**Enterprise II Screenshots – Group 7**

Contents

[**Azure Functions to fetch weather data using Open Weather API** 2](#_Toc101567788)

[Function scheduled to trigger at CRON schedule: “0 5 4 \* \* \*” 2](#_Toc101567789)

[Python file that runs with the function trigger, and calls the Open Weather API to fetch weather data. 2](#_Toc101567790)

[Requirements file containing the required PyPi packages required to run. 3](#_Toc101567791)

[**Azure Storage to store the model selected using TPOT** 3](#_Toc101567792)

[**Azure PostgreSQL storing weather data, test data, model results** 3](#_Toc101567793)

[Table containing weather data generated using function trigger 3](#_Toc101567794)

[Table containing model results 4](#_Toc101567795)

[**Hyperparameter Optimization** 4](#_Toc101567796)

[1. Using Hyperopt 4](#_Toc101567797)

[2. Using optuna 5](#_Toc101567798)

[**Auto ML** 5](#_Toc101567799)

[1. TPOT 5](#_Toc101567800)

[2. H2o.ai 6](#_Toc101567801)

[**Synthetic Data Generation (used as Test Data)** 6](#_Toc101567802)

# **Azure Functions to fetch weather data using Open Weather API**

## Function scheduled to trigger at CRON schedule: “0 5 4 \* \* \*”

This means 4:05 AM UTC every day, or 12:05 AM EST every day.

Graphical user interface, text, application, email

Description automatically generated

## Python file that runs with the function trigger, and calls the Open Weather API to fetch weather data.

Graphical user interface, text, application

Description automatically generated

## Requirements file containing the required PyPi packages required to run.

Graphical user interface, text, application, email

Description automatically generated

# **Azure Storage to store the model selected using TPOT**

Graphical user interface, text, application, email

Description automatically generated

# **Azure PostgreSQL storing weather data, test data, model results**

## Table containing weather data generated using function trigger

In the below weather data, the “updated at” column shows the datetime when the function ran and the weather data last got updated.

**A screenshot of a computer

Description automatically generated**

## Table containing model results

A picture containing graphical user interface

Description automatically generated

# **Hyperparameter Optimization**

1. Using Hyperopt

Using the below space of parameters, we ran Bayesian optimization with hyperopt on Random Forest Classifier.

Text

Description automatically generated

1. Using optuna

Optuna allowed us to further widen the space for hyper parameters for tuning.

Text

Description automatically generated

# **Auto ML**

1. TPOT

Text, application

Description automatically generated

Using TPOT to find best estimator pipeline based on training data.

Due to limited computational resource, limiting the estimator time to 30 mins and using the best estimator recommended within the stipulated run.

1. H2o.ai

Text

Description automatically generated

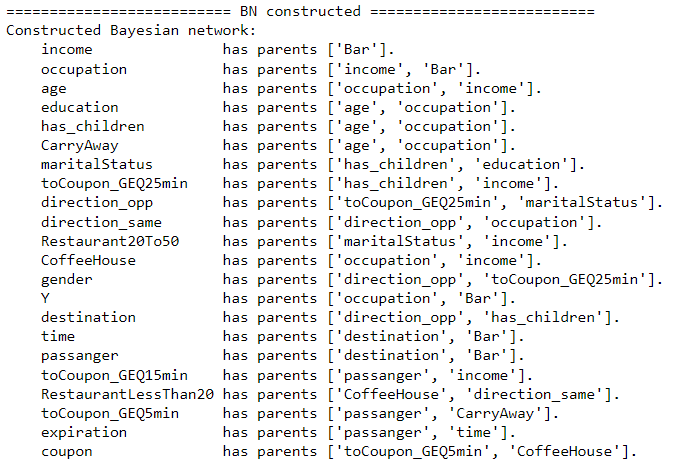
Training H2O on training data to find the best possible estimator.

Running it to get top 10 models, skipping stacked ensemble algos to reduce computational resource strain.

# **Synthetic Data Generation (used as Test Data)**

Following is the Bayesian Network created for the generation of the synthetic data by the describer object with the following hyperparameter:

1. thresholdthreshold\_value = 30 (An attribute is categorical if its domain size is less than this)
2. epsilon = 1 (A parameter in Differential Privacy i.e., noise addition in the synthetic data)
3. degree\_of\_bayesian\_network = 2 (The maximum number of parents in Bayesian network, i.e., the maximum number of incoming edges)
4. num\_tuples\_to\_generate = 21000 (Number of tuples generated in synthetic dataset)



Following is the pairwise comparison between Original and Synthetic dataset generated i.e., pairwise correlation analysis to replicate the original dataset in correlation mode by the Bayesian Network

