```
In [111]: import torch
     ...: import jovian
     ...: import torchvision
     ...: import torch.nn as nn
     ...: import pandas as pd
     ...: import matplotlib.pyplot as plt
     ...: import torch.nn.functional as F
     ...: from torchvision.datasets.utils import download_url
     ...: from torch.utils.data import DataLoader, TensorDataset, random_split

In [112]: DATASET_URL = "https://hub.jovian.ml/wp-content/uploads/2020/05/
insurance.csv"
     ...: DATA_FILENAME = "insurance.csv"
     ...: download_url(DATASET_URL, '.')
Using downloaded and verified file: .\insurance.csv

In [113]: dataframe_raw = pd.read_csv(DATA_FILENAME)
     ...: dataframe_raw.head()
Out[113]:
   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 [114]: your_name = "YashKumarSahu" # at least 5 characters

In [115]: def customize_dataset(dataframe_raw, rand_str):
     ...:     dataframe = dataframe_raw.copy(deep=True)
     ...:     # drop some rows
     ...:     dataframe = dataframe.sample(int(0.95*len(dataframe)),
random_state=int(ord(rand_str[0])))
     ...:     # scale input
     ...:     dataframe.bmi = dataframe.bmi * ord(rand_str[1])/100.
     ...:     # scale target
     ...:     dataframe.charges = dataframe.charges * ord(rand_str[2])/100.
     ...:     # drop column
     ...:     if ord(rand_str[3]) % 2 == 1:
     ...:         dataframe = dataframe.drop(['region'], axis=1)
     ...:     return dataframe
     ...:

In [116]: dataframe = customize_dataset(dataframe_raw, your_name)
     ...: dataframe.head()
Out[116]:
      age     sex       bmi  children smoker     region       charges
290    28  female  32.39800         0     no  southwest   3647.820700
651    53  female  38.41200         1     no  southeast  12166.667650
118    49  female  26.35490         0     no  southeast   9891.528695
255    55  female  24.60405         3     no  northeast  15004.432203
1166   57    male  39.15890         0     no  southeast  12629.876495

In [117]: num_rows = dataframe.shape[0]
     ...: print(num_rows)
1271

In [118]: num_cols = dataframe.shape[1]
     ...: print(num_cols)
7
```

```
In [119]: input_cols = [col for col in dataframe.columns[:-1]]
     ...: input_cols
Out[119]: ['age', 'sex', 'bmi', 'children', 'smoker', 'region']

In [120]: categorical_cols = [col for col in
dataframe.select_dtypes(exclude=["number"]).columns.values]
     ...: categorical_cols
Out[120]: ['sex', 'smoker', 'region']

In [121]: output_cols = [dataframe.columns[-1]]
     ...: output_cols
Out[121]: ['charges']

In [122]: # Write your answer here
     ...: output_col = dataframe[output_cols[0]]
     ...: print ('Minimum: {:.2f}'.format(output_col.min()))
     ...: print ('Maximum: {:.2f}'.format(output_col.max()))
     ...: print ('Average: {:.2f}'.format(output_col.mean()))
Minimum: 1290.15
Maximum: 73335.99
Average: 15386.99

In [123]: def dataframe_to_arrays(dataframe):
     ...:     # Make a copy of the original dataframe
     ...:     dataframe1 = dataframe.copy(deep=True)
     ...:     # Convert non-numeric categorical columns to numbers
     ...:     for col in categorical_cols:
     ...:         dataframe1[col] = dataframe1[col].astype('category').cat.codes
     ...:     # Extract input & outupts as numpy arrays
     ...:     inputs_array = dataframe1[input_cols].to_numpy()
     ...:     targets_array = dataframe1[output_cols].to_numpy()
     ...:     return inputs_array, targets_array
     ...:

In [124]: inputs_array, targets_array = dataframe_to_arrays(dataframe)
     ...: inputs_array, targets_array
Out[124]:
(array([[28.    ,  0.    , 32.398 ,  0.    ,  0.    ,  3.    ],
        [53.    ,  0.    , 38.412 ,  1.    ,  0.    ,  2.    ],
        [49.    ,  0.    , 26.3549,  0.    ,  0.    ,  2.    ],
        ...,
        [53.    ,  1.    , 40.2259,  0.    ,  0.    ,  2.    ],
        [40.    ,  0.    , 27.8293,  3.    ,  0.    ,  1.    ],
        [60.    ,  1.    , 23.5904,  0.    ,  0.    ,  1.    ]]),
 array([[ 3647.8207  ],
        [12166.66765 ],
        [ 9891.528695],
        ...,
        [10929.956845],
        [ 9268.630965],
        [14402.14552 ]]))

In [125]: inputs = torch.from_numpy(inputs_array).type(torch.float32)
     ...: targets = torch.from_numpy(targets_array).type(torch.float32)

In [126]: inputs.dtype, targets.dtype
Out[126]: (torch.float32, torch.float32)

In [127]: dataset = TensorDataset(inputs, targets)

In [128]: val_percent = 0.15 # between 0.1 and 0.2
```

```
     ...: val_size = int(num_rows * val_percent)
     ...: train_size = num_rows - val_size
     ...:
     ...:
     ...: train_ds, val_ds = random_split(dataset, [train_size, val_size]) # Use the
random_split function to split dataset into 2 parts of the desired length

In [129]: batch_size = 128

In [130]: train_loader = DataLoader(train_ds, batch_size, shuffle=True)
     ...: val_loader = DataLoader(val_ds, batch_size)

In [131]: for xb, yb in train_loader:
     ...:     print("inputs:", xb)
     ...:     print("targets:", yb)
     ...:     break
     ...:
inputs: tensor([[41.0000,  0.0000, 31.2340,  1.0000,  0.0000,  3.0000],
        [27.0000,  0.0000, 30.3174,  1.0000,  0.0000,  1.0000],
        [18.0000,  0.0000, 29.2115,  0.0000,  0.0000,  0.0000],
        [19.0000,  1.0000, 32.1070,  0.0000,  0.0000,  3.0000],
        [20.0000,  1.0000, 29.2115,  5.0000,  0.0000,  0.0000],
        [36.0000,  1.0000, 32.8054,  1.0000,  0.0000,  1.0000],
        [51.0000,  0.0000, 33.1740,  1.0000,  0.0000,  3.0000],
        [53.0000,  0.0000, 21.9317,  3.0000,  1.0000,  0.0000],
        [42.0000,  0.0000, 35.1091,  1.0000,  0.0000,  1.0000],
        [39.0000,  1.0000, 34.2410,  2.0000,  1.0000,  3.0000],
        [25.0000,  0.0000, 25.9863,  2.0000,  0.0000,  1.0000],
        [45.0000,  0.0000, 29.9730,  2.0000,  0.0000,  3.0000],
        [38.0000,  0.0000, 29.7693,  1.0000,  0.0000,  2.0000],
        [19.0000,  0.0000, 28.0330,  0.0000,  0.0000,  3.0000],
        [33.0000,  1.0000, 23.8668,  2.0000,  0.0000,  1.0000],
        [41.0000,  1.0000, 35.9385,  2.0000,  0.0000,  1.0000],
        [42.0000,  1.0000, 27.4607,  3.0000,  1.0000,  1.0000],
        [26.0000,  0.0000, 16.6791,  2.0000,  1.0000,  0.0000],
        [62.0000,  1.0000, 29.9487,  3.0000,  1.0000,  1.0000],
        [33.0000,  1.0000, 34.6775,  1.0000,  1.0000,  2.0000],
        [50.0000,  1.0000, 43.4269,  1.0000,  0.0000,  2.0000],
        [45.0000,  1.0000, 38.6109,  0.0000,  0.0000,  0.0000],
        [19.0000,  1.0000, 25.2491,  1.0000,  1.0000,  1.0000],
        [59.0000,  1.0000, 24.6962,  0.0000,  0.0000,  1.0000],
        [24.0000,  1.0000, 31.7190,  0.0000,  1.0000,  3.0000],
        [22.0000,  1.0000, 27.4607,  1.0000,  0.0000,  1.0000],
        [27.0000,  0.0000, 17.4163,  2.0000,  1.0000,  0.0000],
        [44.0000,  0.0000, 31.3698,  1.0000,  0.0000,  2.0000],
        [26.0000,  0.0000, 21.5631,  0.0000,  0.0000,  1.0000],
        [59.0000,  0.0000, 35.4244,  1.0000,  0.0000,  2.0000],
        [42.0000,  0.0000, 22.6689,  0.0000,  1.0000,  0.0000],
        [60.0000,  0.0000, 26.7235,  0.0000,  0.0000,  0.0000],
        [53.0000,  1.0000, 27.7420,  3.0000,  0.0000,  3.0000],
        [51.0000,  0.0000, 33.0770,  0.0000,  0.0000,  2.0000],
        [42.0000,  0.0000, 28.1300,  1.0000,  0.0000,  3.0000],
        [18.0000,  0.0000, 30.9624,  0.0000,  0.0000,  0.0000],
        [44.0000,  0.0000, 36.9182,  0.0000,  1.0000,  2.0000],
        [37.0000,  1.0000, 29.9487,  3.0000,  0.0000,  1.0000],
        [53.0000,  1.0000, 20.2730,  0.0000,  1.0000,  2.0000],
        [58.0000,  1.0000, 29.3958,  0.0000,  0.0000,  0.0000],
        [30.0000,  1.0000, 37.6651,  1.0000,  0.0000,  2.0000],
        [19.0000,  1.0000, 21.9317,  0.0000,  0.0000,  1.0000],
        [61.0000,  1.0000, 22.9454,  0.0000,  0.0000,  0.0000],
        [28.0000,  1.0000, 35.3080,  1.0000,  1.0000,  3.0000],
        [46.0000,  0.0000, 33.5620,  1.0000,  1.0000,  3.0000],
```

```
[28.0000,  1.0000, 26.1706,  2.0000,  0.0000,  0.0000],
[36.0000,  0.0000, 25.1230,  1.0000,  0.0000,  3.0000],
[47.0000,  1.0000, 35.1043,  0.0000,  1.0000,  2.0000],
[27.0000,  1.0000, 32.1604,  2.0000,  0.0000,  1.0000],
[55.0000,  1.0000, 31.6899,  1.0000,  0.0000,  2.0000],
[38.0000,  0.0000, 38.9455,  0.0000,  0.0000,  2.0000],
[48.0000,  0.0000, 26.4471,  1.0000,  0.0000,  0.0000],
[56.0000,  1.0000, 31.1467,  1.0000,  0.0000,  0.0000],
[26.0000,  1.0000, 30.1331,  0.0000,  0.0000,  1.0000],
[28.0000,  0.0000, 25.1569,  1.0000,  0.0000,  1.0000],
[52.0000,  1.0000, 35.5990,  0.0000,  0.0000,  3.0000],
[59.0000,  0.0000, 34.1440,  0.0000,  0.0000,  2.0000],
[52.0000,  1.0000, 46.3078,  1.0000,  0.0000,  2.0000],
[24.0000,  1.0000, 22.6980,  0.0000,  0.0000,  3.0000],
[55.0000,  0.0000, 32.5289,  2.0000,  0.0000,  1.0000],
[30.0000,  0.0000, 32.3301,  1.0000,  0.0000,  2.0000],
[22.0000,  1.0000, 31.1467,  0.0000,  0.0000,  1.0000],
[47.0000,  0.0000, 32.3447,  0.0000,  0.0000,  0.0000],
[20.0000,  1.0000, 27.1842,  1.0000,  1.0000,  1.0000],
[28.0000,  0.0000, 25.7147,  2.0000,  0.0000,  2.0000],
[25.0000,  1.0000, 25.9960,  3.0000,  0.0000,  3.0000],
[54.0000,  1.0000, 28.3240,  1.0000,  0.0000,  3.0000],
[64.0000,  0.0000, 22.3003,  0.0000,  1.0000,  2.0000],
[18.0000,  0.0000, 37.0249,  0.0000,  0.0000,  2.0000],
[30.0000,  1.0000, 27.8293,  3.0000,  1.0000,  1.0000],
[30.0000,  0.0000, 22.9454,  3.0000,  1.0000,  1.0000],
[49.0000,  1.0000, 25.0648,  2.0000,  1.0000,  1.0000],
[18.0000,  1.0000, 24.4198,  0.0000,  1.0000,  0.0000],
[56.0000,  0.0000, 34.7260,  1.0000,  0.0000,  3.0000],
[33.0000,  1.0000, 41.1862,  1.0000,  0.0000,  2.0000],
[20.0000,  0.0000, 35.8900,  5.0000,  0.0000,  3.0000],
[56.0000,  0.0000, 27.9214,  0.0000,  0.0000,  0.0000],
[36.0000,  1.0000, 30.5550,  0.0000,  0.0000,  3.0000],
[53.0000,  1.0000, 30.2252,  1.0000,  0.0000,  1.0000],
[55.0000,  1.0000, 32.0100,  0.0000,  0.0000,  2.0000],
[21.0000,  1.0000, 21.6310,  1.0000,  0.0000,  3.0000],
[63.0000,  0.0000, 35.7445,  0.0000,  0.0000,  2.0000],
[43.0000,  0.0000, 34.5708,  1.0000,  0.0000,  2.0000],
[63.0000,  0.0000, 26.9078,  0.0000,  1.0000,  0.0000],
[62.0000,  0.0000, 24.2500,  0.0000,  0.0000,  3.0000],
[49.0000,  0.0000, 29.8566,  1.0000,  0.0000,  0.0000],
[21.0000,  0.0000, 31.6996,  2.0000,  0.0000,  1.0000],
[50.0000,  1.0000, 24.5410,  0.0000,  0.0000,  2.0000],
[42.0000,  1.0000, 34.7260,  2.0000,  0.0000,  3.0000],
[50.0000,  0.0000, 26.5392,  0.0000,  0.0000,  0.0000],
[37.0000,  1.0000, 28.7508,  0.0000,  0.0000,  1.0000],
[22.0000,  0.0000, 23.5710,  0.0000,  0.0000,  3.0000],
[50.0000,  1.0000, 26.6313,  1.0000,  0.0000,  0.0000],
[19.0000,  1.0000, 29.4880,  0.0000,  0.0000,  3.0000],
[59.0000,  0.0000, 26.8884,  3.0000,  0.0000,  2.0000],
[51.0000,  1.0000, 24.0511,  2.0000,  1.0000,  1.0000],
[20.0000,  1.0000, 34.5563,  3.0000,  1.0000,  1.0000],
[30.0000,  0.0000, 31.4280,  1.0000,  0.0000,  3.0000],
[49.0000,  1.0000, 28.9351,  1.0000,  0.0000,  0.0000],
[32.0000,  1.0000, 45.1341,  2.0000,  0.0000,  2.0000],
[50.0000,  1.0000, 25.8020,  0.0000,  0.0000,  3.0000],
[29.0000,  0.0000, 37.6651,  3.0000,  0.0000,  2.0000],
[39.0000,  0.0000, 24.1433,  3.0000,  1.0000,  0.0000],
[30.0000,  0.0000, 19.3515,  3.0000,  0.0000,  1.0000],
[29.0000,  1.0000, 33.3680,  0.0000,  1.0000,  3.0000],
[23.0000,  1.0000, 33.3680,  0.0000,  0.0000,  3.0000],
[59.0000,  0.0000, 26.9951,  3.0000,  0.0000,  2.0000],
```

```
        [29.0000,  0.0000, 24.8320,  4.0000,  0.0000,  3.0000],
        [44.0000,  1.0000, 35.9870,  2.0000,  0.0000,  3.0000],
        [19.0000,  1.0000, 24.7883,  1.0000,  0.0000,  1.0000],
        [44.0000,  0.0000, 28.9157,  2.0000,  0.0000,  2.0000],
        [27.0000,  0.0000, 29.6723,  1.0000,  0.0000,  0.0000],
        [44.0000,  1.0000, 21.1945,  3.0000,  0.0000,  0.0000],
        [20.0000,  1.0000, 34.2507,  1.0000,  0.0000,  2.0000],
        [39.0000,  0.0000, 23.4982,  5.0000,  0.0000,  1.0000],
        [31.0000,  0.0000, 25.0260,  2.0000,  0.0000,  3.0000],
        [44.0000,  1.0000, 29.2940,  2.0000,  1.0000,  3.0000],
        [62.0000,  1.0000, 36.2780,  0.0000,  0.0000,  3.0000],
        [45.0000,  0.0000, 26.9951,  2.0000,  0.0000,  2.0000],
        [62.0000,  1.0000, 26.7235,  1.0000,  0.0000,  1.0000],
        [22.0000,  0.0000, 20.6416,  3.0000,  0.0000,  1.0000],
        [49.0000,  1.0000, 29.3910,  0.0000,  0.0000,  3.0000],
        [37.0000,  0.0000, 46.1720,  2.0000,  1.0000,  3.0000],
        [27.0000,  1.0000, 32.6502,  0.0000,  0.0000,  2.0000],
        [63.0000,  1.0000, 40.2259,  0.0000,  0.0000,  2.0000],
        [63.0000,  1.0000, 38.6060,  3.0000,  0.0000,  3.0000],
        [58.0000,  0.0000, 22.0869,  0.0000,  0.0000,  2.0000],
        [32.0000,  1.0000, 29.1291,  1.0000,  0.0000,  2.0000]])
targets: tensor([[ 7792.3550],
        [ 4549.4819],
        [ 2533.9927],
        [26545.3984],
        [ 5652.3188],
        [ 6184.0767],
        [11353.6064],
        [28604.3926],
        [ 8560.1895],
        [46119.4727],
        [ 4817.4800],
        [ 9798.0303],
        [ 6873.3560],
        [ 2004.6960],
        [ 6046.1343],
        [ 8355.5576],
        [37705.5781],
        [16623.9902],
        [53725.8867],
        [44025.1602],
        [10417.5400],
        [ 8565.6641],
        [18918.5293],
        [13943.7412],
        [39643.7656],
        [ 3034.8994],
        [17257.5664],
        [ 8778.7783],
        [ 3652.7310],
        [32531.0820],
        [22959.4590],
        [15199.6582],
        [12941.4346],
        [10676.0967],
        [ 8108.2383],
        [ 2536.8779],
        [56217.9062],
        [ 7816.3926],
        [24375.1914],
        [13728.9941],
        [21807.6484],
```

[1872.7416],
[15099.0439],
[58873.7422],
[47910.8438],
[5100.3584],
[6293.3164],
[47927.4922],
[4667.5195],
[12428.6094],
[6211.1274],
[10864.3379],
[13527.4512],
[3104.5037],
[4753.6880],
[10516.2500],
[14081.2109],
[11211.2471],
[2265.0562],
[14110.1416],
[4773.6831],
[2363.6235],
[24010.6016],
[20194.4375],
[4991.5068],
[4492.0459],
[12001.5107],
[31093.6016],
[1876.4186],
[23857.8867],
[21580.7559],
[27378.3262],
[17845.9082],
[13425.2490],
[13025.7217],
[5555.2246],
[13407.1357],
[5062.5679],
[12031.2764],
[23898.7129],
[2418.5420],
[15971.1641],
[8447.5859],
[33951.6406],
[15468.7900],
[11245.0996],
[29921.7930],
[9709.0674],
[8234.1084],
[29505.0625],
[5782.3687],
[2473.0393],
[11060.3115],
[1444.7439],
[16101.3037],
[27562.4902],
[43085.1445],
[4772.1963],
[10681.2305],
[5389.3472],
[9711.1455],
[5908.9951],
[24908.9199],

```
        [ 6547.4453],
        [41627.3555],
        [ 2100.8694],
        [16101.4795],
        [ 6565.1973],
        [ 8901.3877],
        [ 2554.7991],
        [ 9452.0850],
        [19315.8730],
        [10224.8105],
        [31882.9316],
        [10310.6650],
        [ 5674.9106],
        [44848.3281],
        [14926.2617],
        [ 9793.1221],
        [16028.3164],
        [ 4940.7119],
        [ 9334.1816],
        [53030.5391],
        [ 2873.1765],
        [15416.1992],
        [17445.5801],
        [13608.8496],
        [ 4685.6216]])
```

In [**132**]: