# Summary of Anonymous Data Analysis

## Introduction

**Problem statement**: Given a training data set and test data set (no target), find the probability that a row is classified as 1 (vs 0) and use ROC AUC to score models. Create two models and train, save, and retrieve models to apply to testing data to mimic production. Submit a csv file with probabilities that the row belonged to class 1.

## Data

- Training data consisted of 100 features, 1 target (classified as 0 or 1) and 40,000 rows.
- Testing data consisted of 100 features and 10,000 rows.
- Features included currency values, percentages, other numerical values, and categorical values.
- In the training data, here are the number of 0 and 1 target values. From here, we can see that the positive class (1) is not rare, at over 20 percent of the training data, but the data is imbalanced, which is why the ROC AUC metric is appropriate.

  0 31856
  1 8144

### Data Cleaning

- Removed percent signs and converted percentages to decimals.
- Removed currency signs.
- Replaced missing values with most frequent for categorical features and mean for numerical features.
- Converted all categorical values to lowercase.
- After analyzing values for categorical columns, changed values to uniform strings and fixed spellings.
- Used one hot encoding to encode categorical data.
- After encoding, features totaled 122.

## Method

I investigated various methods using the training data and the ROC AUC metric. Using 25 percent of the training data, I tested K Nearest Neighbor, Logistic Regression, Decision Tree Classifier, and Random Forest Classifier and scored models using the ROC AUC metric. Data is scaled using normalization (MinMaxScaler) because algorithms that fit a model that use a weighted sum of input variables (Logistic Regression) and algorithms that use a distance (K Nearest Neighbor) behave best when data is normalized.

### K Nearest Neighbor

Using grid search and visualization, I determined that the best value for k to maximize the ROC AUC score was 27 and the best metric was Manhattan, which is best for high dimensional data.
ROC AUC: 0.6722995899448072
Accuracy score: 0.864125

### Logistic Regression

ROC AUC: 0.7830404594791912
Accuracy score: 0.8895
ROC AUC Cross Val Score Mean: 0.906238884984572
ROC AUC Cross Val Score std: 0.003952097777607287

### Decision Tree Classifier

Using grid search cross validation, I determined that the best criterion was entropy and the best minimum samples split was 114.
ROC AUC: 0.7076502653261465
Accuracy score: 0.836375

### Random Forest Classifier

Using grid search and visualization, the max depth of 20 and estimators set to 30 gives the best results.
ROC AUC: 0.712065366017884
Accuracy score: 0.88125

### Voting Classifier

Using KNN, Logistic Regression, Decision Tree Classifier, and Random Forest Classifier with parameters that were fine tuned in the previous model testing, I used the Voting Classifier to see if this ensemble model would produce better results.
Roc auc: 0.7547016344298747
Accuracy score: 0.897125

## Conclusion

Since the Logistic Regression and Voting Classifier models gave the highest ROC AUC scores, I chose these two models to implement. In the final code, both the training data and testing data are cleaned. Next, both models are created using the parameters from the model testing and fitted with all of the training data. Each model is then saved to separate files. Models are then

retrieved from files and used to predict probabilities for the test data. The probabilities are saved to csv files - one for each model.

## Logistic Regression

Pros: Low cost with short run time; few parameters to fine tune.
Cons: Simple; may not be helpful for more complex cases.

## Voting Classifier

Pros: Using different types of models, this ensemble model is powerful and good for complex cases.
Cons: Higher cost with long run time; more parameters to fine tune with multiple models to use in the Voting Classifier.