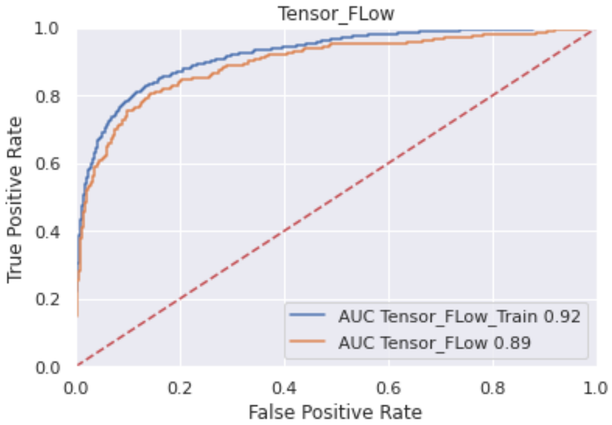
Vivian Xia

Assignment 4: Statistical Output

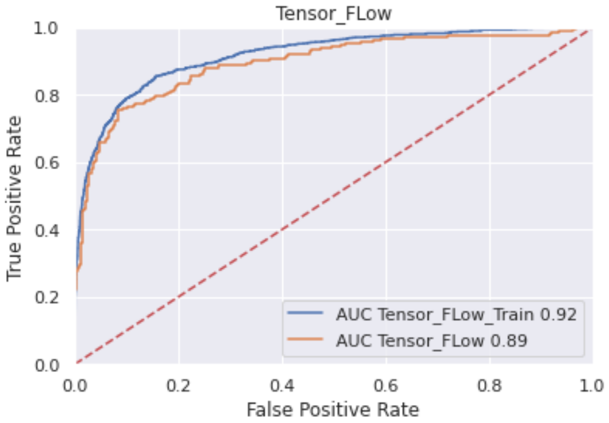
Loan Default Probability

One layer (Dense, activation = relu):



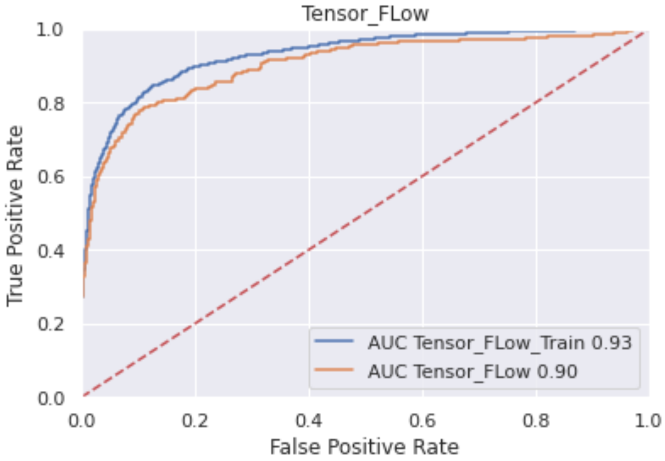
```
Tensor_FLow CLASSIFICATION ACCURACY
======
Tensor_FLow_Train  =  0.8966023489932886
Tensor_FLow  =  0.8800335570469798
------
```

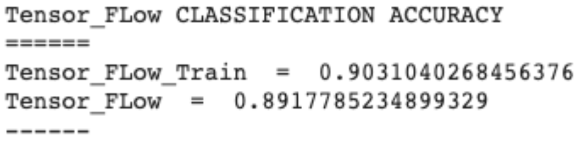
Two layers (Dense and Dropout, activation = relu):



```
Tensor_FLow CLASSIFICATION ACCURACY
======
Tensor_FLow_Train  =  0.8963926174496645
Tensor_FLow  =  0.8859060402684564
------
```

Three layers (Dense, Dropout, Dense and activation = relu):

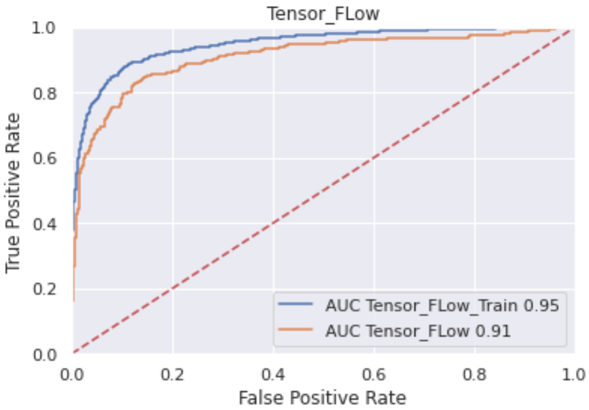


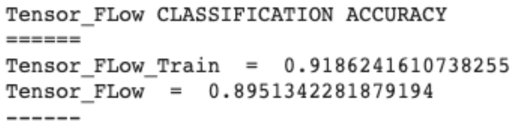


```
Tensor_FLow CLASSIFICATION ACCURACY
======
Tensor_FLow_Train  =  0.9031040268456376
Tensor_FLow  =  0.8917785234899329
------
```

For loan default probability, the tensor flow model has the best AUC and accuracy score with three layers (Dense, Dropout, Dense) with the activation function relu.

Three layers (epoch = 500, activation = relu):

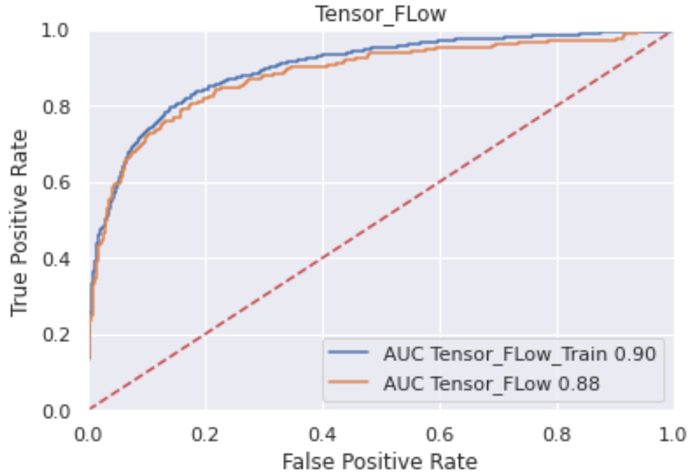


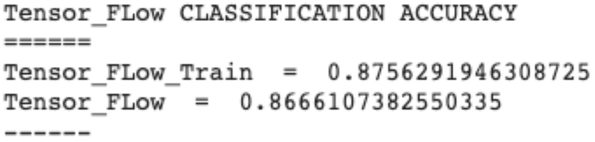


```
Tensor_FLow CLASSIFICATION ACCURACY
======
Tensor_FLow_Train  =  0.9186241610738255
Tensor_FLow  =  0.8951342281879194
------
```

I also increased the number of steps from F_theEpochs = 100 to 500, and the model also did better with more steps, but the AUC and accuracy of the model did not increase by much, so I will stick with 100 steps.

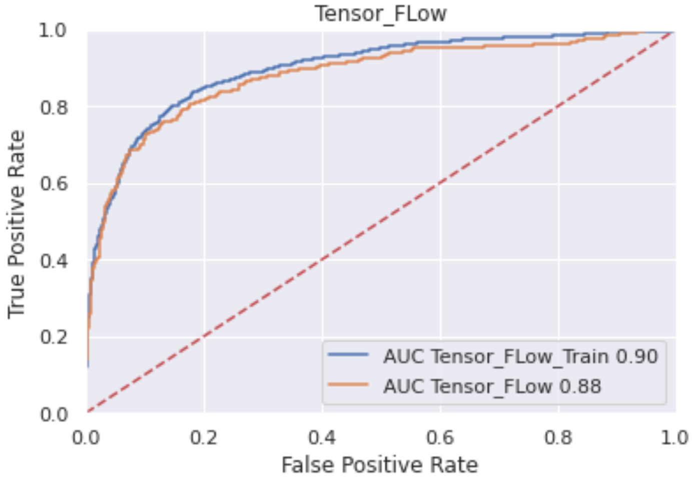
Three layers (epoch = 100, activation = softmax):

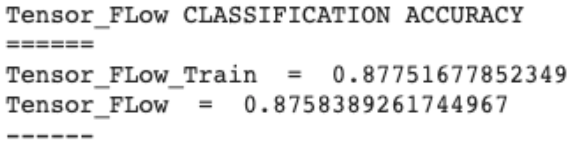




```
Tensor_FLow CLASSIFICATION ACCURACY
======
Tensor_FLow_Train  =  0.8756291946308725
Tensor_FLow  =  0.8666107382550335
------
```

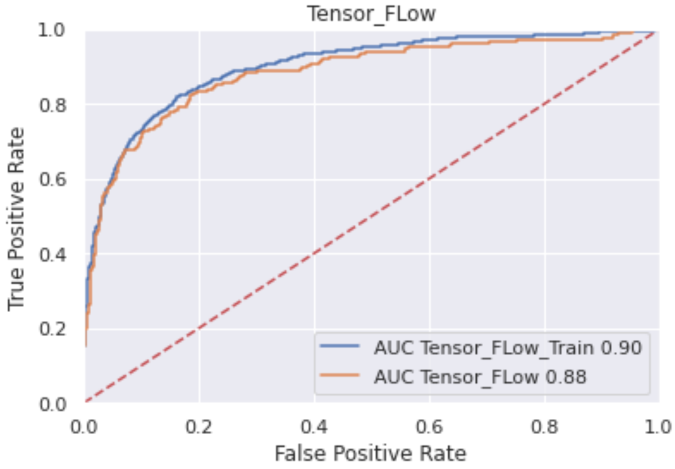
Three layers (epoch = 100, activation = sigmoid):

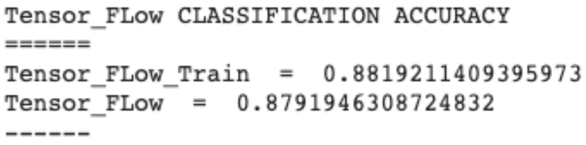




```
Tensor_FLow CLASSIFICATION ACCURACY
======
Tensor_FLow_Train  =  0.87751677852349
Tensor_FLow  =  0.8758389261744967
------
```

Three layers (epoch = 100, activation = softmax):





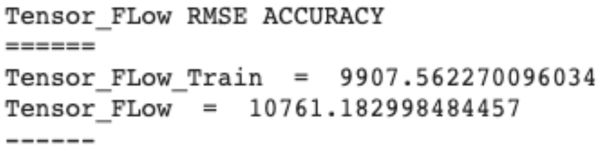
```
Tensor_FLow CLASSIFICATION ACCURACY
======
Tensor_FLow_Train  =  0.8819211409395973
Tensor_FLow  =  0.8791946308724832
------
```

I explored the AUC and accuracy scores of a three layer and epoch = 100 model with different activation functions. The relu activation function had the best AUC and accuracy scores. The softmax, sigmoid, and softplus also provided pretty high accuracy scores respectively, but the relu model does do better by around 0.02, so the relu activation function should be used.

The final tensorflow model had three layers (dense, dropout, dense) with an activation function of relu, and has an epoch of 100.

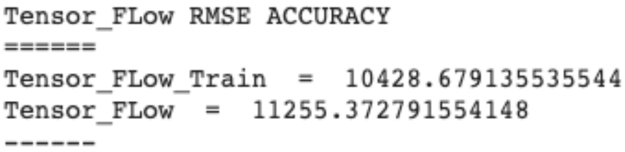
Loss Amount

One layer (Dense, activation function = relu):



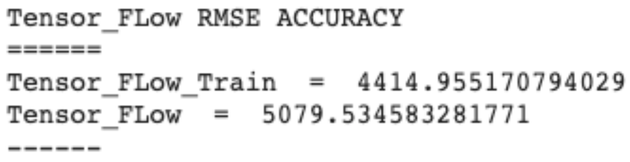
```
Tensor_FLow RMSE ACCURACY
======
Tensor_FLow_Train  =  9907.562270096034
Tensor_FLow  =  10761.182998484457
------
```

Two layers (Dense and Dropout, activation = relu):



```
Tensor_FLow RMSE ACCURACY
======
Tensor_FLow_Train  =  10428.679135535544
Tensor_FLow  =  11255.372791554148
------
```

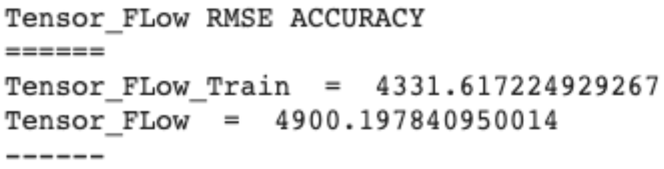
Three layers (Dense, Dropout, Dense and activation = relu):



```
Tensor_FLow RMSE ACCURACY
======
Tensor_FLow_Train  =  4414.955170794029
Tensor_FLow  =  5079.534583281771
------
```

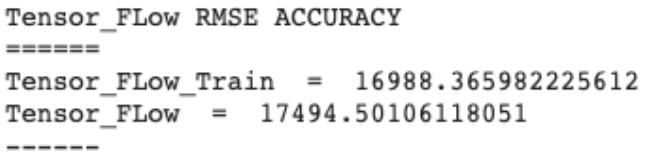
The three-layer model has a much smaller RMSE than the models with one or two layers, so the three layer (dense, dropout, dense) should be used.

Three layers (activation = softplus):



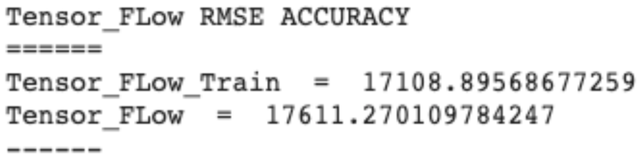
```
Tensor_FLow RMSE ACCURACY
======
Tensor_FLow_Train  =  4331.617224929267
Tensor_FLow  =  4900.197840950014
------
```

Three layers (activation = sigmoid):



```
Tensor_FLow RMSE ACCURACY
======
Tensor_FLow_Train  =  16988.365982225612
Tensor_FLow  =  17494.50106118051
------
```

Three layers (activation = softmax):

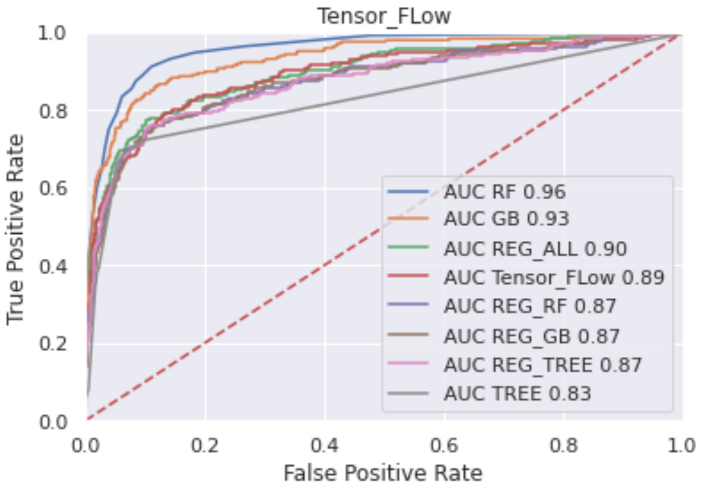


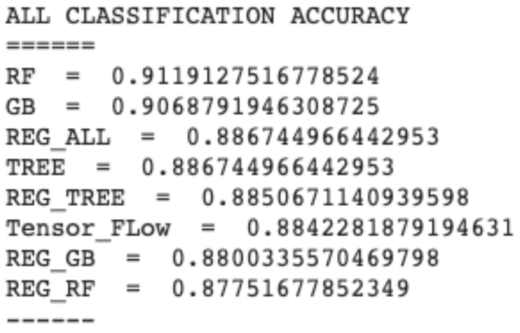
```
Tensor_FLow RMSE ACCURACY
======
Tensor_FLow_Train  =  17108.89568677259
Tensor_FLow  =  17611.270109784247
------
```

The model with the softplus activation function has the lowest RMSE of the other three layer models with other activation functions. The model with the activation of sigmoid or softmax have a much larger RMSE score. The model with relu is a little worse than the model with softplus, so softplus should be used in the final model.

Comparison of Models

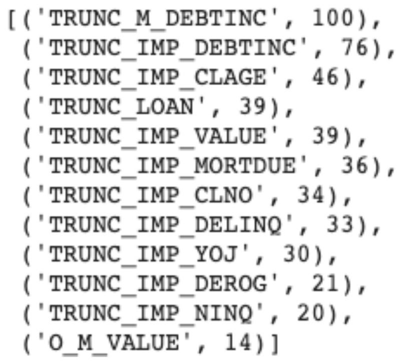
Loan Default Probability:





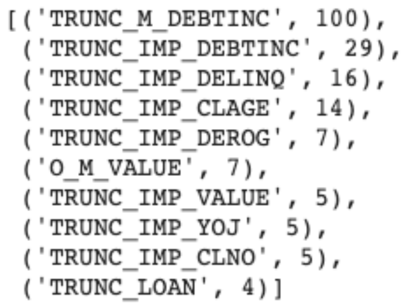
```
ALL CLASSIFICATION ACCURACY
======
RF  =  0.9119127516778524
GB  =  0.9068791946308725
REG_ALL  =  0.886744966442953
TREE  =  0.886744966442953
REG_TREE  =  0.8850671140939598
Tensor_FLow  =  0.8842281879194631
REG_GB  =  0.8800335570469798
REG_RF  =  0.87751677852349
------
```

The most accurate model is Random Forest Classifier model. It has the best AUC and accuracy score compared to the other models. The following is the list of the 12 variables used for the RF Classifier model:



```
[('TRUNC_M_DEBTINC', 100),
 ('TRUNC_IMP_DEBTINC', 76),
 ('TRUNC_IMP_CLAGE', 46),
 ('TRUNC_LOAN', 39),
 ('TRUNC_IMP_VALUE', 39),
 ('TRUNC_IMP_MORTDUE', 36),
 ('TRUNC_IMP_CLNO', 34),
 ('TRUNC_IMP_DELINQ', 33),
 ('TRUNC_IMP_YOJ', 30),
 ('TRUNC_IMP_DEROG', 21),
 ('TRUNC_IMP_NINQ', 20),
 ('O_M_VALUE', 14)]
```

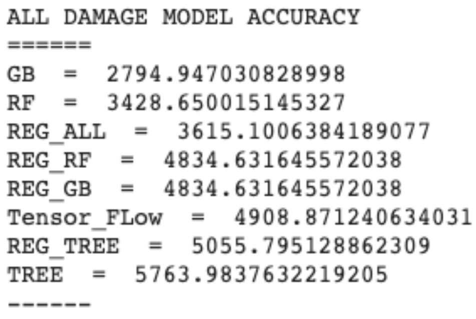
The list of the 10 variables in the GB Classifier model is the following:



```
[('TRUNC_M_DEBTINC', 100),
 ('TRUNC_IMP_DEBTINC', 29),
 ('TRUNC_IMP_DELINQ', 16),
 ('TRUNC_IMP_CLAGE', 14),
 ('TRUNC_IMP_DEROG', 7),
 ('O_M_VALUE', 7),
 ('TRUNC_IMP_VALUE', 5),
 ('TRUNC_IMP_YOJ', 5),
 ('TRUNC_IMP_CLNO', 5),
 ('TRUNC_LOAN', 4)]
```

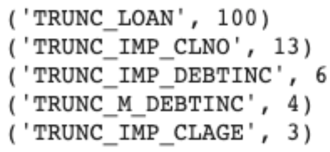
The GB model does not use TRUNC_IMP_MORTDUE and TRUNC_IMP_NINQ that are used in the RF model, but the difference in accuracy accounts for these extra two variables to be considered, so I would still recommend using the RF model. The regression model that has the best accuracy score is the one with all the variables. Even though regression models are easier to deploy than classifier tree models, the difference in accuracy scores is 0.025 and it also uses many more variables, more than three times the number of variables than the RF Classifier model uses.

Loss Amount:



```
ALL DAMAGE MODEL ACCURACY
======
GB  =  2794.947030828998
RF  =  3428.650015145327
REG_ALL  =  3615.1006384189077
REG_RF  =  4834.631645572038
REG_GB  =  4834.631645572038
Tensor_FLow  =  4908.871240634031
REG_TREE  =  5055.795128862309
TREE  =  5763.9837632219205
------
```

The model with the lowest RMSE score is the Gradient Boosting Classifier model. The 5 variables used are the following:



```
('TRUNC_LOAN', 100)
('TRUNC_IMP_CLNO', 13)
('TRUNC_IMP_DEBTINC', 6)
('TRUNC_M_DEBTINC', 4)
('TRUNC_IMP_CLAGE', 3)
```

These variables all make sense in predicting the loss amount. This model also does not use many variables and has a much lower RMSE than the other models, so I would recommend using the GB Classifier model to predict loss amount.