Import Standard Python Modules

In [1]:
```
import datetime
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
import joblib
```

Import Azure ML SDK modules

In [2]:
```
import azureml.core
from azureml.core import Workspace
from azureml.core.model import Model
from azureml.core import Experiment
from azureml.core.webservice import Webservice
from azureml.core.image import ContainerImage
from azureml.core.webservice import AciWebservice
from azureml.core.conda_dependencies import CondaDependencies
```

Create Azure ML Workspace

In [4]:
```
from azureml.core.authentication import InteractiveLoginAuthentication
interactive_auth = InteractiveLoginAuthentication(tenant_id="48a6f912-76ab-4bcd-ad7a-0efa2d1a9b13")
AZ_SUBSCRIPTION_ID='54c4256e-bb50-4fbd-895d-da32982a5dad'
ws = Workspace.create(name='salary',
                      subscription_id=AZ_SUBSCRIPTION_ID,
                      resource_group='ashish',
                      create_resource_group=True,
                      location='southeastasia'
                     )
```

```
Deploying KeyVault with name salarykeyvaultb8d19db9d7.
Deploying StorageAccount with name salarystorage61460099751.
Deploying AppInsights with name salaryinsights189e9a48ba.
Deployed AppInsights with name salaryinsights189e9a48ba. Took 8.52 seconds.
Deployed KeyVault with name salarykeyvaultb8d19db9d7. Took 29.6 seconds.
Deployed StorageAccount with name salarystorage61460099751. Took 37.91 seconds.
Deploying Workspace with name salary.
Deployed Workspace with name salary. Took 99.6 seconds.
```

Write configuration to local file

In [5]:
```
ws.write_config()
```

Create Azure ML Experiment

In [6]:
```
exp = Experiment(workspace=ws, name='salexp')
```

Start logging metrics

In [7]:
```
run = exp.start_logging()
run.log("Experiment start time", str(datetime.datetime.now()))
```

Load salary dataset

In [9]:
```
sal = pd.read_csv(r'C:\Users\Jaswanth Reddy\Downloads\Ashish salary(Deployment)\data\sal.csv',header=0, index_co
X = sal[['x']]
y = sal['y']
```

Split into train and test data

In [10]:
```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.25, random_state=10)
```

Train the model

In [11]:
```
lm = LinearRegression()
lm.fit(X_train,y_train)
```

Out[11]:
```
LinearRegression()
```

Freeze the model

In [12]:
```
filename = 'outputs/sal_model.pkl'
joblib.dump(lm, filename)
```

Out[12]:
```
['outputs/sal_model.pkl']
```

Test the model

In [13]:
```
filename = 'outputs/sal_model.pkl'
loaded_model=joblib.load(filename)
y=loaded_model.predict([[21]])[0]
print(y)
```

```
141760.56910569107
```

log metrics to Azure ML Experiment

In [14]:
```
run.log('Intercept :', lm.intercept_)
run.log('Slope :', lm.coef_[0])
```

End Azure ML Experiment

In [15]:
```
run.log("Experiment end time", str(datetime.datetime.now()))
run.complete()
```

Get Portal URL

In [16]:
```
print(run.get_portal_url())
```

```
https://ml.azure.com/experiments/salexp/runs/54dae4af-0b34-4a4a-bf5f-b20ff912a299?wsid=/subscriptions/54c4256e-bb50-4fbd-895d-da32982a5dad/resourcegroups/ashish/workspaces/salary (https://ml.azure.com/experiments/salexp/runs/54dae4af-0b34-4a4a-bf5f-b20ff912a299?wsid=/subscriptions/54c4256e-bb50-4fbd-895d-da32982a5dad/resourcegroups/ashish/workspaces/salary)
```

Register the model

In [17]:
```
model = Model.register(model_path = "outputs/sal_model.pkl",
                       model_name = "sal_model",
                       tags = {"key": "1"},
                       description = "Salary Prediction",
                       workspace = ws)
```

```
Registering model sal_model
```

Define Azure ML Deployment configuration

In [18]:
```
aciconfig = AciWebservice.deploy_configuration(cpu_cores=1,
                                               memory_gb=1,
                                               tags={"data": "Salary",  "method" : "sklearn"},
                                               description='Predict Stackoverflow Salary')
```

Create environmental configuration

In [19]:
```
salenv = CondaDependencies()
salenv.add_conda_package("scikit-learn")

with open("salenv.yml","w") as f:
    f.write(salenv.serialize_to_string())
with open("salenv.yml","r") as f:
    print(f.read())
```

```
# Conda environment specification. The dependencies defined in this file will

# be automatically provisioned for runs with userManagedDependencies=False.


# Details about the Conda environment file format:

# https://conda.io/docs/user-guide/tasks/manage-environments.html#create-env-file-manually (https://conda.io/docs/user-guide/tasks/manage-environments.html#create-env-file-manually)


name: project_environment
dependencies:
  # The python interpreter version.

  # Currently Azure ML only supports 3.5.2 and later.

- python=3.6.2

- pip:
    # Required packages for AzureML execution, history, and data preparation.

  - azureml-defaults

- scikit-learn
channels:
- anaconda
- conda-forge
```

Create Azure ML Scoring file

In [20]:
```
%%writefile score.py
import json
import numpy as np
import os
import pickle
import joblib
from sklearn.linear_model import LogisticRegression

from azureml.core.model import Model

def init():
    global model
    # retrieve the path to the model file using the model name
    model_path = Model.get_model_path('sal_model')
    model = joblib.load(model_path)

def run(raw_data):
    data = np.array(json.loads(raw_data)['data'])
    # make prediction
    y_hat = model.predict(data)
    return json.dumps(y_hat.tolist())
```

```
Overwriting score.py
```

Deploy the model to Azure Container Instance

In [21]:
```
%%time
image_config = ContainerImage.image_configuration(execution_script="score.py",
                                                  runtime="python",
                                                  conda_file="salenv.yml")
```

```
Wall time: 0 ns

<timed exec>:1: DeprecationWarning: ContainerImage class has been deprecated and will be removed in a future r
elease. Please migrate to using Environments. https://docs.microsoft.com/en-us/azure/machine-learning/how-to-u
se-environments (https://docs.microsoft.com/en-us/azure/machine-learning/how-to-use-environments)
```

In [ ]:
```
t url there open models and deploy(load .py,.yml files) after that deploy and at last check endpoint to know stat
```

Test with Post URL

In [29]:
```
import requests
data={'data':[[20]]}
url="http://5a1b8455-6f29-4312-a514-b057a21944e1.southeastasia.azurecontainer.io/score"
response=requests.post(url,json=data)
response.json()
```

Out[29]:
```
'[139920.39295392955]'
```

In [ ]:
```

```