In order to identify which supervised and unsupervised learning techniques are best suited to classify the Instacart Data, it is important to first identify that which is being classified. In this case, there is a binary classification consisting of products that were reordered and those products that were not reordered by each individual user. There are many different algorithms that can be used for binary classification so further inspection of the data is necessary.

Initially, when dealing with a binary classification problem the first algorithm I think of working with is Logistic Regression, also known as logit. The predictions made by logit are discrete meaning they output specific values or multiple categories. Logit, although having regression in its name, does not have to output continuous variables and can use these discrete values as it's classifier. Binary classification only deals with a classification of two categories either True or False, 1 or 0. Specifically, the two categories within this data set are 'reordered' (1) and 'not reordered' (0). What's more is that logit can output probabilities that can be used with a threshold to alter the classification based on the data. These probabilities are given by use of a sigmoid function which in short, expresses log-odds of an observation using a linear function. There are many assumptions to be taken into account when using logit or things can go wrong. The first assumption to inspect is if some of the input variables are highly correlated. This would cause some coefficients to become extremely large, reaching infinity, have the wrong sign and/or eventually overfit. Therefore it is important to check with a pairwise correlation plot to eliminate such input variables. The next assumption to consider is assuming a linear relationship with the input variables and the output of the model, the log odds. This negatively influences the constant effect that the input variable has on the likelihood that one of the outcome categories will occur. Lastly, only relevant independent variables that have zero errors or noise should be chosen. In this case feature importance would mean identifying those variables that have the most impact on our output. In other words, the number given to an anonymous user should not be included within the input variables because the arbitrary number given has no influence on whether or not they will reorder a product. After the features are chosen and are confirmed to be relevant and meeting all of the conditions for creating a logit model, hyper parameters will be tuned to allow the model to fit properly and to generalize to unseen data more effectively. The two most important parameters to tune within this logit model will be the regularization method used and 'C.' C is the inverse of regularization strength which takes a positive value. The smaller this value is, the stronger regularization occurs; the larger this value is the less regularization occurs meaning that it can be generalized to fit unseen data. Within real world applications such as this data, a large C value will be used from 10,000 to 10,000,000. Another regularization technique used to create a generalized model includes choosing the penalty term between the regularization. There are two kinds of techniques used, Lasso and Ridge Regression. Each, using it's different penalty terms, provides a way to make the model fit properly. When the proper parameters are used, you will not end up with a model that under fits or overfits. It is like a sweet spot in the middle. Evaluation methods for these models will be discussed later.

The next algorithm to be used is a favorite among Kaggle competitors and uses tree based learning for classification. It is called Light Gradient Boosting (LGBM) and is far quicker than similar algorithms which is something that needs to be considered due to the size of the data and computational power required. It is a unique utilization of tree based learning; LGBM grows trees leaf-wise, an order in which the vertical expansion is first. It chooses the leaf's split based on global loss rather than just using the loss along a particular branch. This is one reason why it is often the faster of these models. Using this method LGBM focuses on the accuracy of results using the least amount of memory and will work great with this very large dataset. It can be used for both classification and regression but will again be used for this binary classification. Unlike logit there are fewer assumptions to be made about the data before creating the model. The first assumption is that it requires a data set with more than approximately 10,000 observations. Lastly, it is important just as in logit that the two categories of the output variables are of balanced weight. In order to ensure the target variable is balanced SMOTE will be used on the training and test data to ensure that the categories are equal in proportion. SMOTE, known as Synthetic Minority Oversampling Technique, is used to increase the number of the smaller categories, in this case "reorders" within the data set by keeping the variance of the dataset balanced as the size increases. Both algorithms have similar default ways of balancing out the target data. However, in order to maintain balance and control, SMOTE will be performed before splitting the data for cross validation. There are a number of hyperparameters to be tuned in an LGBM model, over a hundred, but the following are the most important parameters to consider. The most important parameter out of the hundred plus is 'application.' Application specifies the utilization of your model, will the outcome be a classification, multiclass classification or regression? The next in importance for LGBM is the metric used for specifying loss during model building. Because this data is dealing with a binary classification, our parameter will hence be binary classification. To handle overfitting, max depth can be decreased and is an influential factor in model building. Another unique feature of this algorithm is its ability to accept categorical variables. In logit, mean target encoding will be used to incorporate department and aisle information within the model. With LGBM, categorical features only have to be identified in the parameters and LGBM will use it without any encoding needed by the user. Lastly, early_stopping_round and save_binary can be used to increase the speed of your model building. Early stopping will stop the model if a metric hasn't improved in a determined amount of rounds and save_binary increases speed of data read time and memory by saving the dataset to binary file.

The evaluation methods for both algorithms are vast but because Kaggle is using the F1 score to review the final results, this evaluation metric will be used along with AUC-ROC Curve. Area under the curve (AUC) and Receiver Operating Characteristics are one of the most used evaluation metrics for classification models because of its visualized classification thresholds. These two thresholds are the True Positive Rate TPR and the False Positive Rate FPR. Before plotting these parameters however, I usually use a confusion matrix first to gather all true and

false parameters. In order to discuss with lay people, visualizations would work best and therefore the graph will be plotted. Overall, a good model has an AUC closer to 1 which means it has a good measure of separability between the two categories of classification. When the model has a low TPR and high FPR it means that the specificity, or precision, is low and sensitivity, or recall is also low. To find the model's best performance, one will try to maximize the TPR and minimize FPR. The golden median in which the best optimization lies is found by tuning the model's hyperparameters to deliver the best results.

The F1 score also deals with the precision and recall within classification projects. This score is the weighted average of precision and recall. In order to find the best F1 score, a threshold of a determined probability can help us increase the amount of True positives given by our model.

Once all of the features are created the next step is to identify their importance and correlation with other created features. Next, assumptions are determined to be fulfilled for each algorithm as the data is preprocessed and the target variable is balanced for model fitting. After using a grid search to determine the correct hyperparameters to use, the model fitting will begin and different iterations of the model will be used to determine the best F1 score and AUC-ROC curve. These evaluation metrics will delineate how well the performance of our model can classify the data, in this case, what products will be ordered in the next purchase for user 'j' in their next purchase?