SUPERVISED LEARNING CAPSTONE

PREDICTING LOAN DEFAULTS

BACKGROUND – DATASET SELECTION

- The dataset I chose provides credit related data for loans that were issued in 2018
- The dataset contains 855,969 observations across 73 columns
- Originates from Kaggle and can be found in the following link:

  https://www.kaggle.com/manishpthakur/pythonproject

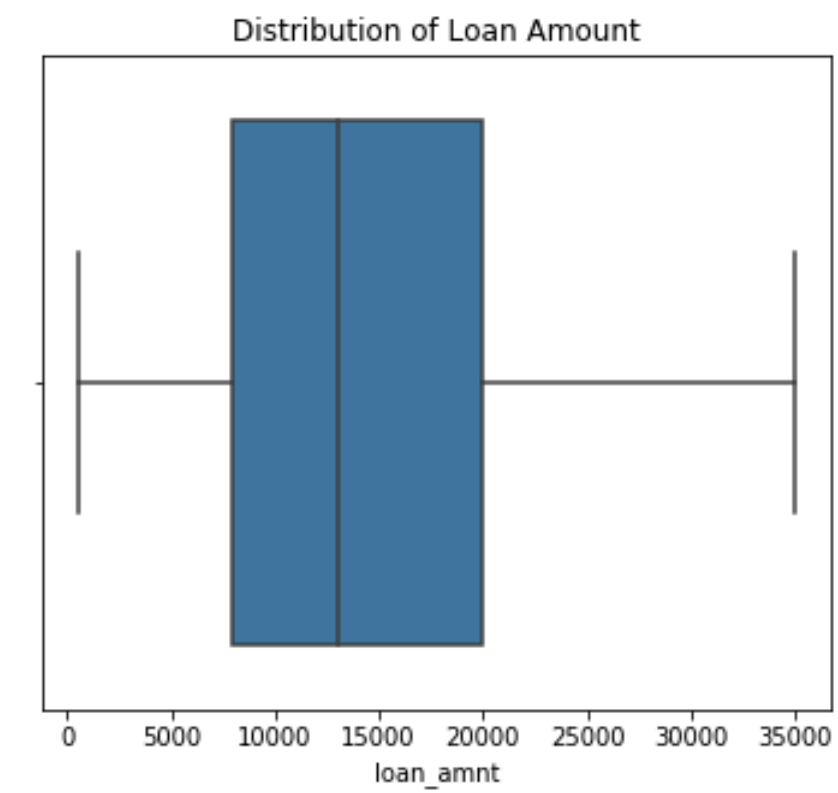
DATA FEATURES & DATA TYPES

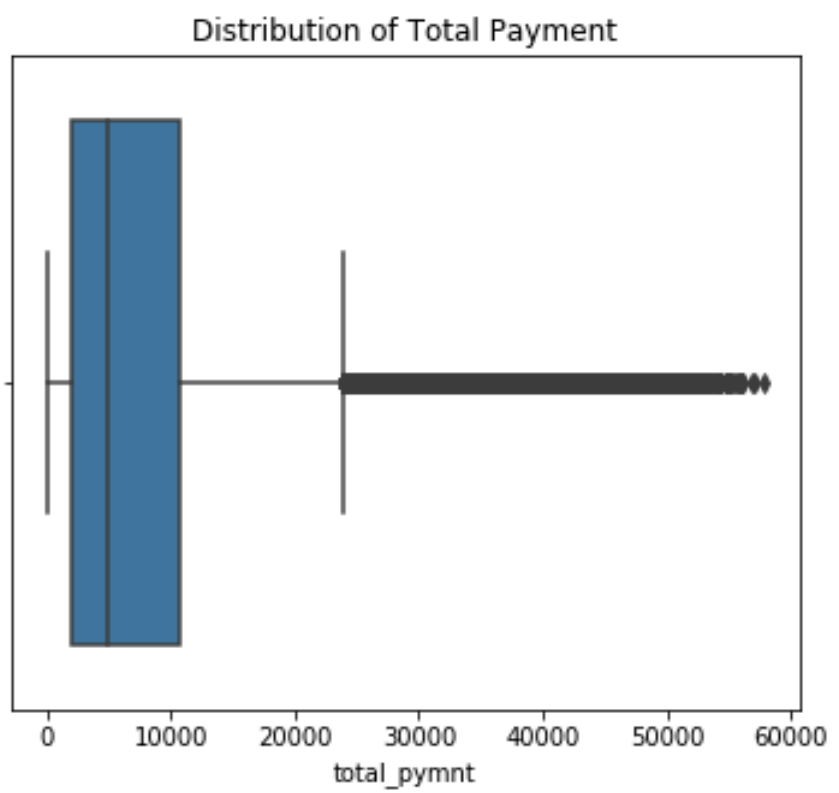
Continuous	Categorical
Loan Amount	Grade
Funded Amount	Sub Grade
Term	Home Owner
Interest Rate	Verification Status
Annual Income	Payment Plan
Issue Date	Purpose
Debt to Income	Zip Code
Delinquent 2 Yrs	State
Accounts Opened	Earliest Credit Line
Revolving Balance	Initial List Status
Total Accounts	Policy Code
Out Principle	Application Type
Out Principle Inv	Account Now Delinquent
Total Payment	Default Indicator
Total Payment Inv	
Total Received Principle	
Total Received Interest	
Total Received Late Fees	
Recoveries	
Collection Recovery Fee	
Last Payment Amount	

PROCESS & METHODOLOGY

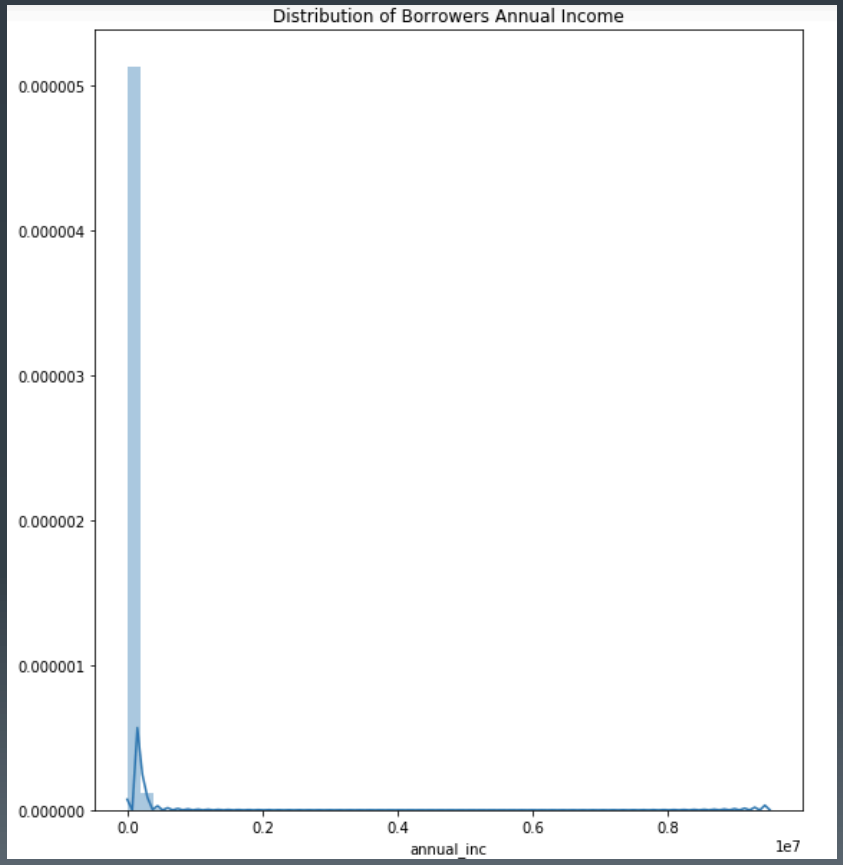
1. Exploratory Analysis – Univariate and Bivariate
2. Data Cleaning
3. Feature Selection & engineering
4. Define Research Question
5. Run Models
    - Using Default Parameters, Tuned Parameters, PCA Components, Rebalance Class, Rebalance Class Tuned, and Revised Feature Set
6. Determine Accuracy on Training Set including Confusion Matrix and Cross Validation
7. Final Model Selection

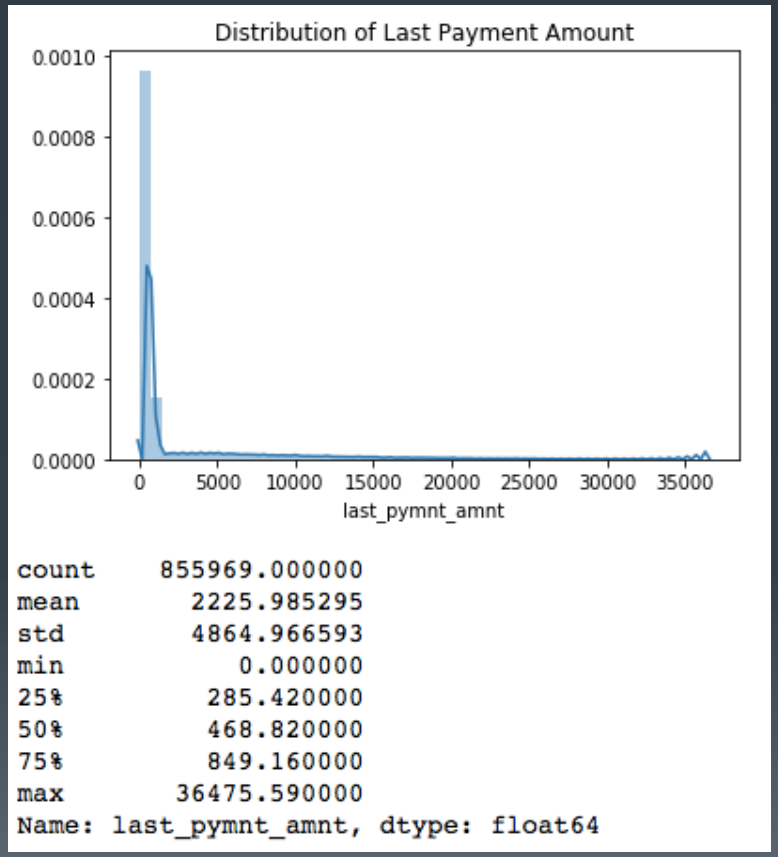
EXPLORATORY ANALYSIS





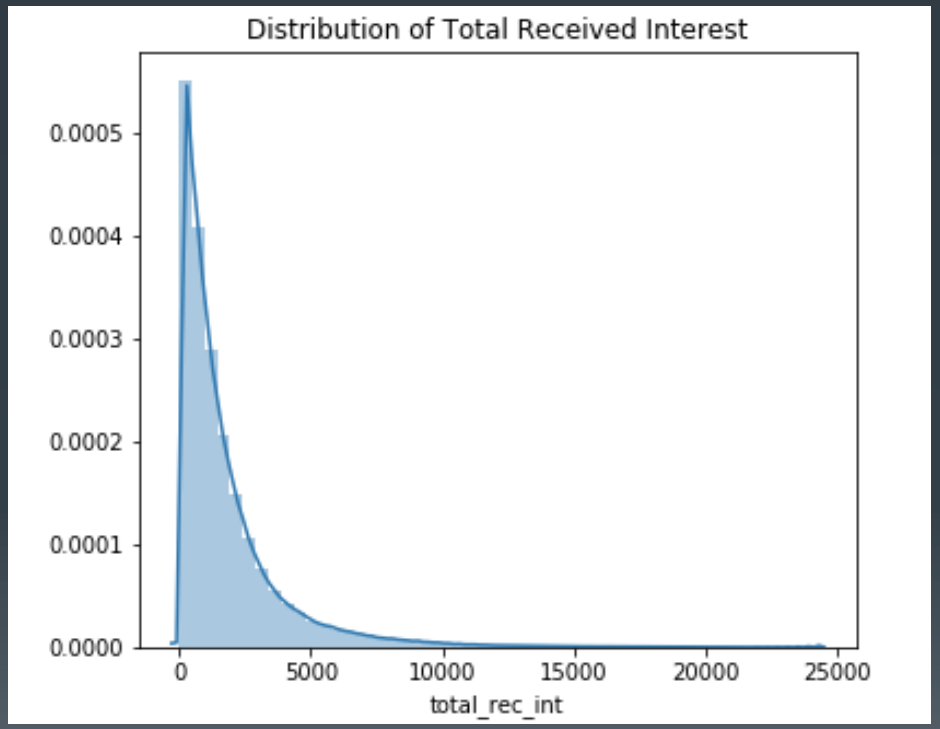
EXPLORATORY ANALYSIS (Continued…)

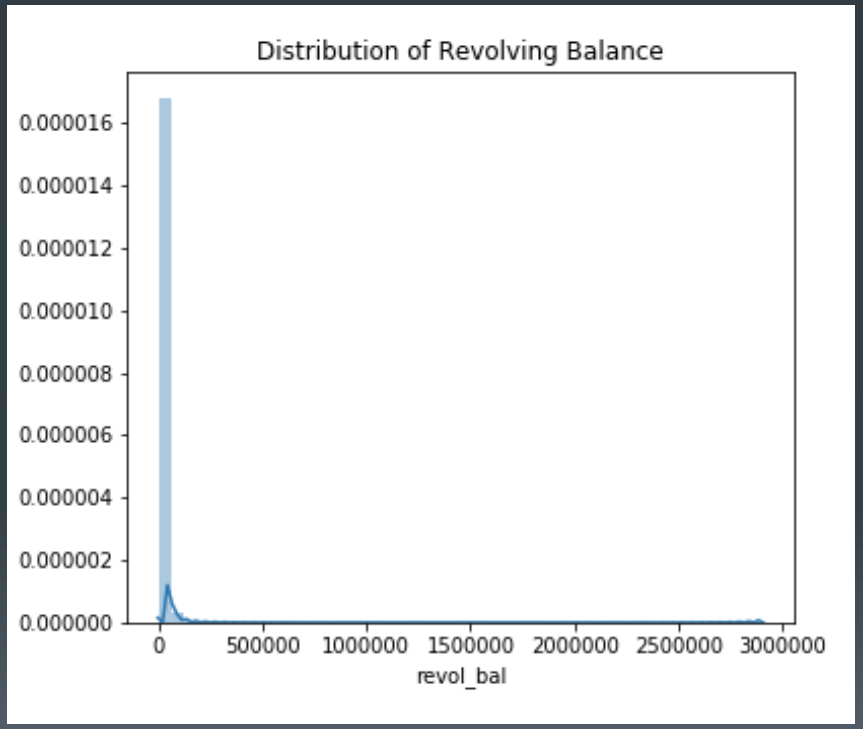




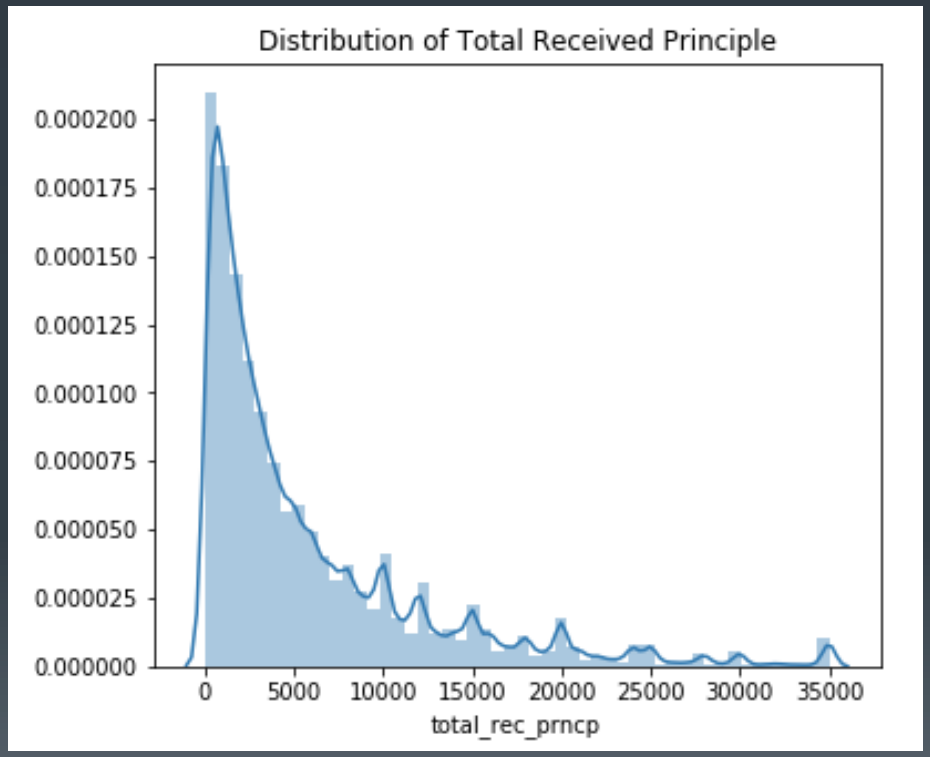

```
count    855969.000000
mean       2225.985295
std        4864.966593
min           0.000000
25%         285.420000
50%         468.820000
75%         849.160000
max       36475.590000
Name: last_pymnt_amnt, dtype: float64
```

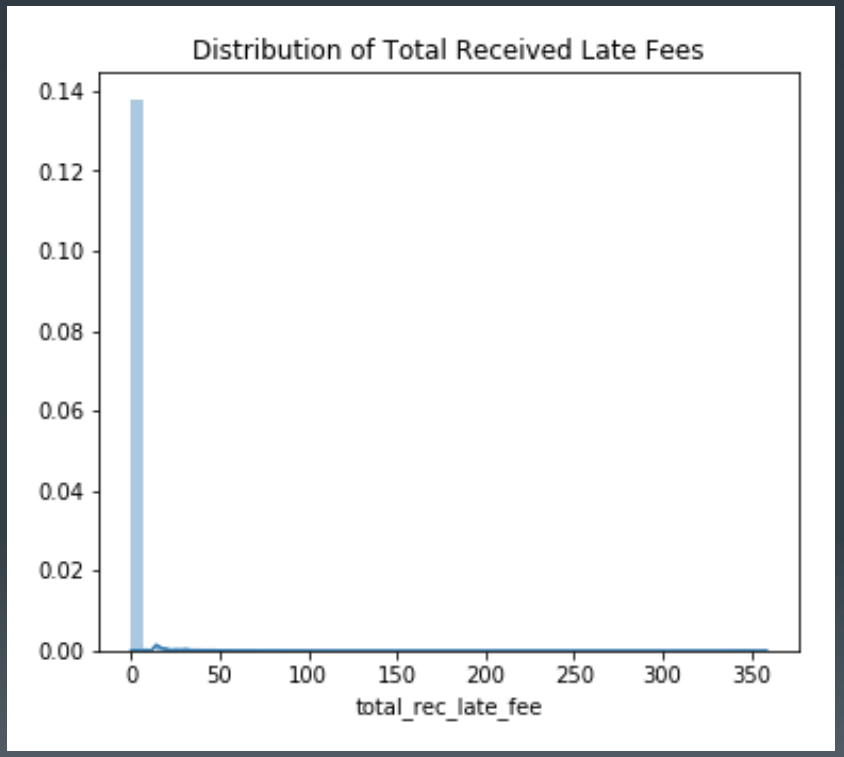
EXPLORATORY ANALYSIS (Continued…)



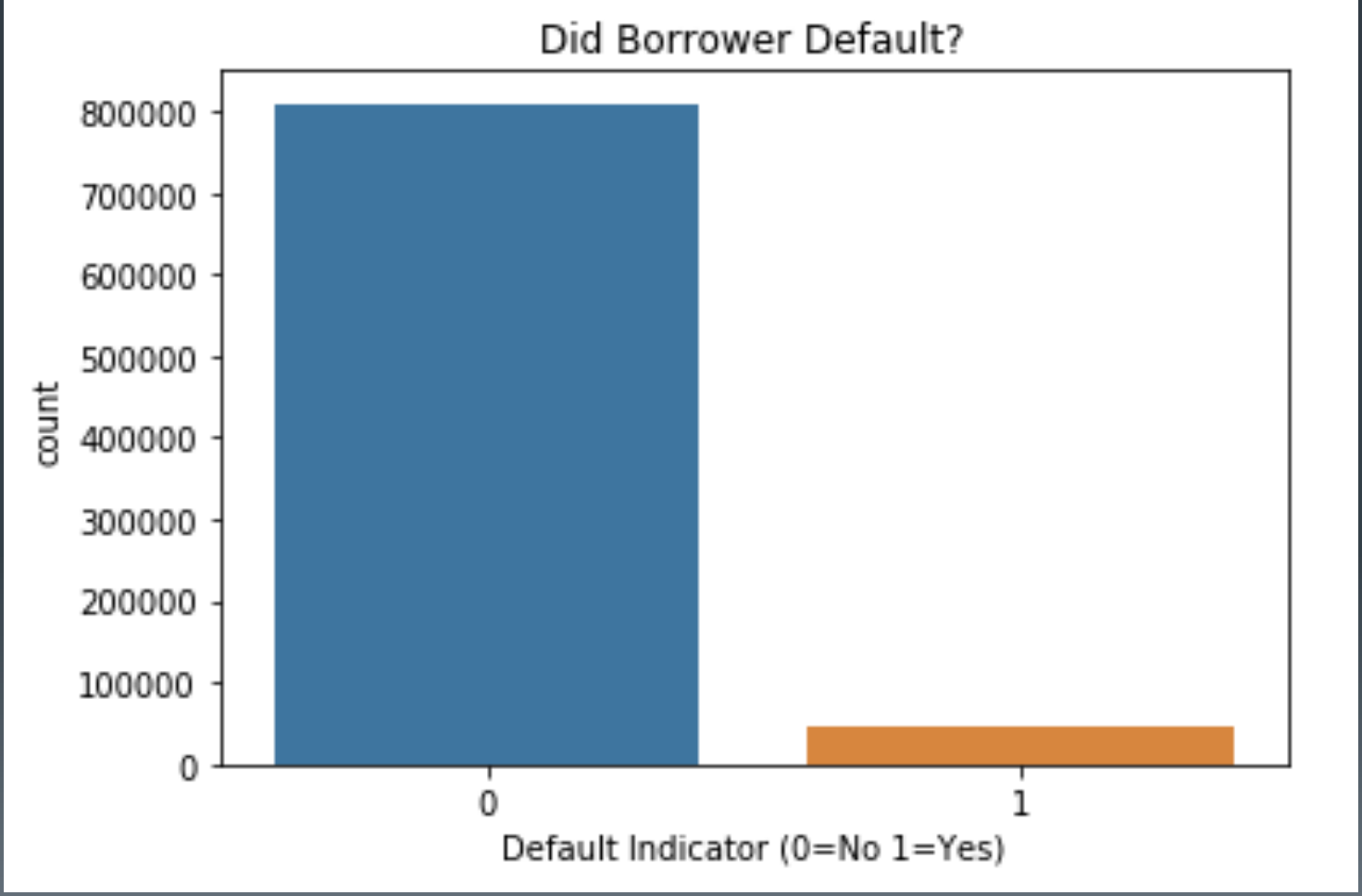


EXPLORATORY ANALYSIS (Continued…)

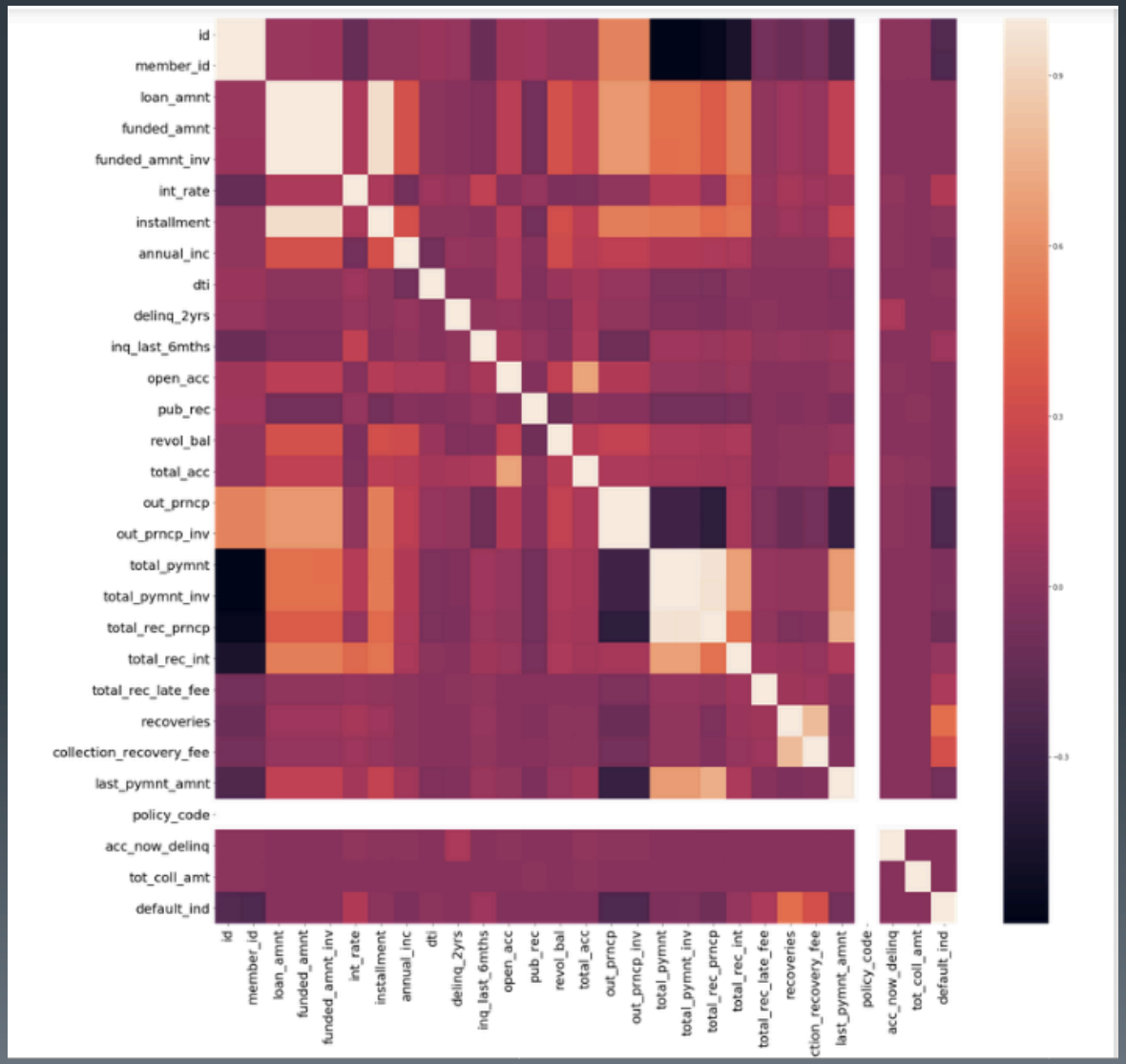




EXPLORATORY ANALYSIS (Continued…)



EXPLORATORY ANALYSIS (Continued…)



RESEARCH QUESTION:

CAN WE PREDICT WHETHER OR NOT A BORROWER OF A LOAN WILL DEFAULT?

FEATURE SELECTION & ENGINEERING

- Original Baseline
  - Class Imbalance – Dominant class is Borrower Not Defaulting at 94.57%
- Revised Baseline
  - Resampled data to create 50/50, balanced dataset of Borrower Defaulting and Borrower Not Defaulting
- Original Feature Set
  - Removed outliers outside the 95% quantile
- Revised Feature Set
  - Last model iterations use this revised feature set

MODEL ITERATIONS

- Model Input
  - 855,969 rows and 73 columns
- Default Parameters
  - For each model, ran the default parameters
- Tuned Parameters
  - Based on iterations of each model type, determine the input values that could best improve performance without overfitting
- Principal Component Analysis (PCA)
  - Determined the number of input components for PCA to account for the most explained variance
- Rebalanced Class
  - Resampled data to create 50/50 balance dataset
- Revised Feature Set
  - Based on median values for continuous variables, created assigned “high” or “low”

NAIVE BAYES CLASSIFIER

Model	Parameter	Training Accuracy	Cross-Validation %	Runtime (seconds)
Default	Bernoulli	97.52%	98.00, 98.05, 97.45, 97.00, 96.79	9.71
Tuned	Gaussian	85.21%	85.20, 86.05, 85,40, 84.70, 84.84	8.79
PCA	Bernoulli	94.77%	94.55, 94.55, 94.70, 94.65, 94.55	0.48
PCA Tuned	Gaussian	94.69%	94.60, 94.70, 94.70, 94.50, 94.70	0.15
Rebalanced Class	Bernoulli	97.07%	79.16, 100.0, 79.16, 87.50, 81.81	0.06
Rebalanced Tuned	Bernoulli	90.30%	90.54, 90,04, 88.50, 89.94, 90.45	0.68
Rebalanced Revised Feat.	Bernoulli	93.96%	83.33, 95.83, 91.66, 86.36, 81.81	0.13

K-NEAREST NEIGHBOR CLASSIFIER

Model	Parameter	Training Accuracy	Cross-Validation %	Runtime (seconds)
Default	K=5, Weights = 'Uniform'	98.05%	97.75, 97.05, 97.40, 96.90, 97.09	54.13
Tuned	K=100, Weights = 'Distance'	100.0%	94.10, 94.10, 94.10, 94.10, 94.14	54.37
PCA	K=5, Weights = 'Uniform'	94.69%	94.70, 94.70, 94.70, 94.70, 94.70	0.14
PCA Tuned	K=100, Weights = 'Distance'	100.0%	94.70, 94.70, 94.70, 94.70, 94.70	1.91
Rebalanced Class	K=5, Weights = 'Uniform'	96.27%	54.16, 66.66, 66.66, 66.66, 72.72	0.09
Rebalanced Tuned	K=5, Weights = 'Uniform'	100.0%	93.53, 93.53, 94.00, 94.47, 94.47	2.35
Rebalanced Revised Feat.	K=5, Weights = 'Uniform'	75.86%	62.50, 75.00, 54.16, 72.72, 77.27	0.10

RANDOM FOREST CLASSIFIER

Model	Parameter	Training Accuracy	Cross-Validation %	Runtime (secs)
Default	n_estimators = 10 Criterion = 'gini' Max_features = 'auto' Max_depth = None	100.0%	99.95, 99.85, 99.75, 99.90, 99.84	4.37
Tuned	n_estimators = 50 Criterion = entropy' Max_features = 'auto' Max_depth = None	99.99%	98.35, 98.30, 98.30, 97.95, 97.99	18.26
PCA	n_estimators = 10 Criterion = 'gini' Max_features = 'auto' Max_depth = None	98.72%	94.65, 94.65, 94.65, 94.60, 94.45	3.26
PCA Tuned	n_estimators = 50 Criterion = entropy' Max_features = 'auto' Max_depth = None	99.87%	94.65, 94.70, 94,70, 94.70, 94.65	25.37
Rebalanced Class	n_estimators = 50 Criterion = entropy' Max_features = 'auto' Max_depth = None	99.15%	87.50, 87.50, 87.50, 87.50, 77.27	0.32
Rebalanced Tuned	n_estimators = 50 Criterion = entropy' Max_features = 'auto' Max_depth = None	100.0%	96.01, 95.52, 97.00, 96.98, 97.98	2.35
Rebalanced Revised Feat.	n_estimators = 50 Criterion = entropy' Max_features = 'auto' Max_depth = None	99.13%	91.66, 91.66, 75.00, 72.72, 90.90	0.32

LOGISTIC REGRESSION

Model	Parameter	Training Accuracy	Cross-Validation %	Runtime (secs)
Default	N/A	98.68%	97.90, 97.55, 97.25, 97.40, 97.74	6.46
PCA	N/A	94.69%	50.45, 94.70, 94.70, 94.70, 94.70	0.14
Rebalanced Class	N/A	95.76%	95.83, 100.0, 95.83, 91.66, 90.90	0.09
Rebalanced Revised Feat.	N/A	96.55%	87.50, 91.66, 91.66, 95.45, 90.90	0.10

LASSO LOGISTIC REGRESSION

Model	Parameter	Training Accuracy	Cross-Validation %	Runtime (secs)
Default	Penalty = 'l1', C=1.0	99.95%	99.85, 99.95, 99.85, 99.90, 99.89	27.4
Tuned	Penalty = 'l1', C=0.2	99.61%	99.60, 99.25, 99.60, 99.50, 99.69	18.26
PCA	Penalty = 'l1', C=1.0	94.69%	94.70, 94.70, 94.70, 94.70, 94.70	88.36
PCA Tuned	Penalty = 'l1', C=1.0	94.69%	94.70, 94.70, 94.70, 94.70, 94.70	92.74
Rebalanced Class	Penalty = 'l1', C=1.0	100.0%	91.66, 100.0, 87.50, 91.66, 81.81	1.87
Rebalanced Tuned	Penalty = 'l1', C=0.5	99.4%	99.50, 98.50, 98.50, 100.0, 99.49	2.49
Rebalanced Revised Feat.	Penalty = 'l1', C=1.0	100.0%	87.50, 95.83, 95,83, 90.90, 90.90	0.14

RIDGE LOGISTIC REGRESSION

Model	Parameter	Training Accuracy	Cross-Validation %	Runtime (secs)
Default	Penalty = 'l2', C=1.0	98.88%	99.25, 98.65, 97.25, 97.45, 99.24	6.8
Tuned	Penalty = 'l2', C=5000	99.29%	99.35, 99.15, 99.45, 98.15, 99.55	4.93
PCA	Penalty = 'l2', C=1.0	94.69%	50.45, 94.70, 94.70, 94.70, 94.70	88.36
PCA Tuned	Penalty = 'l2', C=1.0	94.69%	50.45, 94.70, 94.70, 67.45, 46.85	0.74
Rebalanced Class	Penalty = 'l2', C=1.0	95.76%	95.83, 100.0, 100.0, 91.66, 90.90	0.16
Rebalanced Tuned	Penalty = 'l2', C=1.0	99.10%	99.50, 98.00, 98.00, 100.0, 98.99	0.57
Rebalanced Revised Feat.	Penalty = 'l2', C=1.0	96.55%	95.83, 91.66, 91.66, 95.45, 90.90	0.14

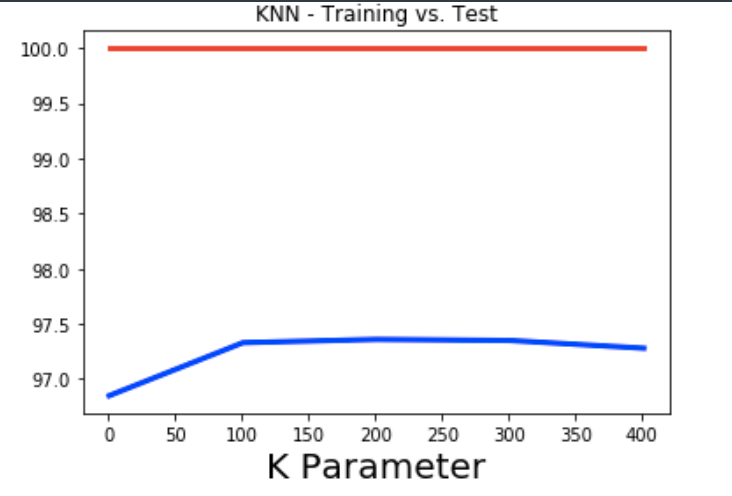
SUPPORT VECTOR REGRESSION

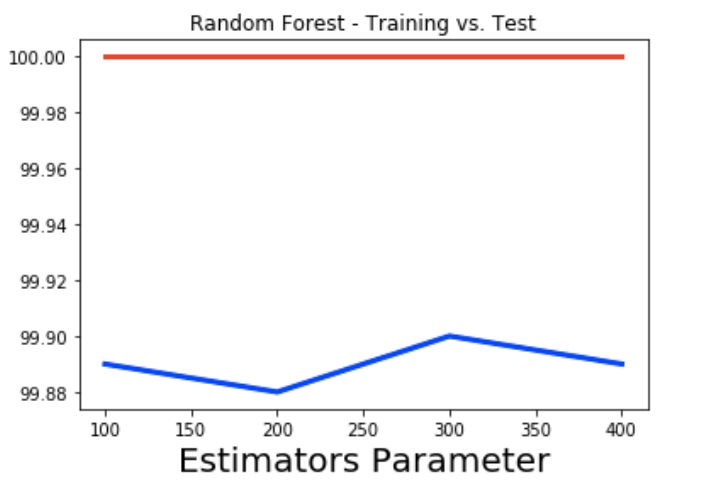
Model	Parameter	Training Accuracy	Cross-Validation %	Runtime (secs)
Default	C=1.0, kernel='rbf'	100.0%	94.65, 94.65, 94.65, 94.65, 94.69	4193.02
Tuned	C=1.0, kernel='linear'	99.29%	99.35, 99.15, 99.45, 94.65, 94.69	4.93
PCA	C=1.0, kernel='rbf'	100.0%	94.70, 94.70, 94.70, 94.70, 94.70	196.12
Rebalanced Revised Feat.	C=1.0, kernel='rbf'	100.0%	50.00, 50.00, 50.00, 50.00, 50.00	0.16

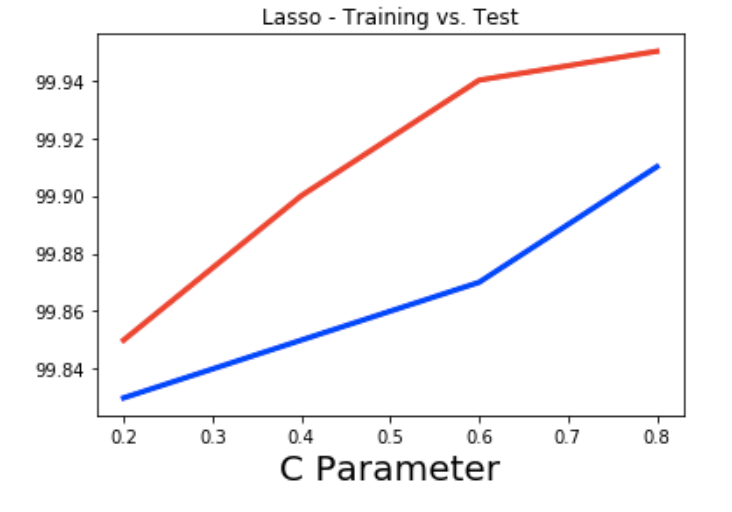
GRADIENT BOOSTING CLASSIFIER

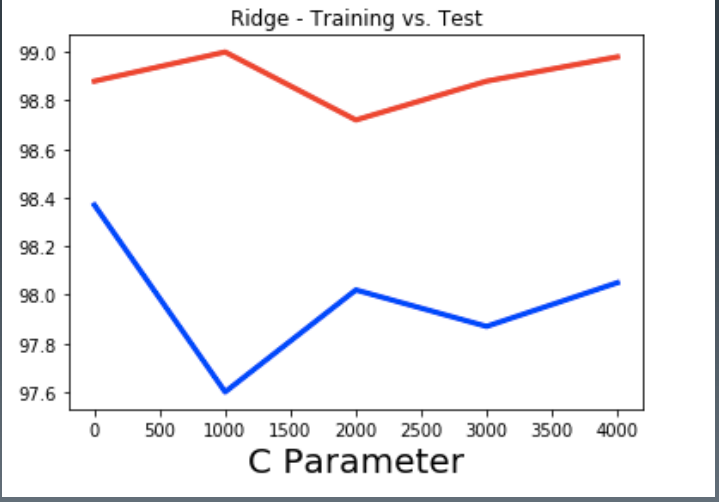
Model	Parameter	Training Accuracy	Cross-Validation %	Runtime (secs)
Default	n_estimators = 100 max_depth = 3 loss='deviance' Subsample=1.0	100.0%	100.0, 99.95, 99.95, 99.90, 99.94	169.84
Tuned	n_estimators = 300 max_depth = 3 loss='deviance' Subsample=0.8	100.0%	98.00, 95.00, 97.00, 97.00, 93.46	42.41
PCA	n_estimators = 100 max_depth = 3 loss='deviance' Subsample=1.0	94.89%	94.50, 94.65, 94.60, 94.55, 94.55	8.57
PCA Tuned	n_estimators = 100 max_depth = 3 loss='deviance' Subsample=1.0	100.0%	99.00, 96.00, 97,50, 96.00, 99.49	44.85
Rebalanced Class	n_estimators = 100 max_depth = 3 loss='deviance'	100.0%	91.66, 100.0, 87.50, 91,66, 81.81	1.59
Rebalanced Tuned	n_estimators = 100 max_depth = 3 loss='deviance'	100.0%	96.51, 96.01, 99.00, 97.48, 99.49	37.24
Rebalanced Revised Feat.	n_estimators = 100 max_depth = 3 loss='deviance'	100.0%	91.66, 95.83, 95.83, 100.0, 90.90	1.54

TRAINING VS. TEST

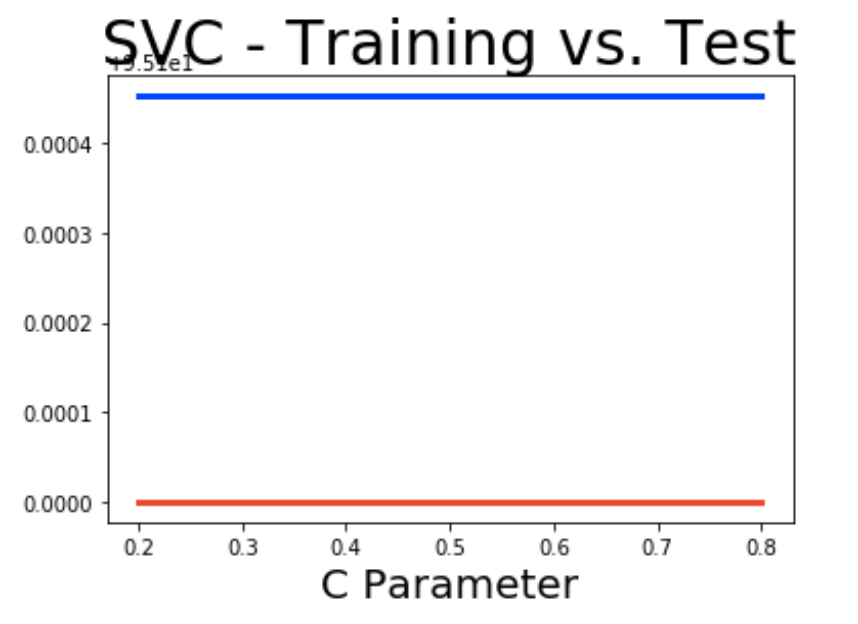


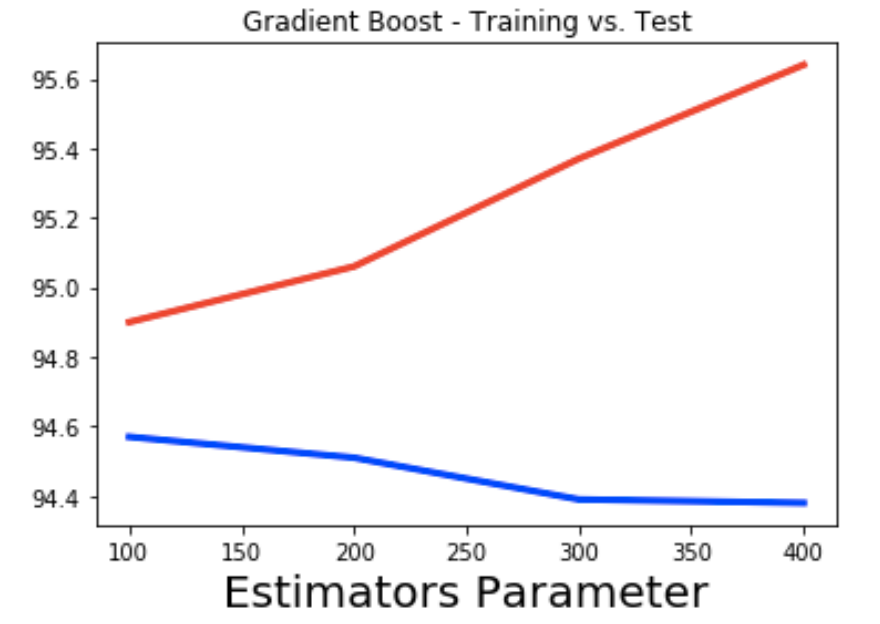






TRAINING VS. TEST





FINAL ANALYSIS & MODEL SELECTION

- All models seem to have very high accuracy training accuracy
  - Gradient Boost model seemed to have the highest overall training accuracy across more models
- Support Vector Regression had a significant drop in Cross Validation % (from mid/ upper 90's to 50) for running the model with rebalanced class and revised features
- There mostly weren't much difference between results before class imbalance was addressed and afterwards except for a few exceptions
  - K-Nearest Neighbor Classifier dropped in Training Accuracy significantly (Ranging from mid 90s to 100 % down to 75%)
  - Lasso Logistic Regression also dropped with regular and revised features
- Tuned models don't seem to have much difference with the non-tuned models
- Support Vector and Gradient Boost models required significantly more processing time
- Various types of Logistic Regression (regular, Lasso, Ridge) produced similar results – don't seem any significant difference

PRACTICAL USE OF THE MODEL

- More accurate prediction on loan defaults can have impact to the financial institutions, financial markets, and borrowers
  - Financial institutions:
    - Determine more suitable loan rates and terms for less risky to more risky borrowers
    - Increase revenue and decrease losses
    - Turn down borrowers with excessive risk
  - Financial markets:
    - Avoid large inter-dependent institutions defaulting impacting stability of the broader market
  - Borrowers:
    - Borrowers understand their default risk to not be offered loans too risky which they can't pay back and get into deeper financial problems

CHALLENGES & SHORTCOMINGS

- Data was only for 2018 and weren't any indication on source of data
- Size of data with large amount of rows and columns impacted exploration and processing time
- Original dataset had many columns that had data that were only sometimes populated due to nature of data
  - Had to evaluate each of those columns in detail to determine validity - whether it was poor data quality or expected, then its applicability to the tests
  - Determined they were okay to remove – no impact to experiment
- Significant performance issues with running Support Vector models
  - Either not enough processing power or data size and complexity of model resulted in extremely long processing time or processes would hang and never complete.
  - Had to shut down processes, restart computer, and rerun steps many times