# Predicting Influencers in a Social Network

**Group 11**

Akshay Mehra

Sayam Ganguly

Patrick Green

# Objective

## Problem Definition

Predict the human judgement about which of two users is more influential in a social network, given twitter activity data for the pair. This is a binary classification problem.

## Data Set

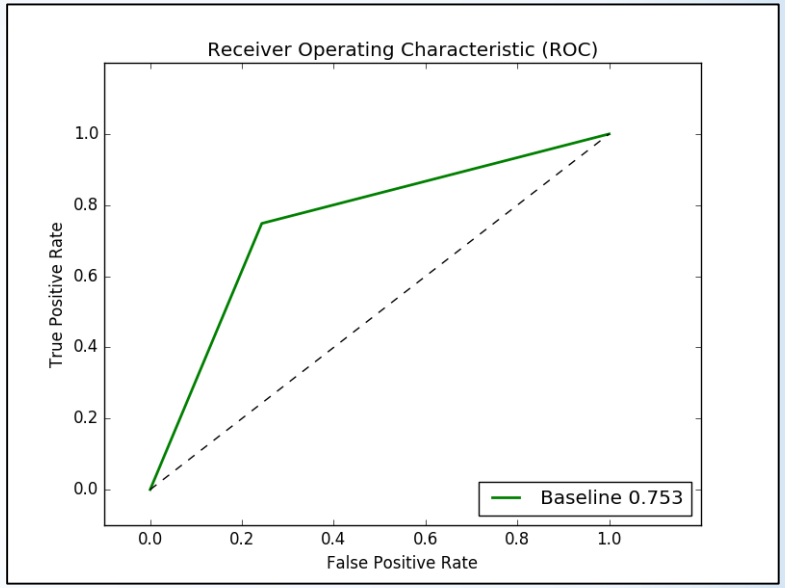
Numeric features for pairs of users (A and B). Both training and test data sets provided.

- Follower/Following Count
- Mentions Sent/Received
- Retweets Sent/Received
- Post Count
- Listed Count (count of inclusion in user-created categories or lists)
- Network Features (proprietary calculated features)

# Approach

## Pre-processing

- Feature Selection – determine which features correlate with influence
- Feature Transformations – normalize data
- Dimensionality Reduction (PCA)




## Modeling Methods

- Logistic Regression
- Neural Network
- Support Vector Machines (SVM)
- Ensemble

## Evaluation

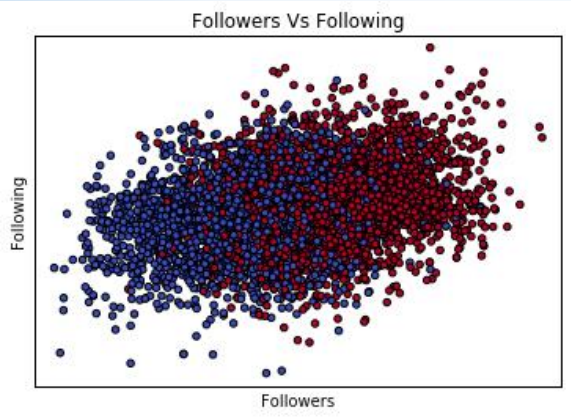
Measuring the area under the Receiver Operating Characteristic (ROC) curve, which plots true positives against false positives for probabilistic outcomes for a range of thresholds.

## Baseline Solution

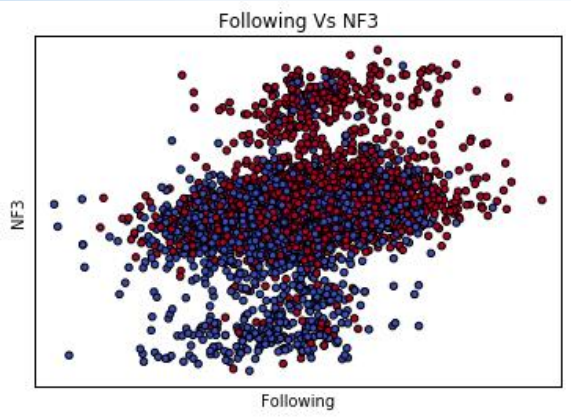
The person who has more followers is the person who is more influential. (Area Under Curve : 0.753)

# Data Visualization

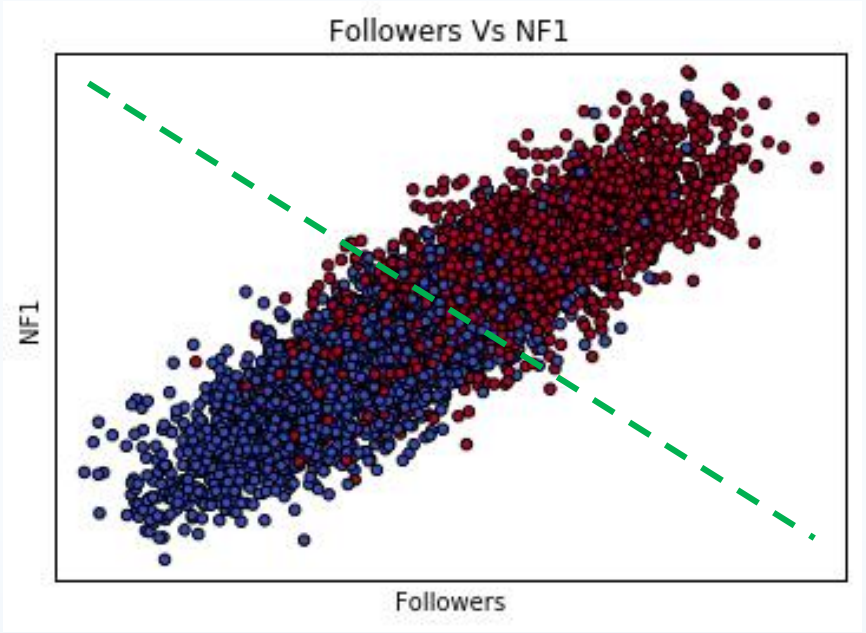
Plotting pairs of features suggests the relative importance of the features by separability of the data. Red points indicate that user A is more influential; Blue that user B is more influential.



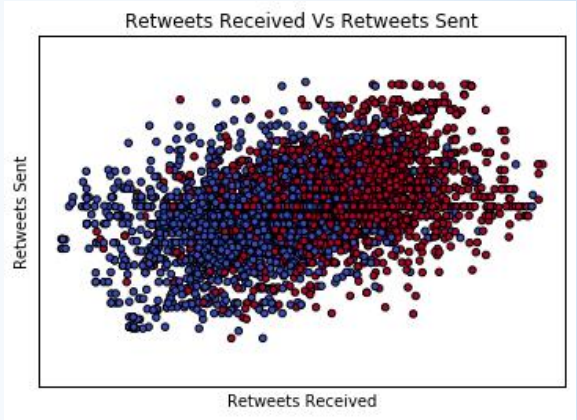

**Follower count correlates with influence, but not Following count.**



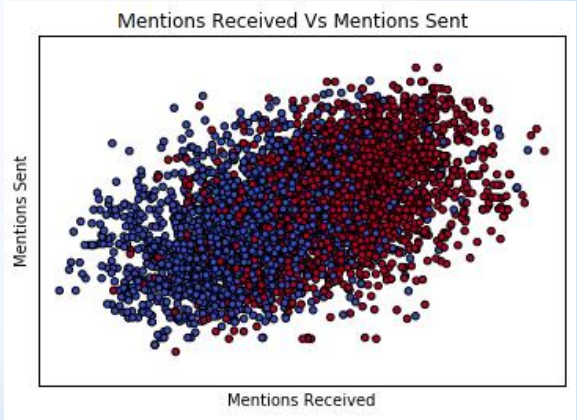

**Network Feature 3 drives influence more than Following.**




**Both Follower count and Network Feature 1 are important to influence, suggested by a diagonal separation.**




**Retweets Received is more important than Retweets sent.**



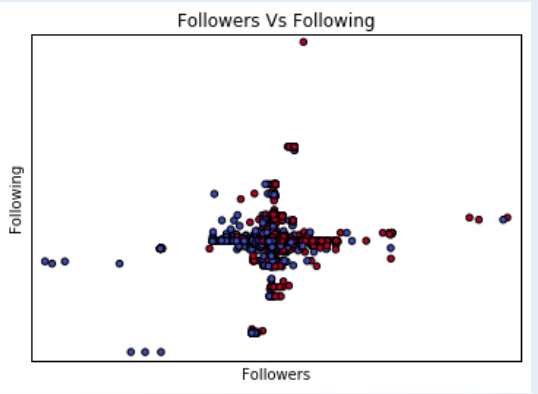

**Mentions Received is more than important than Mentions Sent.**

All plots use the difference of logs transform: log(A.feature) – log(B.feature)

# Feature Transformation

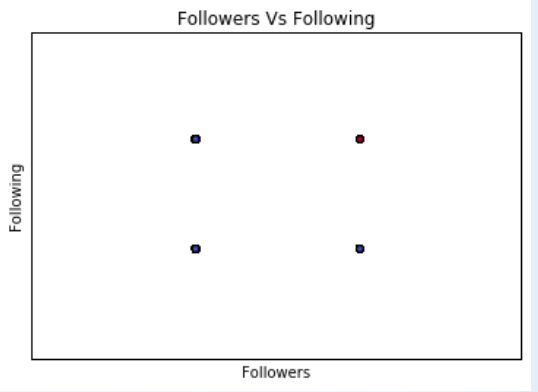
## Delta Transform

- Baseline - absolute difference between values for two users.
- Preserves the meaning of large difference in values.
- To reduce the scale of the data, we normalize it.



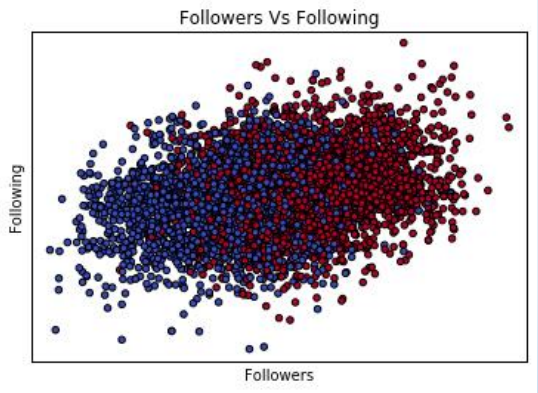

## Discrete/Binary Transform

- Reduce differences to one (larger for user A) or zero (larger for user B).
- Simpler data set, but disregards the magnitude of differences.




## Logarithm

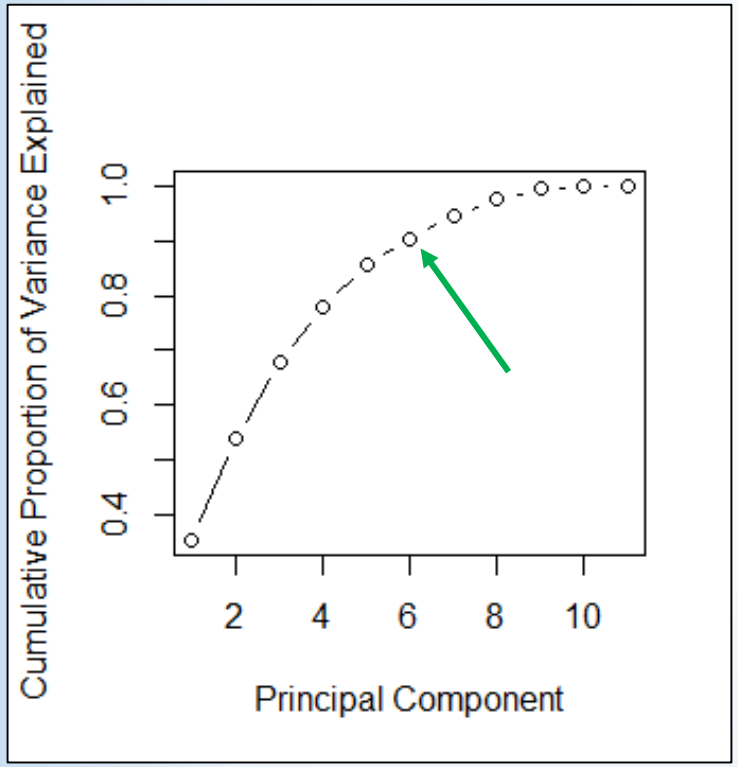
- Reduces large differences in the orders of magnitude in features (ie. follower versus following counts).
- Simplifies non-linear relationships between input and output.



# Dimensionality Reduction

## Principal Component Analysis

Reduce dimensionality by creating orthogonal components from linear combinations of initial features.




**Can reduce dimensionality to six components and capture ~90% of the cumulative variance.**

| Principal Component 1 | |
|---|---|
| A.follower_count – B.follower_count | -0.35497 |
| A.following_count – B.following_count | -0.06158 |
| A.listed_count – B.listed_count | -0.43547 |
| A.mentions_received – B.mentions_received | -0.46721 |
| A.retweets_received – B.retweets_received | -0.46998 |
| A.mentions_sent – B.mentions_sent | -0.05578 |
| A.retweets_sent – B.retweets_sent | -0.07444 |
| A.posts – B.posts | -0.06964 |
| A.network_feature_1 – B.network_feature_1 | -0.47135 |
| A.network_feature_2 – B.network_feature_2 | 0.063936 |
| A.network_feature_3 – B.network_feature_3 | 0.040503 |

**To the left is the composition of the first principal component.**

**Below is a breakdown of how each component is affecting the variance.**

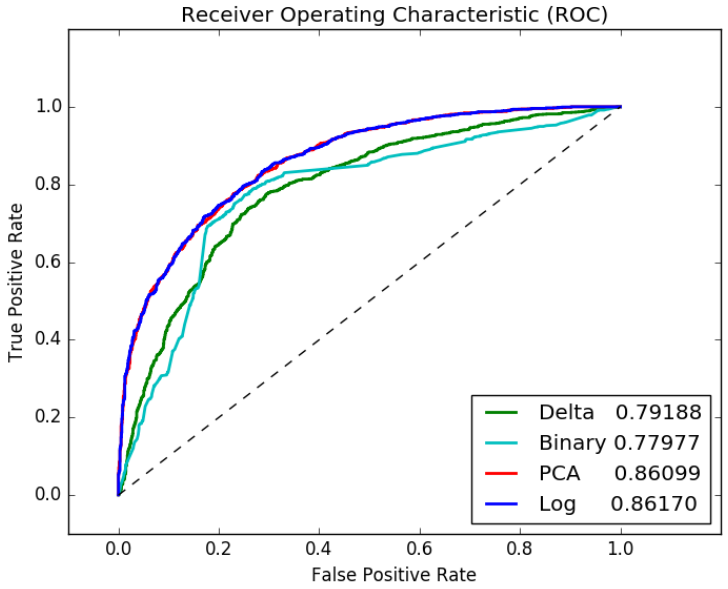
| Principal Components | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | **PC1** | **PC2** | **PC3** | **PC4** | **PC5** | **PC6** | **PC7** | **PC8** | **PC9** | **PC10** | **PC11** |
| Standard Deviation | 1.9736 | 1.4221 | 1.249 | 1.04245 | 0.9197 | 0.73123 | 0.67873 | 0.57052 | 0.45938 | 0.21665 | 0.10378 |
| Proportion of Variance | 0.3541 | 0.1839 | 0.1418 | 0.09879 | 0.0769 | 0.04861 | 0.04188 | 0.02959 | 0.01918 | 0.00427 | 0.00098 |
| Cumulative Proportion | 0.3541 | 0.538 | 0.6798 | 0.77859 | 0.8555 | 0.9041 | 0.94598 | 0.97557 | 0.99475 | 0.99902 | 1 |

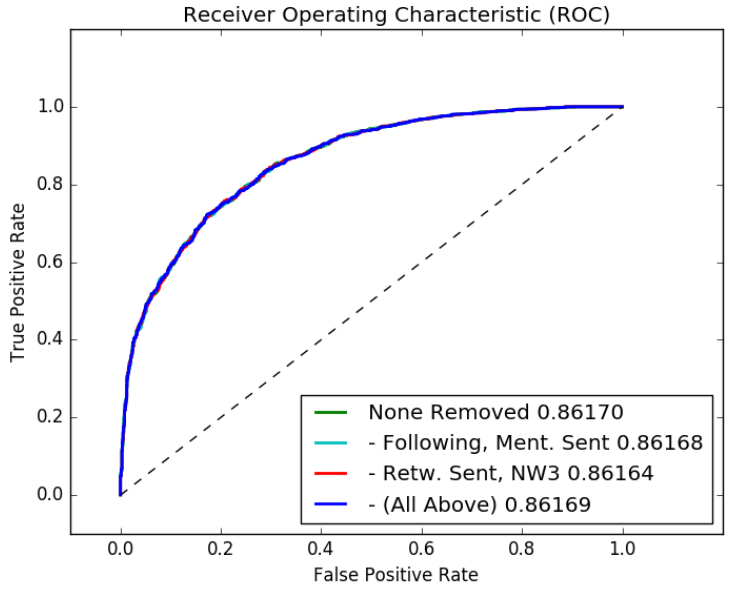
# Modeling Methods

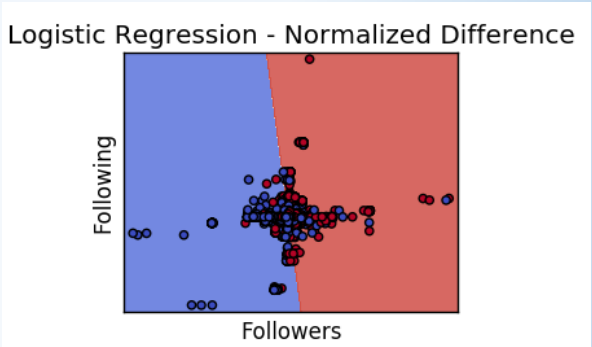
- Logistic Regression
- Support Vector Machines (SVM)
- Neural Network
- Gradient Boosting

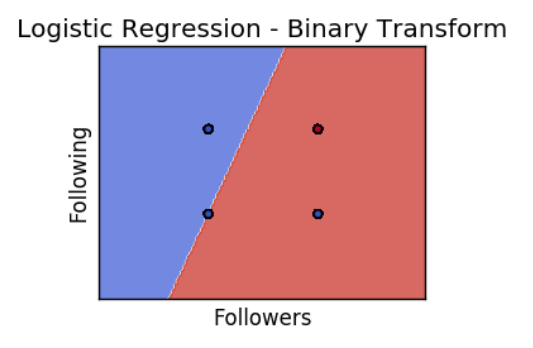
# Logistic Regression

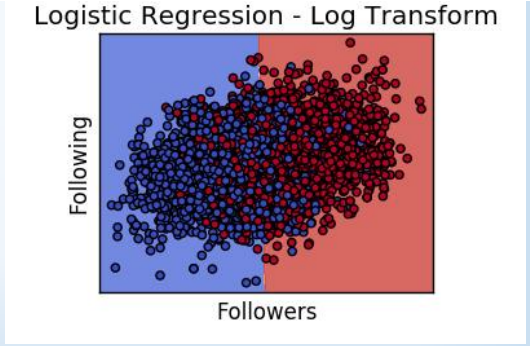
The log transform produces the best results, followed closely by the PCA transform.



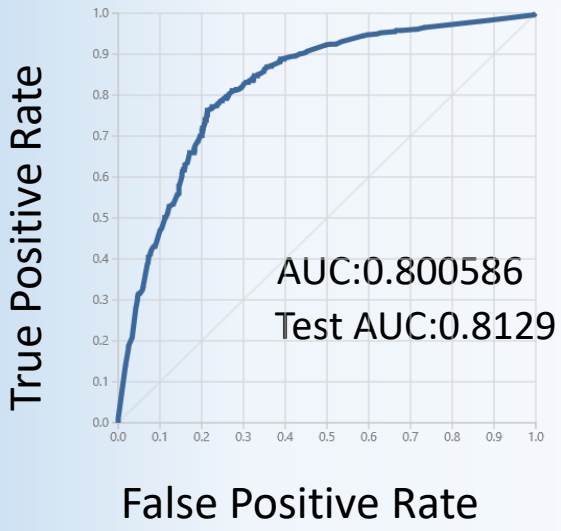



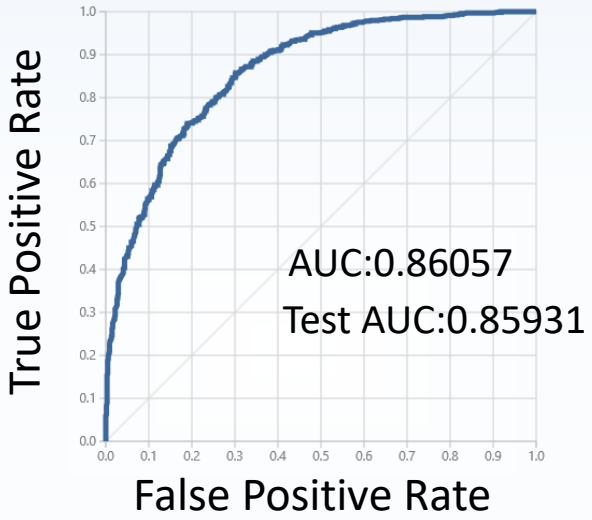



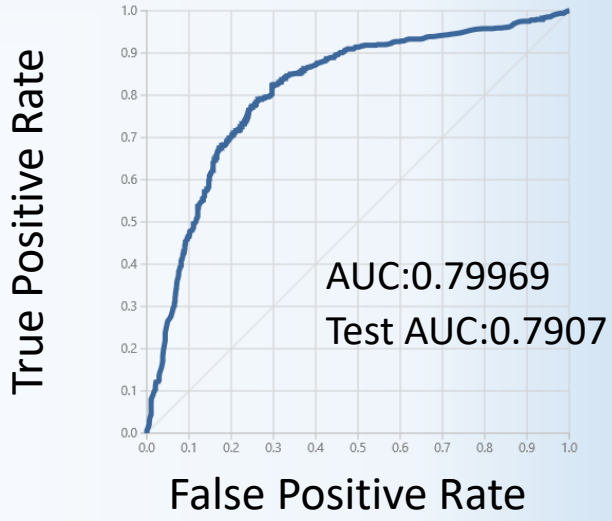

Removing the least important features negatively impacts results. Removing multiple such features works better, but still does not yield results which exceed the inclusion of all features.

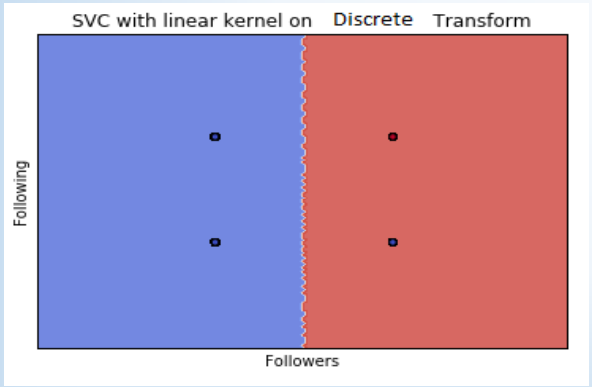
# Support Vector Machines (SVM)

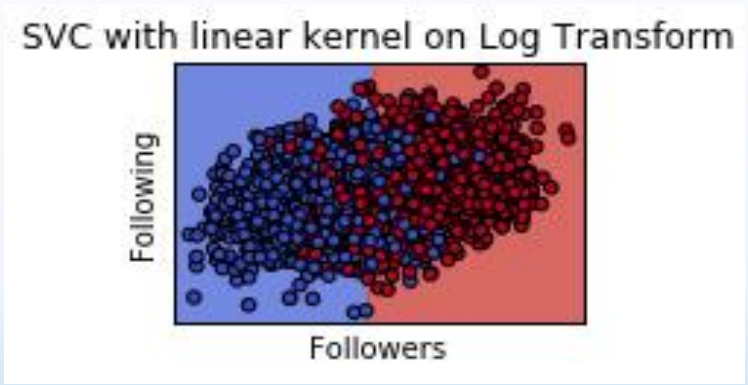
The log transform produces the best results.
The normalized and binary transforms perform similar to each other.

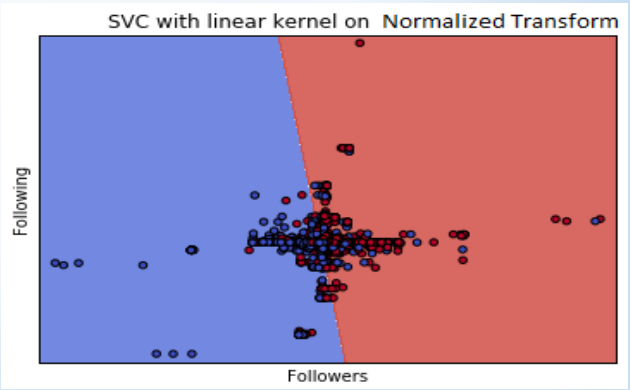


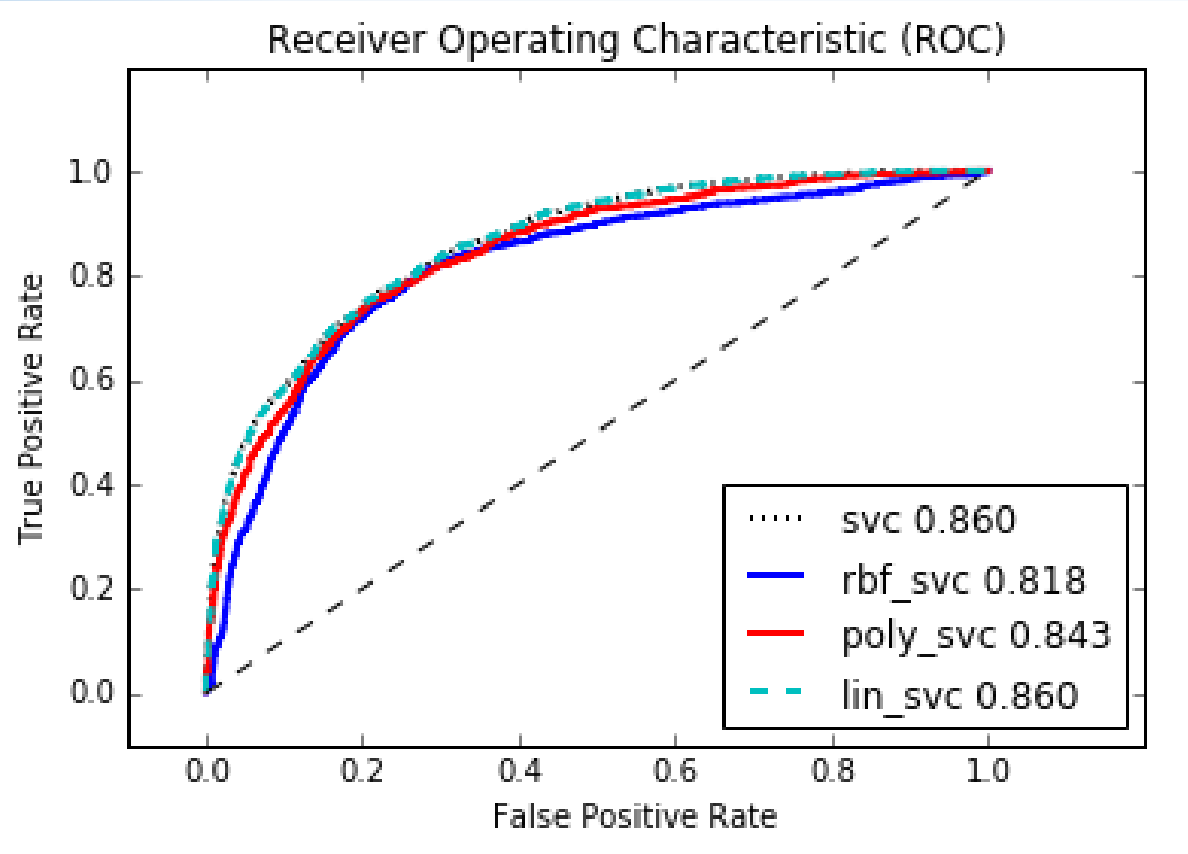



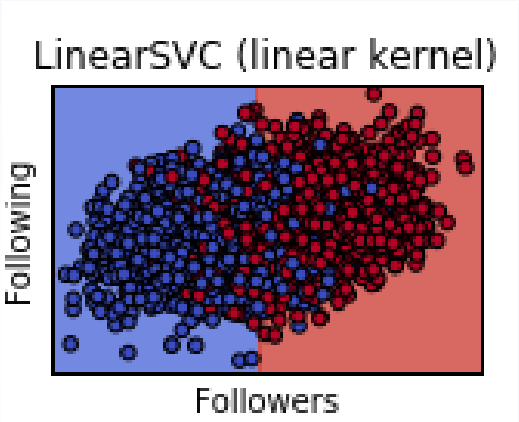



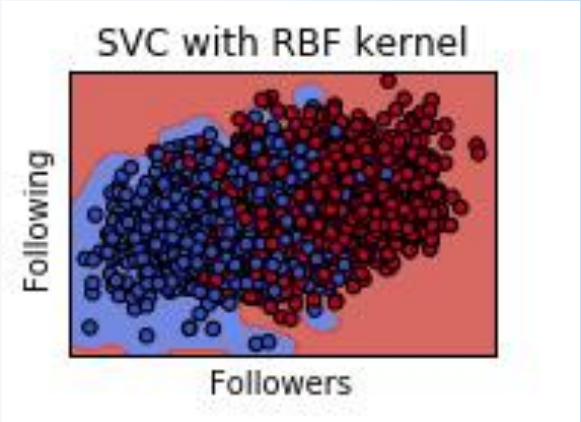



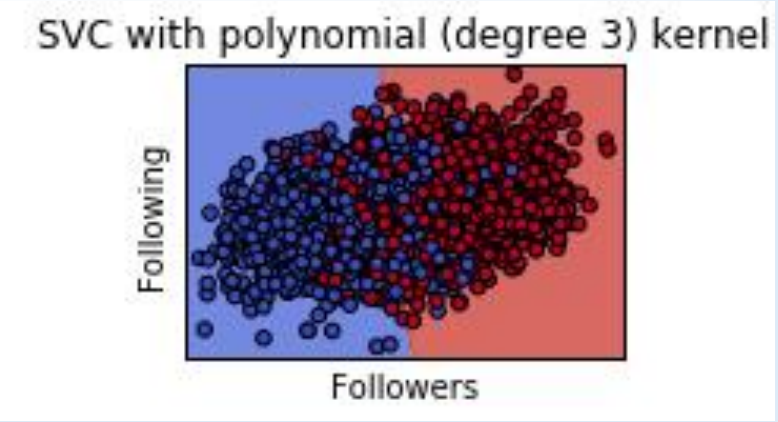
# Support Vector Machines (SVM)

The linear SVM model performs best.
The RBF Kernel and the polynomial kernel do not perform as good as the linear kernel.



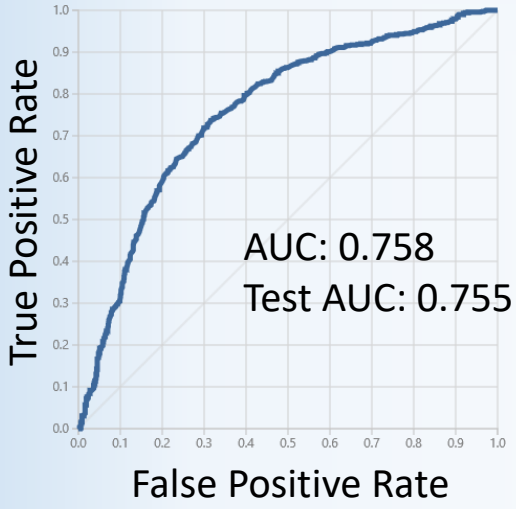



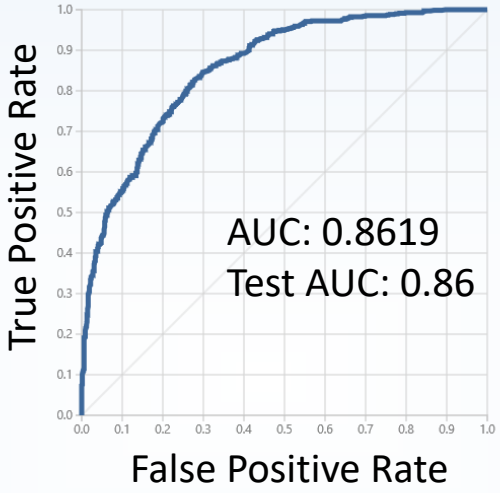



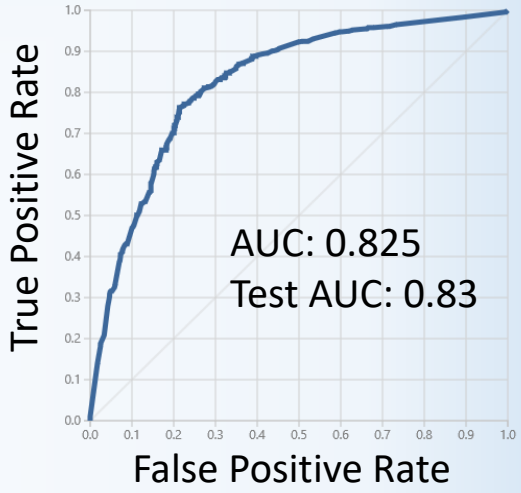

**All models are using the log feature transform.**

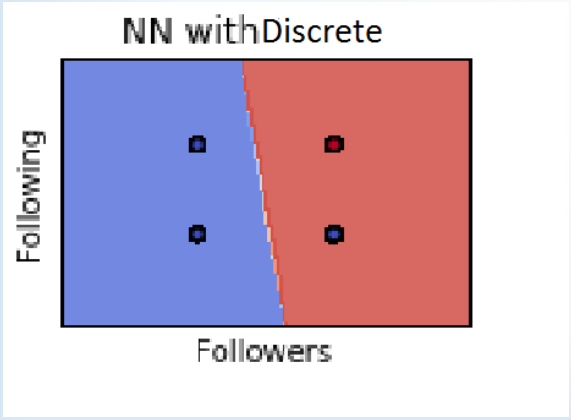
# Neural Network (NN)

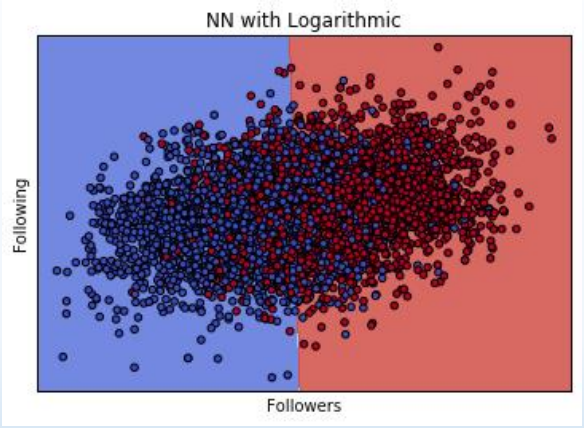
Again, the log transform produces the best results, with the Discrete transform performing poorly.

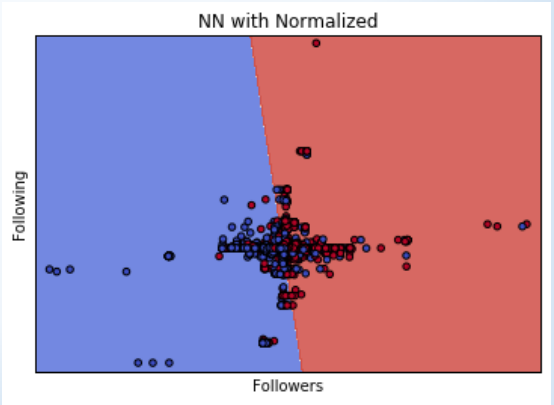


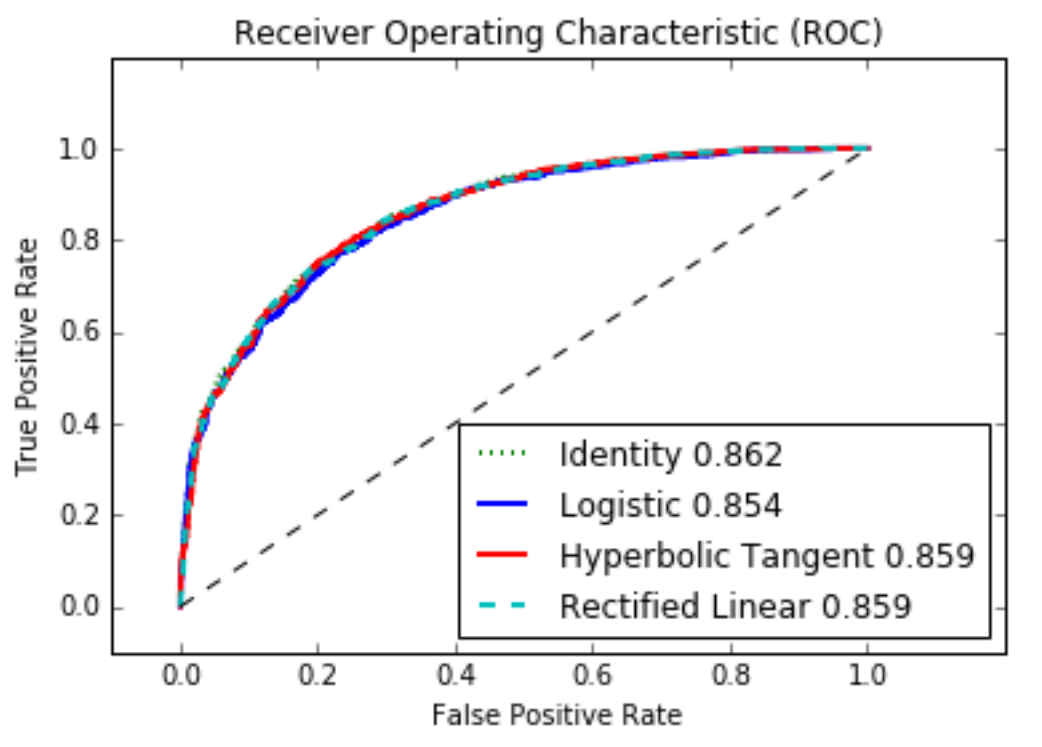



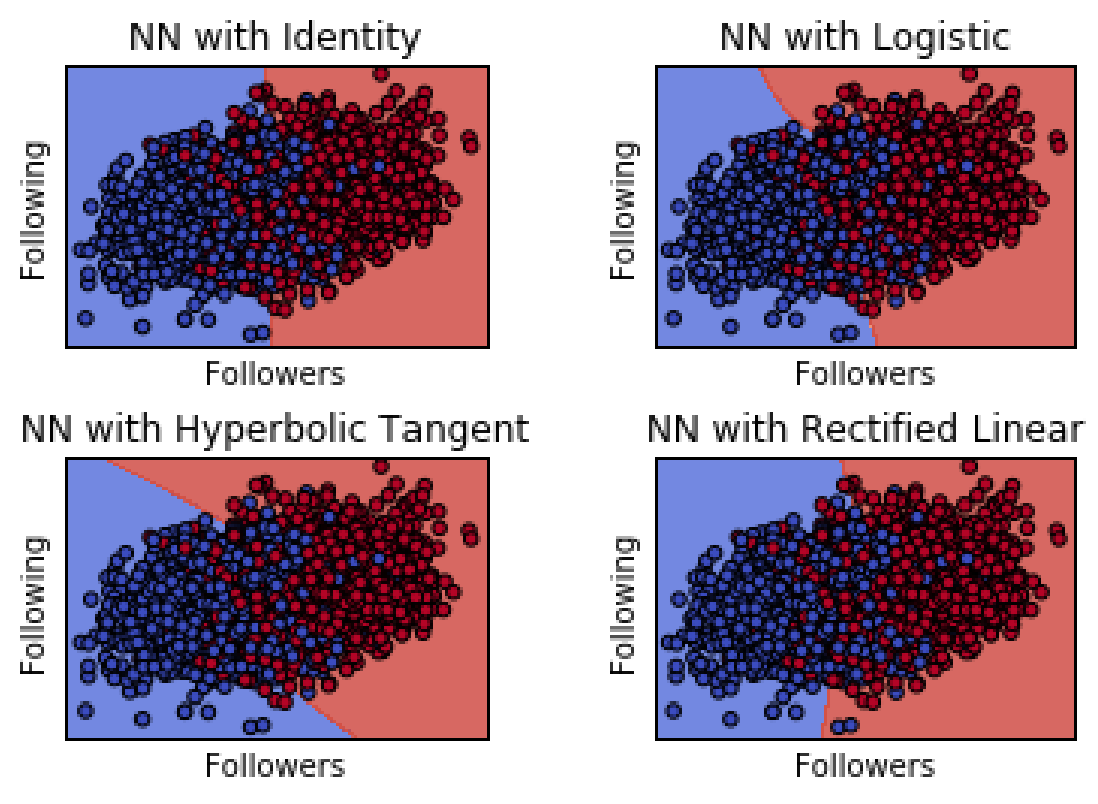



# Neural Network (NN)

Neural Network with Identity performs best.



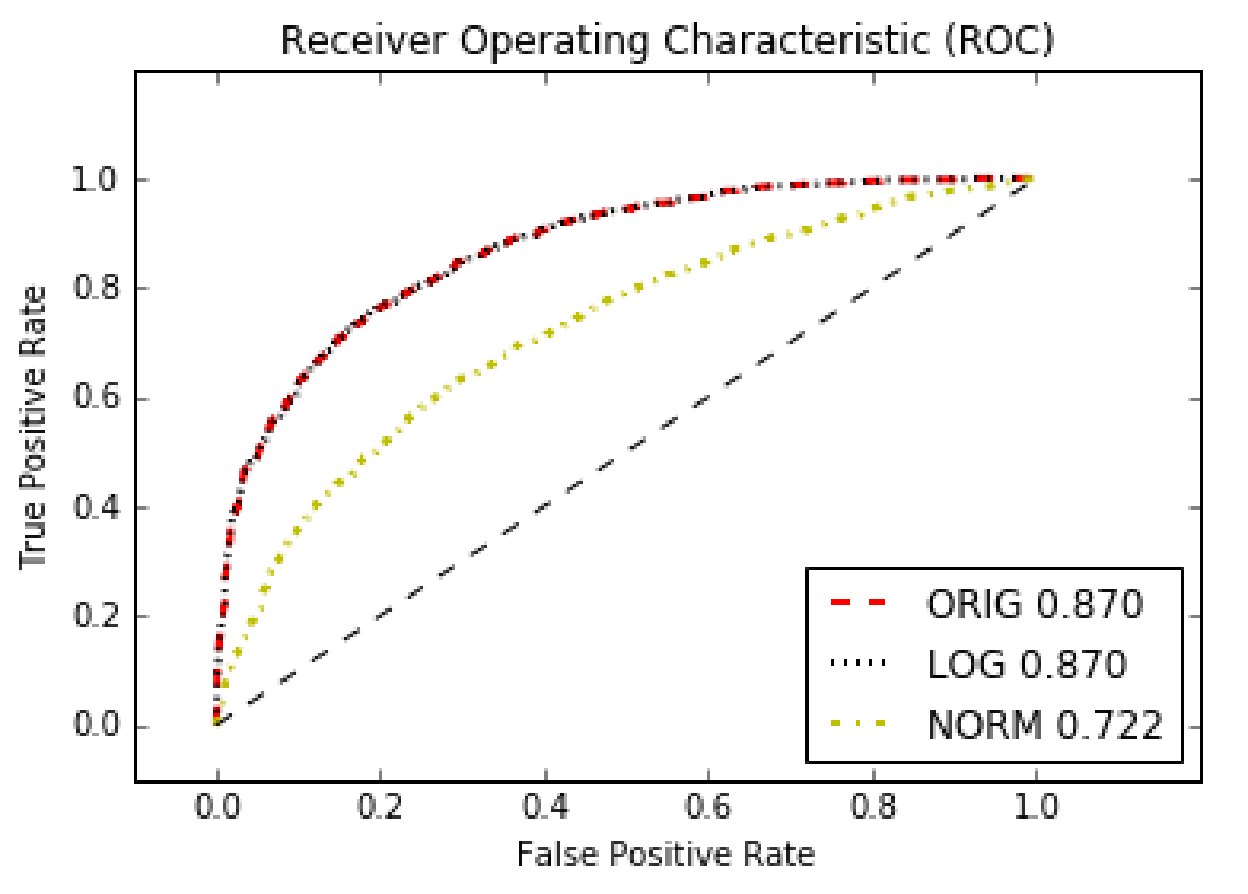



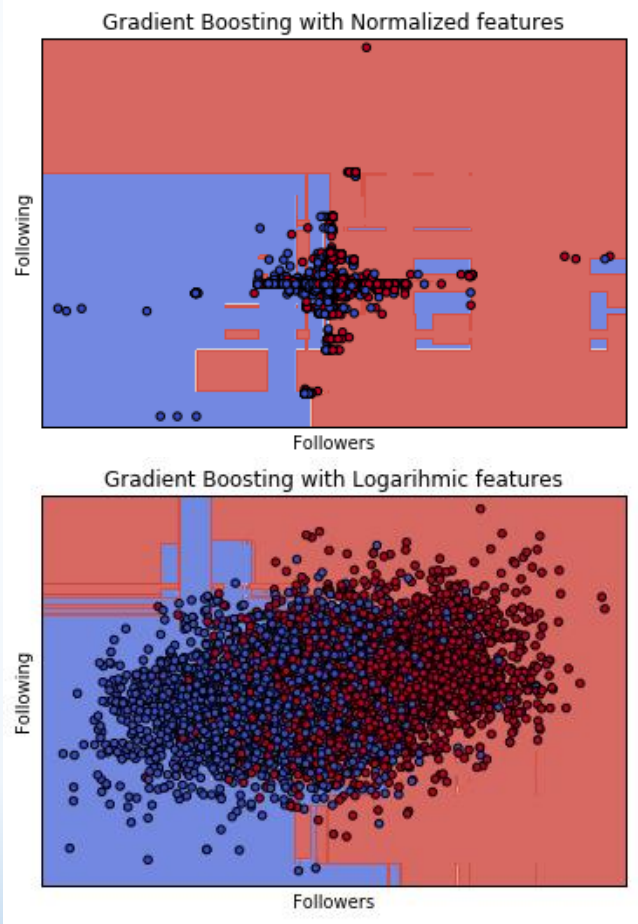

**All models are using the log feature transform.**

# Ensemble Model

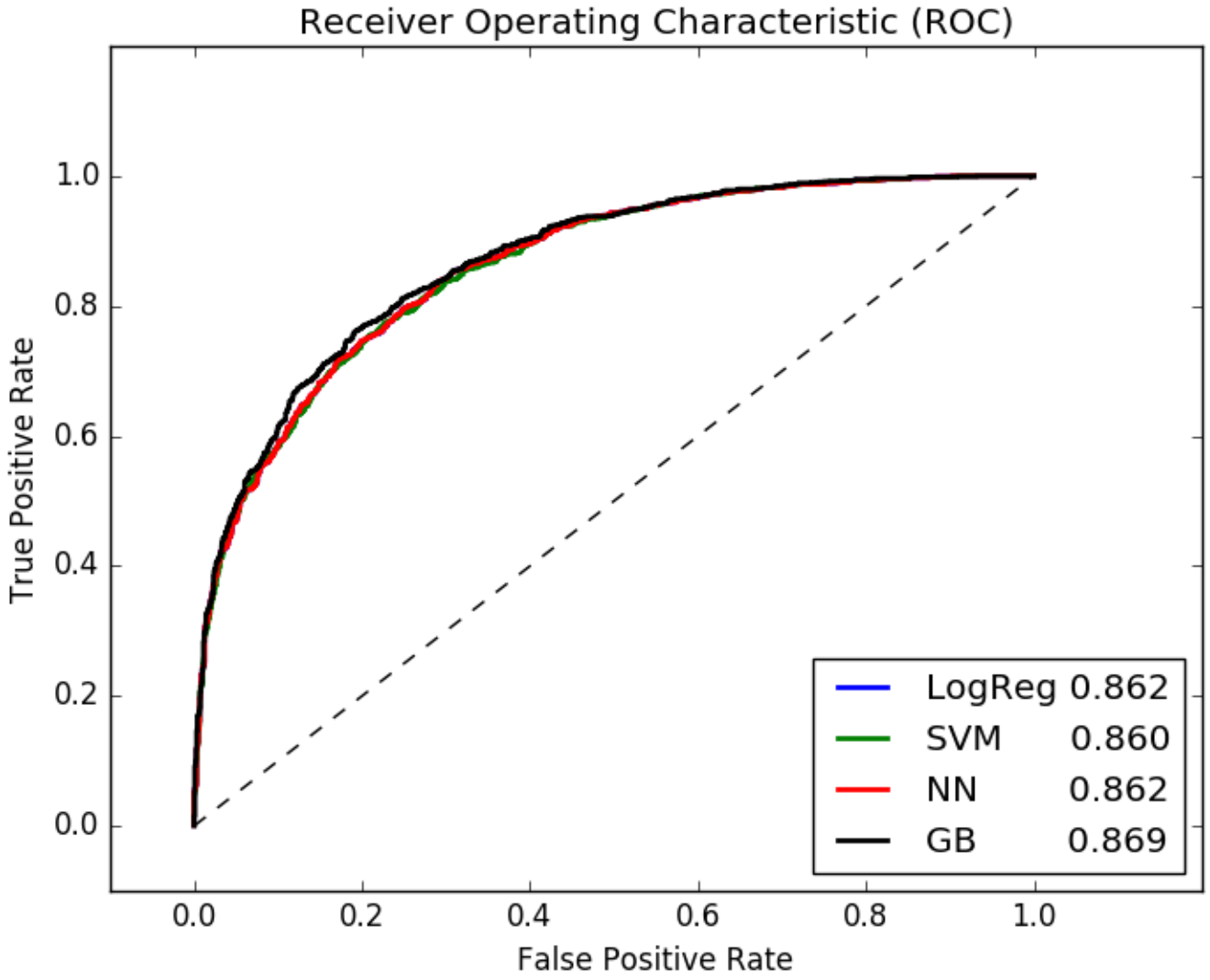
## Gradient Boosting

A weak decision tree learner is iteratively improved upon using gradient descent to minimize loss.



# Conclusion




## What "Worked"

- Ensemble Model with Gradient Boosting - best performing model
- Logarithm feature transform
- Linear-like separation

## What "Didn't Work"

- Discrete/Binary transform
- Feature Selection

# Thanks!!