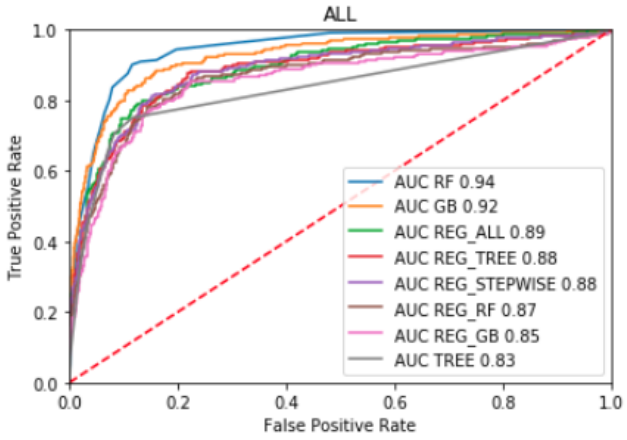
# MSDS 422 Machine Learning

## Assignment 3 – Regression

In addition to our models that we had created in the last assignment (decision tree, random forest, gradient boosting), we have created several logistic regression models with each using a different set of variables that the prior models found to be predictive. To give us an idea as to which model may work the best, we created an ROC curve of each model as well as determined its accuracy. This graphic displays our findings of all the models:




```
ALL CLASSIFICATION ACCURACY
======
RF  =  0.8976510067114094
GB  =  0.8934563758389261
TREE  =  0.8758389261744967
REG_ALL  =  0.875
REG_STEPWISE  =  0.87248322147651
REG_TREE  =  0.8716442953020134
REG_GB  =  0.8615771812080537
REG_RF  =  0.860738255033557
------
```

Based on our findings, the Random Forest model appears to be the most accurate at predicting loan defaults at nearly 90% accuracy. Also, the Random Forest model had the highest area under the ROC curve at .94. Regardless, if we had to pick a model to go to production with, it would be the Regression Model with Tree-based variables. Even though it did not have the best AUC or accuracy, regression models tend to be easy to implement. Additionally, the fact that it has a lesser number of variables (than say the Regression Model with ALL variables) makes it easy to work with.

Now let's look at the coefficients that our Regression Model with Tree-based variables has determined:

```
LOAN DEFAULT
---------
Total Variables:  12
INTERCEPT  =  -4.927671869118972
M_VALUE  =  3.5251186616485497
IMP_VALUE  =  1.4063317605824592e-06
M_YOJ  =  -0.3411002748684892
IMP_YOJ  =  -0.02199967727027834
M_DEROG  =  -0.8318943637727674
IMP_DEROG  =  0.4900442400535972
IMP_DELINQ  =  0.7370471619420972
IMP_CLAGE  =  -0.005452913311168733
IMP_CLNO  =  -0.019038629014586125
M_DEBTINC  =  2.8063075731534575
IMP_DEBTINC  =  0.09834485503094775
```

The variables with the most positive coefficients (M_VALUE and M_DEBTINC) seem to make sense as these variables signify missing info on home value and debt to income ratio. We would suggest looking further into how loans can have these values missing and if this signifies fraudulent activity on the part of the borrower and could lead to a more likely loan default. The variable with the most negative coefficient is M_DEROG. This may not make much sense in that if a borrower's derogatory credit marks were missing one may think they could be susceptible to loan default. Again, could be fraudulent activity here.

For predicting loss amounts, we have created similar models to the ones we used for predicting loan defaults, although instead of logistic regression we used linear regression. Below is our findings for our loss amount predicting models:

```
ALL LOSSES MODEL ACCURACY
======
GB  =  2244.4595716025806
RF  =  2765.465216534271
REG_ALL  =  3761.9751787278515
REG_TREE  =  4209.686119614146
REG_RF  =  4367.626957823208
REG_GB  =  4367.626957823208
REG_STEPWISE  =  4367.626957823208
ALL  =  5993.590136935047
------
```

With its Root Mean Square Error of 2244.46, we would elect to go with the Gradient Boosting method to predict loss amounts. Even though regression models can be easier to work with, the difference in RMSE between Gradient Boosting and the closest regression model (Regression with all variables) is fairly large.

Let's take a look at the coefficients for the Regression Model using Tree variables:

```
LOSSES
---------
Total Variables:  7
INTERCEPT  =  -12802.412181838548
LOAN  =  0.7978092339650397
IMP_VALUE  =  -0.011531816956570233
IMP_DELINQ  =  626.7130931151491
IMP_CLNO  =  224.43206922489586
M_DEBTINC  =  5694.317168306267
IMP_DEBTINC  =  126.51103415440615
```

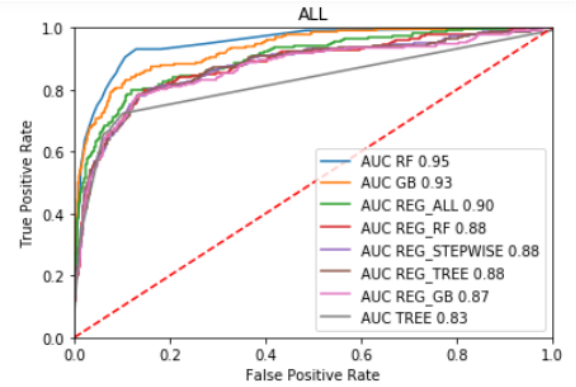
Some of the larger positive coefficients seem to make sense in predicting loss amounts. For example, missing debt to income ratio (M_DEBTINC) can possibly signify fraudulent activity and can lead to larger losses.

Additionally, having delinquencies (IMP_DELINQ) and higher numbers of credit lines (IMP_CLNO) can also lead to larger losses.

**Bingo Bonus** – Tried the following regarding regression parameters:

1. Tried solvers lbfgs, liblinear, sag, and saga for LogisticRegression for all variables . Noticed LARGE drop in AUC and accuracy compared to 'newton-cg' for all of them.
2. Tried max_iter = 100. Noticed a very small drop in AUC and accuracy compared to 1000. Did not observe any increase in AUC or accuracy when going over 1000.
3. Set fit_intercept = 'False' for LinearRegression for all variables. Noticed no impact.
4. Set normalize = 'True', noticed very slight increase in RMSE for the training dataset and very slight decrease in RMSE test dataset.

**Bingo Bonus** – Also tried several different random_state values for where the train/test data were being split. Noticed some minor shifting of models in the overall rankings due to small increases/decreases in AUC and accuracy. I think with many of the models packed tightly around the .87 accuracy range and .88/.87 AUC range we'll see some models shifting order in those ranges. Overall I would think we have a solid model. Below is random_state = 4 results (as opposed to the data earlier in the document which is random_state = 3).




```
ALL CLASSIFICATION ACCURACY
======
RF  =  0.912751677852349
GB  =  0.910234899328859
REG_ALL  =  0.8901006711409396
REG_STEPWISE  =  0.87751677852349
REG_TREE  =  0.8766778523489933
REG_GB  =  0.8766778523489933
TREE  =  0.8766778523489933
REG_RF  =  0.875
------


ALL LOSSES MODEL ACCURACY
======
GB  =  2420.601900970896
RF  =  2951.0813707195125
REG_ALL  =  3634.676632630515
REG_TREE  =  4246.8855302376705
REG_RF  =  4358.06472084949
REG_GB  =  4358.06472084949
REG_STEPWISE  =  4358.06472084949
TREE  =  5722.46895603711
```