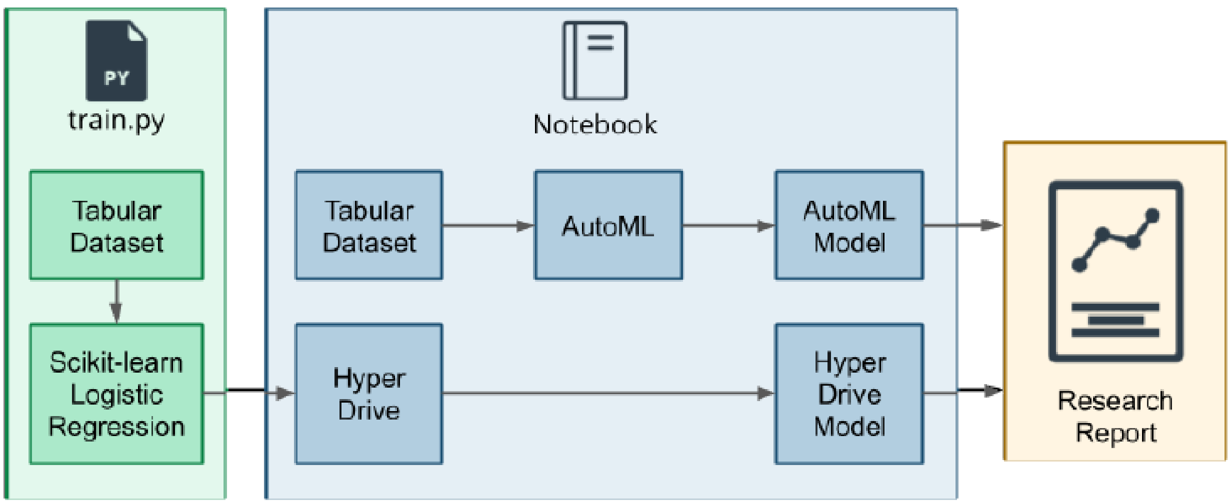
Optimizing an ML Pipeline in Azure
Overview
This project is part of the Udacity Azure ML Nanodegree. In this project, we build and optimize an Azure ML pipeline using the Python SDK and a provided Scikit-learn model. This model is then compared to an Azure AutoML run.

Summary
The data is related with direct marketing campaigns of a Portuguese banking institution. The marketing campaigns were based on phone calls. Often, more than one contact to the same client was required, in order to access if the product (bank term deposit) would be ('yes') or not ('no') subscribed. The classification goal is to predict if the client will subscribe (yes/no) a term deposit (variable y). The best performing model is an ensemble model VotingEnsemble produced by the AutomML run. It has an accuracy rate of 91.60% whereas it is 90.5% incase of HyperDrive assisted Scikit-learn LogicRegression model.

Scikit-learn Pipeline
The main steps and architecture is shown in below diagram.



The pipeline consists of a training script (train.py), a dataset downloaded from Portuguese banking institution, a Scikit-learn Logistic Regression, a HyperDrive for optimizing the hyper parameters. A compute instance is created and a Jupyter Notebook is used to run the training script.
Benefits of the parameter sampler chosen The random parameter sampler for HyperDrive supports discrete and continuous hyper parameters, as well as early termination of low-performance runs. It is Simple to use, eliminates bias and increases the accuracy of the model. Benefits of the early stopping policy chosen, the early termination policy BanditPolicy for HyperDrive automatically terminates poorly performing runs and improves computational efficiency.

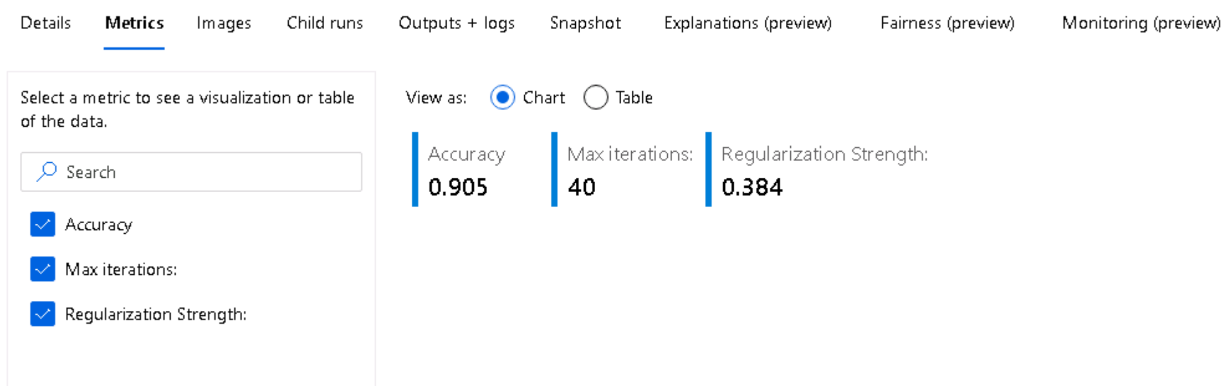
AutoML
The AutoML run was executed with below AutoMLConfig settings:

```
automl_config = AutoMLConfig(
experiment_timeout_minutes=30,
task='classification',
primary_metric='accuracy',
training_data=x,
label_column_name='y',
n_cross_validations=2)
```

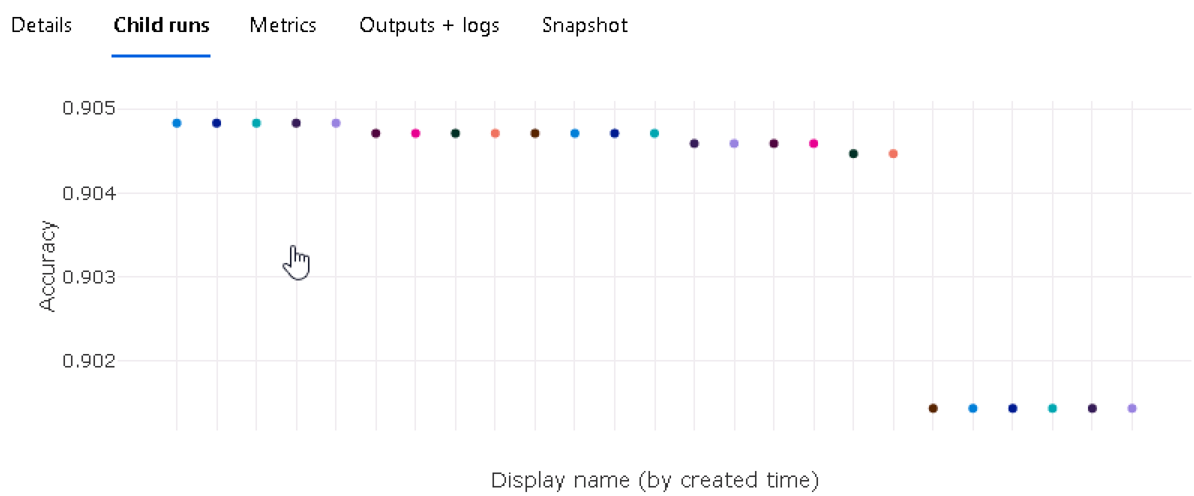
The best model generated from the run was a VotingEnsemble model, It consisted of 9 voting classifiers and weights.
Auto ml combined the predictions of the 9 voting classifiers and achieves the top accuracy rate of 91.60%. VotingEnsemble model also gives lists of hyper parameters of the 9 voting classifiers, below is the example weights and hyper parameters for standard scalar wrapper, XGBoost Classifier:

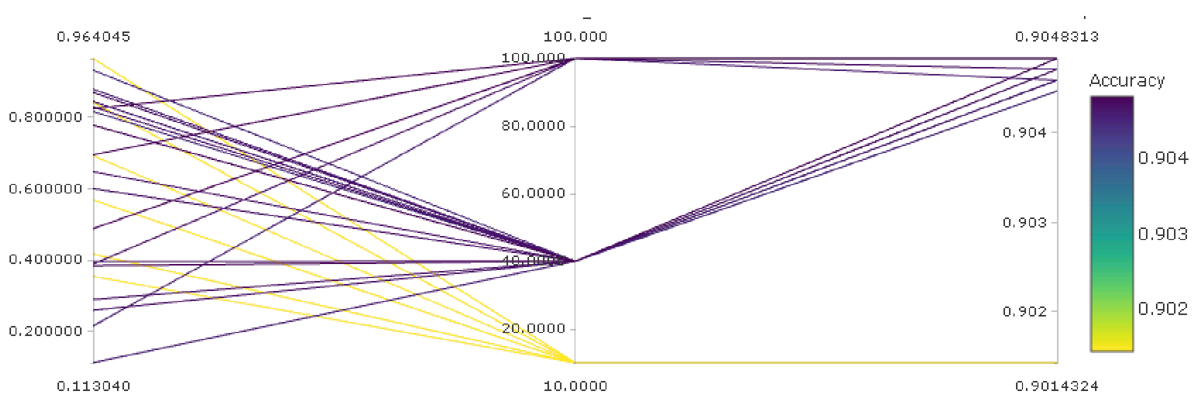
```
Data transformation:
{
    "class_name": "StandardScaler",
    "module": "sklearn.preprocessing",
    "param_args": [],
    "param_kwargs": {
        "with_mean": false,
        "with_std": false
    },
    "prepared_kwargs": {},
    "spec_class": "preproc"
}

Training algorithm:
{
    "class_name": "XGBoostClassifier",
    "module": "automl.client.core.common.model_wrappers",
    "param_args": [],
    "param_kwargs": {
        "booster": "gbtree",
        "colsample_bytree": 0.7,
        "eta": 0.1,
        "gamma": 0.1,
        "max_depth": 9,
        "max_leaves": 511,
        "n_estimators": 25,
        "objective": "reg:logistic",
        "reg_alpha": 0,
        "reg_lambda": 1.7708333333333335,
        "subsample": 0.9,
        "tree_method": "auto"
    },
    "prepared_kwargs": {},
    "spec_class": "sklearn"
}

Ensemble weight: 0.14285714285714285
```

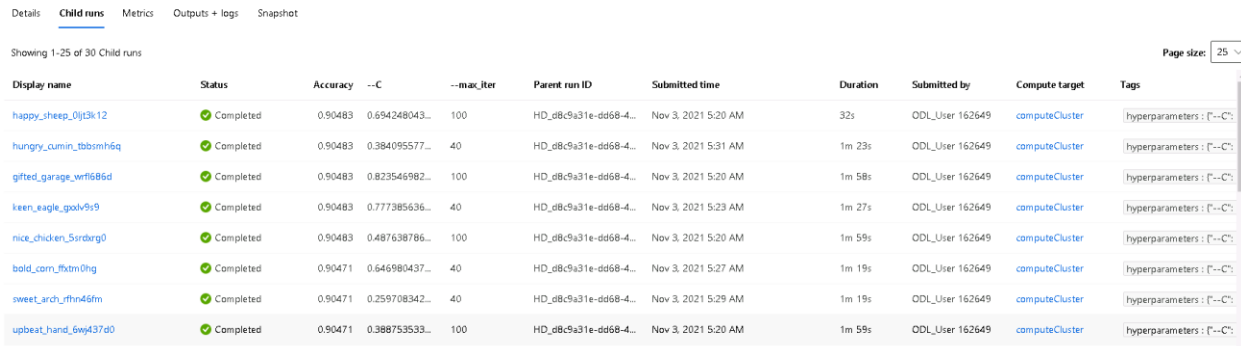
Pipeline comparison
There is a little difference in accuracy and also a little difference between them from architecture point of view. HyperDrive requires, a custom-coded machine learning model whereas AutoML requires selection of few paramters for AutoML config. AutoML model also have a feature for model interpretation.



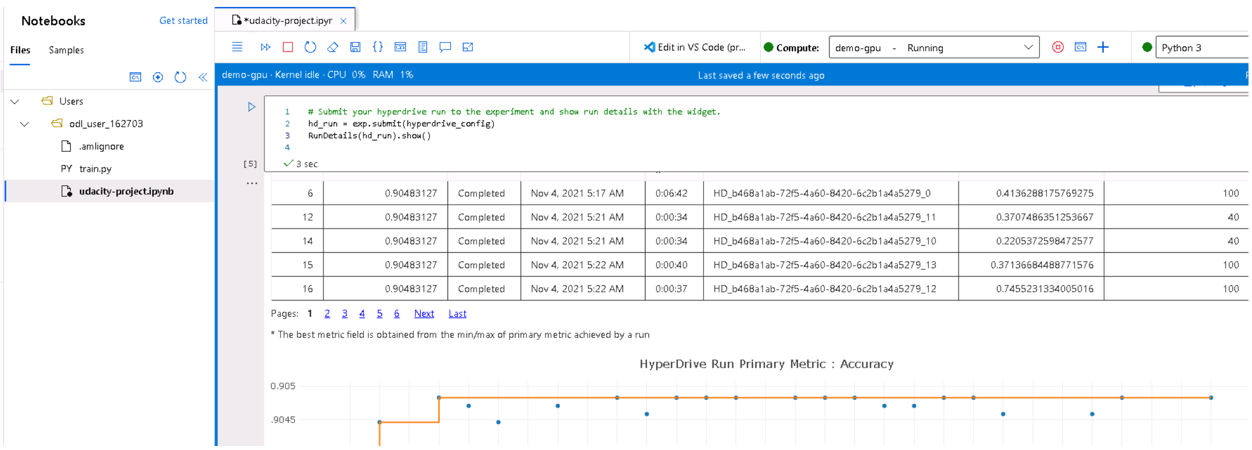
The HyperDrive assisted Scikit-learn LogicRegression model gives the best accuracy of 90.50% as shown below:



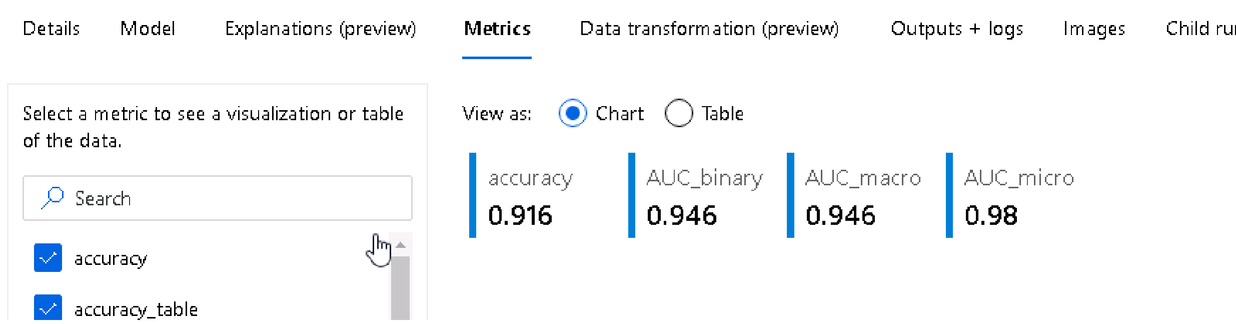


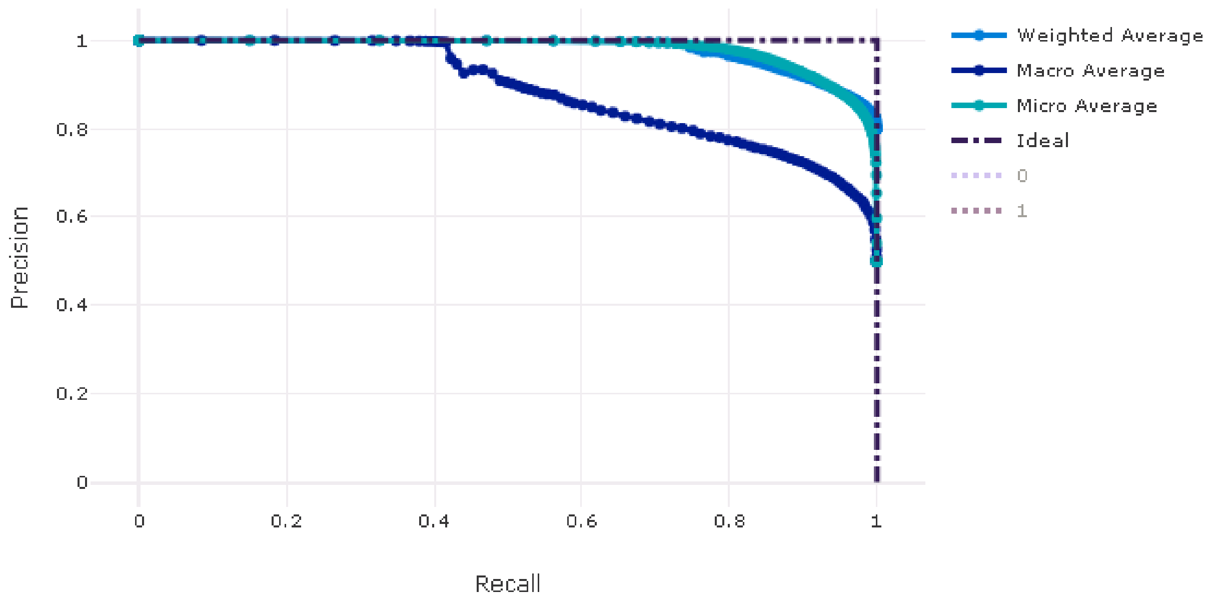


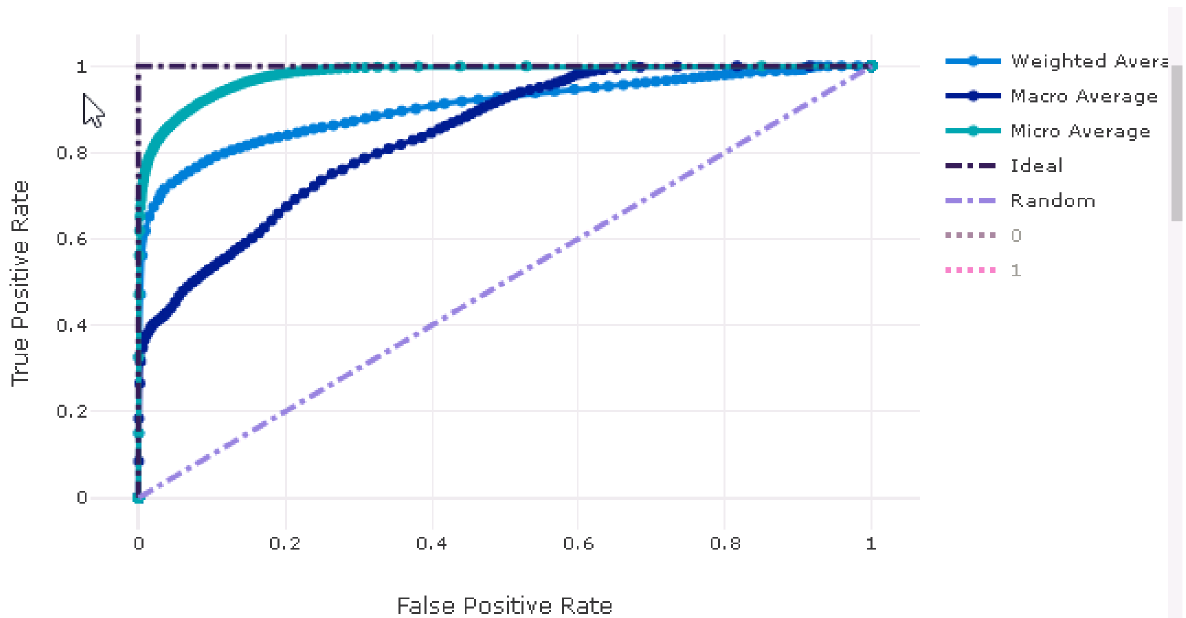
Showing the run details with the widget:



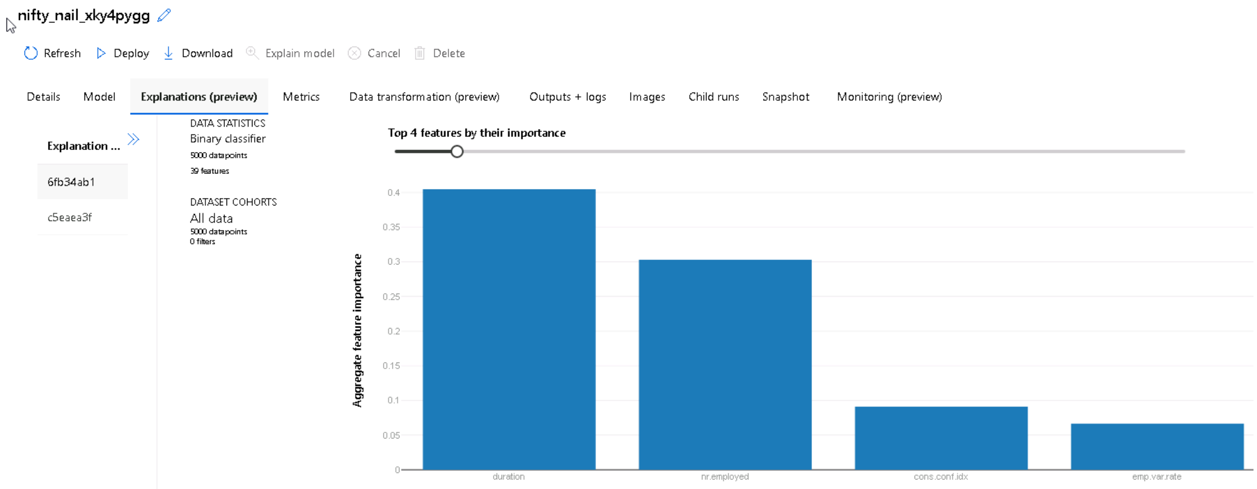
And the AutoML (VotingEnsemble) model gives the best accuracy of 91.60%:

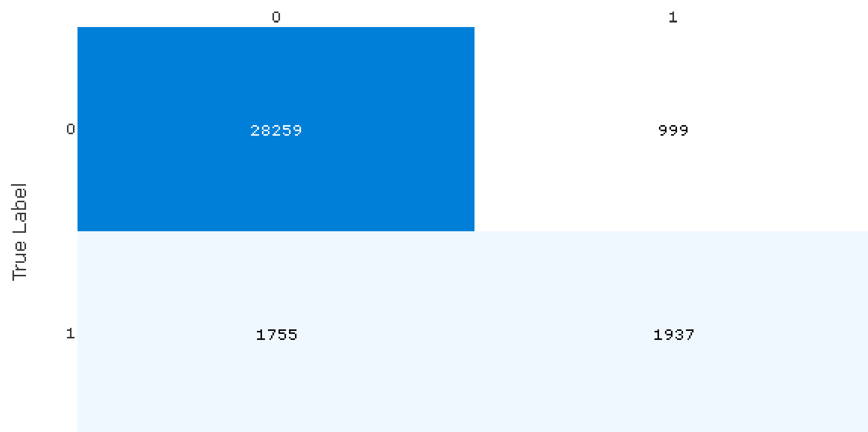






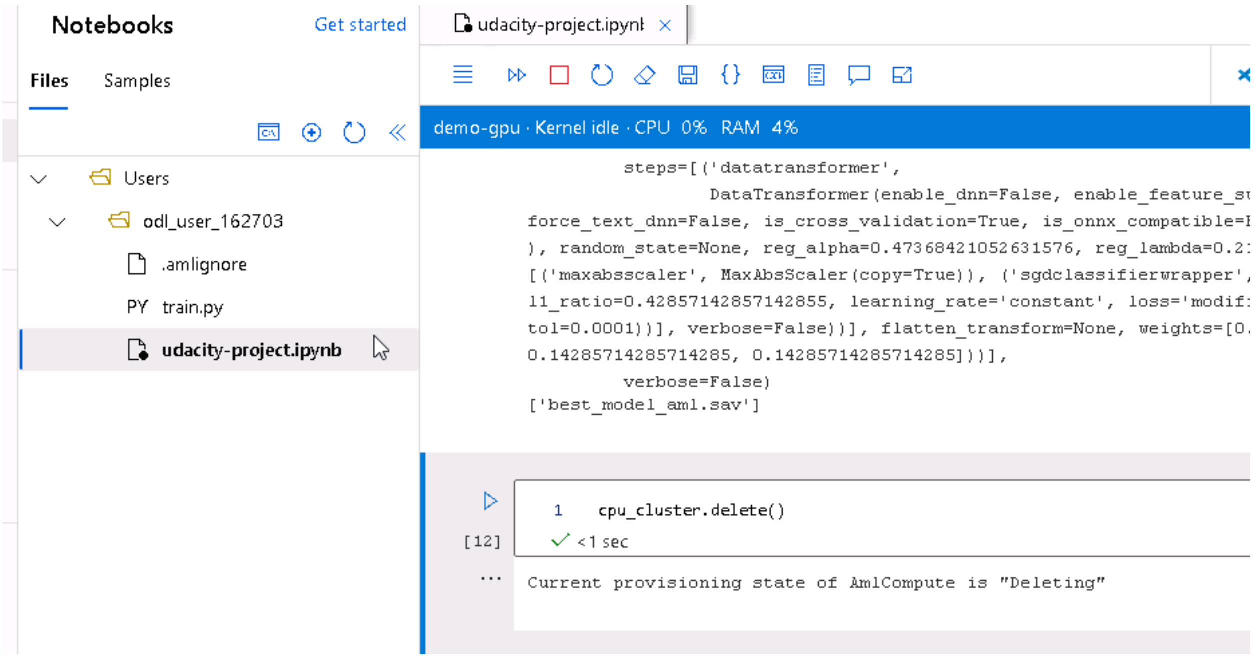
below are the AutoML generated visual feature based explanation and confusion matrix **(not normalized)**:





Future work
Apply model interpretability of AutoML on more complex and larger datasets, to gain speed and valuable insights in feature engineering, which can in turn be used to refine complex model accuracy
Experiment with different hyperparameter sampling methods like Gird sampling or Bayesian sampling on the Scikit-learn LogicRegression model or other customcoded machine learning models.

deleting compute:
Deleted the compute cluster as shown below


Notebooks
Get started
Files
Samples
Users
odl_user_162703
.amlignore
PY train.py
udacity-project.ipynb
udacity-project.ipynb
demo-gpu · Kernel idle · CPU 0% RAM 4%
steps=[('datatransformer',
DataTransformer(enable_dnn=False, enable_feature_s
force_text_dnn=False, is_cross_validation=True, is_onnx_compatible=
), random_state=None, reg_alpha=0.47368421052631576, reg_lambda=0.2
[('maxabsscaler', MaxAbsScaler(copy=True)), ('sgdclassifierwrapper',
l1_ratio=0.42857142857142855, learning_rate='constant', loss='modif
tol=0.0001))], verbose=False))], flatten_transform=None, weights=[0.
0.14285714285714285, 0.14285714285714285]))],
verbose=False)
['best_model_aml.sav']
1 cpu_cluster.delete()
[12]
<1 sec
Current provisioning state of AmlCompute is "Deleting"