# Optimizing an ML Pipeline in Azure

**## Overview**

This project is part of the Udacity Azure ML Nanodegree.

In this project, we build and optimize an Azure ML pipeline using the Python SDK and a provided Scikit-learn model. This model is then compared to an Azure AutoML run.

**## Summary**

\*\*In 1-2 sentences, explain the problem statement: e.g "This dataset contains data about... we seek to predict..."\*\*

This dataset contains customers data of bank it contains variables such as age, job,marital education,default,housing,loan etc. based on these variables y has to be predicted . y is a categorical variable therefore a classification algorithm is implemented.

\*\*In 1-2 sentences, explain the solution: e.g. "The best performing model was a ..."\*\*

Logistic Regression model has been used the hyperparameters tuned are

C = Regurlarization parameter

max\_iter= maximum iterrations

It is observed that at a C=1 and max\_iter=10 we fetched the best accuracy of nearly 90.5 %

**## Scikit-learn Pipeline**

\*\*Explain the pipeline architecture, including data, hyperparameter tuning, and classification algorithm.\*\*

Built a pipeline to perform preprocessing of raw data. The Pipeline includes following steps :

1.Fetching data from hyperlink and transfom to Tabular format.

2.Cleansed dataset using preprocessing techniques by dropping the null values, replacing the values with numbers and get dummy values for all text variables.

3.Splitted the dataset to x\_train and x\_test.

4. Logistic Regression model has been implemented using Sklearn and tuned hyperparameters C

and number of iterations using HyperConfig in Azure ML.

5. Used BanditPolicy which is a early termination policy helps in reaching a nearly accurate solution based on a slack criteria based on a frequency and delay interval.

6. RandomParameter Sampling defines a random sample over a search space. The computation of Random Parameter Sampling is less than Grid Sampling as it reaches fast.

5. Extracted the best model using get\_best\_run\_by\_primary\_metric()

6. Saved model using joblib as pickle file.

**\*\*What are the benefits of the parameter sampler you chose?\*\***

1. Random sampling supports discrete and continuous hyperparameters.

It supports early termination of low-performance runs.

**\*\*What are the benefits of the early stopping policy you chose?\*\***

1. Bandit policy is based on slack factor/slack amount and evaluation interval.

Bandit terminates when the primary metric is not within the specified slack factor/slack amount compared to the best performing run which reduces the computation time and helps in searching in better search spaces.

**## AutoML**

\*\*In 1-2 sentences, describe the model and hyperparameters generated by AutoML.\*\*

VotingEnsemble was the best model as per AutoML it provided an accuracy of nearly 91.5 %

**## Pipeline comparison**

\*\*Compare the two models and their performance. What are the differences in accuracy? In architecture? If there was a difference, why do you think there was one?\*\*

Logistic Regression was having accuracy of 90.5 % and Auto ML model provides an accuracy of 91.5 %.

Voting Ensemple model has done better than Logistic Regression as it is a Emsemble model it handles bias error better.

**## Future work**

\*\*What are some areas of improvement for future experiments? Why might these improvements help the model?\*\*

A more complex model can improve the performance of model. Some more feature engineering data transformation can be done to improve the model performance.

## Proof of cluster clean up

\*\*If you did not delete your compute cluster in the code, please complete this section. Otherwise, delete this section.\*\*

Compute\_name.delete() method is used to delete the resources

\*\*Image of cluster marked for deletion\*\*

