Machine Learning Consumer Loan Processing

Ram Rao

July 15, 2022

DSA 5900 Practicum

Project Definition

- Identify Credit-Worthiness of Loan Applicants at Financial Institutions
  - Apply Machine Learning Models to Evaluate whether Applicants will default on a Loan
- Identify a Process for Remote Machine Learning
  - Distributed System Training
  - Aggregation and Testing on Server
- Stakeholders:
  - Agencies that Process Consumer Loans
- Dr. Radhakrishnan and Dr. Trafalis are my advisors

Data Ingestion

Data Source:

https://www.bondora.com/en/public-reports

Tableau, Python, Sckit Learn, Tensorflow/Keras,

PyTorch and PySft

Overall Class Counts
Defaulted: 1
Not Defaulted: 0

Target Class	Count of Target Class	% of Total Count of Target Class)
0	156,588	66.0%
1	80,635	34.0%
Grand Total	237,223	100.0%

Count of Target Class and % of Total Count of Target Class) broken down by Target Class.

No of Features

111 Predictor Variables

1 Target Variable

- Defaulted : 1
- Non-Defaulted : 0

Tableau : Data Viz

Python: Data Processing

Sckit Learn: ML Models

Tensorflow/Keras: Neural Net

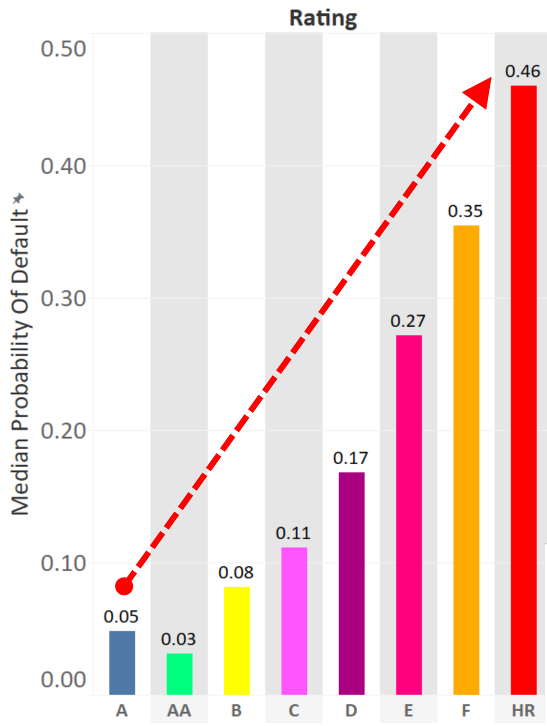
PyTorch, PySft: Remote ML

Data Exploration and Preparation - 1

Exploratory Analysis:

- ❑ Lower Default
  - ➢ Higher Income
  - ➢ Lower Interest Servicing
  - ➢ Better Credit Rating
  - ➢ Higher Previous Credit
  - ➢ Lower PrincipalOverdue
  - ➢ Higher Education
  - ➢ More Prompt Payment
- ❑ No Significant Multicollinearity
- ❑ Correlation Not High Between Predictor and Target

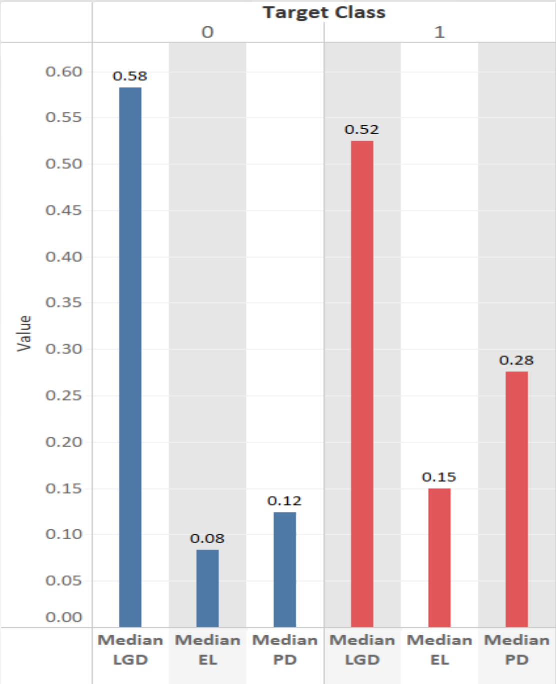
Credit Rating vs Median Probability of Default

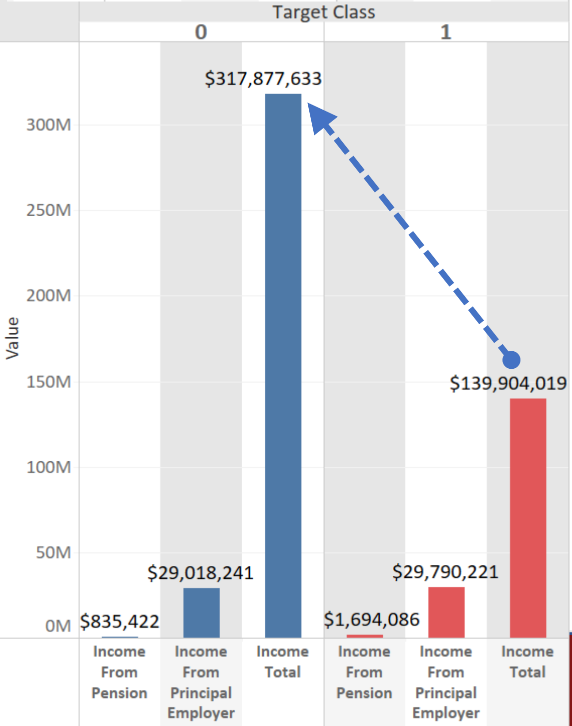


Days to Payments By Class Percentage of Total vs Days Outstanding

Defaulted: 1; Non Defaulted: 0

	Target Class		
Active Late Category	0	1	Grand Total
0-7	95.84%	4.16%	100.00%
8-15	97.51%	2.49%	100.00%
151-180	2.94%	97.06%	100.00%
180+	0.85%	99.15%	100.00%





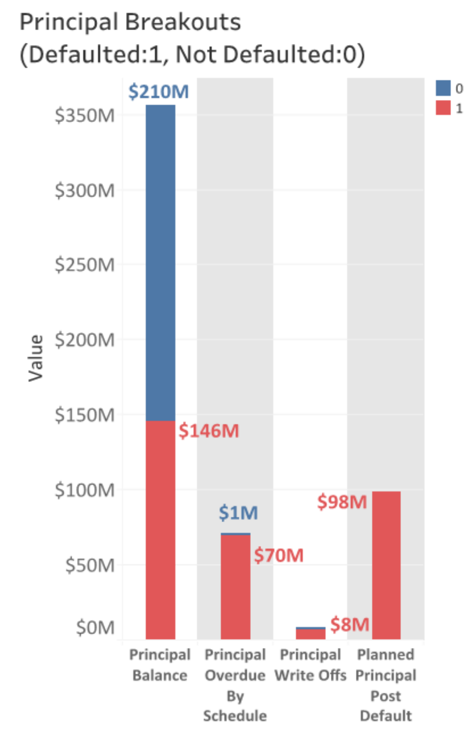
Correlation Coefficient

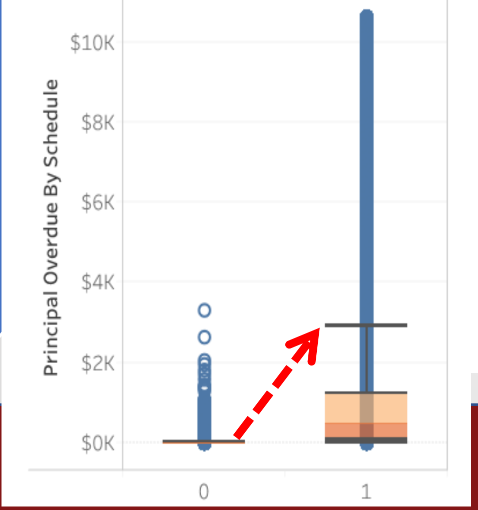
Variable_Name	Defaulted
EmploymentDurationCurrentEmployer_UpTo3Years	0.091
NewCreditCustomer_True	0.102
EmploymentDurationCurrentEmployer_UpTo2Years	0.108
PrincipalBalance	0.111
RefinanceLiabilities	0.119
Rating_E	0.120
IncomeFromPrincipalEmployer	0.144
MonthlyPayment	0.160
PlannedInterestTillDate	0.187
OccupationArea	0.237
DebtToIncome	0.245
Rating_HR	0.249
UseOfLoan	0.254
Rating_F	0.256
ExpectedReturn	0.273
ActiveScheduleFirstPaymentReached_True	0.277
MaritalStatus	0.282
EmploymentStatus	0.286
Country_ES	0.298
Interest	0.354
ExpectedLoss	0.409
ProbabilityOfDefault	0.432
PrincipalOverdueBySchedule	0.487
Status_Late	0.758
Defaulted	1.000

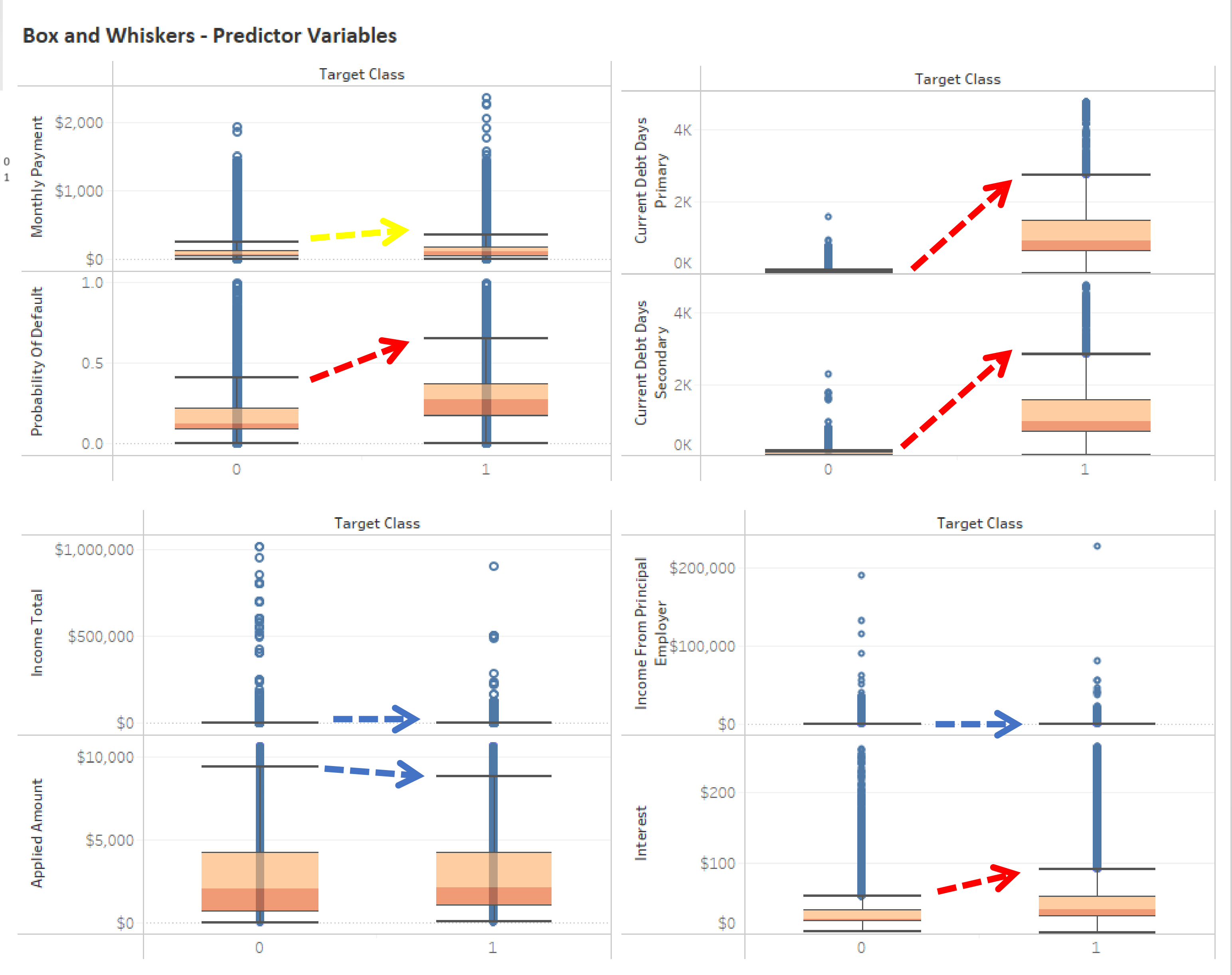
Data Exploration and Preparation - 2

Exploratory Analysis:

- ❑ **Higher Default**
  - ➢ Higher Principal Overdue
- ❑ Higher Spread and Max for Target Class 1
  - ➢ Probability of Default
  - ➢ Debt Types
  - ➢ Interest Servicing
  - ➢ Principal Overdue
- ❑ No Significant Differences Between Classes
  - ➢ Applied Amount
  - ➢ Income Types



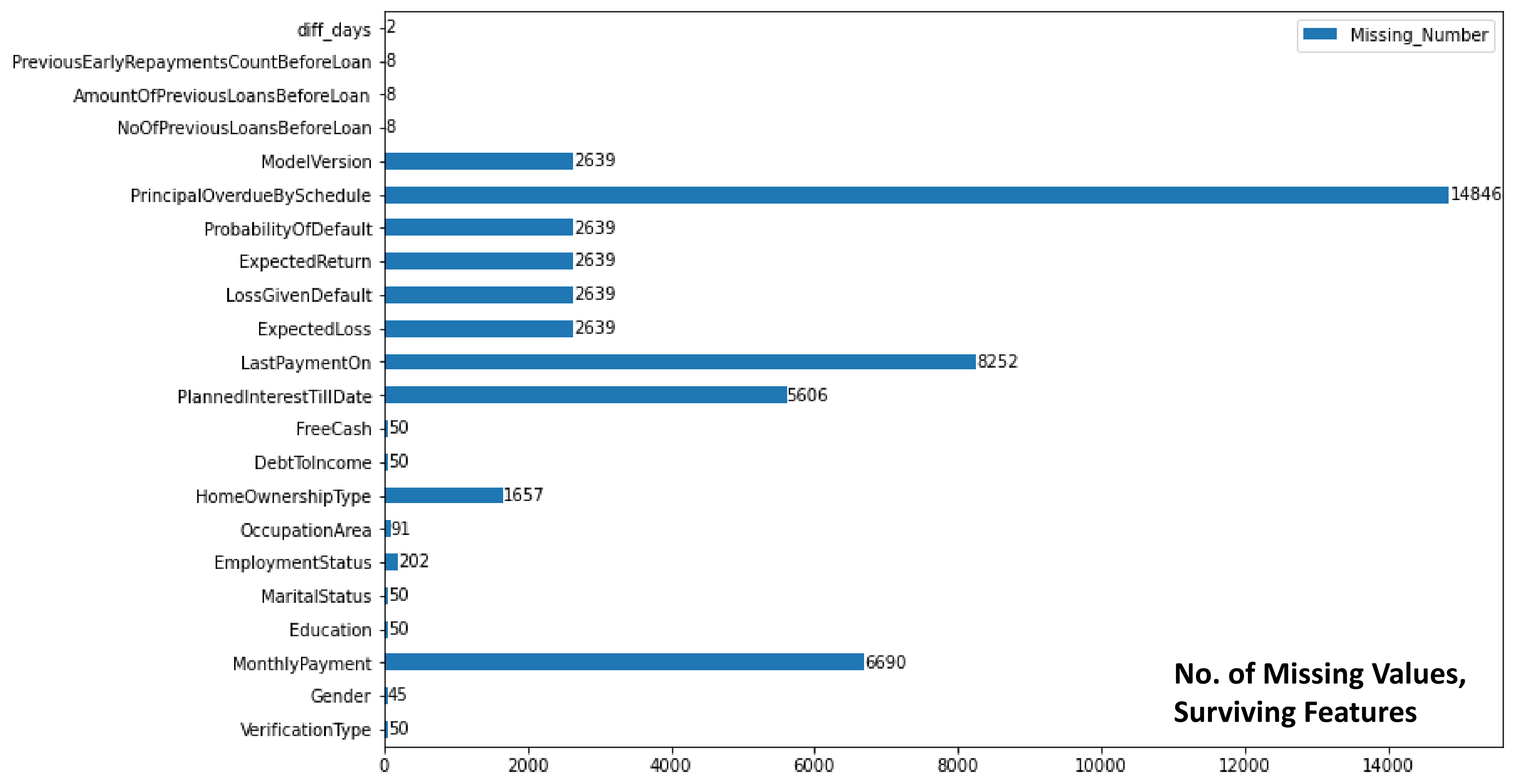




Data Exploration and Preparation - 3

Exploratory Analysis:

- ❑ Missing Value Handling
  - ✓ Removed Categorical Variables with No Numerical Value
  - ✓ Removed Variables with more than 10 pct Missing
  - ✓ Removed Variables Populated Following Default
  - ✓ Removed Rows with Missing Values for Surviving Features
  - ✓ Scaled Continuous Variables
  - ✓ One Hot Encoded Categorical Variables



No. of Missing Values, Surviving Features

Data Cleansing

Dataset ID	No of Features
Original Dataset	112
Final Dataset	59
Final Dataset, Following Scaling and Hot Encoding	72

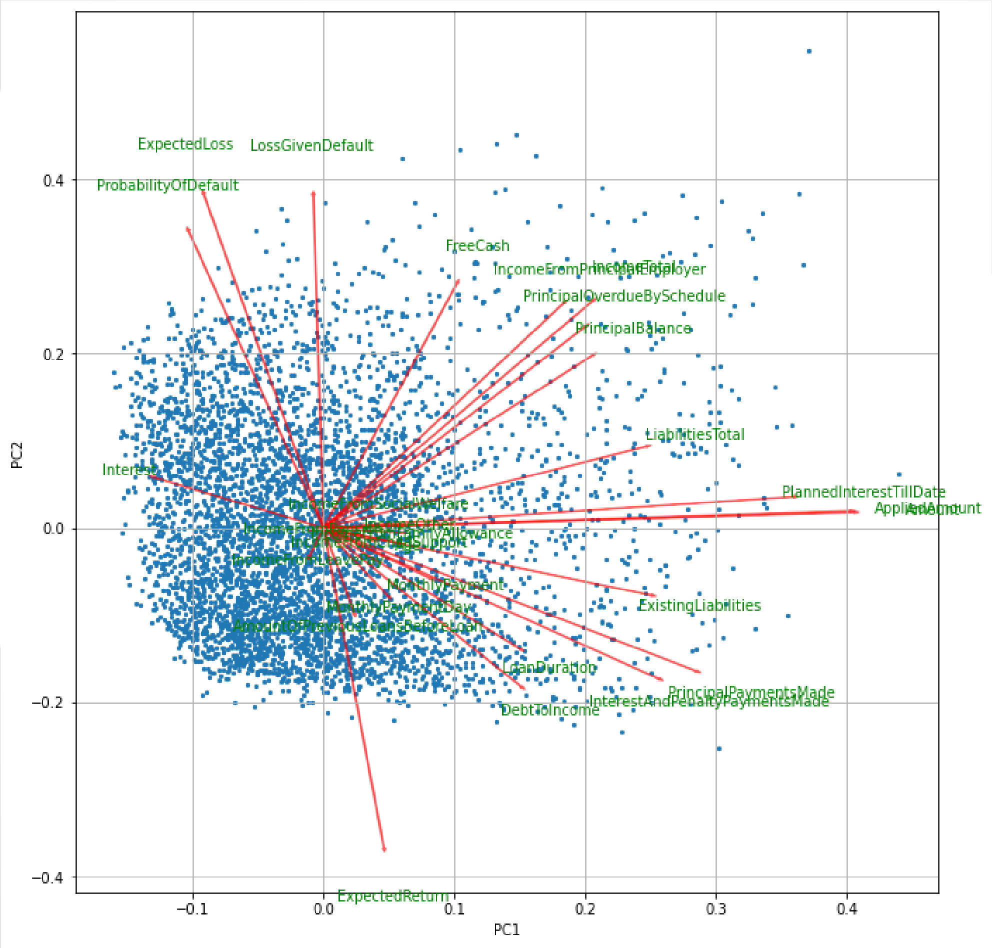
Final Dataset Breakdown

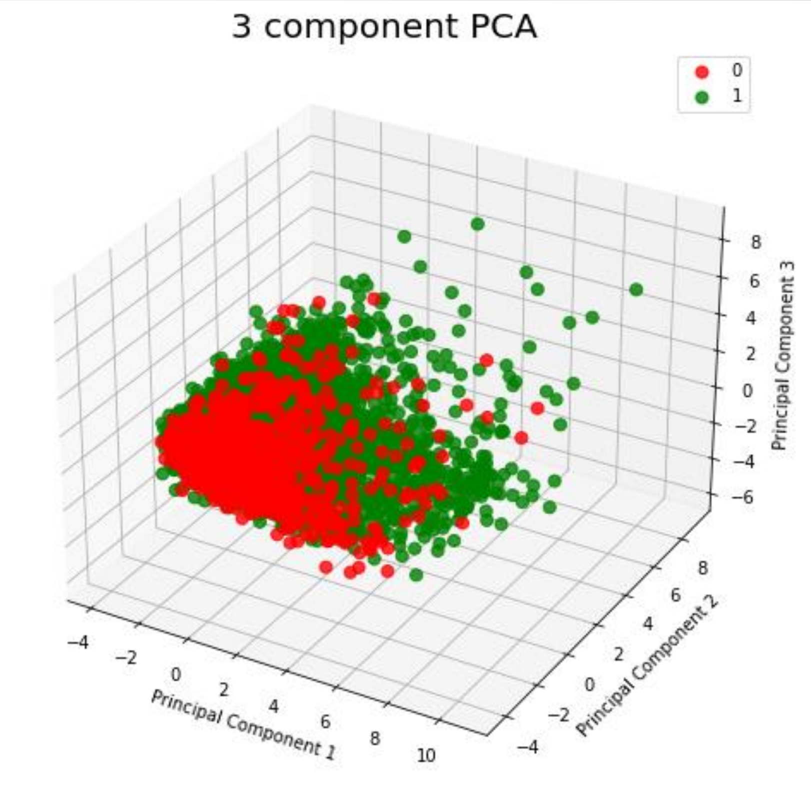
Target Class	Count of Target Class	% of Total Count of Target Class
0	137,895	65.28%
1	73,345	34.72%
Total	211,240	100.00%

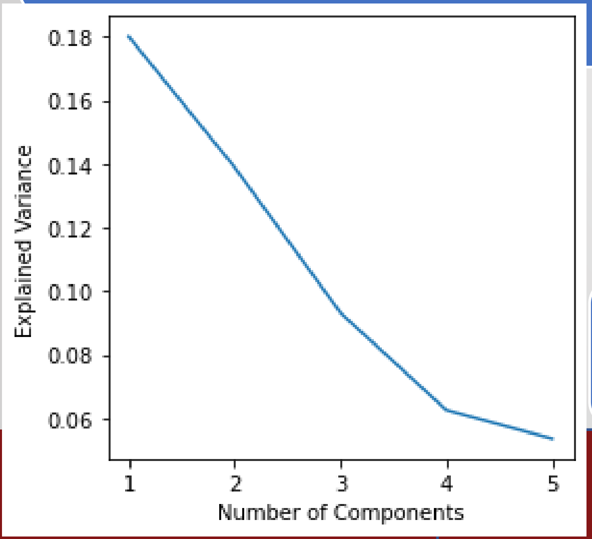
PCA Assessment

PCA Analysis:

- ✓ 5,000 Dataset Points Analyzed
- ✓ No of Continuous Variables Scaled and Transformed: 28
- ✓ Limited Variance Explained by 5 Components
- ✓ No Significant Separation Between Classes Observed from PCA 1, 2, and 3
- ✓ Bi Plot shows Explanation of Few Features from PCA 1 and 2

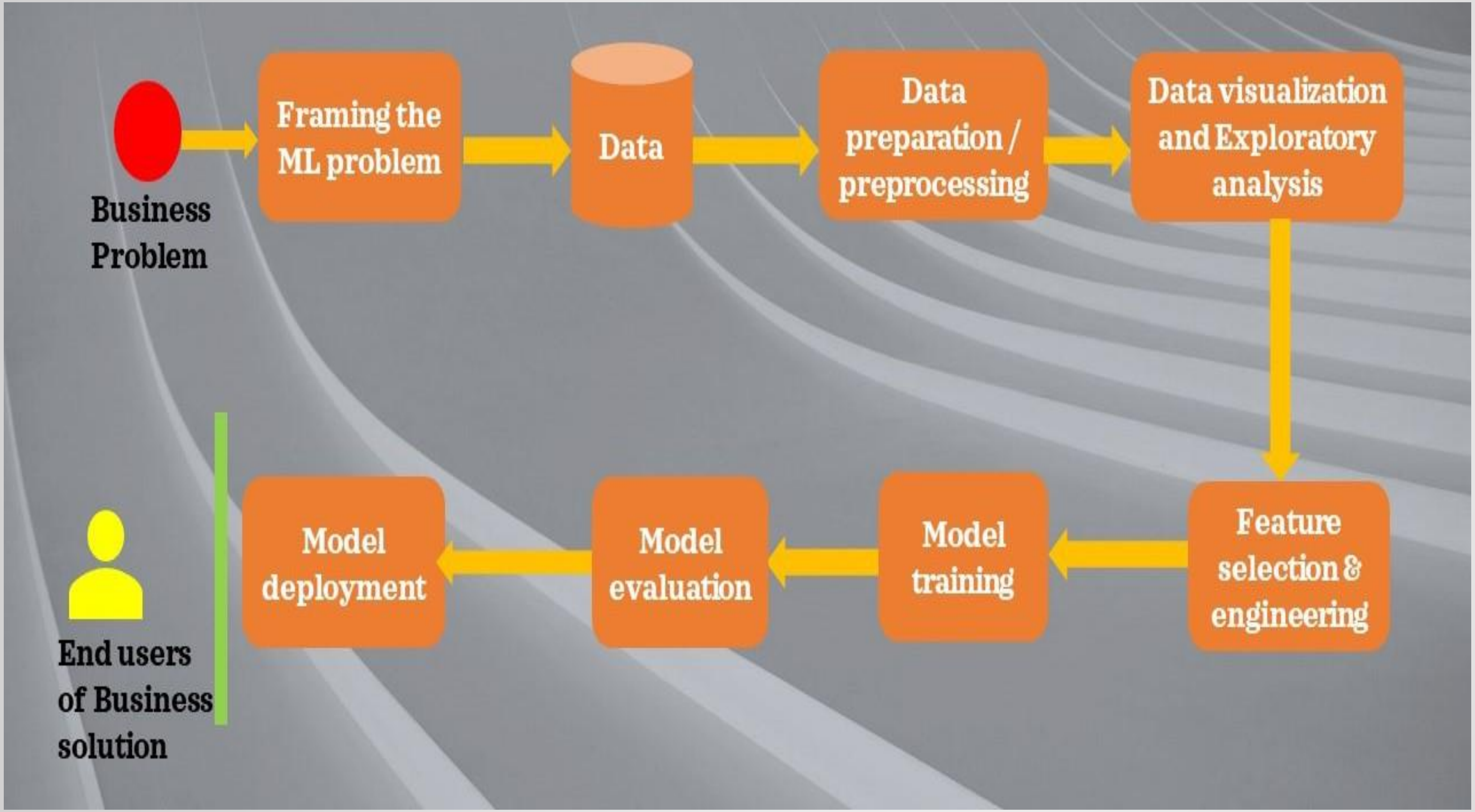






PCA Not a Significant Benefit to Model Predictability, Categorical Count Outweighs Continuous Variables

Modeling Preprocessing And Overview



- ❑ **Preprocessing with Sckit-Learn**
  - ✓ Scaled Continuous Variables
  - ✓ One hot encoded Categorical Variables
- ❑ **Modeling, Training/Testing**
  - ✓ Sckit Learn
  - ✓ Tensorflow Keras
    - ▪ Default
    - ▪ GridSearch CV Optimization
  - ✓ Remote Machine Learning – PyTorch and PySft
  - ✓ Sckit-Learn Metrics for Evaluation

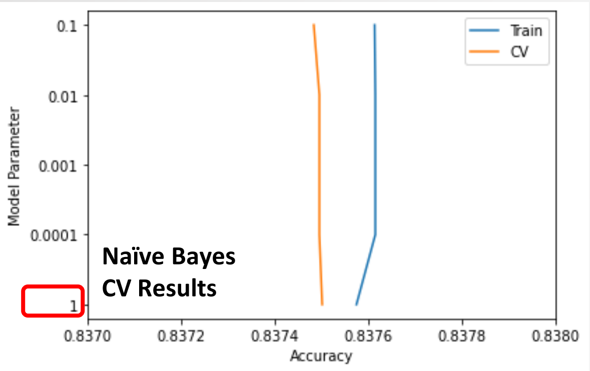
Model Results - Logistic Regression and Naïve Bayes

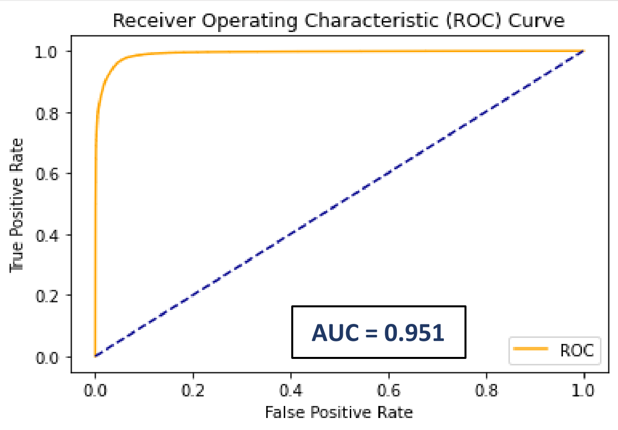
Logistic Regression:

- Grid Search 5-Fold CV
- 200 Iterations
- Hyperparameters
  - ✓ Penalty: L1 and L2, Elasticnet
  - ✓ C : 1, 5, 10
  - ✓ Solver, lbfgs, liblinear and saga
  - ✓ L1_ratio: 0.2, 0.6

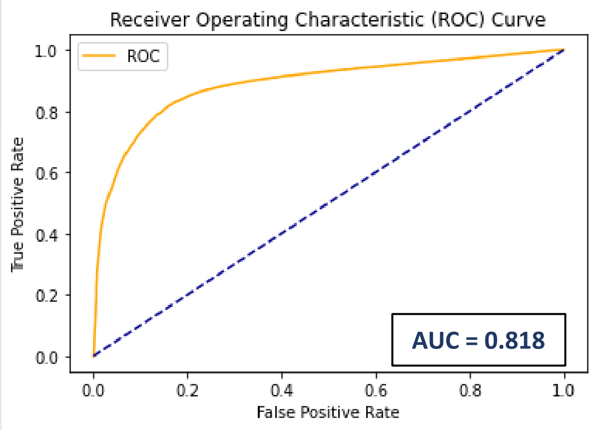
Naïve Bayes:

- Grid Search 5-Fold CV
- Hyperparameters
  - ✓ Alpha: 1E-4, 1E-2, 1E-1, and 1





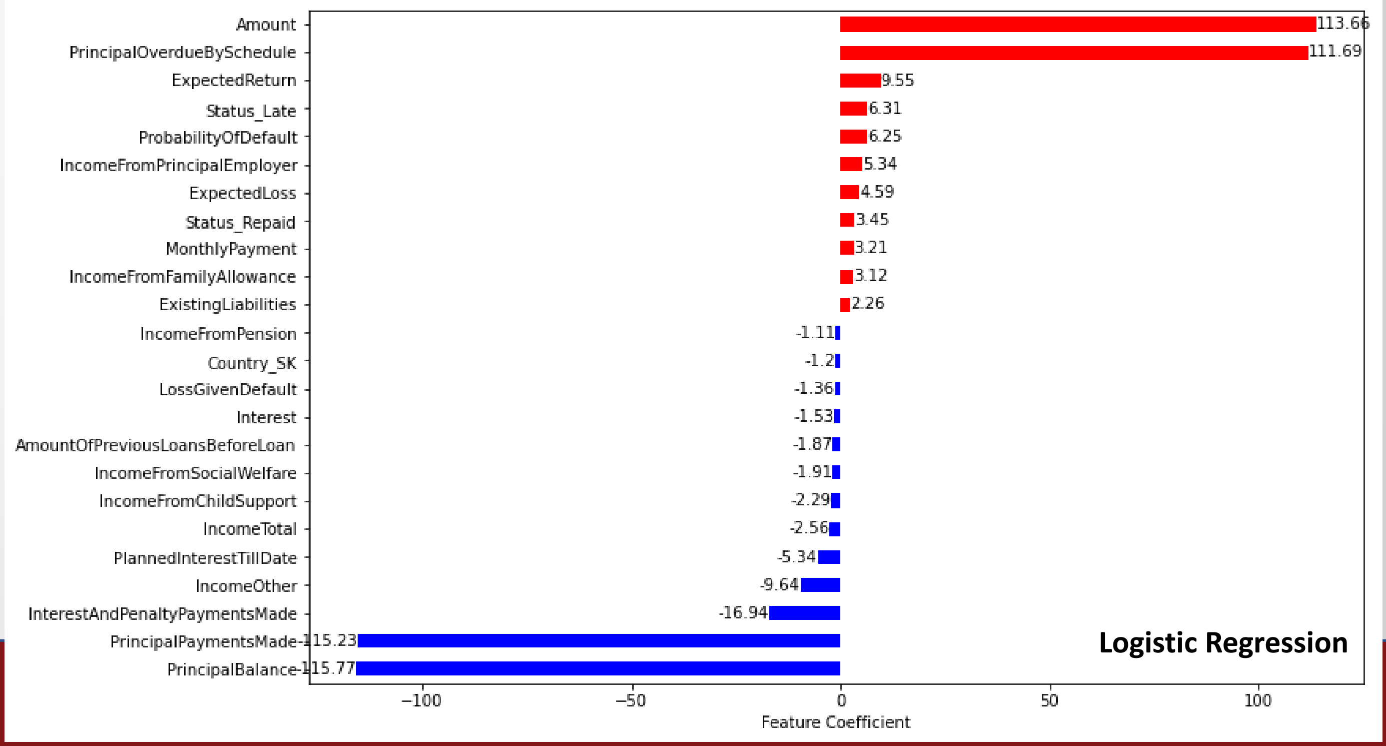
Logistic Regression



Naïve Bayes

Logistic Regression	Class 0 Predicted	Class 1 Predicted
Class 0 Actual	26,280	907
Class 1 Actual	928	13,687

Naïve Bayes	Class 0 Predicted	Class 1 Predicted
Class 0 Actual	24,283	2,904
Class 1 Actual	3,762	10,853



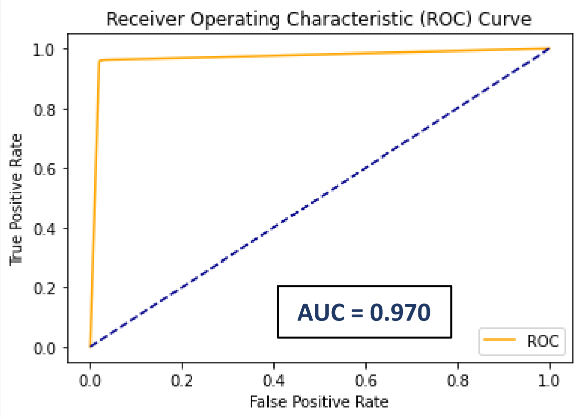
Model Results - Decision Trees and Ensemble Forest

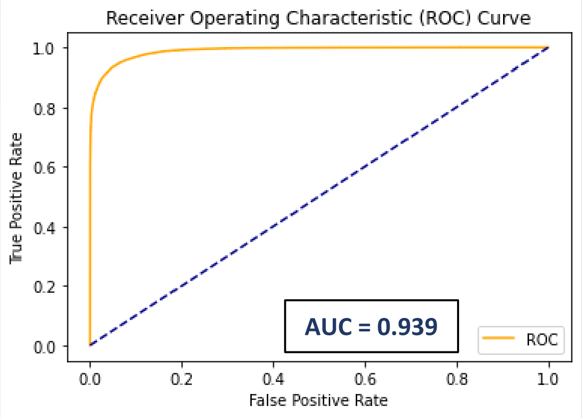
Decision Trees:

- Grid Search 5-Fold CV
- Hyperparameters
  - ✓ Criterion : gini, **entropy**
  - ✓ Max_depth : 5, 10, **20**

Ensemble Forest:

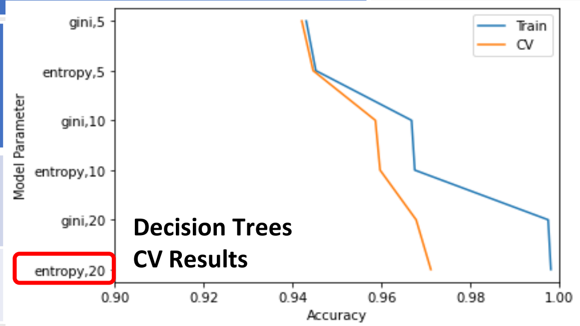
- Grid Search 5-Fold CV
- Hyperparameters
  - ✓ N_estimators: 5,10,20, 50, **100**
  - ✓ Learning_Rate: 0.1, 0.5, **1.0**, 2.0, 5.0

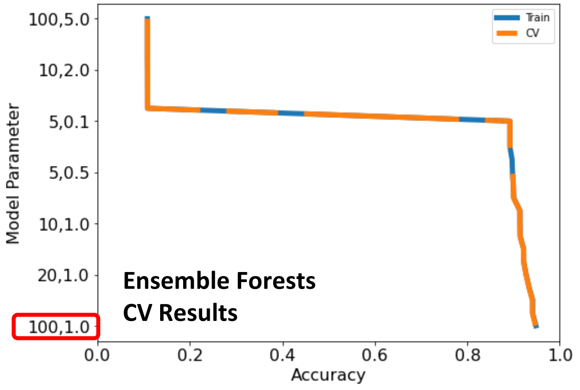




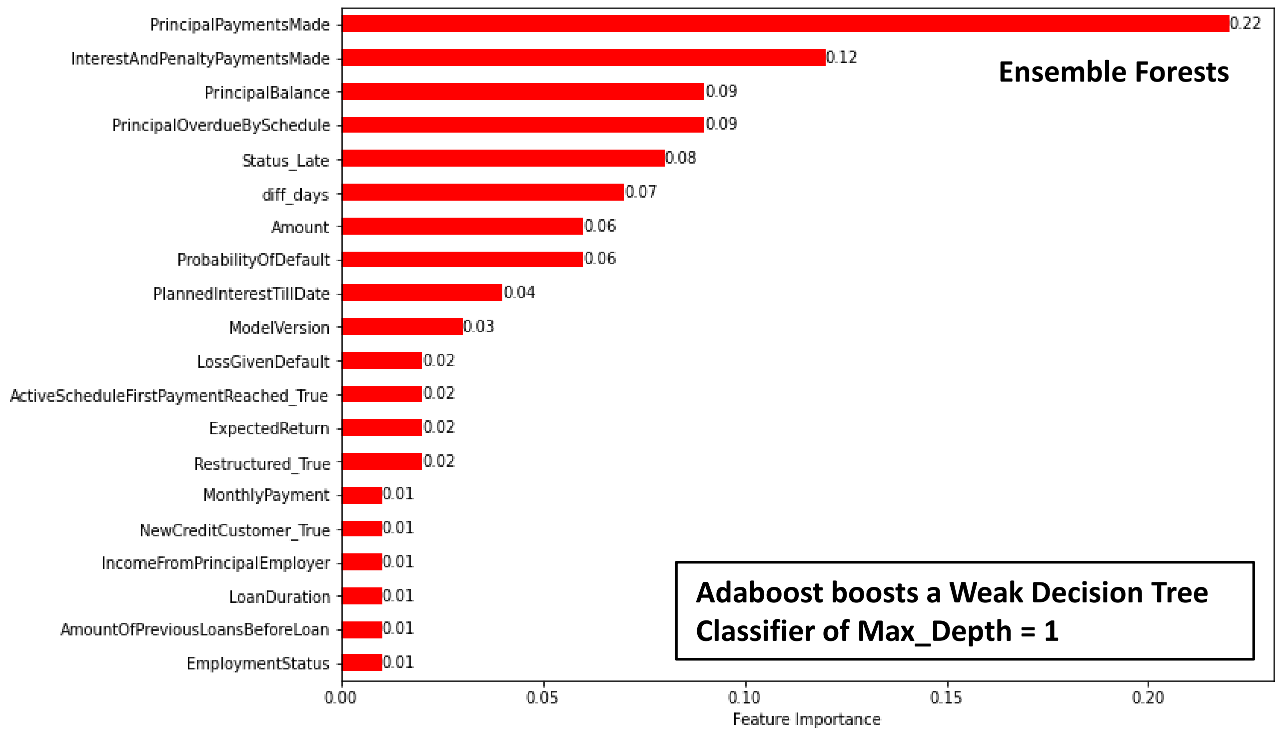
Decision Trees	Class 0 Predicted	Class 1 Predicted
Class 0 Actual	26,663	554
Class 1 Actual	591	14,024

Ensemble Forest	Class 0 Predicted	Class 1 Predicted
Class 0 Actual	26,238	949
Class 1 Actual	1,276	13,339





Decision Trees **Ensemble Forest**

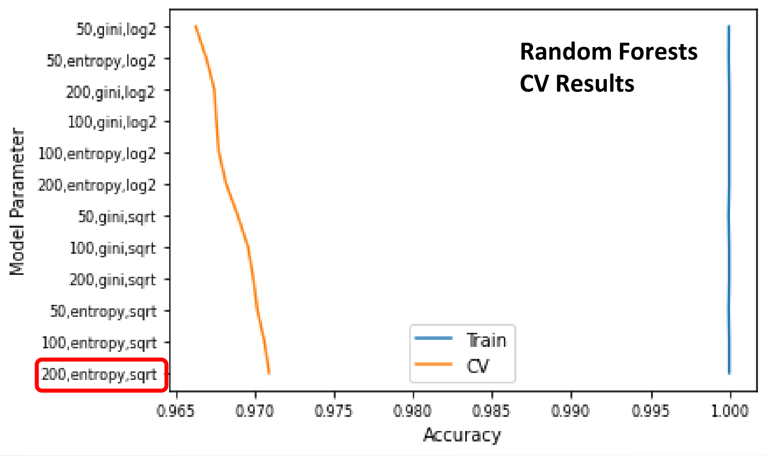


Model Results Random Forest

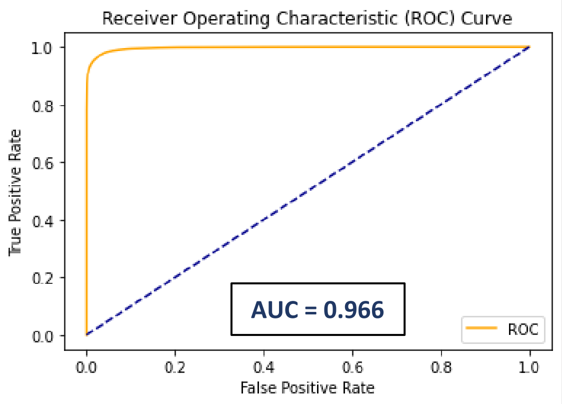
Random Forest:

- Grid Search 5-Fold CV
  - Criterion: gini
  - Max_depth: None
- Hyperparameters
  - ✓ N_estimators: 50, 100, 200
  - ✓ Criterion: gini, entropy
  - ✓ Max_features: sqrt, log2

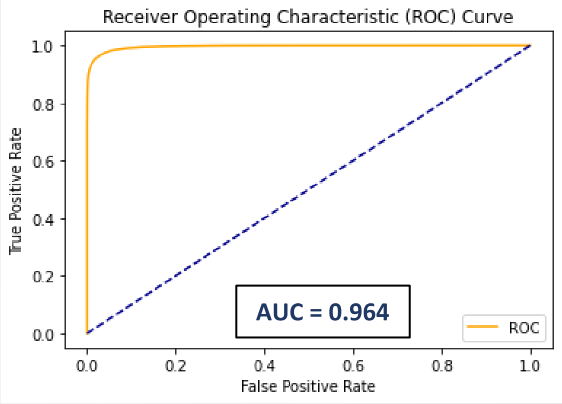
Random Forest	Class 0 Predicted	Class 1 Predicted
Class 0 Actual	**26,854**	**333**
Class 1 Actual	**826**	**13,789**



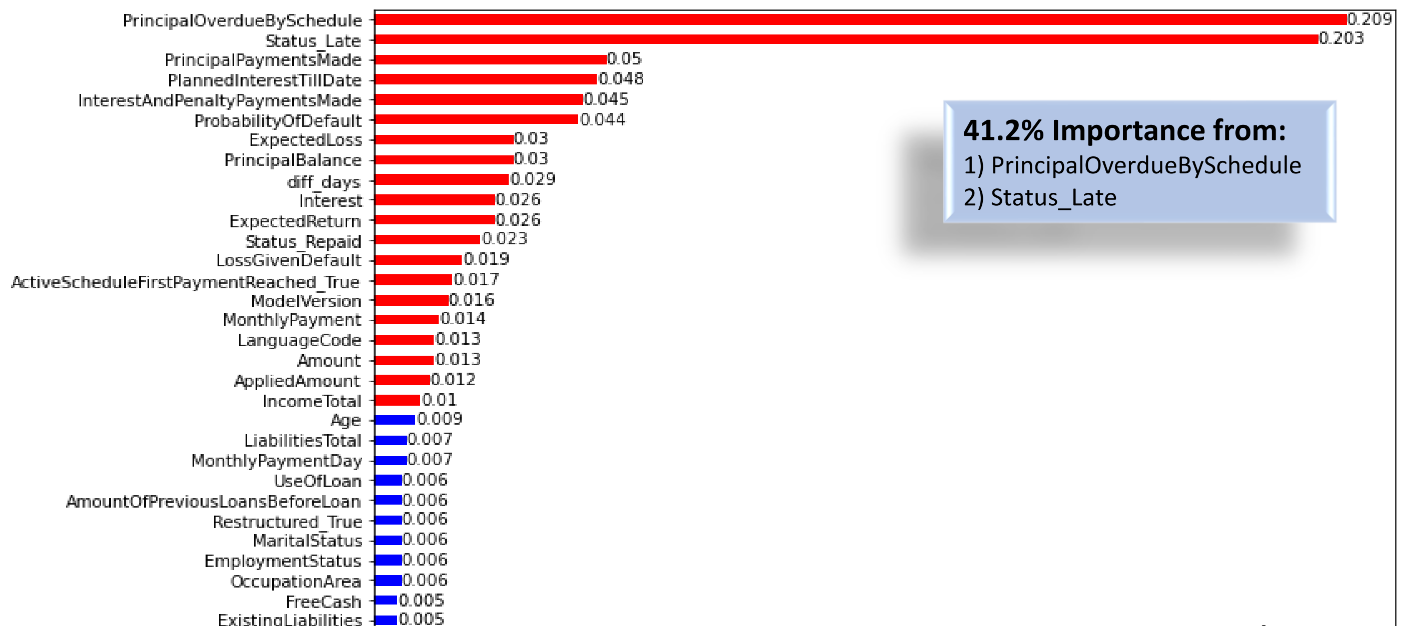
Random Forest w/Tuning



Random Forest w/Tuning



Random Forest w/o Tuning



Random Forest w/Tuning

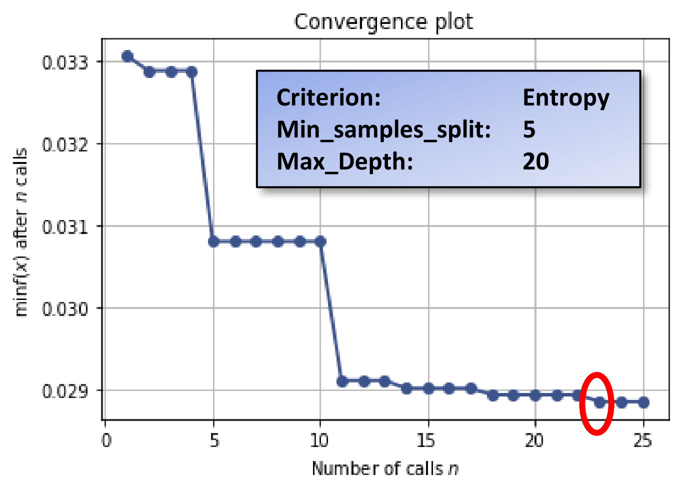
Tree, Boosted Weak Tree, Random Forest Bayesian Optimization

Bayesian Optimization:

- Scikit –Optimize
- Gaussian-Minimize Function
- Objective Function
  - 1-Accuracy
- Surrogate Function
  - Multivariate Gaussian
- Acquisition Function
  - LCB/EI/PI
- 25 Iterations, 5-Fold CV
- Best Result from Search Space Found

Search Space

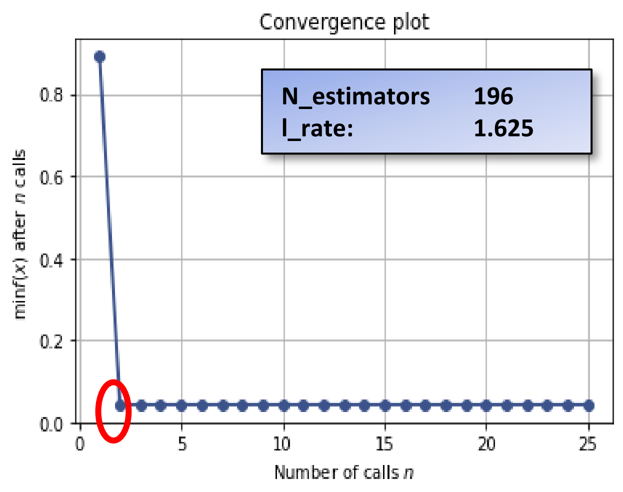
Criterion:	[Gini, Entropy]
Min_samples_split:	[2,5]
Max_Depth:	[5,20]



Decision Tree – BO Results
(1-Accuracy) vs No. of Iterations

Search Space

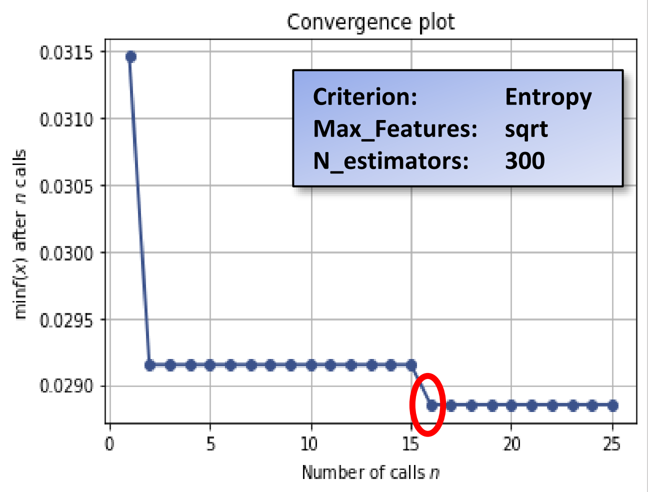
N_estimators:	[5,200]
l_rate:	[0.1, 5]



Ensemble Forest – BO Results
(1-Accuracy) vs No. of Iterations

Search Space

Criterion:	[Gini, Entropy]
Max_Features:	[sqrt, log2]
N_estimators:	[50,300]



Random Forest – BO Results
(1-Accuracy) vs No. of Iterations

Random Forests	Class 0 Predicted	Class 1 Predicted
Class 0 Actual	**26,842**	**345**
Class 1 Actual	**816**	**13,799**

Performance Similar to Gridsearch CV

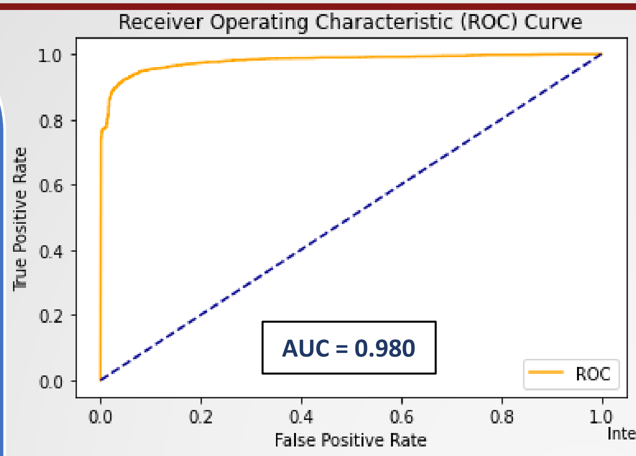
- ✓ Search Space Uniform to Log Uniform Sampling within Provided Bounds for Integer/Real and from Provided List for Categorical
- ✓ Less Expensive And Results could be Better and Reduce Underfitting Depending on Training Set

Model Results - Neural Nets, Keras/Tensorflow

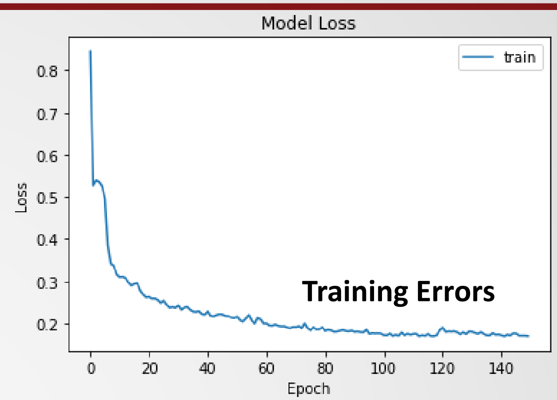
Neural Net:

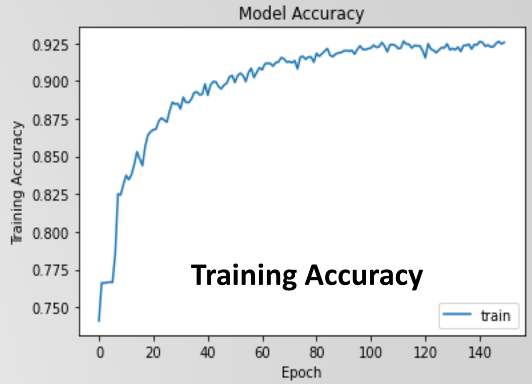
- ✓ 3 Hidden Layers: 100, 50, and 25 Neurons, Relu Activation
- ✓ 1 Output Layer, 1 Neuron, Sigmoid Activation
- ✓ Grid Search CV = 3
- ✓ Hyperparameters
- Optimizer: rmsprop, adam
- inits: glorot_uniform, normal, uniform
- Epochs: 50, 100, 150
- Batches: 5, 20

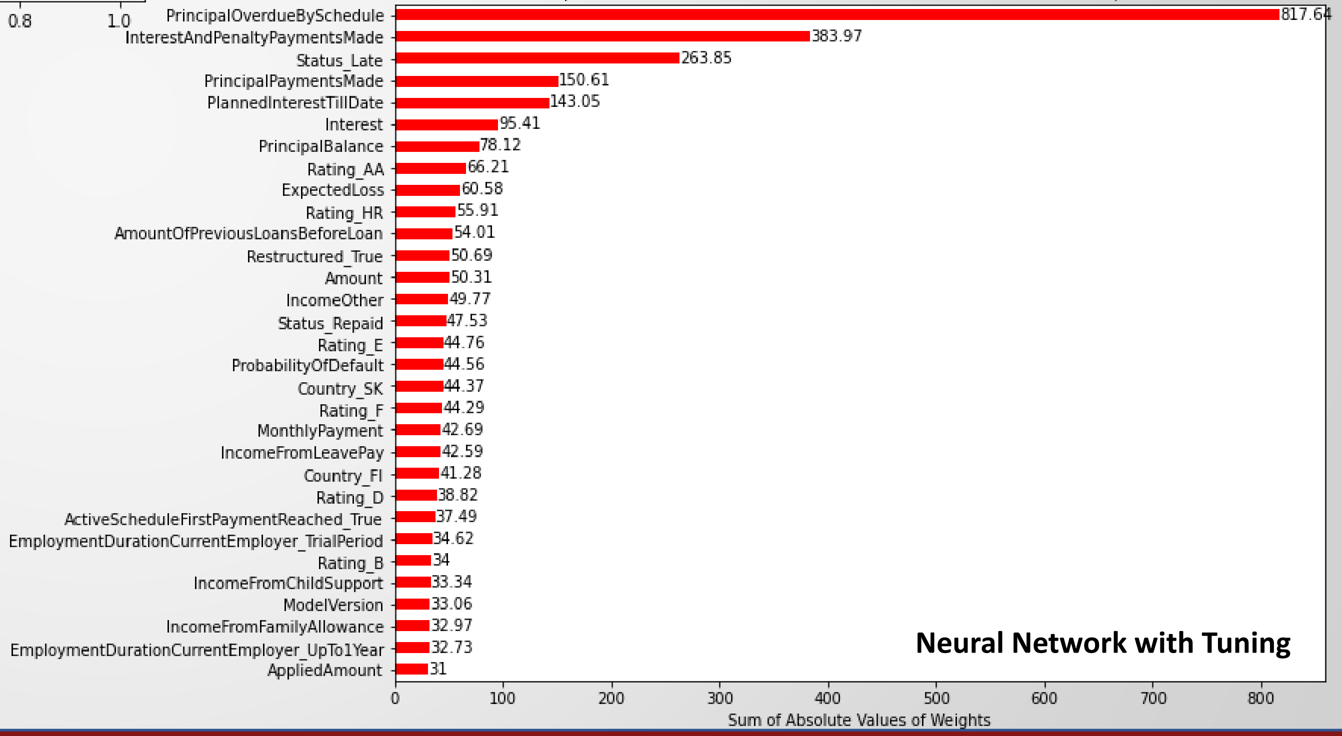
Neural Net	Class 0 Predicted	Class 1 Predicted
Class 0 Actual	**630**	**308**
Class 1 Actual	**44**	**3,018**



Neural Network







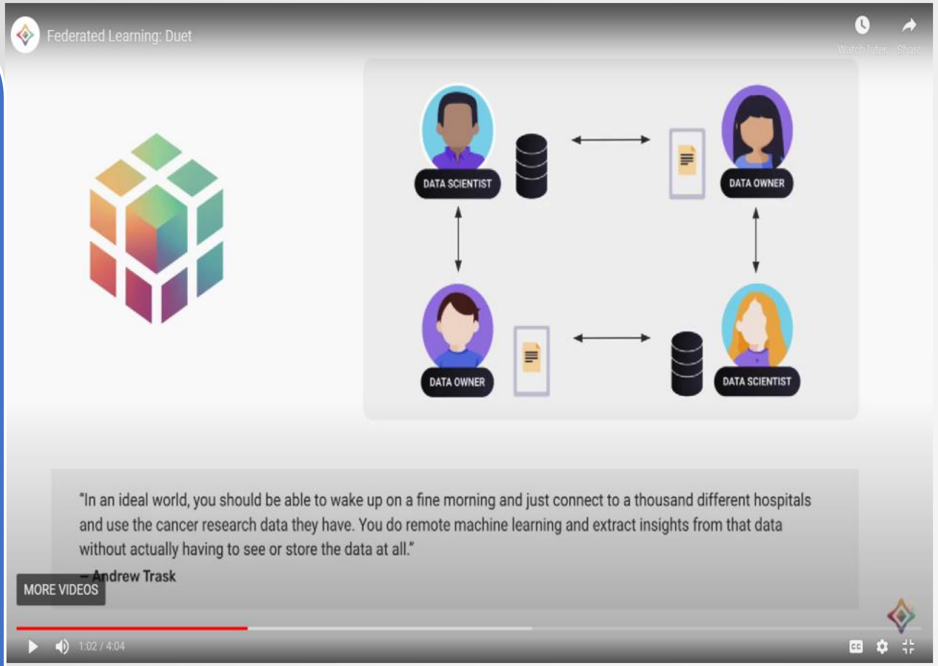
Remote Machine Learning - Overview

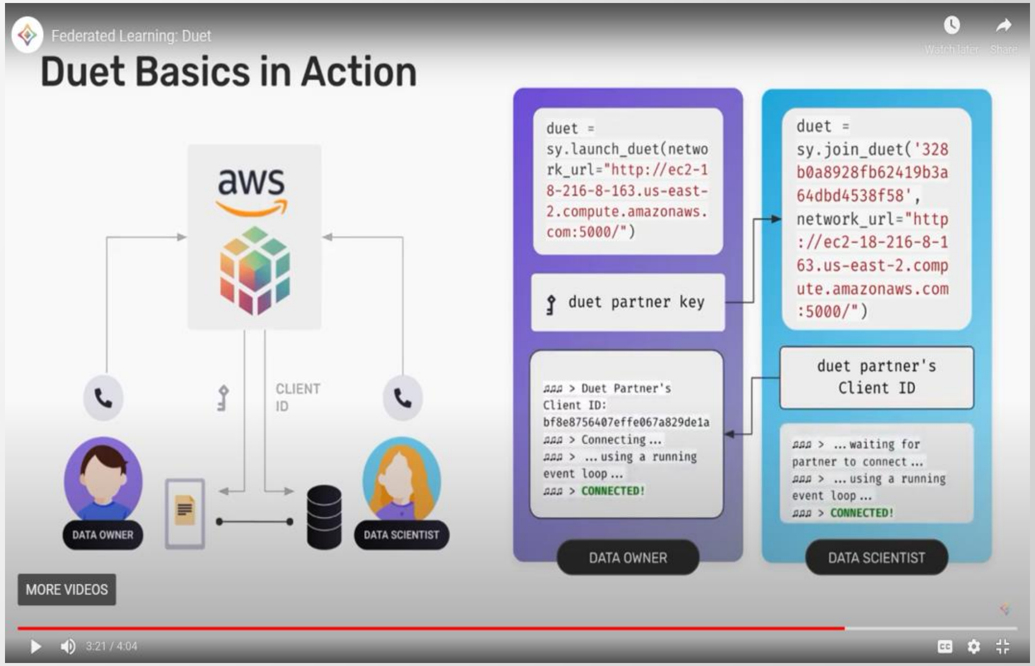
Why Useful?

- ✓ Keeps Data Private
- ✓ Data Owner has Control Over Data
- ✓ Machine Learner Benefits from Access to Distributed Data

Process?

- ✓ PySft Wrapper to ML Package
- ✓ Encryption and Privacy Maintained
- ✓ Machine Learner Can Access Multiple Data Sources Simultaneously
- ✓ Models Trained Remotely and can be Aggregated for Use





Remote Machine Learning –PyTorch/PySft Results

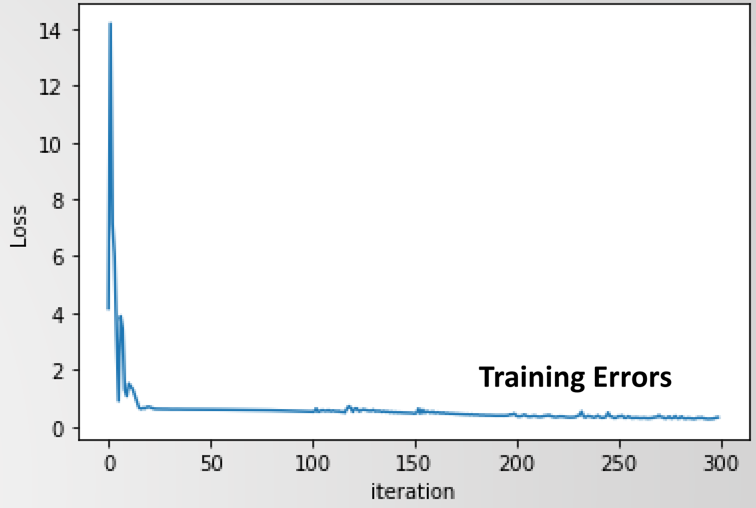
Remote Learning Process:

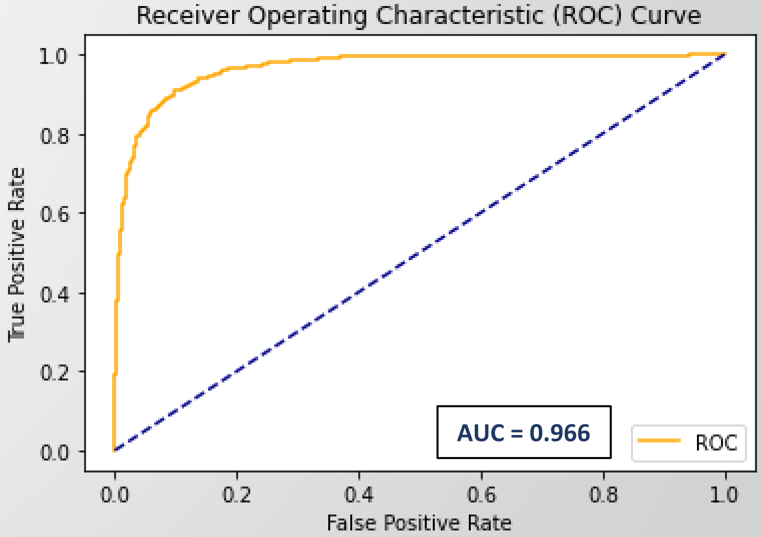
- ✓ Data Owner/Data Scientist interact via PySyft and PyGrid/AWS
- ✓ Data Owner sends data to Data Scientist
- ✓ Data Scientist makes requests via Pysft to Data Owner
- ✓ Data Scientist creates model
- ✓ Data Scientist sends model to Owner
- ✓ Training on Remote Server
- ✓ Model Sent to Data Scientist Once Trained
- ✓ Data Scientist Tests Model – Sckit Learn Packages

PyTorch and PySft:

- ✓ 3 Hidden Layers: 100, 50 and 25 Neurons, Relu Activation
- ✓ 1 Output Layer, 2 Neurons, Log_soft_max Activation
- ✓ 300 Epochs
- ✓ Optimizer: Adam
- ✓ learning_rate = .01
- ✓ nn.functional.nll_loss

PyTorch/ PySft	Class 0 Predicted	Class 1 Predicted
Class 0 Actual	**1,262**	**99**
Class 1 Actual	**97**	**632**

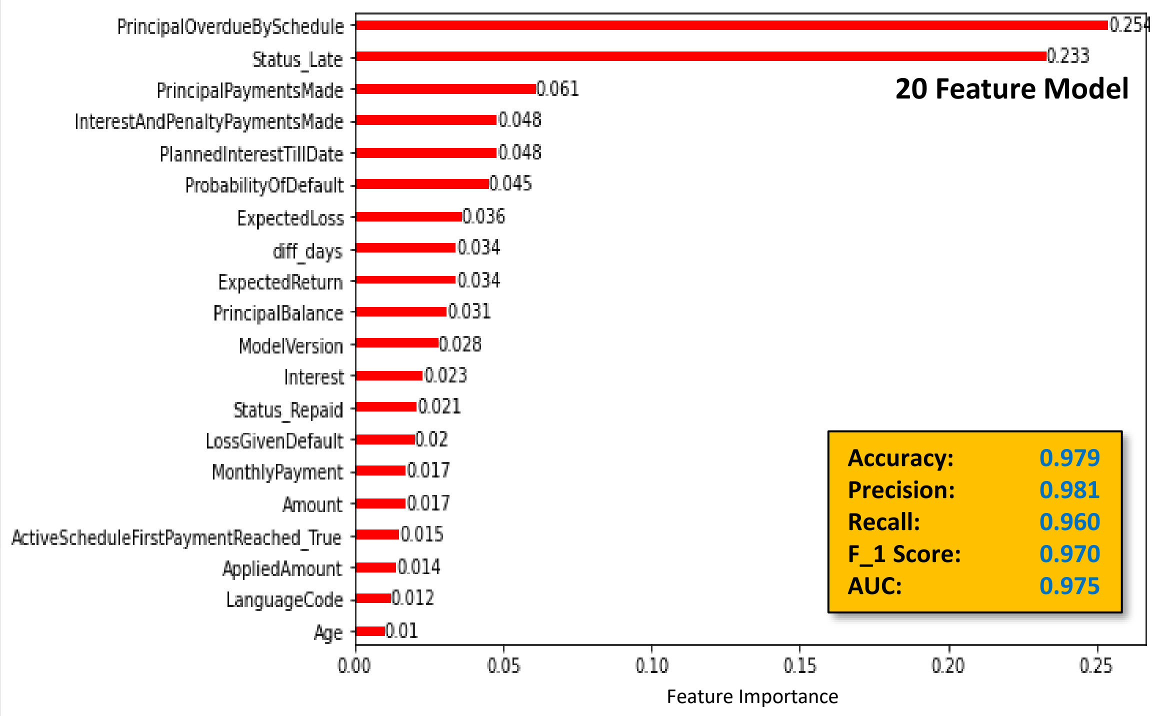




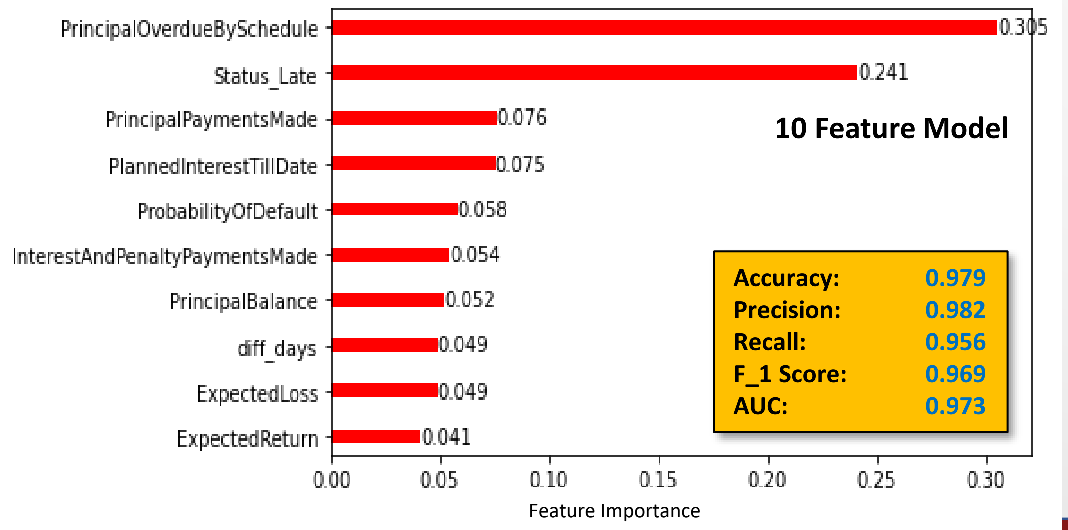
Model Evaluation – Performance Metrics

	Best Hyperparameters	RMSE	Accuracy	Precision	Recall	F_1Score	AUC
Logistic Regression	L1 Penalty, liblinear Solver, C =5	0.209	0.956	0.938	0.936	0.937	0.951
Naïve Bayes	Alpha = 1.0	0.399	0.841	0.789	0.743	0.765	0.818
Decision Tree	Criterion – entropy, Max_depth = 20	0.166	0.973	0.962	0.960	0.961	0.970
Ensemble Forest, Boosts DT of Max_Depth: 1	N_estimators= 196 l_rate = 1.625	0.199	0.960	0.950	0.936	0.943	0.954
Random Forest	N_estimators = 200, Criterion – entropy, Max_ features = sqrt	0.163	0.972	0.976	0.943	0.960	0.966
Neural Net – Keras/Tensorflow	Batch_size = 5, epochs=150, init- glorot_uniform, optimizer= adam	0.249	0.912	0.907	0.986	0.945	0.980
Neural Net - PyTorch	Not Applicable	0.306	0.906	0.865	0.867	0.867	0.966

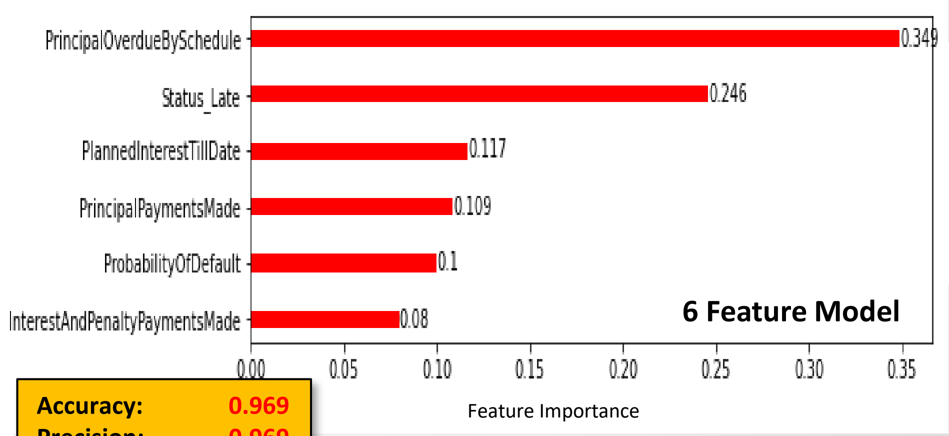
Reduced Features – Random Forest



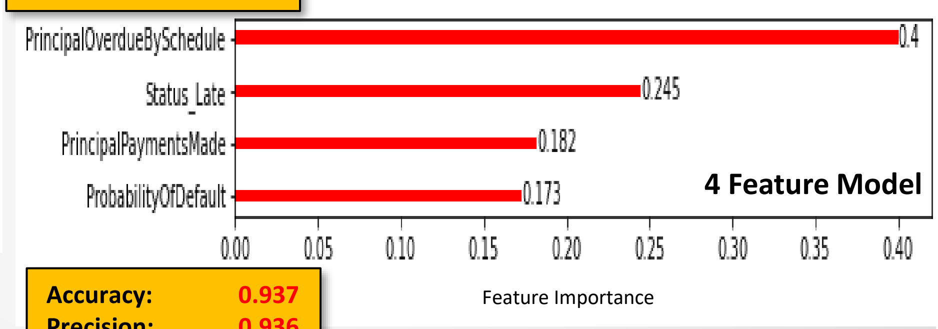
Accuracy:	**0.979**
Precision:	**0.981**
Recall:	**0.960**
F_1 Score:	**0.970**
AUC:	**0.975**



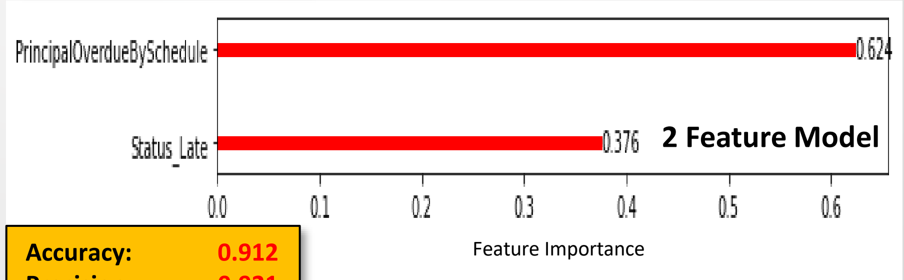
Accuracy:	**0.979**
Precision:	**0.982**
Recall:	**0.956**
F_1 Score:	**0.969**
AUC:	**0.973**



Accuracy:	**0.969**
Precision:	**0.969**
Recall:	**0.941**
F_1 Score:	**0.955**
AUC:	**0.962**



Accuracy:	**0.937**
Precision:	**0.936**
Recall:	**0.879**
F_1 Score:	**0.906**
AUC:	**0.923**



Accuracy:	**0.912**
Precision:	**0.921**
Recall:	**0.819**
F_1 Score:	**0.867**
AUC:	**0.891**

Hyperparameters:

- **Criterion: Entropy**
- **N_estimators: 300**
- **Max_Features: sqrt**

BASE MODEL

Accuracy:	**0.972**
Precision:	**0.976**
Recall:	**0.944**
F_1 Score:	**0.960**
AUC:	**0.966**

Results:

- **No Loss in Prediction Power for 10 & 20 Feature Relative to Base Model**
- **More Interpretable than Base Model**
- **Model Predictability Not as Good for 6, 4 & 2 Feature Models**

Conclusions

All Models, Except Naïve Bayes, Provided Consistent Results – 5-Fold CV

Precision, Accuracy, Recall, and F1_Scores were all above 0.90

Random Forest and Decision Tree had best RMSEs of 0.163/0.166

Neural Nets with Tensorflow/Keras had best AUC of 0.980

- 3-Fold Grid Search CV was Trained on 10 Pct of Dataset as it was Expensive to Train on Full Dataset

Remote (Federated) ML with PyTorch/PySft Provided Good Results

Performance Similar to Other Models

Can be Trained Remotely on Multiple Distributed Systems and Model Results can be Aggregated on Server for Testing

Smaller Feature Set Models Random Forest; Comparison with Base Model – 71 Input Features

- **No Loss in Prediction Power, 20 & 10 Input Feature Models; More Interpretable**
- Predictive Power Less Reliable, 6, 4, & 2 Input Feature Models

Questions