Homework 2 Answers
Kelly Scott Sims

Q1.1a

$$I.\quad NLL(D,w)=-\sum_{i=1}^{N}[(1-y_i)\log(1-\sigma(w^T x_i))+y_i\log(\sigma(w^T x_i))]$$

Take the partial of both sides of the plus (+) sign as

$$II.\quad \frac{\partial NLL}{\partial W}=\frac{-\partial}{\partial w}(1-y)\log(1-\sigma(w^T x_i))-\frac{\partial}{\partial w}y_i\log(\sigma(w^T x_i))$$

Knowing that the derivative of a*log(x) = $\frac{a}{x}$ and with the chain rule being $\frac{d}{dx}[f(g(x))]=f'(g(x))g'(x)$ both sides of the negative (-) sign with a partial in II. can be reduced *to*

$$III.\quad [\frac{-(1-y)}{(1-\sigma(w^T x_i))}]\frac{\partial}{\partial w}(\sigma(w^T x_i))-[\frac{y_i}{\sigma(w^T x_i)}]\frac{\partial}{\partial w}(\sigma(w^T x_i))$$

Since both sides of the minus (-) sign contains a partial, it can be pulled out from both sides

$$IV.\quad \frac{\partial}{\partial w}(\sigma(w^T x_i))[\frac{-(1-y)}{(1-\sigma(w^T x_i))}-\frac{y_i}{\sigma(w^T x_i)}]$$

Now we must handle the partial derivative with respect to w of $\sigma(w^T x_i)$ where (σ) is $\frac{1}{1+e^{(-w^T x_i)}}$.

Again using the chain rule, first we will take the derivative of the "outside" sigma function $\sigma(z)$ where $z=w^T x_i$

$$V.\quad \frac{\partial}{\partial z}\sigma(z)=\frac{\partial}{\partial z}\frac{1}{1+e^{-z}}=\frac{\partial}{\partial z}(1+e^{-z})^{-1}$$

$$=\quad -(1+e^{-z})^{-2}\frac{\partial}{\partial z}(1+e^{-z})=\frac{-e^{-z}}{(1+e^{-z})^2}$$

$$=\quad \sigma(z)[1-\sigma(z)]$$

Now taking the derivative of the "inside" z and combining with the results of equation V.

$$\text{VI.} \quad \frac{\partial}{\partial w} z = \frac{\partial}{\partial w}(w^T x_i) = x_i$$

Combining equation V and VI yields the partial with respect to w of sigma

$$\text{VII.} \quad \frac{\partial}{\partial w}\sigma(w^T x_i) = \sigma(w^T x_i)[1-\sigma(w^T x_i)]x_i$$

Substituting Equation VII into Equation IV yields

$$\text{VIII.} \quad [\frac{-(1-y)}{(1-\sigma(w^T x_i))} - \frac{y_i}{\sigma(w^T x_i)}]\sigma(w^T x_i)[1-\sigma(w^T x_i)]x_i$$

performing some algebra results

$$\text{IX.} \quad [\sigma(w^T x_i) - y_i]x_j$$

Q1.2a

$$LL = y\log(\sigma(w^T x_i)) + (1-y)\log(1-\sigma(w^T x_i))$$

or

When y = 0 $\log(1-w^T x_i)$
and
When y = 1 $\log(w^T x_i)$

Q1.2b

$$w_t = w_{(t-1)} - [\eta * (\sigma(w^T x_i) - y_i)x_t]$$

Q1.2c

If x_t is sparse, and the update rule is dependent on the value of x_t as seen in 1.2b. That means all values of w will not be updated hence not all values of the array will be accessed and complexity should be

O(log(n))

Q1.2d

When learning rate η is too big, gradient descent will take large steps towards convergence. However, one risks the possibility of overshooting a local or global minimum with large updates (large learning rates). This could potentially result in a model never converging.

When learning rate is too small, it could take too many iterations to drive convergence and reach a global/local minimum. This is because each update step is assumed to be very minimal (small) in the direction of the minimum. This too could result in a model never converging or could greatly increase training time due to having to increase the amount of iterations performed during the training process.

Q1.2e

L2 norm is defined as

$$(N)LL+\mu\|w\|$$

to account for the addition of regularization, the derivative must be taken with respect to w and added to equation IX in 1.1a

$$\frac{\partial}{\partial w}\mu\|w\|_2^2=\frac{\partial}{\partial w}\mu w^2=2\mu w$$

So the update with regularization now becomes

$$w_t=w_{(t-1)}-\eta*((\sigma(w^T x_i)-y_i)x_t+2\mu w_t)$$

if x_t is not sparse, then each value in the array must be accessed and would be

O(n)

But if x_t is sparse, each value is not accessed in the array, but each corresponding weight and x_t is accessed making complexity
O(nlog(n))

Q2.1 Descriptive Statistics

Metric	Deceased patients	Alive patients
Event Count		
1. Average Event Count	1027.7385229540919	683.1552587646077
2. Max Event Count	16829	12627
3. Min Event Count	2	1
Encounter Count		
1. Average Encounter Count	24.839321357285428	18.695492487479132
2. Median Encounter Count	14.0	9.0
3. Max Encounter Count	375	391
4. Min Encounter Count	1	1
Record Length		
1. Average Record Length	157.04191616766468	194.70283806343906
2. Median Record Length	25.0	16.0
3. Max Record Length	5364	3103
4. Min Record Length	0	0
Common Diagnosis	DIAG320128,416 DIAG319835,413 DIAG313217,377 DIAG197320,346 DIAG132797,297	DIAG320128,1018 DIAG319835,721 DIAG317576,719 DIAG42872402,674 DIAG313217,641
Common Laboratory Test	LAB3009542,32765 LAB3023103,28395 LAB3000963,28308 LAB3018572,27383 LAB3016723,27060	LAB3009542,66937 LAB3000963,57751 LAB3023103,57022 LAB3018572,54721 LAB3007461,53560
Common Medication	DRUG19095164,6396 DRUG43012825,5451 DRUG19049105,4326 DRUG956874,3962 DRUG19122121,3910	DRUG19095164,12468 DRUG43012825,10389 DRUG19049105,9351 DRUG19122121,7586 DRUG956874,7301

2.3 SGD Logistic Regression

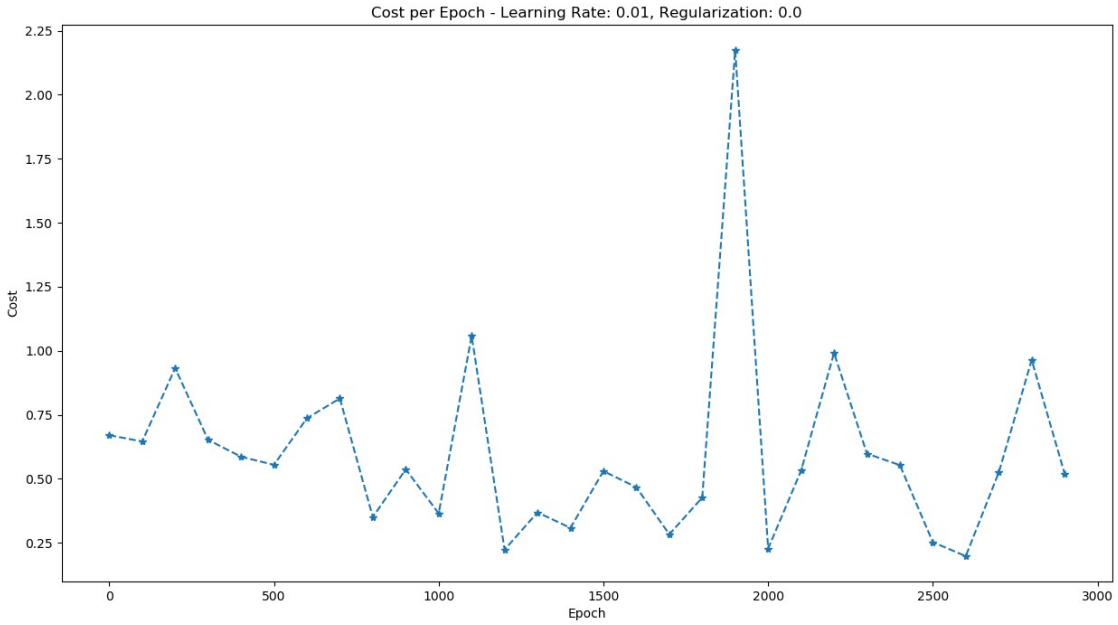
For this part, two major differences were employed: bias term in the activation function, and no bias term.

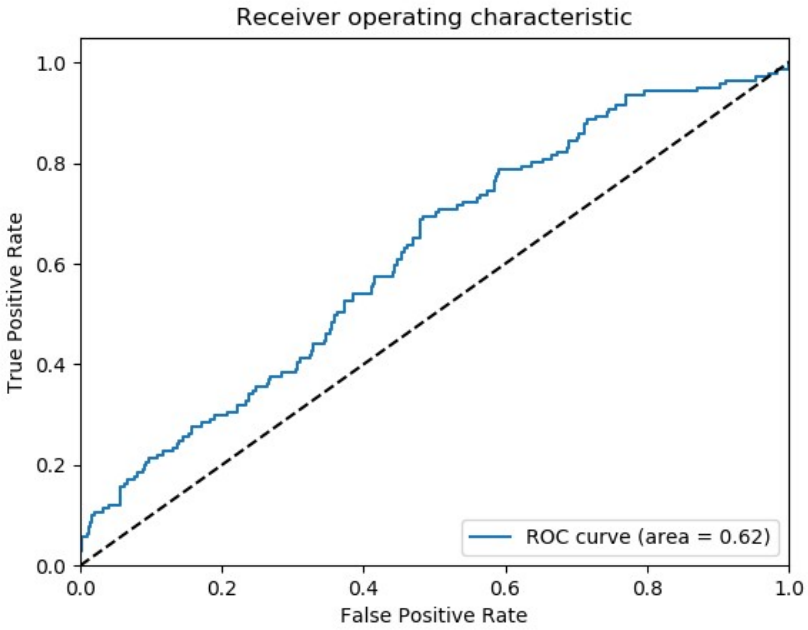
I. Activation function (sigmoid) = $\frac{1}{1+e^{-z}}$

II. Where $z = W^T X$ no bias

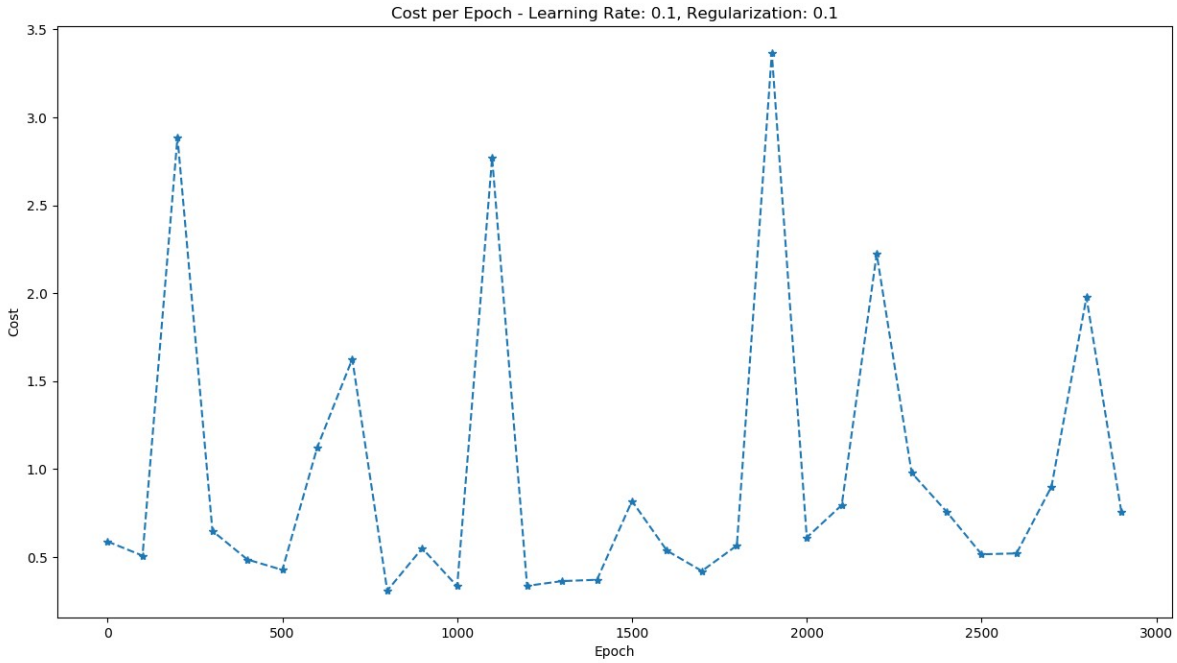
III. $z = W^T X + b$ with bias

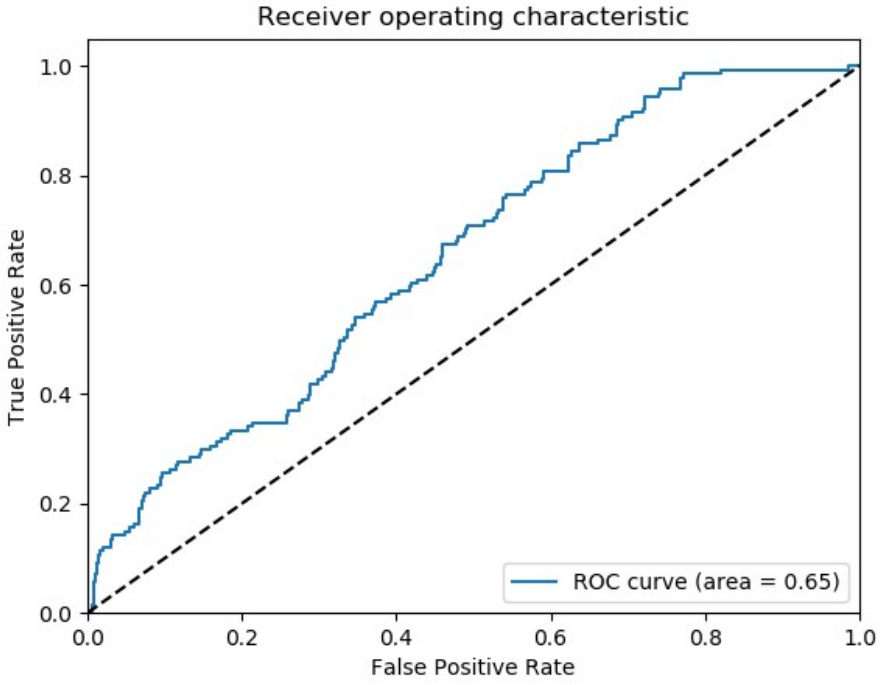
Starting with NO bias term, a model was trained with the default hyperparameters of 0.01 learning rate and no regularization with the following results.



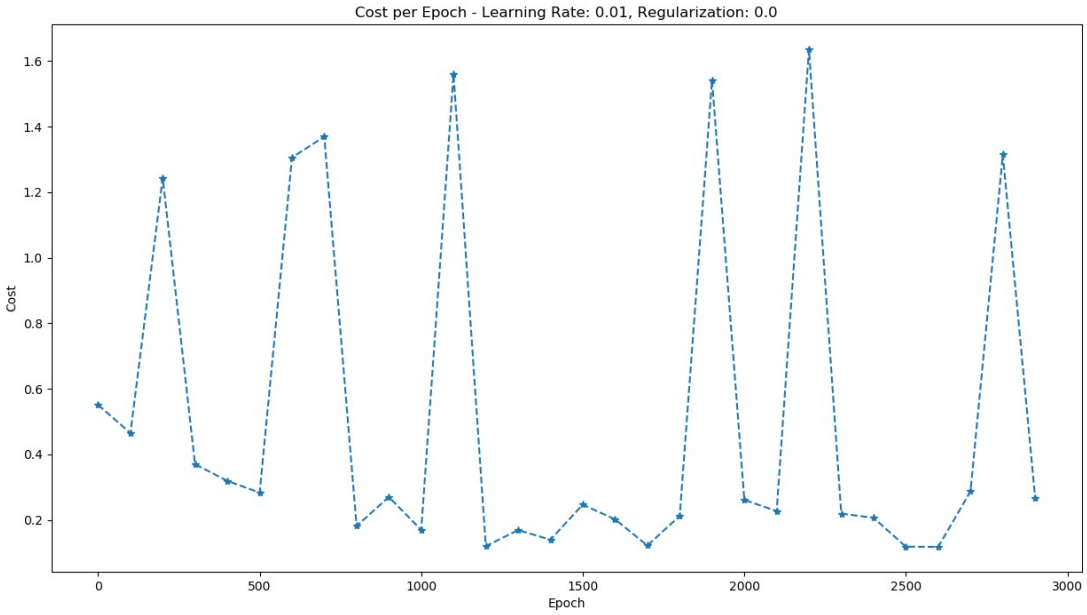


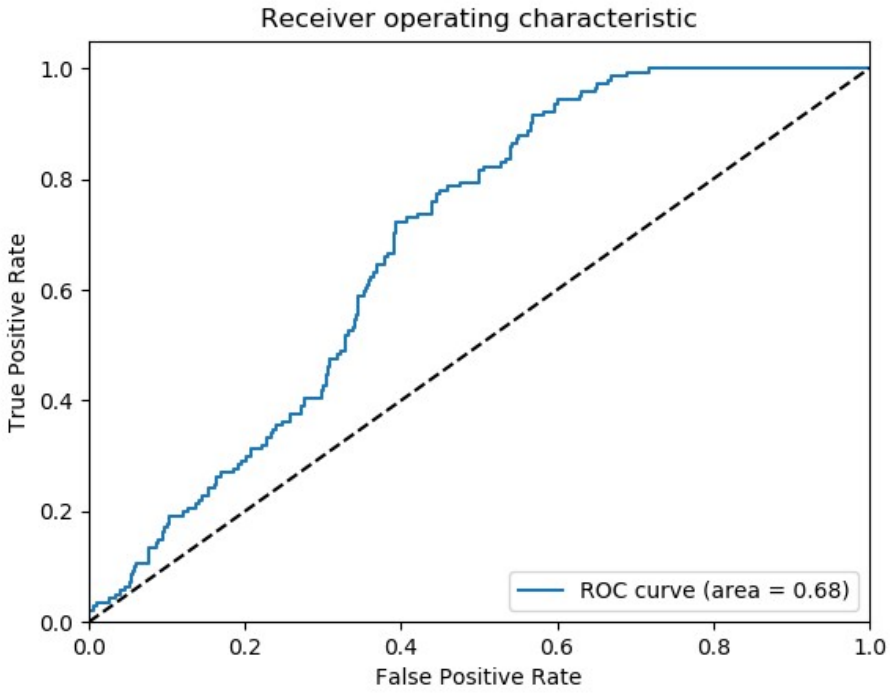
The cost had a downward trend with each iteration. It had some spikes here and there, which is usually an indication that the learning rate is too high. But It could also be due to the randomness of that observation. A few other hyperparameter combinations were test to gauge performance and metrics but it was pretty clear that the models were doing just better than “slightly guessing”. Best achieved model is seen below with a learning rate of 0.1 and regularization parameter of 0.1





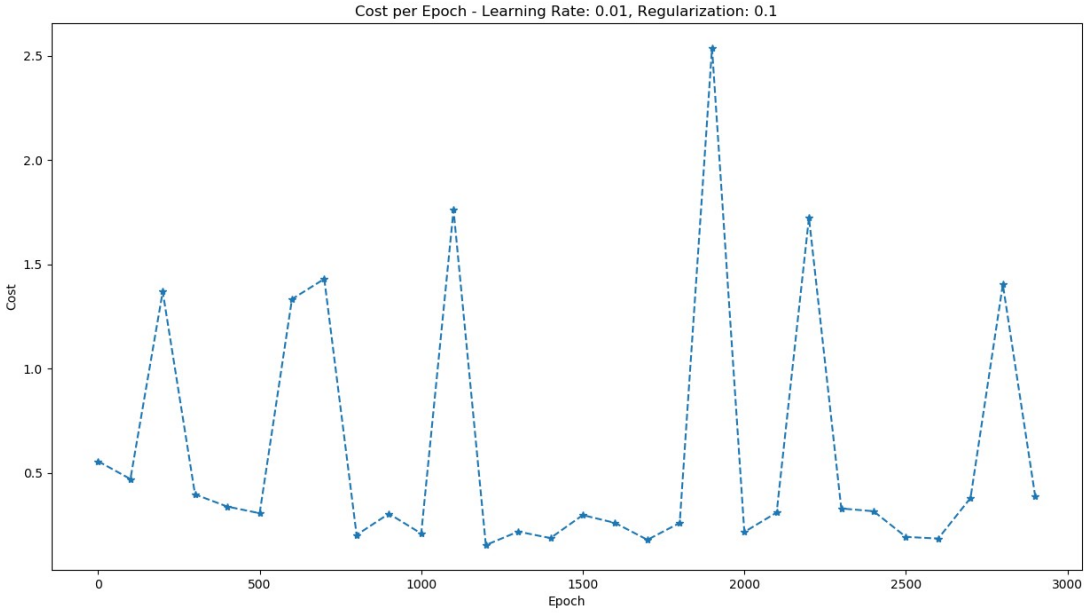
Next, the bias term was added to further tune the model. With default parameters of 0.01 learning rate and no regularization, model performance was the following

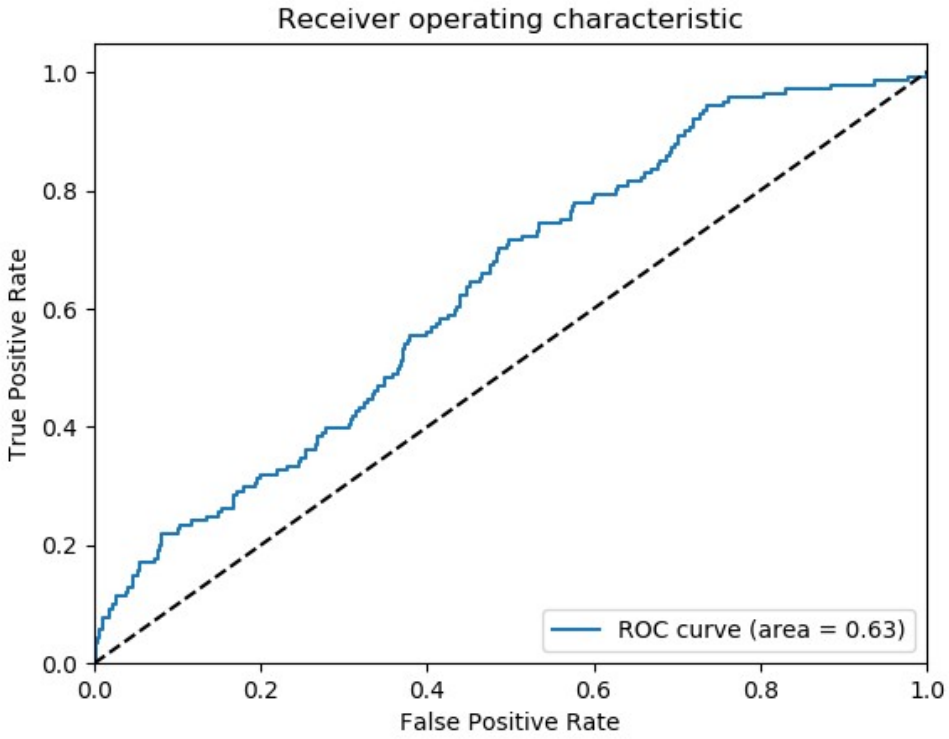




Again, the cost per epoch shows an overall downward trend with some spikes here and there. The AUC was also registering better than any model trained without the bias term. This bias model too, was

tuned for better performance, but ultimately any change to the regularization parameter only reduced AUC





The fact that performance only continued to decrease with increasing regularization, this hints that the models suffer from a high bias problem (under fitting). More data is needed to drive convergence further.

2.4 Hadoop

While training multiple models using Hadoop, the previous issue of high bias only continued. No matter how many models trained, or what hyperparameters were tuned, the AUC didn't significantly improve. As a matter of fact, the AUC was worse than the single trained model with bias that was trained on all the data. Since more data was needed to combat the high bias model, training multiple models on fewer data points only amplified the high bias problem. First, 5 models were trained with bias term and default hyperparameters. As hyperparameters were tuned, model performance decreased.

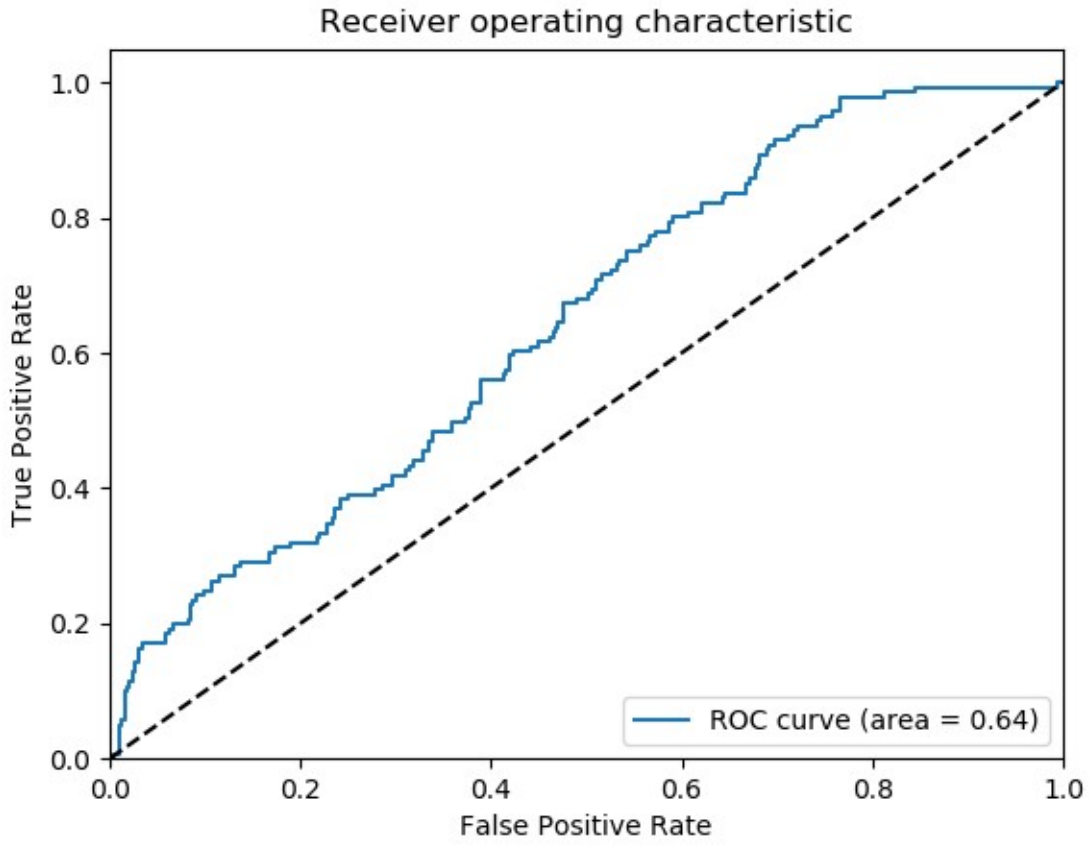


Illustration 1: 5 model ensemble

Below is the results of the best 5 model ensemble.

Next, due to perceived high bias, the ensemble was reduced so fewer models were trained on more data. The performance increased slightly, but still under performed the single model trained on all data. Below is the best trained reduced model ensemble. 3 models were trained:

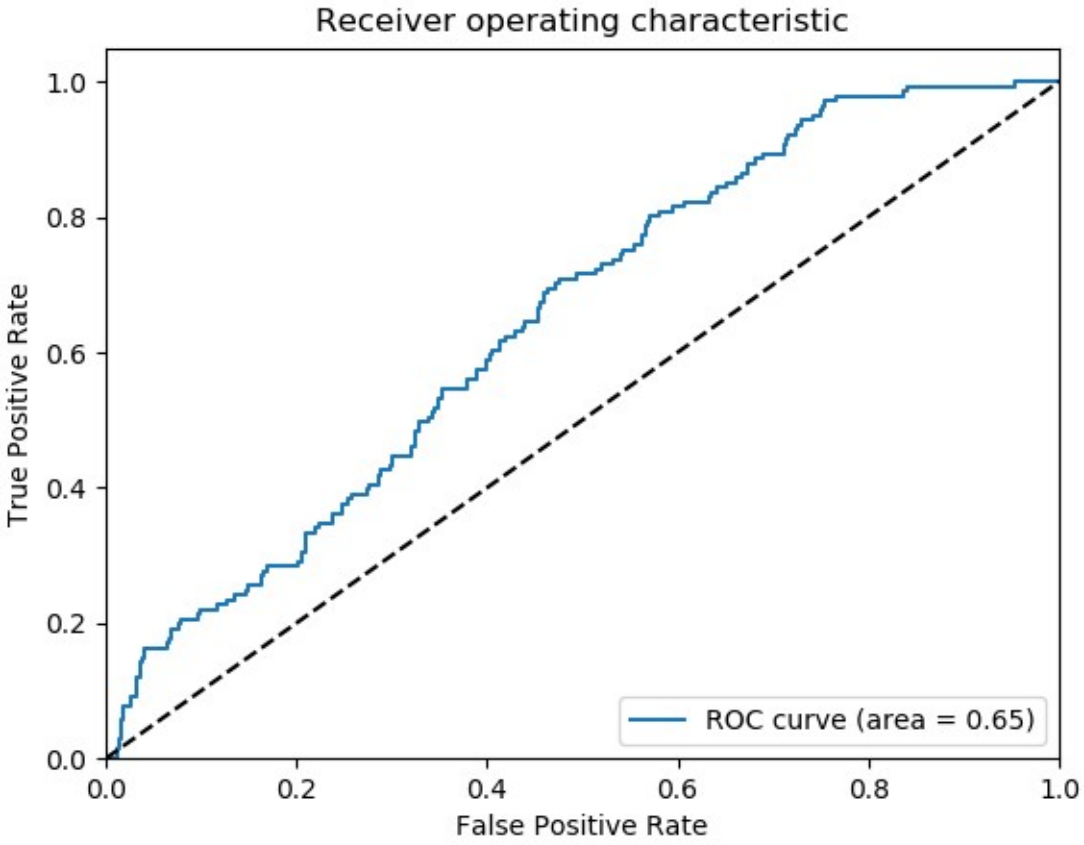


Illustration 2: 3 Model Ensemble