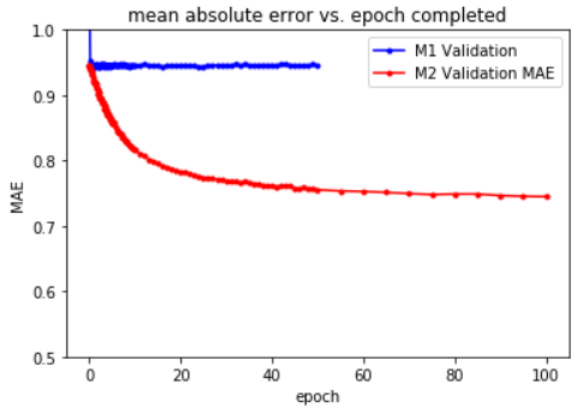


There appears to be a slight amount of overfitting. The training MAE ends up to be around 0.723 while test MAE is around 0.74. This was done with a step size of 0.25.

2.2 Part B

M2 vs M1

Model Comparison between Model and Model 2

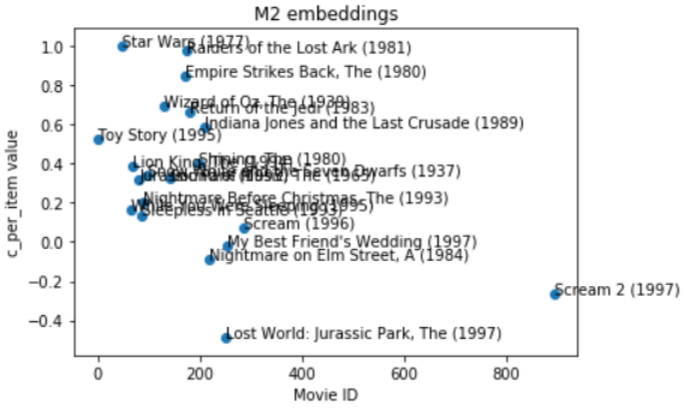


The validation error for M2 is much lower than that of M1. This is because in M1, the ratings of other movies affect each movie's predicted rating given a user. Therefore, the scalars are more focused on each individual movie, and outside data does not affect it.

2.3 Part C

Movie Embeddings

Learned Per-Movie Rating Adjustment Parameter vs ID



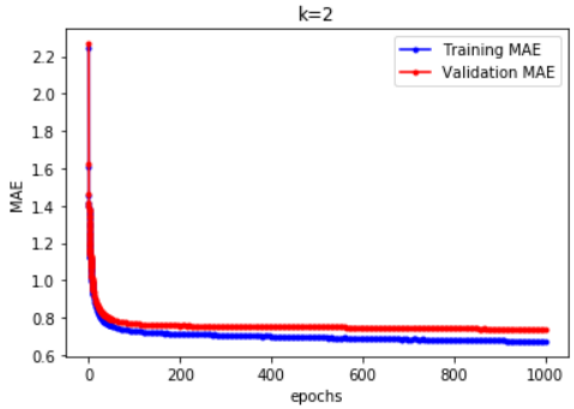
The trend seems to be, the higher the c value, the higher the movie rating. A movie with a large positive value (Star Wars) is regarded to be a very good movie. The movie with the lowest C value is Jurassic Park, the Lost World, which based on IMDB reviews, is not a good movie. This contrasts with the movie with the largest C value which was Star Wars, which is regarded as one of the best movies of all time potentially. Therefore, a higher c value means a better movie.

3 Report for Problem 3

3.1 Part A

M3 n factors = 2

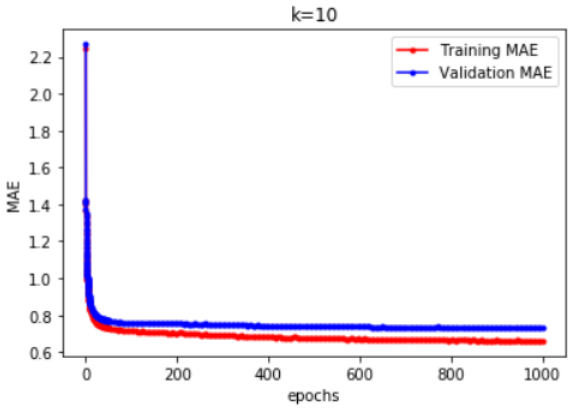
Training and Validation



This is run on 1000 epochs, and shows overfitting. The training MAE is around 0.67, and the validation MAE is around 0.739.

M3 n factors = 10

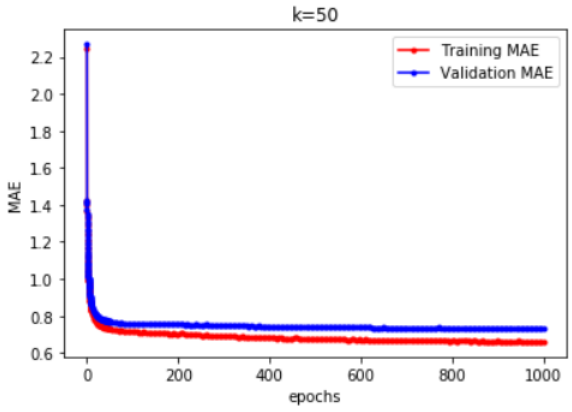
Training and Validation



This is run on 1000 epochs, and shows overfitting. The training MAE is 0.66, while the validation MAE is 0.732.

M3 n factors = 50

Training and Validation

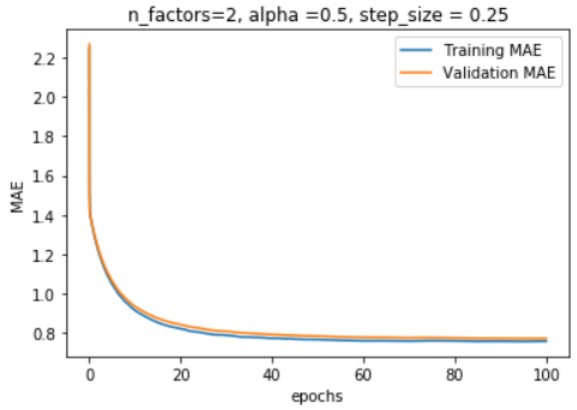


This is run on 1000 epochs, and shows overfitting. The training MAE is 0.6609, while the validation MAE is 0.732.

3.2 Part B: Adding Regularization

M3 n factors = 2

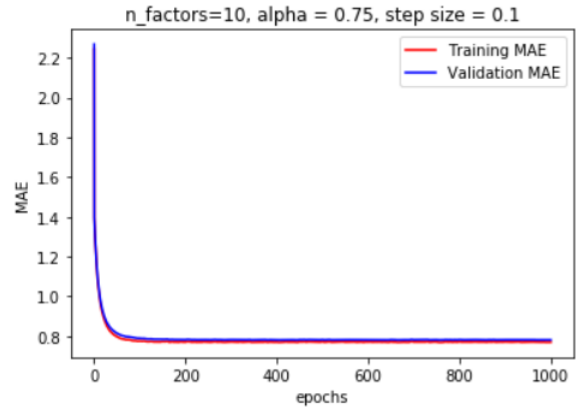
Training and Validation



This is run on 100 epochs, and no longer shows strong overfitting. The training MAE is around 0.76, and the validation MAE is around 0.77. While this MAE is higher, it does not overfit.

M3 n factors = 10

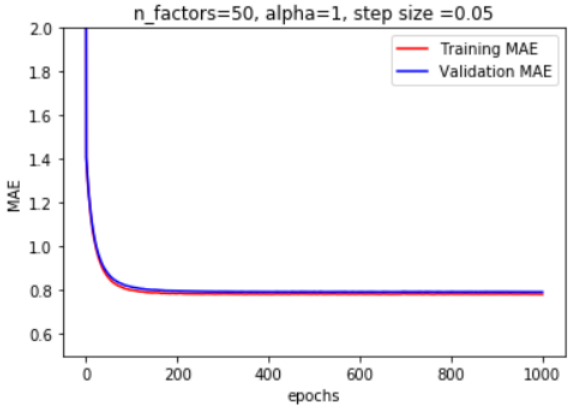
Training and Validation



This is run on 1000 epochs, and no longer shows overfitting. The training MAE is 0.77, while the validation MAE is 0.78. While the MAE is higher, it does not overfit.

M3 n factors = 50

Training and Validation



This is run on 1000 epochs, and no longer shows overfitting. The training MAE is 0.78, while the validation MAE is 0.79. The MAE is higher than its non regularized counterpart, but it is no longer overfitting.

3.3 Part C

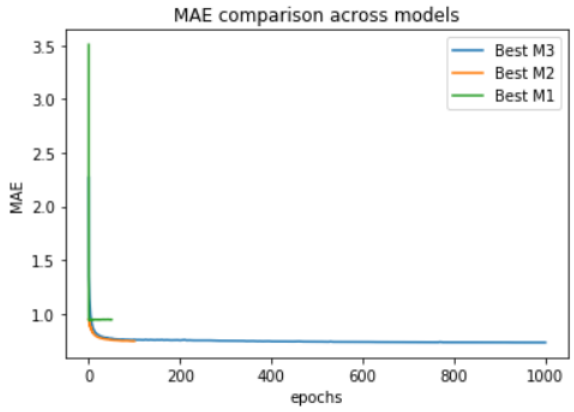
We can reduce K to decrease the number of factors to avoid overfitting when using the same model M3 while using SGD. The larger K is, the higher chance it can overfit because each user and item vector can hold more features than potentially necessary.

3.4 Part D

Minimum Validation MAE across models

```
M3, K=2, regularized 0.7720784952451926
M3, K=10, regularized 0.7810769035947673
M3, K=50, regularized 0.7919037217149163
M3, K=2 0.7392416242473782
M3, K=10 0.732368640716035
M3, K=50 0.732368640716035
M2 0.744843941220189
M1 0.9433754963030364
```

Validation MAE Comparison across models

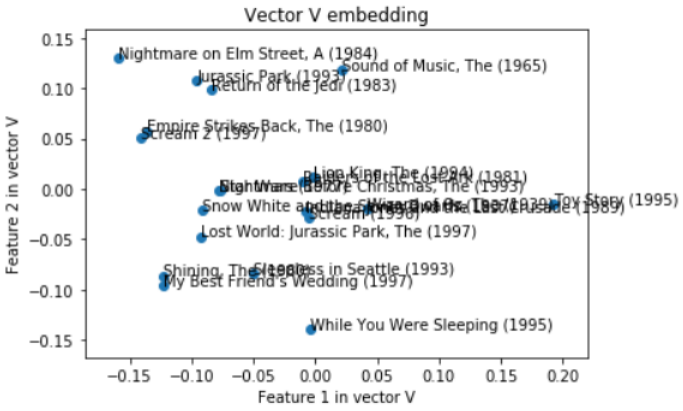


My best M3 is either k=10 and k=50, which is marginally better than M2, and much better than M1. In terms of predictive performance, the improvement between k=2 and k=10 is not much. I would recommend either k=10 or k=2 because it is faster than k=50 with not much improvement. Note that the small one on the graph is M1 because I ran it for 50 epochs because it is constant and will not change once the average is calculated. On the graph it is not exactly clear that M3 is the minimum, which is why I included the minimums for each model.

3.5 Part E

Vector Embeddings

Item Vector



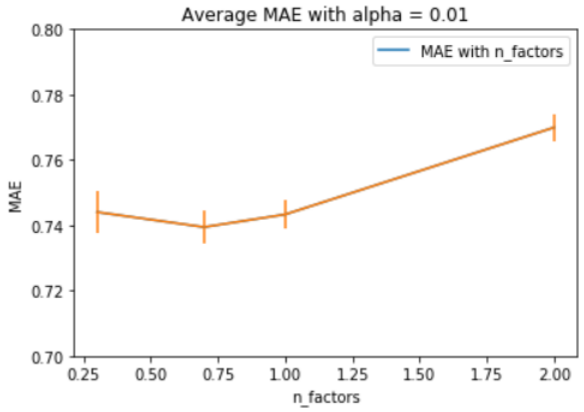
An interpretable trend on the graph appears to be 1st feature of vector v decreases and the 2nd feature of the vector increases, the more violent the movie is. This is shown with movies such as Nightmare on Elm Street and Jurassic Park.

4 Report for Problem 4

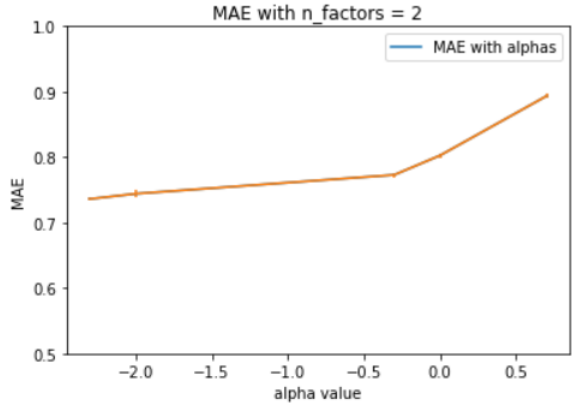
4.1 Part A

I used GridsearchCv to hypertune parameters such as K and alpha. The following are graphs holding a variable fixed. The graphs show the validation MAE

MAE vs hyperparameters



Based on this graph, the best k = 2 when there is a fixed alpha of 0.01. Note that the graph is log10 of k instead of the actual value. Based on the standard deviation error bars, I am fairly certain that k = 2 is optimal factors.



Based on the graph with a fixed k = 2, the optimal alpha value is 0,01. It appears as though as the alpha increases, the validation MAE increases.

Based on a gridsearchcv, I would use the parameters K=2, alpha = 0.01. I used a learning rate of 0.01, which is slightly lower than the default. I chose this learning rate with gridsearch CV. Based

on the plots of number of factors given the learning rate, and vice versa, the parameter values I have chosen are somewhat optimal.

4.2 Part B

Comparison to other models

Cross Validation

```
Evaluating MAE of algorithm SVD on 5 split(s).

                  Fold 1  Fold 2  Fold 3  Fold 4  Fold 5  Mean    Std
MAE (testset)     0.7273  0.7289  0.7417  0.7341  0.7403  0.7344  0.0058
Fit time          1.19    1.20    1.20    1.18    1.28    1.21    0.04
Test time         0.11    0.11    0.11    0.22    0.10    0.13    0.04
```

With an average MAE of 0.7344, the surprise SVD performs around the same as the best models in M3. In both cases, the alpha/regularization value is much lower, which does not punish overfitting as much. Both models outperform the ones that use regularization to prevent overfitting.

5 Report for Problem 5

5.1 Method

My method for the open ended task was to use the KNN approach to collaborative filtering from surprise. More specifically, I decided to use the KNN Baseline algorithm because essentially, M2 is a sort of baseline, and since it performed relatively well compared to more complicated models, I wanted to use it.

My approach to finding the optimal parameters was using a careful gridsearchcv with many different hyperparameters. I used two different metrics to measure the gridsearchcv: MAE and FCP. FCP stands for fraction of concordant pairs which was used in collaborative filtering with ordinal user; however, I ended up using the hyperparamters that minimized MAE due to the two results performing very similarly but the MAE optimal one with a lower MAE. The two results are pictured below.

Result using FCP optimal hyperparameters

18	darren	0.7175	0.8634	0.8235	4.0893	3.9902

Result using MAE optimal hyperparameters

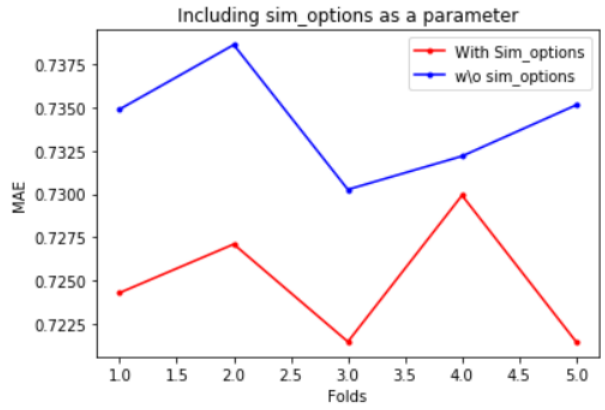
23	darren	0.7096	0.8634	0.8235	4.102	3.9902

As shown by these two results, the MAE optimized hyperparameters are better.

5.2 Hyperparameters

Similarity Measure Configuration

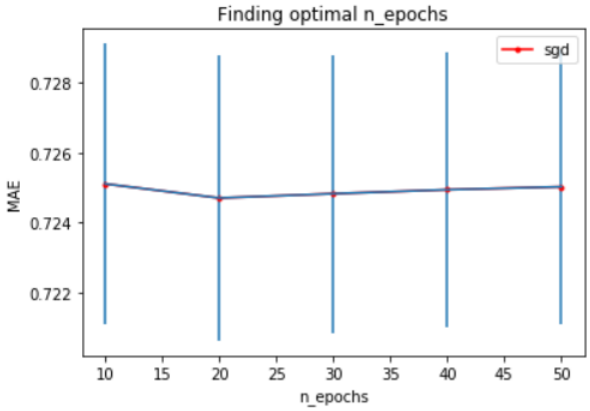
With and without sim options



One hyperparameter I tuned was the $sim_options$. On the documentation, it recommends to use sim measure of a pearson baseline. I plotted cross validation with and without the sim options, and the sim options clearly improves MAE.

number of epochs

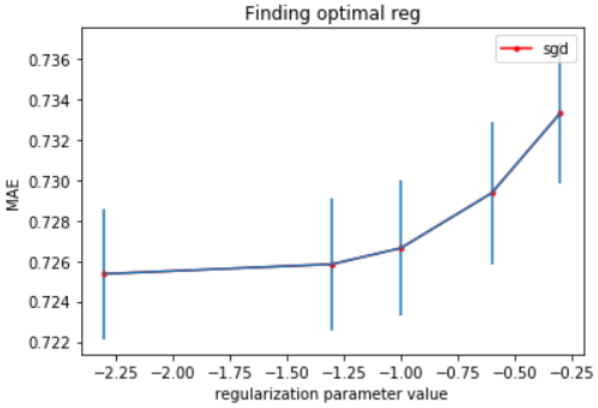
Epochs vs MAE



Based on the graph and standard deviation bars, having 20 epochs instead of any other results in a slight improvement in the algorithm's MAE.

Regularization

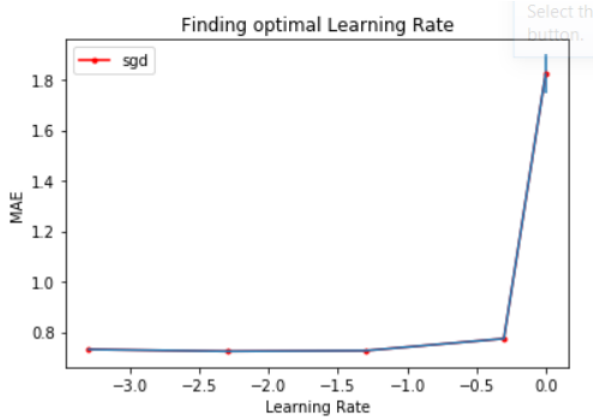
regularization vs MAE



The optimal regularization value is 0.05 based on the graph (the graph shows the log of the regularization term).

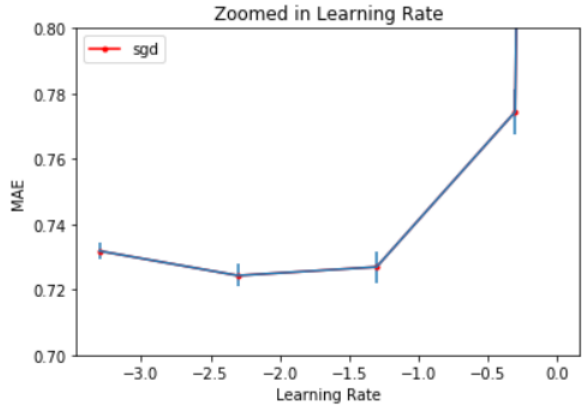
Learning Rate

learning rate vs MAE



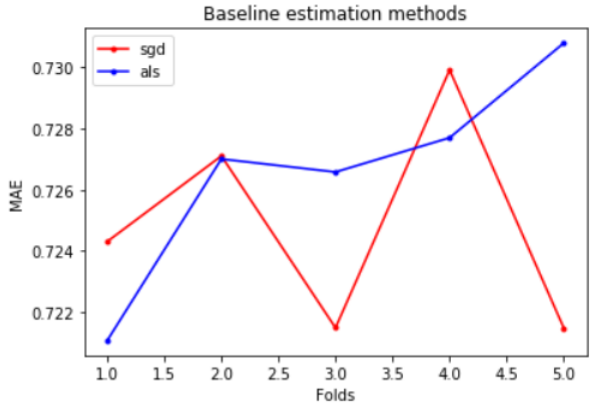
Based on this graph, it may not be particularly clear what the optimal learning rate is, but zoomed in shows it clearer.
Based on this zoomed in graph, the optimal learning rate is 0.005. Once again the graph shows the log10 of the learning rate.

learning rate vs MAE



Baseline Method

SGD vs ALS



This graph compares the ALS method and the SGD method for calculating the baselines in the KNN. On average, the SGD has lower MAE, so it is the optimal method to calculate the baselines.

5.3 Choice and results

Choice
The hyperparameters I used:
sim options: pearson baseline
bsl options are the following:
method = sgd
learning rate = 0.05
reg = 0.005
Result

Leaderboard

23	darren	0.7096	0.8634	0.8235	4.102	3.9902

This is around 3 percent better than my internal validation efforts. This may be because I did a much more serious grid search and hyperparameter tuning, and there were hyperparameters that were made for collaborative filtering improvements.

6 Report for Bonus Problem

6.1 Part A: Analysis

The problem is essentially, given an estimated user preferences (through the U vector), what is the gender of the user? I chose the Random Forest Classifier for this problem because I imagined that similar user vectors would be grouped together, and not really have a sort of linear decision boundary. I would think that it would be in classifiable regions. I tuned hyperparameters using a GridSearch CV. I tuned n estimators, bootstrap, warm start and criterion, and found that 10 bootstrapped trees without warm start measuring with a gini criterion was better. In my gridsearch CV, I looked for the results with the highest balanced accuracy because the data was not evenly distributed. When using gridsearch CV, I tested it with the entire data set and used cross validation with 5 folds. The metric I used to select the best hyperparameters was balanced accuracy because I assumed that I was using an unbalanced dataset.

6.2 Part B: Results

Confusion Matrix

Predicted	0	1
True		
0	20	127
1	37	288

I split the training set in half. I trained my classifier on half the data and used the confusion matrix on the other half of the data. My accuracy was 0.65, so my error rate was 0.35. Predicting on chance would be having a 0.5 error rate, so I would think a 0.15 improvement from that

is somewhat significant. Interestingly enough, when I was scoring my gridsearch CV, many test scores ended to be around 0.5, and the classifier actually performs better on the test set. Given the fact that U vectors may not even be completely accurate because they are estimations, this classifier can potentially perform better on more accurate input data.

Chance vs Classifier