

Automated ML

TODO: Import Dependencies. In the cell below, import all the dependencies that you will need to complete the project.

```
In [1]: import os
import time
import joblib
import requests
from matplotlib import pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error
from azureml.widgets import RunDetails
from azureml.core import Workspace, Experiment
from azureml.core.model import Model, InferenceConfig
from azureml.core.compute import ComputeTarget, AmlCompute
from azureml.core.environment import Environment
from azureml.core.authentication import InteractiveLoginAuthentication
from azureml.core.webservice import AciWebservice
from azureml.data.dataset_factory import TabularDatasetFactory
from azureml.train.automl import AutoMLConfig

print("SDK version:", azureml.core.VERSION)
```

SDK version: 1.20.0

```
In [2]: time.strftime('%Y-%m-%d %H:%M:%S')
```

Out[2]: '2021-02-10 14:33:59'

```
In [3]: ws = Workspace.from_config()
print(ws.name, ws.resource_group, ws.location, ws.subscription_id, sep = '\n')
```

```
quick-starts-ws-138398
aml-quickstarts-138398
southcentralus
1b944a9b-fdae-4f97-aeb1-b7eea0beac53
```

Set up a compute instance to train the model.

```
In [4]: compute_name = os.environ.get("AML_COMPUTE_CLUSTER_NAME", "auto-ml")
vm_size = os.environ.get("AML_COMPUTE_CLUSTER_SKU", "STANDARD_D2_V2")
compute_min_nodes = os.environ.get("AML_COMPUTE_CLUSTER_MIN_NODES", 2)
compute_max_nodes = os.environ.get("AML_COMPUTE_CLUSTER_MAX_NODES", 4)

compute_names = [cn for cn in ws.compute_targets if cn in [compute_name, 'auto-ml', 'aml-compute']]

if compute_names:
    compute_target = ws.compute_targets[compute_names[0]]
    if compute_target and type(compute_target) is AmlCompute:
        print('Using existing compute target: ' + compute_names[0])
else:
    compute_config = AmlCompute.provisioning_configuration(
        vm_size=vm_size,
        min_nodes=compute_min_nodes,
        max_nodes=compute_max_nodes
    )
    compute_target = ComputeTarget.create(ws, compute_name, compute_config)
```

Dataset

Overview

We will be using the Wine Quality dataset made accessible [here \(https://archive.ics.uci.edu/ml/datasets/wine+quality\)](https://archive.ics.uci.edu/ml/datasets/wine+quality).

The dataset contains the physicochemical properties of 1599 red wine samples. The data includes, in column order: fixed acidity, volatile acidity, citric acid, residual sugar, chlorides, free sulfur dioxide, total sulfur dioxide, density, pH, sulphates, alcohol, and the final column is the target variable, quality (score between 0 and 10).

We will carry out an AutoML regression analysis on this dataset to predict the quality of a particular sample of wine from measured properties.

```
In [5]: data_file_source = "https://archive.ics.uci.edu/ml/machine-learning-databases/wine-quality/winequality-red.csv"
ds_data = TabularDatasetFactory.from_delimited_files(
    path=data_file_source,
    separator=';'
)
ds_data = ds_data.register(
    workspace=ws,
    name='wine_quality_data',
    description='Wine Quality data from ' + data_file_source
)
df = ds_data.to_pandas_dataframe()
x = df.drop(columns='quality')
y = df['quality']
```

Split the dataset into a training set and a test set, using 70% of the dataset for training.

```
In [6]: x_train, x_test, y_train, y_test = train_test_split(x, y, train_size=0.7, random_state=42)
datastore = ws.get_default_datastore()
x_train.assign(target=y_train).to_csv('df_train.csv', index=False)
datastore.upload_files(['df_train.csv'], overwrite=True)
train_ds = TabularDatasetFactory.from_delimited_files((datastore, 'df_train.csv'))

Uploading an estimated of 1 files
Uploading df_train.csv
Uploaded df_train.csv, 1 files out of an estimated total of 1
Uploaded 1 files
```

AutoML Configuration

TODO: Explain why you chose the automl settings and cofiguration you used below.

```
In [7]: experiment_name = 'winequality-automl'
project_folder = './winequality-automl-pipeline'
experiment=Experiment(ws, experiment_name)
```

Create AutoML settings object

```
In [8]: aml_regression_metrics = [
'spearman_correlation',
'normalized_root_mean_squared_error',
'r2_score',
'normalized_mean_absolute_error'
]

automl_settings = {
'experiment_timeout_minutes': 15,
'max_concurrent_iterations': 5,
'n_cross_validations': 5,
'primary_metric' : 'normalized_root_mean_squared_error'
}

automl_config = AutoMLConfig(
compute_target=compute_target,
task='regression',
training_data=train_ds,
label_column_name='target',
path=project_folder,
enable_early_stopping=True,
featurization='auto',
debug_log='automl_errors.log',
**automl_settings
)
```

Create pipeline and AutoMLStep

```
In [9]: compute_target.wait_for_completion(show_output=True)
```

Creating
Succeeded.....
AmlCompute wait for completion finished

Minimum number of nodes requested have been provisioned

```
In [10]: automl_run = experiment.submit(config=automl_config)
automl_run
```

Running on remote.

Out[10]:	Experiment	Id	Type	Status	Details Page	Docs Page
	winequality-automl	AutoML_09d54a55-51c6-499f-a116-05e9e7189e72	automl	NotStarted	Link to Azure Machine Learning studio https://ml.azure.com/experiments/winequality-automl/runs/AutoML_09d54a55-51c6-499f-a116-05e9e7189e72?wsid=/subscriptions/1b944a9b-fdae-4f97-aeb1-b7eea0beac53/resourcegroups/aml-quickstarts-138398/workspaces/quickstarts-ws-138398	Link to Documentation https://docs.microsoft.com/en-us/python/api/overview/azure/ml/view=azure-ml-py

Run Details

```
In [11]: RunDetails(automl_run).show()
```

AutoML_09d54a55-51c6-499f-a116-05e9e7189e72:
Status: Completed

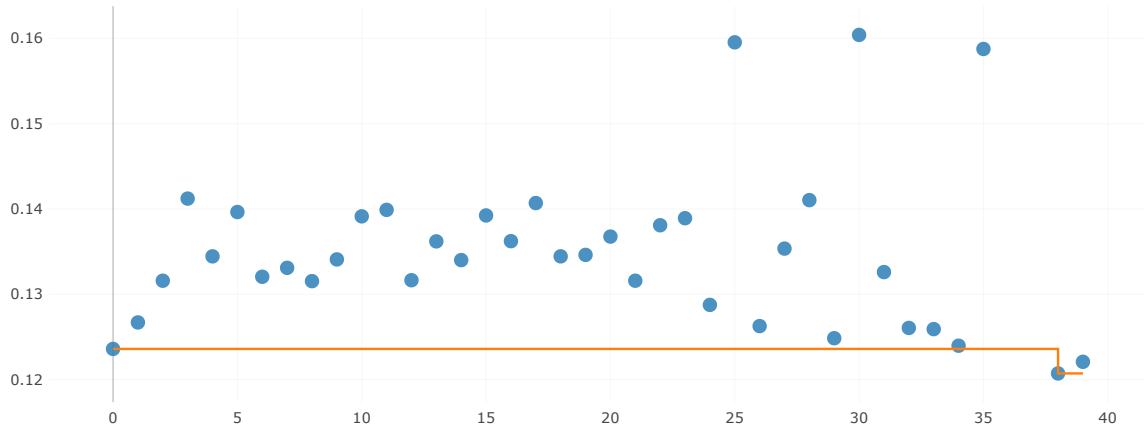


Iteration	Pipeline	Iteration metric	Best metric	Status	Duration	Started	Run Id
37	Canceled			Canceled	0:00:16	Feb 10, 2021 9:57 AM	
36	Canceled			Canceled	0:00:42	Feb 10, 2021 9:57 AM	
38	VotingEnsemble	0.12072878	0.12072878	Completed	0:01:50	Feb 10, 2021 9:58 AM	
39	StackEnsemble	0.12209073	0.12072878	Completed	0:01:44	Feb 10, 2021 9:58 AM	
0	MaxAbsScaler, LightGBM	0.12360182	0.12360182	Completed	0:01:50	Feb 10, 2021 9:44 AM	

Pages: 1 2 3 4 5 6 7 8 5 per page
Next Last

normalized_root_mean_squared_error

AutoML Run with metric : normalized_root_mean_squared_error



Click here to see the run in Azure Machine Learning studio (https://ml.azure.com/experiments/winequality-automl/runs/AutoML_09d54a55-51c6-499f-a116-05e9e7189e72?wsid=/subscriptions/1b944a9b-fdae-4f97-aeb1-b7eea0beac53/resourcegroups/aml-quickstarts-138398/workspaces/quick-starts-ws-138398)

In [12]: `automl_run.wait_for_completion()`

```
Out[12]: {'runId': 'AutoML_09d54a55-51c6-499f-a116-05e9e7189e72',
'target': 'auto-ml',
'status': 'Completed',
'startTimeUtc': '2021-02-10T14:36:46.448406Z',
'endTimeUtc': '2021-02-10T14:59:56.430023Z',
'properties': {'num_iterations': '1000',
'training_type': 'TrainFull',
'acquisition_function': 'EI',
'primary_metric': 'normalized_root_mean_squared_error',
'train_split': '0',
'acquisition_parameter': '0',
'num_cross_validation': '5',
'target': 'auto-ml',
'AMLSettingsJsonString': '{"path":null,"name":"winequality-automl","subscription_id":"1b944a9b-fdae-4f97-aeb1-b7eea0beac53","resource_group":"aml-quickstarts-138398","workspace_name":"quick-starts-ws-138398","region":"southcentralus","compute_target":"auto-ml","spark_service":null,"azure_service":"remote","many_models":false,"pipeline_fetch_max_batch_size":1,"iterations":1000,"primary_metric":"normalized_root_mean_squared_error","task_type":"regression","data_script":null,"validation_size":0.0,"n_cross_validations":5,"y_min":null,"y_max":null,"num_classes":null,"featurization":"auto","ignore_package_version_incompatibilities":false,"is_timeseries":false,"max_cores_per_iteration":1,"max_concurrent_iterations":5,"iteration_timeout_minutes":null,"mem_in_mb":null,"enforce_time_on_windows":false,"experiment_timeout_minutes":15,"experiment_exit_score":null,"whitelist_models":null,"blacklist_algos":["TensorFlowDNN","TensorFlowLinearRegressor"],"supported_models":["ExtremeRandomTrees","ElasticNet","DecisionTree","LightGBM","LassoLars","KNN","SGD","OnlineGradientDescentRegressor","TensorFlowLinearRegressor","GradientBoosting","TensorFlowDNN","XGBoostRegressor","FastLinearRegressor","RandomForest"],"auto_blacklist":true,"blacklist_samples_reached":false,"exclude_nan_labels":true,"verbosity":20,"debug_log":"azureml_automl.log","show_warnings":false,"model_explainability":true,"service_url":null,"sdk_url":null,"sdk_packages":null,"enable_onnx_compatible_models":false,"enable_split_onnx_featurizer_estimator_models":false,"vm_type":"STANDARD_D2_V2","telemetry_verbosity":20,"send_telemetry":true,"enable_dnn":false,"scenario":"SDK-1.13.0","environment_label":null,"force_text_dnn":false,"enable_feature_sweeping":true,"enable_early_stopping":true,"early_stopping_n_iters":10,"metrics":null,"enable_ensembling":true,"enable_stack_ensembling":true,"ensemble_iterations":15,"enable_tf":false,"enable_subsampling":null,"subsample_seed":null,"enable_nimbusml":false,"enable_streaming":false,"force_streaming":false,"track_child_runs":true,"allowed_private_models":[],"label_column_name":"","target","weight_column_name":null,"cv_split_column_names":null,"enable_local_managed":false,"local_managed_run_id":null,"cost_mode":1,"lag_length":0,"metric_operation":"minimize","preprocess":true}',
'DataPrepJsonString': '{"training_data": [{"blocks": [{"id": "c4e856cd-98c9-44ed-89d6-20c8e03f8269", "type": "Microsoft.DPrep.GetDatastoreFilesBlock", "arguments": {"datastores": [{"datastoreName": "workspaceblobstore", "path": "df_train.csv", "resourceGroup": "aml-quickstarts-138398", "subscription": "1b944a9b-fdae-4f97-aeb1-b7eea0beac53", "workspaceName": "quick-starts-ws-138398"}]}], "localData": {"isEnabled": true, "name": "null", "annotation": null, "id": "7af68e2e-9ba9-4e77-bbcb-798f60f85961", "type": "Microsoft.DPrep.ParseDelimitedBlock", "arguments": {"columnHeadersMode": 3, "fileEncoding": 0, "handleQuotedLineBreaks": false, "preview": false, "separator": ",", "skipRows": 0, "skipRowsMode": 0}, "localData": {"isEnabled": true, "name": null, "annotation": null, "id": "22c6750c-c56e-42fc-86f0-bc23756a5e9f", "type": "Microsoft.DPrep.DropColumnBlock", "arguments": {"columns": {"type": 0, "details": {"selectedColumns": [{"Path": ""}], "isEnabled": true, "name": null, "annotation": null, "id": "3ee009af-6fae-4df3-be2f-445e64715706", "type": "Microsoft.DPrep.SetColumnTypesBlock", "arguments": {"columnConversion": [{"column": {"type": 2, "details": {"selectedColumn": {"fixed acidity": {"typeProperty": 3, "column": {"type": 2, "details": {"selectedColumn": {"volatile acidity": {"typeProperty": 3, "column": {"type": 2, "details": {"selectedColumn": {"citric acid": {"typeProperty": 3, "column": {"type": 2, "details": {"selectedColumn": {"residual sugar": {"typeProperty": 3, "column": {"type": 2, "details": {"selectedColumn": {"chlorides": {"typeProperty": 3, "column": {"type": 2, "details": {"selectedColumn": {"free sulfur dioxide": {"typeProperty": 3, "column": {"type": 2, "details": {"selectedColumn": {"total sulfur dioxide": {"typeProperty": 3, "column": {"type": 2, "details": {"selectedColumn": {"density": {"typeProperty": 3, "column": {"type": 2, "details": {"selectedColumn": {"pH": {"typeProperty": 3, "column": {"type": 2, "details": {"selectedColumn": {"sulphates": {"typeProperty": 3, "column": {"type": 2, "details": {"selectedColumn": {"alcohol": {"typeProperty": 3, "column": {"type": 2, "details": {"selectedColumn": {"target": {"typeProperty": 2, "localData": {"isEnabled": true, "name": null, "annotation": null}, "inspectors": [], "meta": {"saveDatasetId": "242fd539-a946-4bae-be61-7ac8cddf0169", "datasetType": "tabular", "subscriptionId": "1b944a9b-fdae-4f97-aeb1-b7eea0beac53", "workspaceId": "a064dfc8-3ed1-42fd-a27c-2c0d6e5664ef", "workspaceLocation": "southcentralus"}}, "activities": {}}, "EnableSubsampling": None, "runTemplate": 'AutoML', 'azureml.runsource': 'automl', 'display_task_type': 'regression', 'dependencies_versions': {'azureml-widgets': "1.20.0", "azureml-train": "1.20.0", "azureml-train-restclients-hyperdrive": "1.20.0", "azureml-train-core": "1.20.0", "azureml-train-automl": "1.20.0", "azureml-train-automl-runtime": "1.20.0", "azureml-train-automl-client": "1.20.0", "azureml-tensorboard": "1.20.0", "azureml-telemetry": "1.20.0", "azureml-samples": "0+unknown", "azureml-pipeline": "1.20.0", "azureml-pipeline-steps": "1.20.0", "azureml-pipeline-core": "1.20.0", "azureml-opendatasets": "1.20.0", "azureml-model-management-sdk": "1.0.1b6.post1", "azureml-mflow": "1.20.0.post1", "azureml-interpret": "1.20.0", "azureml-explain-model": "1.20.0", "azureml-defaults": "1.20.0", "azureml-dataset-runtime": "1.20.0", "azureml-dataprep": "2.7.3", "azureml-dataprep-rslex": "1.5.0", "azureml-dataprep-native": "27.0.0", "azureml-datadrift": "1.20.0", "azureml-core": "1.20.0", "azureml-contrib-services": "1.20.0", "azureml-contrib-server": "1.20.0", "azureml-contrib-reinforcementlearning": "1.20.0", "azureml-contrib-pipeline-steps": "1.20.0", "azureml-contrib-notebook": "1.20.0", "azureml-contrib-interpret": "1.20.0", "azureml-contrib-gbdt": "1.20.0", "azureml-contrib-fairness": "1.20.0", "azureml-contrib-dataset": "1.20.0", "azureml-cli-common": "1.20.0", "azureml-automl-runtime": "1.20.0", "azureml-automl-core": "1.20.0", "azureml-accel-models": "1.20.0"}}, 'aml_system_scenario_identification': 'Remote.Parent', 'ClientType': 'SDK', 'environment_cpu_name': 'AzureML-AutoML', 'environment_cpu_label': 'prod', 'environment_gpu_name': 'AzureML-AutoML-GPU', 'environment_gpu_label': 'prod', 'root_attribution': 'automl', 'attribution': 'AutoML', 'Orchestrator': 'AutoML', 'CancelUri': 'https://southcentralus.experiments.azureml.net/jasmine/v1.0/subscriptions/1b944a9b-fdae-4f97-aeb1-b7eea0beac53/resourceGroups/aml-quickstarts-138398/providers/Microsoft.MachineLearningServices/workspaces/quick-starts-ws-138398/experimentids/ea4af114-a339-492e-85c2-79c5112bce33/cancel/AutoML_09d54a55-51c6-499f-a116-05e9e7189e72', 'ClientSdkVersion': '1.21.0', 'snapshotId': '00000000-0000-0000-0000-000000000000', 'SetupRunId': 'AutoML_09d54a55-51c6-499f-a116-05e9e7189e72_setup', 'SetupRunContainerId': 'dcid.AutoML_09d54a55-51c6-499f-a116-05e9e7189e72_setup',
```

```
'FeaturizationRunJsonPath': 'featurizer_container.json',
'FeaturizationRunId': 'AutoML_09d54a55-51c6-499f-a116-05e9e7189e72_featurize',
'ProblemInfoJsonString': '{"dataset_num_categorical": 0, "is_sparse": false, "subsampling": false, "dataset_classes": 6, "dataset_features": 11, "dataset_samples": 1119, "single_frequency_class_detected": false}',
'ModelExplainRunId': 'AutoML_09d54a55-51c6-499f-a116-05e9e7189e72_ModelExplain',
'inputDatasets': [{'dataset': {'id': '242fd539-a946-4bae-be61-7ac8ccdf0169'}, 'consumptionDetails': {'type': 'RunInput', 'inputName': 'training_data', 'mechanism': 'Direct'}}],
'outputDatasets': [],
'logFiles': {},
'submittedBy': 'ODL_User 138398'}
```

Best Model

TODO: In the cell below, get the best model from the automl experiments and display all the properties of the model.

```
In [13]: # Retrieve and save your best automl model.
automl_best_run = automl_run.get_best_child()
automl_best_run_metrics = automl_best_run.get_metrics()

print('Best Run Id: ', automl_best_run.id)
[(k, round(v, 4)) if type(v) != str else (k, v) for k, v in automl_best_run_metrics.items()]
```

Best Run Id: AutoML_09d54a55-51c6-499f-a116-05e9e7189e72_38

```
Out[13]: [('normalized_median_absolute_error', 0.064),
('median_absolute_error', 0.3199),
('r2_score', 0.4403),
('normalized_mean_absolute_error', 0.0895),
('mean_absolute_percentage_error', 8.2792),
('spearman_correlation', 0.6714),
('root_mean_squared_error', 0.6036),
('explained_variance', 0.4483),
('root_mean_squared_log_error', 0.094),
('mean_absolute_error', 0.4475),
('normalized_root_mean_squared_error', 0.1207),
('normalized_root_mean_squared_log_error', 0.1159),
('residuals',
'aml://artifactId/ExperimentRun/dcid.AutoML_09d54a55-51c6-499f-a116-05e9e7189e72_38/residuals'),
('predicted_true',
'aml://artifactId/ExperimentRun/dcid.AutoML_09d54a55-51c6-499f-a116-05e9e7189e72_38/predicted_true')]
```

```
In [14]: model = automl_run.register_model(
description='Wine quality AutoML best model',
tags={'area': 'wine-quality'})
```

```
In [15]: model.download()
```

Out[15]: 'model.pkl'

```
In [16]: local_model = joblib.load('model.pkl')
```

```
In [17]: local_model.steps[1][1].get_params()
```

```
Out[17]: {'estimators': [(0,
 Pipeline(memory=None,
 steps=[('maxabsscaler', MaxAbsScaler(copy=True)),
 ('lightgbmregressor',
 LightGBMRegressor(boosting_type='gbdt', class_weight=None,
 colsample_bytree=1.0,
 importance_type='split', learning_rate=0.1,
 max_depth=-1, min_child_samples=20,
 min_child_weight=0.001, min_split_gain=0.0,
 n_estimators=100, n_jobs=1, num_leaves=31,
 objective=None, random_state=None,
 reg_alpha=0.0, reg_lambda=0.0, silent=True,
 subsample=1.0, subsample_for_bin=200000,
 subsample_freq=0, verbose=-1))),
 verbose=False)),
 (34,
 Pipeline(memory=None,
 steps=[('standardscalerwrapper',
 <azureml.automl.runtime.shared.model_wrappers.StandardScalerWrapper object at 0x7f2f3c3184e0>),
 ('gradientboostingregressor',
 GradientBoostingRegressor(alpha=0.9, ccp_alpha=0.0,
 criterion='friedman_mse', init=None,
 learning_rate=0.046415888336127774,
 loss='huber', max_depth=10,
 max_features=0.8,
 max_leaf_nodes=None,
 min_impurity_decrease=0.0,
 min_impurity_split=None,
 min_samples_leaf=0.01321775243133542,
 min_samples_split=0.0021800253234900498,
 min_weight_fraction_leaf=0.0,
 n_estimators=50,
 n_iter_no_change=None,
 presort='deprecated',
 random_state=None, subsample=0.9,
 tol=0.0001, validation_fraction=0.1,
 verbose=0, warm_start=False))),
 verbose=False)),
 (29,
 Pipeline(memory=None,
 steps=[('minmaxscaler', MinMaxScaler(copy=True, feature_range=(0, 1))),
 ('randomforestregressor',
 RandomForestRegressor(bootstrap=False, ccp_alpha=0.0,
 criterion='mse', max_depth=None,
 max_features=0.5, max_leaf_nodes=None,
 max_samples=None,
 min_impurity_decrease=0.0,
 min_impurity_split=None,
 min_samples_leaf=0.0023646822772690063,
 min_samples_split=0.052853885930792446,
 min_weight_fraction_leaf=0.0,
 n_estimators=100, n_jobs=1,
 oob_score=False, random_state=None,
 verbose=0, warm_start=False))),
 verbose=False)),
 (33,
 Pipeline(memory=None,
 steps=[('standardscalerwrapper',
 <azureml.automl.runtime.shared.model_wrappers.StandardScalerWrapper object at 0x7f2f3ba86f98>),
 ('randomforestregressor',
 RandomForestRegressor(bootstrap=False, ccp_alpha=0.0,
 criterion='mse', max_depth=None,
 max_features='sqrt', max_leaf_nodes=None,
 max_samples=None,
 min_impurity_decrease=0.0,
 min_impurity_split=None,
 min_samples_leaf=0.019375063213477914,
 min_samples_split=0.00630957344480193,
 min_weight_fraction_leaf=0.0,
 n_estimators=25, n_jobs=1,
 oob_score=False, random_state=None,
 verbose=0, warm_start=False))),
 verbose=False)),
 (32,
 Pipeline(memory=None,
 steps=[('robustscaler',
 RobustScaler(copy=True, quantile_range=[25, 75],
 with_centering=False, with_scaling=False)),
 ('gradientboostingregressor',
 GradientBoostingRegressor(alpha=0.7000000000000001,
 ccp_alpha=0.0, criterion='mse',
 init=None,
 learning_rate=0.021544346900318822,
 loss='ls', max_depth=4,
 max_features=0.2,
 max_leaf_nodes=None,
 min_impurity_decrease=0.0,
 min_impurity_split=None,
 min_samples_leaf=0.01321775243133542,
 min_samples_split=0.001529732116091358,
 min_weight_fraction_leaf=0.0,
 n_estimators=600,
 n_iter_no_change=None,
```

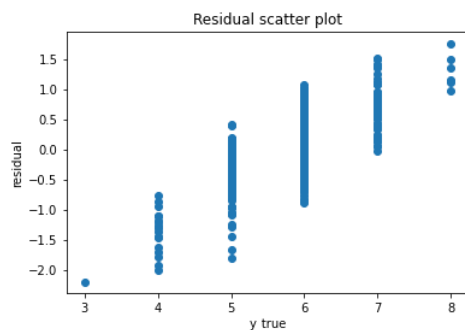
```

presort='deprecated',
random_state=None, subsample=0.1,
tol=0.0001, validation_fraction=0.1,
verbose=0, warm_start=False)),
(verbose=False)),
('26',
 Pipeline(memory=None,
 steps=[('robustscaler',
 RobustScaler(copy=True, quantile_range=[10, 90],
 with_centering=True, with_scaling=False)),
 ('extratreesregressor',
 ExtraTreesRegressor(bootstrap=True, ccp_alpha=0.0,
 criterion='mse', max_depth=None,
 max_features=0.8, max_leaf_nodes=None,
 max_samples=None,
 min_impurity_decrease=0.0,
 min_impurity_split=None,
 min_samples_leaf=0.004196633747563344,
 min_samples_split=0.018261584682702607,
 min_weight_fraction_leaf=0.0,
 n_estimators=25, n_jobs=1, oob_score=False,
 random_state=None, verbose=0,
 warm_start=False))),
(verbose=False)),
('3',
 Pipeline(memory=None,
 steps=[('robustscaler',
 RobustScaler(copy=True, quantile_range=[10, 90],
 with_centering=True, with_scaling=False)),
 ('decisiontreeregressor',
 DecisionTreeRegressor(ccp_alpha=0.0, criterion='mse',
 max_depth=None, max_features=0.9,
 max_leaf_nodes=None,
 min_impurity_decrease=0.0,
 min_impurity_split=None,
 min_samples_leaf=0.007594568292413517,
 min_samples_split=0.0037087774117744725,
 min_weight_fraction_leaf=0.0,
 presort='deprecated', random_state=None,
 splitter='best'))]),
(verbose=False)),
('22',
 Pipeline(memory=None,
 steps=[('minmaxscaler', MinMaxScaler(copy=True, feature_range=(0, 1))),
 ('decisiontreeregressor',
 DecisionTreeRegressor(ccp_alpha=0.0, criterion='friedman_mse',
 max_depth=None, max_features=None,
 max_leaf_nodes=None,
 min_impurity_decrease=0.0,
 min_impurity_split=None,
 min_samples_leaf=0.002742668396429022,
 min_samples_split=0.026024633095283807,
 min_weight_fraction_leaf=0.0,
 presort='deprecated', random_state=None,
 splitter='best'))]),
(verbose=False))),
'weights': [0.4666666666666667,
0.06666666666666667,
0.06666666666666667,
0.06666666666666667,
0.13333333333333333,
0.06666666666666667,
0.06666666666666667,
0.06666666666666667]}

```

```
In [18]: y_pred = local_model.predict(x_test)
```

```
In [19]: plt.scatter(y_test, y_test-y_pred)
plt.title('Residual scatter plot')
plt.xlabel('y_true')
plt.ylabel('residual')
plt.show()
```



Model Deployment

```
In [20]: env = Environment.get(workspace=ws, name='AzureML-AutoML')

inference_config = InferenceConfig(
    entry_script='score_automl.py',
    environment=env
)
aci_config = AciWebService.deploy_configuration(
    cpu_cores=1,
    memory_gb=1
)
```

```
In [21]: service_name = 'wine-quality-predictor-aml'

service = Model.deploy(
    workspace=ws,
    name=service_name,
    models=[model],
    inference_config=inference_config,
    deployment_config=aci_config,
    overwrite=True
)
```

```
In [22]: service.wait_for_deployment(show_output=True)
print("State: " + service.state)
print("Scoring URI: " + service.scoring_uri)

Tips: You can try get_logs(): https://aka.ms/debugimage#dockerlog (https://aka.ms/debugimage#dockerlog) or local deployment: https://aka.ms/debugimage#debug-locally (https://aka.ms/debugimage#debug-locally) to debug if deployment takes longer than 10 minutes.
Running.....
Succeeded
ACI service creation operation finished, operation "Succeeded"
State: Healthy
Scoring URI: http://b7cd6463-b056-4b3a-859a-2e8ceb5e4423.southcentralus.azurecontainer.io/score (http://b7cd6463-b056-4b3a-859a-2e8ceb5e4423.southcentralus.azurecontainer.io/score)
```

Send a request to the deployed web service

```
In [23]: # select a few random rows from the test set to score
random_data = df.sample(5, random_state=42).values
random_data.shape
```

Out[23]: (5, 12)

```
In [24]: x_test_run = random_data[:, :-1].tolist()
y_test_run = random_data[:, -1].tolist()

input_data = "{\"data\": " + str(x_test_run) + "}"
headers = {'Content-Type': 'application/json'}

resp = requests.post(service.scoring_uri, input_data, headers=headers)

print("POST to url", service.scoring_uri)
print("input data:", input_data)
print("label:", y_test_run)
print("prediction:", resp.text)

POST to url http://b7cd6463-b056-4b3a-859a-2e8ceb5e4423.southcentralus.azurecontainer.io/score (http://b7cd6463-b056-4b3a-859a-2e8ceb5e4423.southcentralus.azurecontainer.io/score)
input data: {"data": [[7.7, 0.56, 0.08, 2.5, 0.114, 14.0, 46.0, 0.9971, 3.24, 0.66, 9.6], [7.8, 0.5, 0.17, 1.6, 0.082, 21.0, 102.0, 0.996, 3.39, 0.48, 9.5], [10.7, 0.67, 0.22, 2.7, 0.107, 17.0, 34.0, 1.0004, 3.28, 0.98, 9.9], [8.5, 0.46, 0.31, 2.25, 0.078, 32.0, 58.0, 0.998, 3.33, 0.54, 9.8], [6.7, 0.46, 0.24, 1.7, 0.077, 18.0, 34.0, 0.9948, 3.39, 0.6, 10.6]]}
label: [6.0, 5.0, 6.0, 5.0, 6.0]
prediction: [5.35, 5.04, 5.19, 5.29, 5.91]
```

```
In [25]: time.strftime('%Y-%m-%d %H:%M:%S')
```

Out[25]: '2021-02-10 15:03:45'

TODO: In the cell below, print the logs of the web service and delete the service


```
In [26]: logs = service.get_logs()
for line in logs.split('\n'):
    print(line)
```

2021-02-10T15:03:25,373777168+00:00 - nginx/run
2021-02-10T15:03:25,373777468+00:00 - iot-server/run
/usr/sbin/nginx: /azureml-envs/azureml_7ade26eb614f97df8030bc480da59236/lib/libcrypto.so.1.0.0: no version information available (required by /usr/sbin/nginx)
/usr/sbin/nginx: /azureml-envs/azureml_7ade26eb614f97df8030bc480da59236/lib/libcrypto.so.1.0.0: no version information available (required by /usr/sbin/nginx)
/usr/sbin/nginx: /azureml-envs/azureml_7ade26eb614f97df8030bc480da59236/lib/libssl.so.1.0.0: no version information available (required by /usr/sbin/nginx)
/usr/sbin/nginx: /azureml-envs/azureml_7ade26eb614f97df8030bc480da59236/lib/libssl.so.1.0.0: no version information available (required by /usr/sbin/nginx)
2021-02-10T15:03:25,375284814+00:00 - rsyslog/run
2021-02-10T15:03:25,373993375+00:00 - gunicorn/run
rsyslogd: /azureml-envs/azureml_7ade26eb614f97df8030bc480da59236/lib/libuuid.so.1: no version information available (required by rsyslogd)
EdgeHubConnectionString and IOTEDGE_IOTHUBHOSTNAME are not set. Exiting...
2021-02-10T15:03:25,571769409+00:00 - iot-server/finish 1 0
2021-02-10T15:03:25,573333157+00:00 - Exit code 1 is normal. Not restarting iot-server.
Starting gunicorn 19.9.0
Listening at: http://127.0.0.1:31311 (http://127.0.0.1:31311) (14)
Using worker: sync
worker timeout is set to 300
Booting worker with pid: 42
SPARK_HOME not set. Skipping PySpark Initialization.
Generating new fontManager, this may take some time...
Initializing logger
2021-02-10 15:03:27,076 | root | INFO | Starting up app insights client
2021-02-10 15:03:27,076 | root | INFO | Starting up request id generator
2021-02-10 15:03:27,076 | root | INFO | Starting up app insight hooks
2021-02-10 15:03:27,077 | root | INFO | Invoking user's init function
2021-02-10 15:03:29,501 | root | INFO | Users's init has completed successfully
/azureml-envs/azureml_7ade26eb614f97df8030bc480da59236/lib/python3.6/site-packages/sklearn/externals/joblib/__init__.py:15: FutureWarning: sklearn.externals.joblib is deprecated in 0.21 and will be removed in 0.23. Please import this functionality directly from joblib, which can be installed with: pip install joblib. If this warning is raised when loading pickled models, you may need to re-serialize those models with scikit-learn 0.21+.
warnings.warn(msg, category=FutureWarning)
2021-02-10 15:03:29,503 | root | INFO | Skipping middleware: dbg_model_info as it's not enabled.
2021-02-10 15:03:29,504 | root | INFO | Skipping middleware: dbg_resource_usage as it's not enabled.
2021-02-10 15:03:29,505 | root | INFO | Scoring timeout is found from os.environ: 60000 ms
2021-02-10 15:03:43,119 | root | INFO | Swagger file not present
2021-02-10 15:03:43,120 | root | INFO | 404
127.0.0.1 - - [10/Feb/2021:15:03:43 +0000] "GET /swagger.json HTTP/1.0" 404 19 "-" "Go-http-client/1.1"
2021-02-10 15:03:45,009 | root | INFO | Swagger file not present
2021-02-10 15:03:45,009 | root | INFO | 404
127.0.0.1 - - [10/Feb/2021:15:03:45 +0000] "GET /swagger.json HTTP/1.0" 404 19 "-" "Go-http-client/1.1"
2021-02-10 15:03:45,512 | root | INFO | Validation Request Content-Type
2021-02-10 15:03:45,512 | root | INFO | Scoring Timer is set to 60.0 seconds
2021-02-10 15:03:45,577 | root | INFO | 200
127.0.0.1 - - [10/Feb/2021:15:03:45 +0000] "POST /score HTTP/1.0" 200 30 "-" "python-requests/2.25.1"
2021-02-10 15:11:24,068 | root | INFO | Swagger file not present
2021-02-10 15:11:24,068 | root | INFO | 404
127.0.0.1 - - [10/Feb/2021:15:11:24 +0000] "GET /swagger.json HTTP/1.0" 404 19 "-" "Go-http-client/1.1"
2021-02-10 15:11:51,759 | root | INFO | Validation Request Content-Type
2021-02-10 15:11:51,760 | root | INFO | Scoring Timer is set to 60.0 seconds
2021-02-10 15:11:51,867 | root | INFO | 200
127.0.0.1 - - [10/Feb/2021:15:11:51 +0000] "POST /score HTTP/1.0" 200 30 "-" "python-requests/2.23.0"

```
In [27]: try:
        service.delete()
        compute_target.delete()
except:
    print('Already deleted')
else:
    compute_target.wait_for_completion(show_output=False, is_delete_operation=True)
```

Current provisioning state of AmlCompute is "Deleting"

Current provisioning state of AmlCompute is "Deleting"

Current provisioning state of AmlCompute is "Deleting"

Current provisioning state of AmlCompute is "Deleting"

Current provisioning state of AmlCompute is "Deleting"

Provisioning operation finished, operation "Succeeded"