In [44]:

```python
import tensorflow as tf
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

In [45]:

```python
# Importing csv file and stroring in df
df=pd.read_csv(r"C:\Users\Jaswanth Reddy\Downloads\insurance.csv")
df.head()
```

Out[45]:

	age	sex	bmi	children	smoker	region	charges
0	19	female	27.900	0	yes	southwest	16884.92400
1	18	male	33.770	1	no	southeast	1725.55230
2	28	male	33.000	3	no	southeast	4449.46200
3	33	male	22.705	0	no	northwest	21984.47061
4	32	male	28.880	0	no	northwest	3866.85520

In [46]:

```
# Converting categorical column values into one hot encoding
categorical_columns = ['sex', 'smoker', 'region']
df = pd.get_dummies(data = df, columns = categorical_columns)
df
```

Out[46]:

	age	bmi	children	charges	sex_female	sex_male	smoker_no	smoker_yes	region_northeast	region_northwest	region_sou
0	19	27.900	0	16884.92400	1	0	0	1	0	0	
1	18	33.770	1	1725.55230	0	1	1	0	0	0	
2	28	33.000	3	4449.46200	0	1	1	0	0	0	
3	33	22.705	0	21984.47061	0	1	1	0	0	1	
4	32	28.880	0	3866.85520	0	1	1	0	0	1	
...	...	...	...	...	...	...	...	...	...	...	
1333	50	30.970	3	10600.54830	0	1	1	0	0	1	
1334	18	31.920	0	2205.98080	1	0	1	0	1	0	
1335	18	36.850	0	1629.83350	1	0	1	0	0	0	
1336	21	25.800	0	2007.94500	1	0	1	0	0	0	
1337	61	29.070	0	29141.36030	1	0	0	1	0	1	

1338 rows × 12 columns

In [47]:

```
x=df.drop(['charges'],axis="columns")
target=df['charges']
```

In [48]:

```
x=x.drop(columns=['sex_female','region_southeast'])
x
```

Out[48]:

	age	bmi	children	sex_male	smoker_no	smoker_yes	region_northeast	region_northwest	region_southwest
0	19	27.900	0	0	0	1	0	0	1
1	18	33.770	1	1	1	0	0	0	0
2	28	33.000	3	1	1	0	0	0	0
3	33	22.705	0	1	1	0	0	1	0
4	32	28.880	0	1	1	0	0	1	0
...	...	...	...	...	...	...	...	...	...
1333	50	30.970	3	1	1	0	0	1	0
1334	18	31.920	0	0	1	0	1	0	0
1335	18	36.850	0	0	1	0	0	0	0
1336	21	25.800	0	0	1	0	0	0	1
1337	61	29.070	0	0	0	1	0	1	0

1338 rows × 9 columns

In [49]:

```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,target,test_size=0.2,random_state=1)
tf.keras.utils.normalize(x_train)
```

Out[49]:

	age	bmi	children	sex_male	smoker_no	smoker_yes	region_northeast	region_northwest	region_southwest
216	0.893498	0.448435	0.000000	0.000000	0.016858	0.000000	0.000000	0.016858	0.000000
731	0.926698	0.374176	0.017485	0.017485	0.017485	0.000000	0.000000	0.000000	0.017485
866	0.434455	0.900046	0.000000	0.024136	0.024136	0.000000	0.000000	0.000000	0.000000
202	0.928068	0.371768	0.000000	0.000000	0.015468	0.000000	0.000000	0.015468	0.000000
820	0.799921	0.599052	0.017776	0.017776	0.017776	0.000000	0.000000	0.000000	0.017776
...	...	...	...	...	...	...	...	...	...
715	0.900632	0.433804	0.000000	0.015011	0.015011	0.000000	0.000000	0.000000	0.015011
905	0.661743	0.747134	0.050903	0.000000	0.025452	0.000000	0.025452	0.000000	0.000000
1096	0.824169	0.564959	0.032320	0.000000	0.000000	0.016160	0.016160	0.000000	0.000000
235	0.873136	0.485027	0.043657	0.000000	0.000000	0.021828	0.000000	0.000000	0.000000
1061	0.897594	0.439979	0.015747	0.015747	0.015747	0.000000	0.000000	0.000000	0.000000

1070 rows × 9 columns

In [50]:

```
from sklearn.linear_model import LinearRegression

model = LinearRegression()
```

In [51]:

```
model.fit(x_train,y_train)
```

Out[51]:

```
LinearRegression()
```

```
In [52]: model.predict(x_test)
```

```
Out[52]: array([ 4383.68089988, 12885.03892192, 12589.21653212, 13286.22919217,
                  544.72832757, 32117.58400779, 12919.04237221, 12318.62183013,
                 3784.29145555, 29468.45725408, 11002.8139431 , 17539.69473777,
                 8681.35471964,  8349.04325528,  3130.12725504, 10445.83896118,
                 3863.74357865,  6944.62510786, 15009.63121084, 14441.59911874,
                12543.65768867, 32958.72553095,  9072.63608136,  8986.85860053,
                 3022.85773294,  8164.97136102,  9556.07558002, 10743.20363927,
                 7694.01743692,  4373.43771674, 14140.93557984,  5811.78545062,
                34631.91316718, 27009.11191231, 33348.14098668,  9532.96786929,
                30421.65017927, 26648.91186842, 15157.78333287, 33895.76121465,
                 6303.38552088, 14059.15156303, 10713.4467824 , 15089.36171493,
                 4187.95334069, 13106.4297513 ,  4336.19603407, 28607.05556216,
                 7243.57117377, 14269.4643165 , 13282.36924936, 12329.61280721,
                 1851.87215658,  8876.2837892 , 26089.18341811, 10125.8221046 ,
                34218.77265378, 14537.70022165,  3232.07805794,  5889.64309508,
                 6558.45711628, 14952.73214832, 26943.84457634,  3272.57672674,
                15795.18877494, 11220.12036023, 11132.67761401, 10461.51218201,
                 1520.17580687, 25268.32319722, 37555.4332681 , 33131.32070966,
                 1986.54437212, 11348.45648105, 13683.62487834, 34970.76597049,
                 3194.05204265,  3875.19388449, 10355.84468565, 10429.85383112,
                  -74.18168095, 14069.96921025, 10335.95235396,  3160.49129709,
                33495.55139469, 33108.38629603,  7159.042252  , 37712.17792565,
                12860.01613403, 10312.33535752, 30118.39165257, 33999.155218  ,
                14744.35977759, 10797.48057723,   228.32604517, 10550.25751993,
                 9637.2654186 , 14963.62716464, 14973.49438453,  6077.52837971,
                13679.44499708, 26048.6188477 , 28140.15460801, 27428.44651929,
                35323.96326034, 27120.17093173,   635.73242244,  9265.30720109,
                 4700.17995399, 12458.33462103,  5334.04136712,  4797.80959774,
                 1053.28620015, 18801.23368294,  3268.21781045,  1680.06692797,
                11731.45541277, 12594.4560403 , 11876.24500234,  3722.26917923,
                 8907.38977334, 13909.79277731,  7727.28039545,  6573.92347482,
                36668.28291771, 12172.54974158, 12246.4759298 , 29298.69540744,
                36065.08836969, 11635.06903459, 28119.47917939,  -420.5228157 ,
                 8255.48679122, 31611.56891923,  8278.51950655,  -682.91733795,
                 1175.50251941,  4610.52460783,  7592.72365991, 12602.74525758,
                14871.84794414,  8696.2661006 , 28916.17140639, 15712.12938325,
                14688.56307722, 11117.34115616,  1910.78149758, 10065.51386262,
                 3785.83713249,  6165.85822972, 11400.42215978,  5505.08475585,
                14580.76982237, 13691.35579602, 12694.51188244,  7023.42319484,
                12388.68766385, 10922.09183278, 10269.55783904,  4543.27270357,
                 5648.10144357, 40390.9900769 , 13059.47316213,  4308.66813543,
                 8433.53823713,  4680.92297563, 32207.14761827, 11261.09752853,
                10966.92628193,  6893.83017801,  6439.49932262,  6698.81354717,
                33082.53354683, 34892.66990169,  2163.75212652,  7664.10129233,
                 5208.63123781, 15537.4388228 ,  1472.95942494, 11431.38761905,
                13442.52462926, 11497.84155642, 10547.85065715, 13216.06609157,
                 2392.9275311 , 27535.86192673,  2350.29363146, 14750.02090702,
                 6294.4943912 , 10590.51504221, 14975.55458721, 38857.75707767,
                 2100.48817818,  1489.62172706,  5170.63120404,  7556.77055613,
                 7905.80683902,  4503.61764622, 10680.78553577,  8938.12057203,
                 9389.70713251, 11104.75136012, 10325.31689891,  9247.40925093,
                 8075.54835929,   895.79174623, 10136.82246673,  7306.72664577,
                 6626.07986045, 11706.84936779,  5409.99685749, 32864.25315855,
                 7088.39118065,  6309.6941707 ,  7934.10447803, 38948.10610123,
                11941.19483711, 28316.17975841,  2882.4783976 , 33202.36401978,
                 3690.60862539, 31577.22772525, 13825.53657174,  2716.91852953,
                 1908.80043495,  1262.92212969,  6109.40830379,  4463.80387639,
                25580.05728181, 15737.66640221,  5345.8549026 , 13030.85900261,
                38954.05091304,  4792.05740177, 12711.42561622, 11335.66208015,
                27785.54316341,  2794.86874955, 13392.79241645,  5727.91540048,
                15215.43600554,  5772.15783816, 16929.82927411,  3896.74375465,
                12197.3470759 , 34682.24329155, 10666.53272796, 10601.36016707,
                 4875.20490336, 16734.59399629, 14399.64496923,  5497.30018065,
                11149.82336777, 12497.70437379,  4626.74808217,  7169.33486073,
                27667.13758601, 32240.5545494 ,  -474.41779055, 40306.05467371,
                 9397.25562995,  7750.27185181, 10671.66257411, 33555.1844395 ,
                35949.5230514 , 36650.46723087,  4961.92884343,  6116.92057448])
```

```
In [53]: import numpy as np
         a=pd.DataFrame([18  ,33.770 ,1  ,1  ,1  ,0  ,0  ,0  ,0])
         a=np.array(a)
         a
```

```
Out[53]: array([[18.  ],
                [33.77],
                [ 1.  ],
                [ 1.  ],
                [ 1.  ],
                [ 0.  ],
                [ 0.  ],
                [ 0.  ],
                [ 0.  ]])
```

In [54]:

```
a=a.reshape(1,-1)
a.shape
```

Out[54]:

```
(1, 9)
```

In [55]:

```
model.predict(a)
```

Out[55]:

```
array([3325.91782581])
```

In [21]:

```
import joblib
joblib.dump(model,"insurance_prediction.pkl")
```

Out[21]:

```
['insurance_prediction.pkl']
```

In [22]:

```
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
```

In [23]:

```
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
```

In [24]:

```

AZ_SUBSCRIPTION_ID='54c4256e-bb50-4fbd-895d-da32982a5dad'
ws = Workspace.create(name='insurance_data',
 subscription_id=AZ_SUBSCRIPTION_ID,
resource_group='Jaswanth_4',
create_resource_group=True,
location='centralindia'
 )
```

```
UserWarning: The resource group doesn't exist or was not provided. AzureML SDK is creating a resource group=Ja
swanth_4 in location=centralindia using subscription=54c4256e-bb50-4fbd-895d-da32982a5dad.

Deploying StorageAccount with name insurancstorage1780f4f78.
Deploying AppInsights with name insurancinsights5176ac46.
Deployed AppInsights with name insurancinsights5176ac46. Took 6.97 seconds.
Deploying KeyVault with name insuranckeyvault74420c05.
Deployed KeyVault with name insuranckeyvault74420c05. Took 21.57 seconds.
Deployed StorageAccount with name insurancstorage1780f4f78. Took 22.25 seconds.
Deploying Workspace with name insurance_data.
Deployed Workspace with name insurance_data. Took 41.84 seconds.
```

In [25]:

```
ws.write_config()
```

In [26]:

```
exp = Experiment(workspace=ws, name='insexp')
```

In [27]:

```
run = exp.start_logging(snapshot_directory=None)
run.log("Experiment start time", str(datetime.datetime.now()))
```

In [28]:

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

In [29]:

```
run.log("Experiment end time", str(datetime.datetime.now()))
run.complete()
```

In [30]:

```
print(run.get_portal_url())
```

```
https://ml.azure.com/experiments/insexp/runs/c7f7b381-38ae-4421-84f1-181d893f8340?wsid=/subscriptions/54c4256e
-bb50-4fbd-895d-da32982a5dad/resourcegroups/Jaswanth_4/workspaces/insurance_data (https://ml.azure.com/experim
ents/insexp/runs/c7f7b381-38ae-4421-84f1-181d893f8340?wsid=/subscriptions/54c4256e-bb50-4fbd-895d-da32982a5da
d/resourcegroups/Jaswanth_4/workspaces/insurance_data)
```

In [33]:

```
model = Model.register(model_path = "insurance_prediction.pkl",
                       model_name = "insurance",
                       tags = {"key": "1"},
                       description = "insurance_chun Prediction",
                       workspace = ws)
```

```
Registering model insurance
```

In [34]:

```
aciconfig = AciWebservice.deploy_configuration(cpu_cores=1,
                                               memory_gb=1,
                                               tags={"data": "insurance",  "method" : "sklearn"},
                                               description='Predict insurance_chun')
```

In [36]:

```python
insurancenv = CondaDependencies()
insurancenv.add_conda_package("scikit-learn")

with open("insurancenv.yml","w") as f:
    f.write(insurancenv.serialize_to_string())
with open("insurancenv.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
```

In [37]:

```
%%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('insurance')
    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())
```

```
Writing score.py
```

In [57]:

```
import requests

data={'data':[[18   ,33.770 ,1  ,1  ,1  ,0  ,0  ,0  ,0]]}
url="http://94539de5-9081-4eb3-8b8a-bf9ff00f26c1.centralindia.azurecontainer.io/score"
response=requests.post(url,json=data)
response.json()
```

Out[57]:

```
'[3325.917825813391]'
```

In [ ]: