

```
In [4]: greeting = "Assalam-o-Alaikum!"
print(greeting)
```

Assalam-o-Alaikum!

Import Libraries

```
In [6]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

Import Dataset

```
In [7]: df = pd.read_csv("insurance.csv")
df
```

```
Out[7]:
```

	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
...
1333	50	male	30.970	3	no	northwest	10600.54830
1334	18	female	31.920	0	no	northeast	2205.98080
1335	18	female	36.850	0	no	southeast	1629.83350
1336	21	female	25.800	0	no	southwest	2007.94500
1337	61	female	29.070	0	yes	northwest	29141.36030

1338 rows × 7 columns

Firstly, we change "sex" categories into Numeric Values

```
In [11]: df["sex"] = df["sex"].replace({"female":0, "male": 1})
df
```

```
Out[11]:
```

	age	sex	bmi	children	smoker	region	charges
0	19	0	27.900	0	yes	southwest	16884.92400
1	18	1	33.770	1	no	southeast	1725.55230
2	28	1	33.000	3	no	southeast	4449.46200
3	33	1	22.705	0	no	northwest	21984.47061
4	32	1	28.880	0	no	northwest	3866.85520
...
1333	50	1	30.970	3	no	northwest	10600.54830
1334	18	0	31.920	0	no	northeast	2205.98080
1335	18	0	36.850	0	no	southeast	1629.83350
1336	21	0	25.800	0	no	southwest	2007.94500
1337	61	0	29.070	0	yes	northwest	29141.36030

1338 rows × 7 columns

```
In [14]: df["smoker"] = df["smoker"].replace({"yes": 1, "no": 0})
df
```

Out[14]:

	age	sex	bmi	children	smoker	region	charges
0	19	0	27.900	0	1	southwest	16884.92400
1	18	1	33.770	1	0	southeast	1725.55230
2	28	1	33.000	3	0	southeast	4449.46200
3	33	1	22.705	0	0	northwest	21984.47061
4	32	1	28.880	0	0	northwest	3866.85520
...
1333	50	1	30.970	3	0	northwest	10600.54830
1334	18	0	31.920	0	0	northeast	2205.98080
1335	18	0	36.850	0	0	southeast	1629.83350
1336	21	0	25.800	0	0	southwest	2007.94500
1337	61	0	29.070	0	1	northwest	29141.36030

1338 rows × 7 columns

In [15]: df["region"].unique()

Out[15]: array(['southwest', 'southeast', 'northwest', 'northeast'], dtype=object)

In [16]: df["region"] = df["region"].replace({"southwest": 3, "southeast": 2, "northwest": 1, "northeast": 0})
df

Out[16]:

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

1338 rows × 7 columns

In [17]: df["region"] = df["region"].astype("category")
df

Out[17]:

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

1338 rows × 7 columns

In [22]: df.isnull().sum()

```
Out[22]: age      (
sex      (
bmi      (
children (
smoker   (
region   (
charges  (
dtype: int64
```

```
In [23]: x = df.drop(columns = "charges")
          x
```

age	sex	bmi	children	smoker	region	
0	19	0	27.900	0	1	3
1	18	1	33.770	1	0	2
2	28	1	33.000	3	0	2
3	33	1	22.705	0	0	1
4	32	1	28.880	0	0	1
...
1333	50	1	30.970	3	0	1
1334	18	0	31.920	0	0	0
1335	18	0	36.850	0	0	2
1336	21	0	25.800	0	0	3
1337	61	0	29.070	0	1	1

1338 rows × 6 columns

Label

```
In [24]: y = df["charges"]
```

```
In [27]: from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(x, y, train_size = 0.3, random_state = 0)
```

```
In [28]: from sklearn.linear_model import LinearRegression
lr = LinearRegression()
```

```
In [30]: lr.fit(x_train, y_train)
```

```
Out[30]: ▼ LinearRegression
LinearRegression()
```

```
In [31]: c = lr.intercept_
```

In [32]: C

```
Out[32]: -12391.345036658937
```

```
In [33]: m = lr.coef_  
m
```

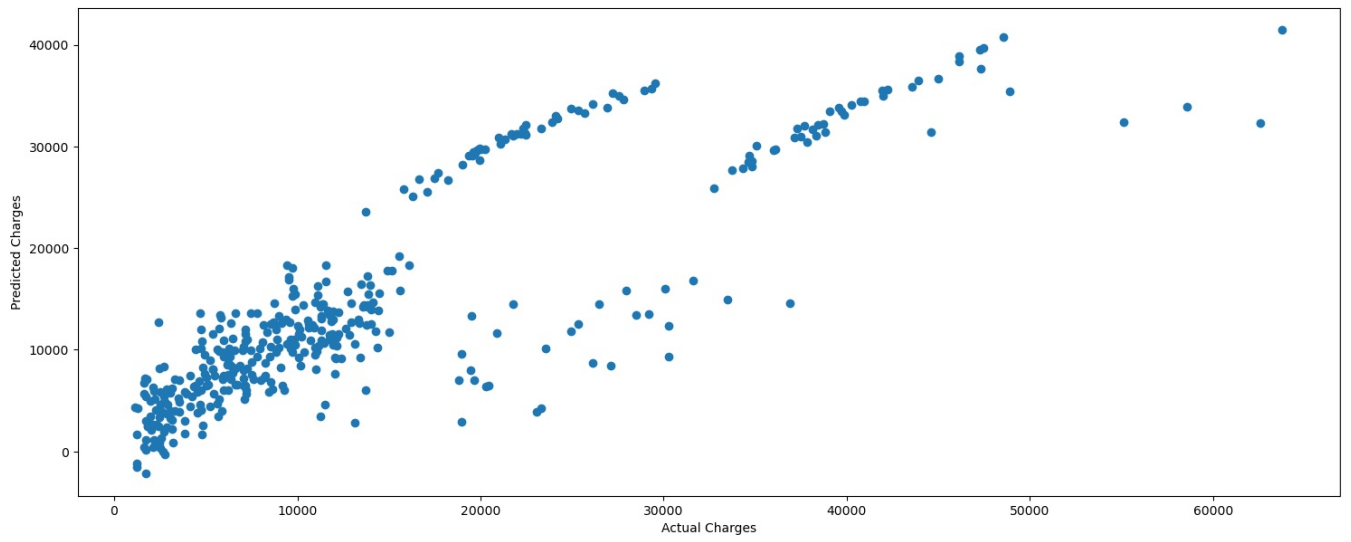
```
Out[33]: array([ 220.96928098, -186.2373625, 408.42427239, 480.17567513,
                23394.65853377, -418.77781747])
```

```
In [35]: y_pred_train = lr.predict(x_train)
y_pred_train
```

```
Out[35]: array([ 8.31662964e+03,  2.95965503e+04,  5.25648956e+03,  1.80536150e+04,
  1.15886877e+04,  8.73738276e+03,  3.55248873e+04,  1.33206749e+04,
  1.83026282e+04,  1.60011987e+04,  3.88814024e+04,  1.36202117e+04,
  3.10667013e+04,  7.63767408e+03,  4.44967619e+03,  9.37406164e+03,
  1.22123505e+04,  1.23676358e+04,  1.00746688e+04,  1.59904934e+04,
  4.36528921e+02,  6.92141513e+03,  1.09625723e+04,  4.64418930e+03,
  1.27126018e+04,  3.38924595e+04,  1.05369031e+04,  2.55038033e+04,
  2.76336174e+04,  1.43820150e+04,  1.46687452e+04,  1.42663114e+04,
  8.20303659e+03,  3.12273226e+04,  1.46352213e+04,  2.96522085e+04,
  7.01183245e+03,  2.14704502e+03,  3.10323960e+04,  8.13595688e+03,
  3.12086569e+04,  3.30102121e+04,  3.34412253e+04,  3.43617807e+03,
  8.41492935e+03,  9.28317555e+03,  1.00969725e+04,  3.21969428e+04,
  2.94706027e+04,  6.85686622e+03,  3.04166165e+04,  1.10130191e+04,
  9.10931735e+03,  4.46672627e+03,  2.98288985e+04,  1.26346161e+04,
  1.55179651e+04,  1.63592007e+04,  7.99100764e+03,  5.12740877e+03,
  5.86994976e+03,  3.18071510e+04,  6.40538988e+03,  6.03384120e+03,
  1.05207170e+04,  1.43833271e+04,  6.03529485e+03,  1.19717837e+04,
  1.55488741e+04,  1.16131185e+04,  4.20884881e+03,  5.13679473e+03])
```

2.91055413e+04,	3.38257152e+04,	1.02876301e+04,	9.99486637e+03,
7.09249178e+03,	1.27237459e+04,	7.46382188e+03,	1.72599105e+04,
3.09414838e+04,	1.54846192e+04,	1.24627912e+04,	1.91774838e+03,
1.54088336e+04,	3.03995642e+03,	8.44499790e+03,	1.26666164e+04,
1.42372778e+04,	7.38644673e+03,	1.38283659e+04,	6.55473317e+03,
1.03558596e+04,	1.09117043e+04,	4.67881221e+03,	1.01453616e+04,
2.86106559e+03,	1.44508172e+04,	4.07903477e+03,	2.87049460e+03,
7.14399748e+03,	1.57738934e+04,	8.44150950e+03,	1.20111376e+04,
3.23878363e+04,	6.52040303e+03,	1.76776468e+03,	1.71240138e+04,
7.42622335e+03,	1.19452615e+04,	1.10046323e+04,	1.03891808e+04,
8.23296998e+03,	1.25008694e+04,	1.68434447e+04,	7.43282847e+03,
1.20275261e+04,	3.57070562e+04,	5.89425353e+03,	7.22453226e+03,
4.07576749e+04,	9.26889568e+03,	5.38968188e+03,	1.12277025e+04,
4.07624514e+03,	3.97307065e+04,	3.34632272e+04,	1.44783395e+04,
-1.18111707e+03,	1.39305561e+04,	3.07293311e+04,	2.91240220e+04,
1.02380552e+04,	5.90042271e+03,	2.51518998e+03,	1.45024248e+04,
3.85690661e+03,	7.28519861e+03,	2.63016286e+03,	2.97645507e+04,
3.18042502e+04,	3.54962880e+04,	1.26456576e+04,	1.92433646e+04,
7.24811309e+03,	3.95024090e+04,	9.89882792e+03,	3.98050850e+03,
1.20674187e+04,	8.48678166e+03,	3.83241902e+04,	1.08932830e+04,
7.56614549e+03,	5.41901909e+03,	3.39588290e+03,	6.07555664e+03,
1.27832286e+04,	2.93464274e+04,	1.17248879e+04,	7.08072596e+03,
2.74012014e+04,	3.02171272e+04,	1.06693471e+04,	2.92809417e+03,
1.29001339e+04,	3.44583105e+04,	3.35578692e+04,	3.13950462e+04,
3.23279907e+03,	1.27345204e+04,	7.72931343e+03,	2.49448289e+03,
4.86657163e+03,	9.55323057e+03,	1.08067158e+04,	9.69559043e+03,
7.67284103e+03,	4.29176818e+03,	1.00902830e+04,	1.42370164e+04,
9.35600251e+03,	6.18083208e+03,	1.57942057e+04,	7.95751089e+03,
-2.17931010e+03,	1.20407692e+04,	1.24871956e+04,	3.27705778e+04,
2.80131661e+04,	1.02513499e+04,	8.17709248e+03,	1.29646319e+04,
2.82398246e+04,	1.78140156e+04,	7.01267264e+03,	1.32955730e+04,
8.72828974e+03,	2.91276931e+04,	8.95739420e+03,	3.12233131e+04,
1.29549169e+04,	6.32641125e+03,	1.83184276e+04,	2.86557026e+04,
1.68733787e+02,	-3.43020181e+01,	4.30987003e+03,	6.07273283e+03,
2.84910312e+04,	4.90734564e+03,	9.11894797e+03,	8.04880898e+03,
1.15297863e+04,	2.67768684e+04,	1.69320631e+04,	3.55812001e+04,
3.21371501e+04,	6.59227201e+03,	3.37480724e+04,	1.53277284e+04,
7.07027143e+03,	7.58594476e+03,	9.76533468e+03,	8.05854845e+03,
1.26984595e+04,	1.00316930e+04,	1.15961842e+04,	8.28428358e+03,
4.88607703e+03,	3.08410416e+04,	3.77996051e+03,	9.54207949e+03,
5.86223468e+03,	4.58596073e+03,	1.18198263e+04,	1.06766278e+03,
1.09994113e+03,	1.21415306e+04,	1.64650893e+04,	8.50016078e+03,
1.01672203e+04,	3.52464184e+04,	3.49715049e+04,	1.25703348e+04,
7.00368854e+03,	2.44392550e+03,	1.25457364e+04,	3.24103917e+04,
1.34724822e+04,	3.49361344e+04,	1.38143728e+04,	9.94274586e+03,
1.25472156e+04,	5.91691950e+03,	5.80087906e+03,	6.12581127e+03,
1.67405984e+04,	1.14987501e+04,	8.79394727e+03,	3.32829994e+04,
3.07680761e+03,	2.50725114e+04,	3.76330451e+04,	1.24036037e+04,
1.18618538e+04,	1.83665480e+04,	3.54330340e+04,	3.41556397e+04,
8.31188121e+03,	9.81864817e+03,	1.34257574e+04,	3.44062583e+04,
1.45630251e+04,	4.63255042e+03,	6.51299086e+03,	7.49313443e+03,
1.58298535e+04,	3.46033274e+04,	6.01899918e+03,	1.34615569e+04,
3.59062850e+04,	9.77643485e+03,	4.96178567e+03,	1.03142706e+04,
4.02871237e+03,	6.44634108e+03,	1.30254920e+04,	4.58919504e+03,
1.35806973e+04,	1.62624403e+04,	3.14282649e+04,	1.20877637e+04,
3.38240011e+04,	1.39404553e+04,	8.72408800e+03,	3.13394201e+04,
3.16810178e+04,	1.14711279e+04,	3.20333475e+04,	1.11123112e+04,
3.03407305e+03,	9.28267119e+03,	1.31664835e+04,	1.10213148e+04,
3.01115675e+04,	1.07623426e+04,	7.09192088e+03,	9.37213685e+03,
2.38721789e+03,	3.65078863e+04,	6.58233864e+03,	3.40754683e+03,
2.55037608e+03,	6.46287966e+03,	2.97034539e+04,	2.68652341e+04,
1.36638031e+04,	1.63701040e+03,	2.97599192e+04,	-1.54869892e+03,
3.76801911e+03,	2.67255461e+04,	1.05628199e+04,	3.15666479e+02,
6.04454737e+03,	8.51954076e+02,	1.41863835e+04,	1.44701121e+04,
1.09918418e+04,	3.68333351e+03,	3.08678762e+04,	3.30963595e+04,
6.49481329e+03,	1.77829619e+04,	9.92267022e+03,	1.17241309e+04,
9.50830455e+03,	1.02394683e+04,	3.88334390e+03,	1.05103047e+04,
9.37145624e+03,	6.41639786e+03,	2.78561999e+04,	1.15661665e+04,
7.16016709e+03,	3.23228991e+04,	1.33818799e+04,	2.16955125e+03,
1.43987555e+04,	1.01643457e+04,	6.97905736e+03,	2.69405631e+03,
5.11390322e+03,	2.58715612e+04,	3.94475998e+02,	-3.33435104e+02,
3.21173588e+04,	1.28948113e+03,	1.33913764e+04,	5.67732625e+03,
9.32158834e+03,	3.62540675e+04,	8.85972810e+02,	4.14614938e+04,
1.12744782e+04,	1.46274054e+04,	3.41206044e+04,	1.08048070e+04,
5.44034812e+03,	3.90763079e+03,	3.11139385e+04,	1.38810994e+04,
4.06878278e+03,	3.44253148e+03,	1.46187081e+04,	2.58401638e+04,
3.17504760e+04,	7.28045121e+03,	6.52370567e+03,	1.35896220e+04,
1.23594665e+04,	1.36233634e+04,	1.09809425e+04,	3.66772824e+04,
9.81167281e+03,	1.25430558e+04,	6.11204725e+03,	9.83574145e+03,
5.65783052e+03,	8.04181769e+03,	2.85376728e+04,	6.74244089e+03,
1.30853862e+04,	1.49667519e+04,	1.13835551e+03,	2.36021926e+04,
4.01987200e+03,	1.15169442e+04,	5.66691396e+03,	5.71337052e+03,
1.68033250e+03])			

```
In [37]: plt.figure(figsize = (18,7))
plt.scatter(y_train, y_pred_train)
plt.xlabel("Actual Charges")
plt.ylabel("Predicted Charges")
plt.show()
```



In []:

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js