UDACITY

Return to Classroom

Optimizing an ML Pipeline in Azure

REVIEW

CODE REVIEW

HISTORY

Meets Specifications

Greetings Learner,

Well done! Good work on covering the requirements and passing the project.

Best part of your submission

- Readme file is well documentation on required details and comparative analysis on both experiments are provided.
- Model experiment runs executed without error and computes target deleted after experiments.
- Correct use of Configuration parameters in HyperDrive and Automl.

Regards

Additional References

- Microsoft Azure Well-Architected Framework
  The Azure Well-Architected Framework is a set of guiding tenets that can be used to improve the quality of a workload. The framework consists of five pillars of architecture excellence: Cost Optimization, Operational Excellence, Performance Efficiency, Reliability, and Security.
- Good reference from Azure on Architectures Reference

Documentation

The README contains an explanation of:

- The pipeline architecture, including data, hyperparameter tuning, and classification algorithm.
- The benefits of the chosen parameter sampler.
- The benefits of the chosen early stopping policy.

Good work. You have updated all required information. Pictorial representation of architecture will be very helpful in understanding the components involved.

The pipeline architecture, including data, hyperparameter tuning, and classification algorithm. ✅

Summary

The dataset Bank Marketing dataset, contains about 32,950 history of distinct customer rows and 21 features of those customers including class label. We seek to predict a customer will default on loan payment or not.

We seek to create and optimize an ML Pipeline using this dataset by:

- Optimize custom model with HyperDrive
- Optimize using Automated machine learning(AutoML)

And Compare the results of the 2 methods

Creating and Optimizing an ML Pipeline

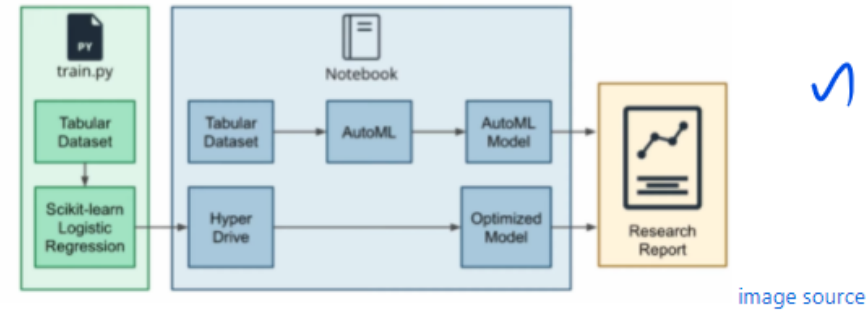

image source from udacity ML Engineer

Azure

In both AUC and Accuracy, AutoML performed better than HyperDrive.

Additional References

- Good reference from Azure on Architectures Reference

The README contains:

- One or more sentences describing the model and parameters generated by AutoML.
- Two or more sentences comparing the two models and their performance.

Good work. You have identified the model selected by AutoML and described it.

- One or more sentences describing the model and parameters generated by AutoML. ✅
- Two or more sentences comparing the two models and their performance. ✅
- You have compared the models generated with two experiments based on accuracy metrics.

Pipeline comparison

Accuracy

- HyperDrive bankmarketing_model version: 2 Regularization Strength: : 0.00551584512091742 Max iterations: : 100 Accuracy : 0.9151239251390997 AUC : 0.9269997696302712
- AutoML automl_bankmarketing_model version: 1 Training context : Auto ML AUC : 0.9475641425618313 Accuracy : 0.9176631259484067

The performace of AutoML is better for both AUC and Accuracy metrics.

Architecture

AutoML Architecture is simpler when compared with HyperDrive.

- In HyperDrive, I have to do some Feature Engineering, Hyperparameter selection, Training and Tuning, but these steps are Automated with AutoML.
- with AutoML, you can create hundreds of models a day,can get better model accuracy and deploy models faster.

Differences

- Reason for difference between AutoML and HyperDrive Architecture is that you need to fix several parameters in HyperDrive, while that is AutoML is already automated for several Machine learning algorithms.

The README contains two or more sentences explaining potential improvements for a future experiment and why these improvements might improve the model.

Well done, you have added ideas on how to improve the experiment in the future

However, you can elaborate on these steps and how Azure Configuration setup can help as well

---

Additional References

- For AutoML , you can select on few model types to experiment

Training Pipeline and AutoML

All specifiable parameters of the training script are specified in the hyperdrive config.

Good work. You have utilized parameter sampler and also early stopping framework for the HyperDrive experiments

A hyperdrive config is used and includes:

- A parameter sampler
- A policy for early stopping

Well done, You have correctly aligned the problem type and used the correct metric for evaluation. Good work on utilizing the parameters for Hyperdrive confiq.

Additional References

- Hyperparameter tune a Keras model using HyperDrive

The hyperdrive run is passed to the *RunDetails* widget.

`.get_best_run_by_primary_metric()` is used on the hyperdrive run to retrieve the best run.

Good work. You have provided the correct steps for registering the model after retrieving the best model

```
# Get the best run, and its metrics and arguments
best_run = run.get_best_run_by_primary_metric()
best_run_metrics = best_run.get_metrics()
script_arguments = best_run.get_details() ['runDefinition']['arguments']
print('Best Run Id: ', best_run.id)
#print(' -AUC:', best_run_metrics['AUC'])
print(' -Accuracy:', best_run_metrics['Accuracy'])
print(' -Arguments:',script_arguments)
```

```
{'run_id': 'HD_8ab725d0-b67d-49dc-96d8-d0bb6fff0d75_4', 'hyperparameters': '{"--C": 0.00551584512091742, "--max_iter": 100}', 'best_primary_metric': 0.9
151239251390997, 'status': 'Completed'}
{'run_id': 'HD_8ab725d0-b67d-49dc-96d8-d0bb6fff0d75_0', 'hyperparameters': '{"--C": 0.00970210967937277, "--max_iter": 100}', 'best_primary_metric': 0.9
150227617602428, 'status': 'Completed'}
{'run_id': 'HD_8ab725d0-b67d-49dc-96d8-d0bb6fff0d75_5', 'hyperparameters': '{"--C": 0.00696169463961035, "--max_iter": 130}', 'best_primary_metric': 0.9
147192716236723, 'status': 'Completed'}
{'run_id': 'HD_8ab725d0-b67d-49dc-96d8-d0bb6fff0d75_2', 'hyperparameters': '{"--C": 0.009713527785810062, "--max_iter": 70}', 'best_primary_metric': 0.9
145169448659585, 'status': 'Completed'}
{'run_id': 'HD_8ab725d0-b67d-49dc-96d8-d0bb6fff0d75_1', 'hyperparameters': '{"--C": 0.0051698545135988065, "--max_iter": 70}', 'best_primary_metric': 0.
9136064744562469, 'status': 'Completed'}
{'run_id': 'HD_8ab725d0-b67d-49dc-96d8-d0bb6fff0d75_3', 'hyperparameters': '{"--C": 0.0071010297796242645, "--max_iter": 100}', 'best_primary_metric':
0.9133029843196763, 'status': 'Completed'}
Best Run Id:  HD_8ab725d0-b67d-49dc-96d8-d0bb6fff0d75_4
 -Accuracy: 0.9151239251390997
 -Arguments: ['--C', '0.00551584512091742', '--max_iter', '100']
```

The solution notebook includes an AutoML config, which contains the following parameters:

- `task`
- `primary_metric`
- `experiment_timeout_minutes`
- `training_data`
- `label_column_name`
- `n_cross_validations`

Good work, All required parameters for AutoML config utilized for model experiments.

Infrastructure

A compute cluster is created using the Azure SDK and the `ComputeTarget` and `AmlCompute` objects.

Correctly done. You have allotted compute cluster resources for Experiments.

Additional References

- There are various compute targets available in Azure , which can be used based on requirements

A `TabularDatasetFactory` is used to create a dataset from the provided link.

Well done. Data imported correctly from the provided URL

The `delete` method of the `AmlCompute` object is used to remove the cluster following training.

OR

An image of the compute cluster being selected for deletion is included in the README.

Good work. You have performed the delete compute step in the notebook.

It is important to free up the compute targets in cloud to release the resources as well as to avoid unwanted cost

DOWNLOAD PROJECT

RETURN TO PATH

Rate this review

START