```
import tensorflow as tf
import pandas as pd
import matplotlib.pyplot as plt
from tensorflow import keras
from tensorflow.keras import layers
```

+ Code + Text

```
# Importing csv file and stroring in df
df=pd.read_csv("/content/insurance.csv")
```

```
# Displaying first five rows from the df
df.head()
```

| | 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 |

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

| | age | bmi | children | charges | sex_female | sex_male | smoker_no | smoker_yes | region_northeast | region_northwe |
|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 19 | 27.900 | 0 | 16884.92400 | 1 | 0 | 0 | 1 | 0 | |
| **1** | 18 | 33.770 | 1 | 1725.55230 | 0 | 1 | 1 | 0 | 0 | |
| **2** | 28 | 33.000 | 3 | 4449.46200 | 0 | 1 | 1 | 0 | 0 | |
| **3** | 33 | 22.705 | 0 | 21984.47061 | 0 | 1 | 1 | 0 | 0 | |
| **4** | 32 | 28.880 | 0 | 3866.85520 | 0 | 1 | 1 | 0 | 0 | |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... | |
| **1333** | 50 | 30.970 | 3 | 10600.54830 | 0 | 1 | 1 | 0 | 0 | |
| **1334** | 18 | 31.920 | 0 | 2205.98080 | 1 | 0 | 1 | 0 | 1 | |

```
# Setting input and output columns
model=df.drop(['charges'],axis="columns")
target=df['charges']
```

```
# Displaying target
target
```

```
0       16884.92400
1        1725.55230
2        4449.46200
3       21984.47061
4        3866.85520
           ...
1333    10600.54830
1334     2205.98080
1335     1629.83350
1336     2007.94500
1337    29141.36030
Name: charges, Length: 1338, dtype: float64
```

```
# Displaying input columns
model
```

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

1338 rows × 11 columns

```
# Splitting data into training and testing
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(model,target,test_size=0.2,random_state=1)

tf.keras.utils.normalize(x_train)
```

| | age | bmi | children | sex_female | sex_male | smoker_no | smoker_yes | region_northeast | region_northwest | re |
|---|---|---|---|---|---|---|---|---|---|---|
| **216** | 0.893371 | 0.448371 | 0.000000 | 0.016856 | 0.000000 | 0.016856 | 0.000000 | 0.000000 | 0.016856 | |
| **731** | 0.926698 | 0.374176 | 0.017485 | 0.000000 | 0.017485 | 0.017485 | 0.000000 | 0.000000 | 0.000000 | |
| **866** | 0.434329 | 0.899784 | 0.000000 | 0.000000 | 0.024129 | 0.024129 | 0.000000 | 0.000000 | 0.000000 | |
| **202** | 0.927957 | 0.371724 | 0.000000 | 0.015466 | 0.000000 | 0.015466 | 0.000000 | 0.000000 | 0.015466 | |
| **820** | 0.799921 | 0.599052 | 0.017776 | 0.000000 | 0.017776 | 0.017776 | 0.000000 | 0.000000 | 0.000000 | |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... | |

```
# Training the model
model=tf.keras.Sequential()
model.add(tf.keras.layers.Dense((64),input_shape=(11,),activation='relu'))
model.add(tf.keras.layers.Dense((32),activation='relu'))
model.add(keras.layers.Dense(1)) # number of output layer(column)
```

```
# 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'])
```

```
num_epochs=300
training_history=model.fit(x_train,y_train,epochs=num_epochs,validation_split=0.2,verbose=True)
```

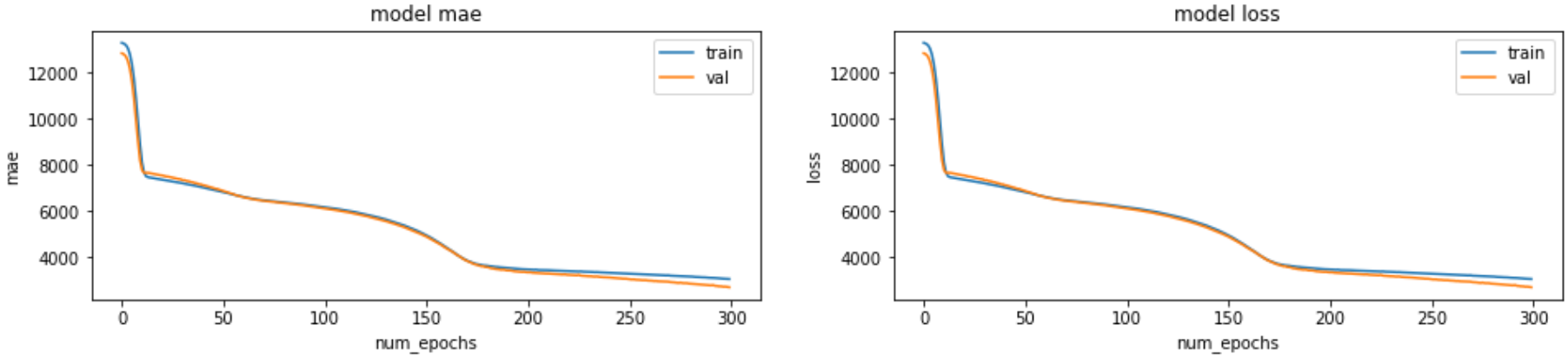
```
Epoch 1/300
27/27 [==============================] - 0s 7ms/step - loss: 13306.5869 - mae: 13306.5869 - mse: 316735392.0000 - va
Epoch 2/300
27/27 [==============================] - 0s 2ms/step - loss: 13277.4346 - mae: 13277.4346 - mse: 315917920.0000 - va
Epoch 3/300
27/27 [==============================] - 0s 2ms/step - loss: 13217.8213 - mae: 13217.8213 - mse: 314244352.0000 - va
Epoch 4/300
27/27 [==============================] - 0s 2ms/step - loss: 13079.8867 - mae: 13079.8867 - mse: 310345120.0000 - va
Epoch 5/300
27/27 [==============================] - 0s 2ms/step - loss: 12819.2021 - mae: 12819.2021 - mse: 303383392.0000 - va
Epoch 6/300
27/27 [==============================] - 0s 2ms/step - loss: 12389.9678 - mae: 12389.9678 - mse: 291914272.0000 - va
Epoch 7/300
27/27 [==============================] - 0s 2ms/step - loss: 11753.2217 - mae: 11753.2217 - mse: 275669696.0000 - va
```

```
Epoch 8/300
27/27 [==============================] - 0s 2ms/step - loss: 10896.9961 - mae: 10896.9961 - mse: 253736432.0000 - va
Epoch 9/300
27/27 [==============================] - 0s 2ms/step - loss: 9891.3145 - mae: 9891.3145 - mse: 228687056.0000 - val_
Epoch 10/300
27/27 [==============================] - 0s 2ms/step - loss: 8901.4365 - mae: 8901.4365 - mse: 203387632.0000 - val_
Epoch 11/300
27/27 [==============================] - 0s 2ms/step - loss: 8099.4224 - mae: 8099.4224 - mse: 181048480.0000 - val_
Epoch 12/300
27/27 [==============================] - 0s 2ms/step - loss: 7667.4185 - mae: 7667.4185 - mse: 164700224.0000 - val_
Epoch 13/300
27/27 [==============================] - 0s 2ms/step - loss: 7509.9214 - mae: 7509.9214 - mse: 154990768.0000 - val_
Epoch 14/300
27/27 [==============================] - 0s 2ms/step - loss: 7467.7461 - mae: 7467.7461 - mse: 151063200.0000 - val_
Epoch 15/300
27/27 [==============================] - 0s 2ms/step - loss: 7452.6279 - mae: 7452.6279 - mse: 149586080.0000 - val_
Epoch 16/300
27/27 [==============================] - 0s 2ms/step - loss: 7436.2095 - mae: 7436.2095 - mse: 149947216.0000 - val_
Epoch 17/300
27/27 [==============================] - 0s 2ms/step - loss: 7420.5410 - mae: 7420.5410 - mse: 150023584.0000 - val_
Epoch 18/300
27/27 [==============================] - 0s 2ms/step - loss: 7408.5098 - mae: 7408.5098 - mse: 149562288.0000 - val_
Epoch 19/300
27/27 [==============================] - 0s 2ms/step - loss: 7392.2886 - mae: 7392.2886 - mse: 149947744.0000 - val_
Epoch 20/300
27/27 [==============================] - 0s 2ms/step - loss: 7376.1787 - mae: 7376.1787 - mse: 149952640.0000 - val_
Epoch 21/300
27/27 [==============================] - 0s 2ms/step - loss: 7360.8076 - mae: 7360.8076 - mse: 149579232.0000 - val_
Epoch 22/300
27/27 [==============================] - 0s 2ms/step - loss: 7346.2720 - mae: 7346.2720 - mse: 149000896.0000 - val_
Epoch 23/300
27/27 [==============================] - 0s 2ms/step - loss: 7329.8037 - mae: 7329.8037 - mse: 149683248.0000 - val_
Epoch 24/300
27/27 [==============================] - 0s 2ms/step - loss: 7314.4883 - mae: 7314.4883 - mse: 149417808.0000 - val_
Epoch 25/300
27/27 [==============================] - 0s 2ms/step - loss: 7298.8950 - mae: 7298.8950 - mse: 148884640.0000 - val_
Epoch 26/300
27/27 [==============================] - 0s 2ms/step - loss: 7284.6108 - mae: 7284.6108 - mse: 149200192.0000 - val_
Epoch 27/300
27/27 [==============================] - 0s 2ms/step - loss: 7269.1748 - mae: 7269.1748 - mse: 149830000.0000 - val_
Epoch 28/300
27/27 [==============================] - 0s 2ms/step - loss: 7251.9697 - mae: 7251.9697 - mse: 149571904.0000 - val_
Epoch 29/300
```

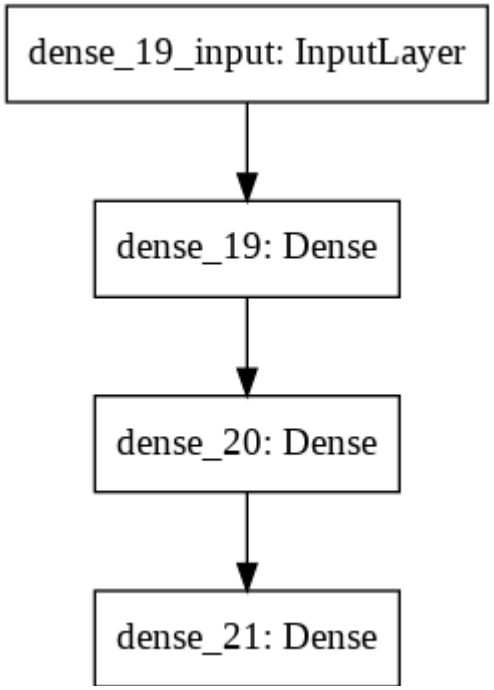
```
# Comparing trained data(MAE) with test data
plt.figure(figsize=(16,3))
plt.subplot(1,2,1)
plt.plot(training_history.history['mae'])
plt.plot(training_history.history['val_mae'])
plt.title("model mae")
plt.ylabel('mae')
plt.xlabel('num_epochs')
plt.legend(['train','val'])

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'])
```

```
<matplotlib.legend.Legend at 0x7f2401ed9e48>
```




```
# Displaying sequential layers graphically
tf.keras.utils.plot_model(model)
```




```
# Evaluating Test data
model.evaluate(x_test,y_test)
```

```
9/9 [==============================] - 0s 1ms/step - loss: 2912.7661 - mae: 2912.7661 - mse: 34377688.0000
[2912.76611328125, 2912.76611328125, 34377688.0]
```