**Summary**

Our recommendation is <best model> with <data set>. The classification accuracy was <best accuracy>.

**Our Process**

The objective is to suggest/recommend Machine Learning models that would best classify the outcome ‘y.’ Our process was as follows:

1. Clean Data
2. Feature Select
3. Model
4. Analyze
5. Evaluate

**Clean Data**

The raw data set was unlabeled. It contained a total of 160,000 records and 51 features. Using a Python library called ‘Pandas\_Profiling’, we uncovered the following:

1. Datatypes
   1. Numeric: 45
   2. Categorical: 5
   3. Boolean: 1
2. High Correlation
   1. x41 – Dropped x41 from dataset
   2. x6 – Dropped x6 from dataset
3. High Cardinality
   1. x37 – The values had incorrect datatype – causing an incorrect assesessment of high cardinality. To clean it, we removed the commas and dollar sign, and replaced the parenthesis with a minus symbol. Converted datatype to numeric.
4. Data augmentation
   1. Misspellings were present in x24. Corrected ‘euorpe’ to ‘europe’
   2. Standardize values
      1. x29 – Replaced all month abbreviations to full spelling (i.e. Aug to August)
      2. x30 – Corrected misspelling of ‘thurday’ to ‘thursday’
   3. NaN
      1. Filled missing data with ‘other’ value
5. Data Distribution was normal. No data transformation was required.

**Feature Select**

This portion took a bulk of our time. The process of feature selection was a continuous cycle between ‘feature select’ and ‘modeling.’ The first pass was to use the complete dataset with each subsequent iteration removing features and/or records.

Because this was a classification assignment, we selected a few classification models to find the best model for the data set (refer to ‘Model’ section for more details). In the end, we decided that all the categorical features made little to no impact on the accuracy scores – therefore, dropped. Using Random Forest, we found the following features to be of importance:

**Model**

Logistic Regression

Decision Tree

K-NN

SVM

Naïve Bayes

**Analyze**

**Evaluate**