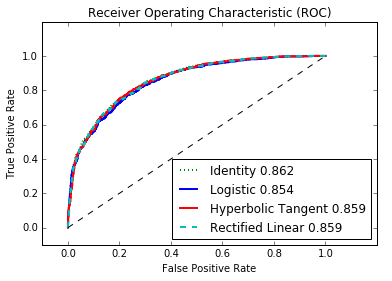
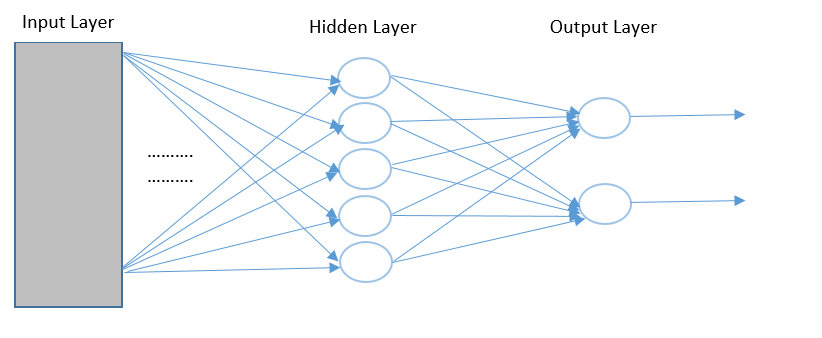
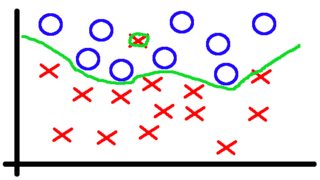
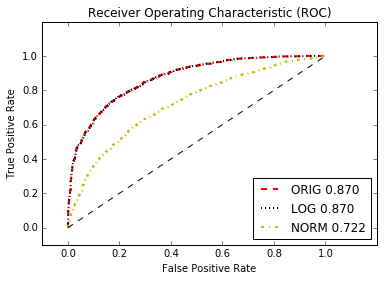
**Methods – Neural Network** Although non-linear SVM did not perform as well as expected but we still wanted to try another non-linear classifier for this problem. We decided to go with neural network which might be a good choice because of its high capacity.

We use Python Scikit-Learn Library’s Multi-Layer Perceptron algorithm which is designed for classification problems. We are using a single hidden layer neural network where the hidden layer has 5 perceptrons. The output layer has 2 perceptrons. The network structure is shown in figure # with an “Identity” activation function in the output layer. The activation of the hidden layer is customizable and we will try out different activation for the hidden layer and compare their performances.



**Experiments – Neural Network** In this model we tried four different activations for the hidden layer – Identity, Sigmoid, Rectified Linear and Hyperbolic Tangent. Our experiment shows that the model performs best with Identity activation which effectively makes the Neural Network a linear classifier and we are getting an accuracy of 86.2%. The non-linear activations are also performing quite well with an accuracy of around 85.6%. The comparison is shown in the ROC curve of figure #. We also tried with different transforms and among them the Logarithmic transform performed the best.

**Methods - Gradient Boosting** Experiments with non-linear SVM and Neural Networks with non-linear activations indicated that non-linear models were getting affected by overfitting. Hence, we used Scikit-Learn’s Gradient Boosting which is an ensemble model that combines many weak learners to form a strong learning hypothesis and is known for its robustness to overfitting. Boosting algorithms treat the learning problem as an optimization problem where the objective function is the error function and the goal is to minimize it. Although there hasn’t been any theoretical explanation so as to why Gradient Boosting does not overfit, empirical results show that due to extremely high level of non-linearity the decision boundaries around the outlying points become so narrow that no other point can fit into that. Hence it does not lead the model to incorrect predictions.

**Experiments – Gradient Boosting** We used Gradient Boosting with the original data, the logarithmic transform and the normalized transform and we got the best results of 87% accuracy using the former two transforms. We used a model with 100 layers of decision tree which produced an extremely non-linear boundary and yet did not overfit owing to the property of Gradient Boosing. The comparison of the results can be seen from the ROC curve of figure #.