### **Data Exploration:**

### Looking at the training data, there are 614 rows and 13 columns, including the target variable Loan Status. Of the 12 other variables, 3 are numeric and continuous. The distribution of each can be seen above in both boxplots and histograms. Each are largely right skewed. These columns were standardized, and the testing data was standardized based on the training set's means and standard deviations to ensure the transformation of the data remained consistent between train and test set.

### Some values initially appeared numeric, but fell into discrete buckets. These include Dependents, Credit History, and Loan Amount Term. Dependents and Credit History were treated as categorical as they had 4 and 2 possible values respectively. Loan Amount Term was tested both as a categorical and numeric continuous variable, since there were so many buckets present. However, this did not noticeably affect any of the models so the variable was kept numeric since some buckets existed in the training set but not the test set (and therefore produced different columns when one hot encoded in each set).

### Five columns were purely categorical and one hot encoded for analysis. Loan ID was dropped as it is unique to each row of data. The target variable, Loan Status, was converted from Yes/No to a binary.

### There was missing data scattered pretty randomly throughout both the training and test sets of data. For columns with less than 3% missing data, the mode (categorical) and median (numeric) was used to impute values. Self Employed and Credit History were 5 and 8 percent missing, so instead of mode I chose to create a separate category for the missing data as to not greatly disrupt the distribution of the categories in each column. I also tested both imputation and adding an additional missing category label, and found no noticeable difference in the model performance statistics.

### In the numeric variables, there was a pretty strong correlation between Loan Amount and Application Income. This violates the multicollinearity requirement of logistic regression, however in thinking about the use case of this model we are really prioritizing performance rather than if models violate any assumptions. With this in consideration I will still consider Logistic Regression models.

### **Hypothesis:**

### Based on the data, I expect a Random Forest model to perform best. Random Forest is sometimes treated as a baseline model, to be created initially for the performance statistics. Then usually more complex models are tested and try to outperform the Random Forest. This is why I expect an outperformance, and I will be testing the following models to see if this is the case and get the best predictions possible out of the data:

### Logistic Regression

### Random Forest

### XGBoost

### SGD Classifier

### Linear SVC

### Dense Neural Network

### **Analysis:**

1. **Analyze and explain the output of your model.**

From the basic logistic regression model chosen, predictions were outputed that match the training dataset in the format of 'Y' for yes and 'N' for no. These values predict whether the test set of clients should be approved for a loan, based off the data given. finalpredictions is an array with only the Loan Status values, and test\_with\_preds is the entire test dataset with the addition of the Loan Status predictions added in.

1. **Why did you use that model?**

Based on all the performance statistics of the above models, I chose to go with the basic logistic regression model. The majority of the models were giving almost the exact same statistics (accuracy, performance, recall). So if all models are performing the same, it's generally better to go with the less complex models. This is because they are more interpretable and easily explained and analyzed. With the logistic regression model, we can use L1 and L2 to see what features are most important, and use that to draw insights about the data.

1. **What could you do to improve your model?**

If given more time, there are lots of model tuning steps that can be taken. While Random Search and Grid Searches are very useful, they tend to take time. It would be ideal to use these more exhaustive searches on each model, to find the very best parameters.

I would also dive a bit into feature creation. Some areas that could be explored include squaring or combining the numeric features with each other, or changing around the buckets for the categorical variables. Both of these could potentially increase the performance of the models.

For the missing values, I would also explore more complex methods of imputation to see if they improve performance. Rather than more simple methods that I used, more complex models like KNN or even Linear/Logistic Regressions could be explored.

1. **What next steps would you take to?**

The next steps with this process would include taking all the improvements listed in part 2, and conducting them to see if model performance improves at all.

Then, whatever model and features chosen can be interpreted for their other value, for example knowing that Applicant Income is one of the most important factors in whether or not a loan is approved, or other information like that.

In order to present findings, I would create a presentation with many more visualizations, key takeaways, and easily interpretable results so they can be communicated outside of the Technology Team and actions can start to be taken off what is learned.