In [1]:
```
import tensorflow as tf
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from tensorflow import keras
from tensorflow.keras import layers
```

In [2]:
```
# Importing csv file and stroring in df
df=pd.read_csv(r"C:\Users\Jaswanth Reddy\Downloads\insurance.csv")
df.head()
```

Out[2]:

	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 [3]:
```
# Converting categorical column values into one hot encoding
categorical_columns = ['sex', 'smoker', 'region']
df = pd.get_dummies(data = df, columns = categorical_columns)
df
```

Out[3]:

	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 [4]:
```
x=df.drop(['charges'],axis="columns")
target=df['charges']
```

In [5]:
```
x=x.drop(columns=['sex_female','region_southeast'])
x
```

Out[5]:

	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 [6]:
```
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[6]:

	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 [7]:
```
model=tf.keras.Sequential()
model.add(tf.keras.layers.Dense((64),input_shape=(9,),activation='relu'))
model.add(tf.keras.layers.Dense((32),activation='relu'))
model.add(keras.layers.Dense(1))
```

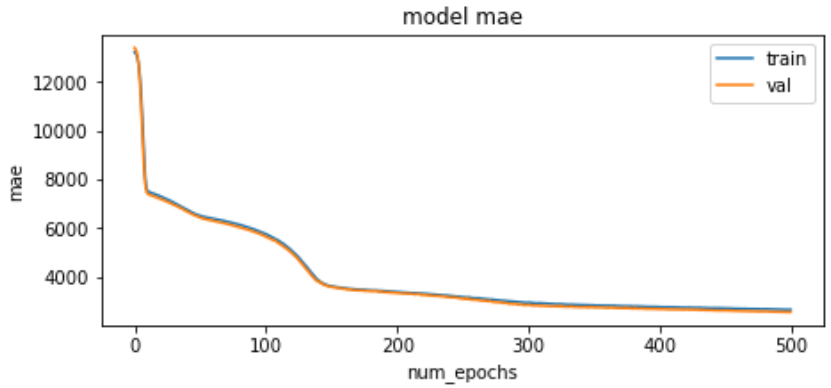
In [8]:
```
# Applying the optimizer,loss and metrics function in order to minimize the error
model.compile(optimizer=keras.optimizers.Adam(0.001),loss='mae',metrics=['mae','mse'])
```

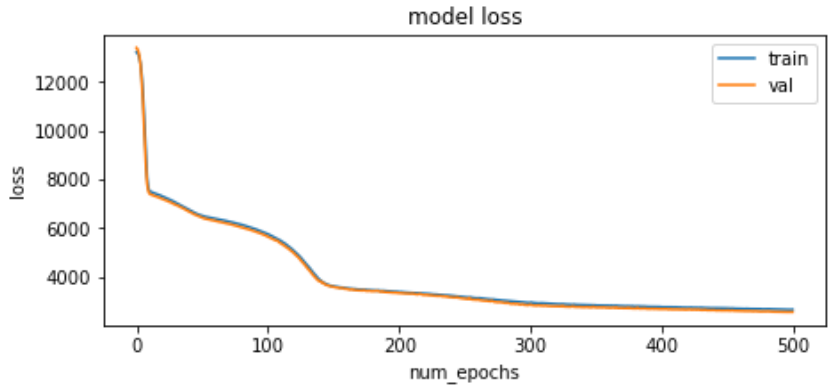
In [9]:
```
num_epochs=500
training_history=model.fit(x_train,y_train,epochs=num_epochs,validation_data=(x_test,y_test),verbose=1)
```

```
34/34 [==============================] - 0s 3ms/step - loss: 2681.3650 - mae: 2681.3650 - mse: 33950064.0000
- val_loss: 2594.3962 - val_mae: 2594.3962 - val_mse: 33770764.0000
Epoch 482/500
34/34 [==============================] - 0s 2ms/step - loss: 2684.7256 - mae: 2684.7256 - mse: 34079344.0000
- val_loss: 2586.7402 - val_mae: 2586.7402 - val_mse: 33786368.0000
Epoch 483/500
34/34 [==============================] - 0s 3ms/step - loss: 2678.0486 - mae: 2678.0486 - mse: 34099672.0000
- val_loss: 2585.7400 - val_mae: 2585.7400 - val_mse: 33685740.0000
Epoch 484/500
34/34 [==============================] - 0s 2ms/step - loss: 2679.1719 - mae: 2679.1719 - mse: 34115976.0000
- val_loss: 2581.1765 - val_mae: 2581.1765 - val_mse: 33741708.0000
Epoch 485/500
34/34 [==============================] - 0s 2ms/step - loss: 2673.8538 - mae: 2673.8538 - mse: 34008844.0000
- val_loss: 2581.5078 - val_mae: 2581.5078 - val_mse: 33679608.0000
Epoch 486/500
34/34 [==============================] - 0s 3ms/step - loss: 2676.6274 - mae: 2676.6274 - mse: 34124080.0000
- val_loss: 2576.7031 - val_mae: 2576.7031 - val_mse: 33683744.0000
Epoch 487/500
34/34 [==============================] - 0s 3ms/step - loss: 2674.5569 - mae: 2674.5569 - mse: 33863832.0000
- val_loss: 2576.7424 - val_mae: 2576.7424 - val_mse: 33668684.0000
```

In [10]:
```
# Comparing trained data(MAE) with test data
plt.figure(figsize=(16,3))
plt.subplot(1,2,1)
plt.plot(training_history.history['mae'])
plt.plot(training_history.history['val_mae'])
plt.title("model mae")
plt.xlabel('num_epochs')
plt.ylabel('mae')
plt.legend(['train','val'])
plt.subplot(1,2,2)
plt.plot(training_history.history['loss'])
plt.plot(training_history.history['val_loss'])
plt.title("model loss")
plt.ylabel('loss')
plt.xlabel('num_epochs')
plt.legend(['train','val'])
```

Out[10]: <matplotlib.legend.Legend at 0x209c9396a30>





In [11]:
```
a=pd.DataFrame([28  ,33.000 ,3  ,1  ,1  ,0  ,0  ,0  ,0])
a=np.array(a)
a
```

Out[11]:
```
array([[28.],
       [33.],
       [ 3.],
       [ 1.],
       [ 1.],
       [ 0.],
       [ 0.],
       [ 0.],
       [ 0.]])
```

In [12]:
```
a=a.reshape(1,-1)
a.shape
```

Out[12]:
```
(1, 9)
```

In [28]:
```
model.predict(a)
```

Out[28]:
```
array([[4389.7236]], dtype=float32)
```

In [14]:
```
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 [15]:
```
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 [16]:
```
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='insurance_data',
 subscription_id=AZ_SUBSCRIPTION_ID,
resource_group='JaswanthReddy',
create_resource_group=True,
location='centralindia'
 )
```

```
Deploying KeyVault with name insuranckeyvaultf9f74d16.
Deploying StorageAccount with name insurancstorage1a0f28b01.
Deploying AppInsights with name insurancinsightsd724a426.
Deployed AppInsights with name insurancinsightsd724a426. Took 5.68 seconds.
Deployed KeyVault with name insuranckeyvaultf9f74d16. Took 20.21 seconds.
Deployed StorageAccount with name insurancstorage1a0f28b01. Took 27.41 seconds.
Deploying Workspace with name insurance_data.
Deployed Workspace with name insurance_data. Took 40.16 seconds.
```

In [17]:
```
ws.write_config()
```

In [18]:
```
exp = Experiment(workspace=ws, name='salexp')
```

In [20]:
```
run = exp.start_logging(snapshot_directory=None)
run.log("Experiment start time", str(datetime.datetime.now()))
```

In [29]:
```
filename = 'outputs/insurane_model.pkl'
joblib.dump(model, filename)
```

```
    968         self.memoize(obj)
--> 969         self._batch_setitems(obj.items())
    970
    971     dispatch[dict] = save_dict

D:\Python\lib\pickle.py in _batch_setitems(self, items)
    993                 for k, v in tmp:
    994                     save(k)
--> 995                     save(v)
    996                 write(SETITEMS)
    997             elif n:

D:\Python\lib\site-packages\joblib\numpy_pickle.py in save(self, obj)
    280             return
    281
--> 282         return Pickler.save(self, obj)
    283
    284
```

In [25]:
```
run.log("Experiment end time", str(datetime.datetime.now()))
run.complete()
```

In [26]:
```
print(run.get_portal_url())
```

```
https://ml.azure.com/experiments/salexp/runs/021dcafe-25d9-48cb-9ccb-d08087c1917c?wsid=/subscriptions/54c4256e-bb50-4fbd-895d-da32982a5dad/resourcegroups/JaswanthReddy/workspaces/insurance_data (https://ml.azure.com/experiments/salexp/runs/021dcafe-25d9-48cb-9ccb-d08087c1917c?wsid=/subscriptions/54c4256e-bb50-4fbd-895d-da32982a5dad/resourcegroups/JaswanthReddy/workspaces/insurance_data)
```

In [30]:
```
model = Model.register(model_path = 'outputs/insurane_model.pkl',
 model_name = "insurance_model",
 tags = {"key": "1"},
 description = "Insurance Prediction",
 workspace = ws)
```

```
Registering model insurance_model
```

In [31]:
```
aciconfig = AciWebservice.deploy_configuration(cpu_cores=1,
 memory_gb=1,
tags={"data": "Insurance", "method" : "sklearn"},
 description='Insurance calculator')
```

In [32]:
```
salenv = CondaDependencies()
salenv.add_conda_package("scikit-learn")
with open("insurance.yml","w") as f:
    f.write(salenv.serialize_to_string())
with open("insurance.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 [34]:
```
%%writefile insurance.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')
    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 insurance.py
```

In [36]:
```
%%time
image_config = ContainerImage.image_configuration(execution_script="insurance.py",
 runtime="python",
conda_file="insurance.yml")
```

```
Wall time: 6.97 ms

<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 [48]:
```
import requests
import json
data={'data':[[20]]}
#data=json.dumps({"a":28,"b":33,"c":3,"d":1,"e":1,"f":0,"g":0,"h":0,"i":0})
url="http://54bc1126-2176-49a4-a5b0-9199ed6fdd2b.centralindia.azurecontainer.io/score"
response=requests.post(url,json=data)
response.json()
```

```
D:\Python\lib\site-packages\requests\sessions.py in request(self, method, url, params, data, headers, cookie
s, files, auth, timeout, allow_redirects, proxies, hooks, stream, verify, cert, json)
    528         }
    529         send_kwargs.update(settings)
--> 530         resp = self.send(prep, **send_kwargs)
    531
    532         return resp

D:\Python\lib\site-packages\requests\sessions.py in send(self, request, **kwargs)
    641
    642         # Send the request
--> 643         r = adapter.send(request, **kwargs)
    644
    645         # Total elapsed time of the request (approximately)

D:\Python\lib\site-packages\requests\adapters.py in send(self, request, stream, timeout, verify, cert, proxi
es)
    514                 raise SSLError(e, request=request)
    515
```

In [ ]: