COMPANY X
BINARY CLASSIFIER
MACHINE LEARNING ASSESSMENT

SUMMARY

After a thorough cleansing and analysis of the provided data, I created several machine learning predictive models. As per the provided instructions, I used the Area Under the Curve (AOC) as an error metric.

The best performing model was a Neural Network Classifier. The second best performing model was a Random Forest Classifier. Enclosed is a Jupyter Notebook file (in html format) containing all Python sourcecode, as well as the predicted probabilities for the test dataset from both of the top-performing models (in csv format).

NEURAL NETWORKS CLASSIFIER

This model utilizes a neural network with two hidden layers, each with 32 nodes.

Positives
- Greatly improved accuracy
- Decreased aoc variance

Negatives
- Difficult to understand/interpret
- Requires extensive computing power
- **Could be difficult to deploy in real-world scenario**

RANDOM FOREST CLASSIFIER

This model implements a random forest algorithm of randomized decision trees.

Positives
- Requires less computing power
- Easier to replicate and tune
- **Easier to deploy in real-world scenario**

Negatives
- Still complicated to understand/interpret
- Increased error compared to Neural Network model

REAL-WORLD APPLICATION

This project differed significantly from a real-world business scenario, in several ways that could directly impact the effectiveness of these models.

Domain Knowledge: In a real-world situation, I would have more understanding of the business domain this dataset represents. This would help significantly with feature selection, knowing which features could potentially leak information about the target and which features are irrelevant to the overall project.

Feature Engineering: Typically, engineering new features can be a powerful way to optimized machine learning models. Without an understanding of what the features and target class represents, this is impossible.

Computing Power: In a real-world scenario, I could take advantage of enterprise-level compute power, especially with distributed systems such as Apache Spark. These models were run on my local machine, and much more experimentation could have been done with greater computational capacity.

Error Metric Selection: Typically part of the machine learning process is selecting an appropriate error metric. In this case the metric was prescribed. Knowing the domain could help select a more appropriate metric, depending on whether false-positives or false-negatives are more consequential.

eddie@kdsolutions.co | 404-644-6068 | kdsolutions.co