HI3_modeling

November 7, 2017

1 Hand-in 3, Part 2: Data Modeling

In this part you will take the csv file "reduced_field_data.csv" from Part 1, and use it to estimate the line criticality indices.

```
In [1]: import pandas as pd
        import numpy as np
        import matplotlib.pyplot as plt

        %matplotlib inline
```

1.1 Score function

Below is a scoring function we have written for you. The output of this function is a score of how well your procedure is doing. Higher scores are worse!

As you can see, it penalizes heavily when you don't predict accurately values when they are being overloaded, i.e. false negatives.

```
In [2]: def score_func(y_est, y_real):
            """
                This function takes your estimates y_est and
                scores them against the real data y_real.
                You should use this function to show how good your estimation method is.
            """
            # Square deviation
            sqr_err = np.sum(((y_est - y_real)**2).sum())
            # Penalty for not estimating a critical value above 0.95 when it occurs.
            false_negative = np.sum(np.where(np.logical_and(y_est < 0.95, y_real > 0.95), 10*np
            return sqr_err + false_negative
```

1.2 Load and clean data

Load your data from the previous exercise as well as the criticality data.

```
In [3]: # Data is loaded here
        field_data = pd.read_csv("reduced_field_data.csv", index_col=0)
        crit_data = pd.read_csv("flow_criticality_data.csv", index_col=0)
```

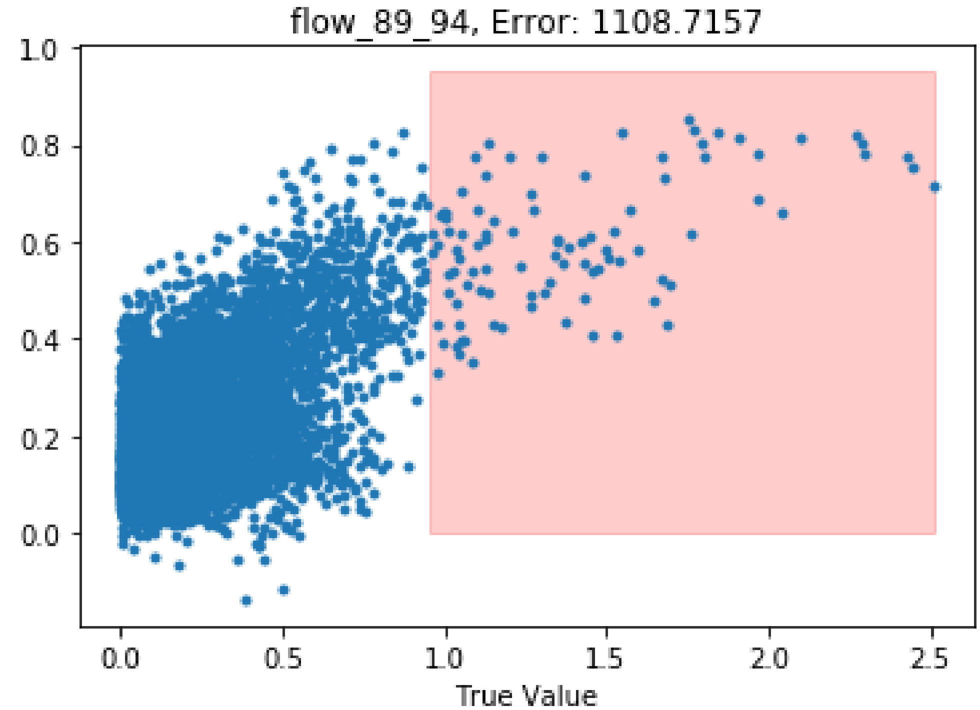
1.3 Linear regression

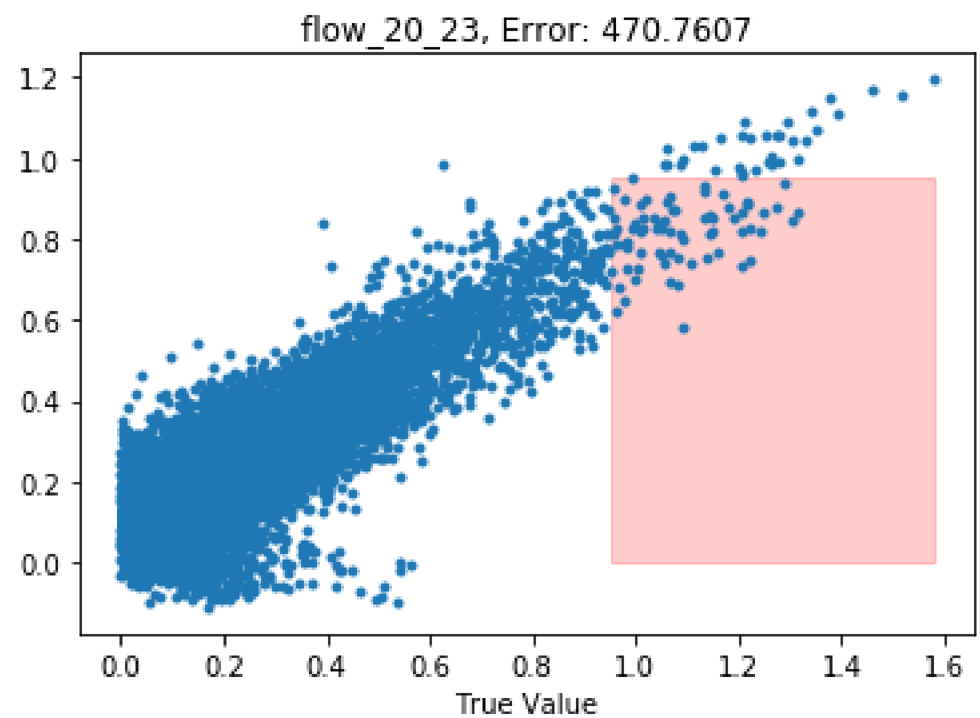
We have implemented a simple linear regression to apply to your data. You should use this as a benchmark for your neural network below.

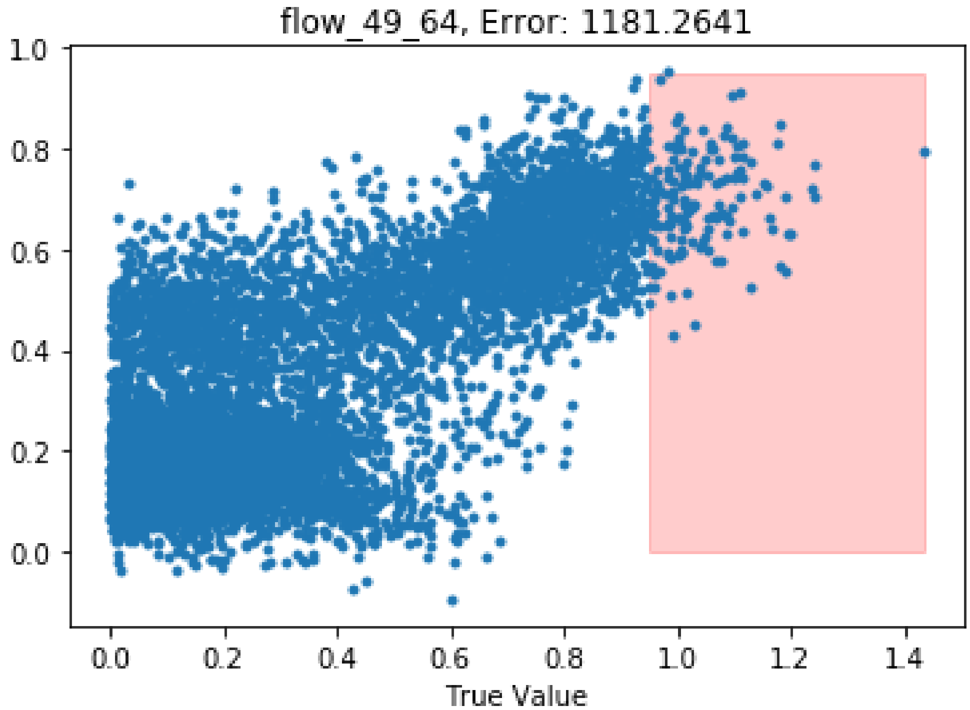
```
In [4]: # Slicing the data, because we have removed the NaN's of the field_data, which is ther
        # Using the time column, from field_data
        crit_data = crit_data[crit_data.index.isin(field_data['time'])].dropna(axis=0)
        # We no longer need the time column, therefore we remove it
        field_data = field_data.drop('time',1)
```

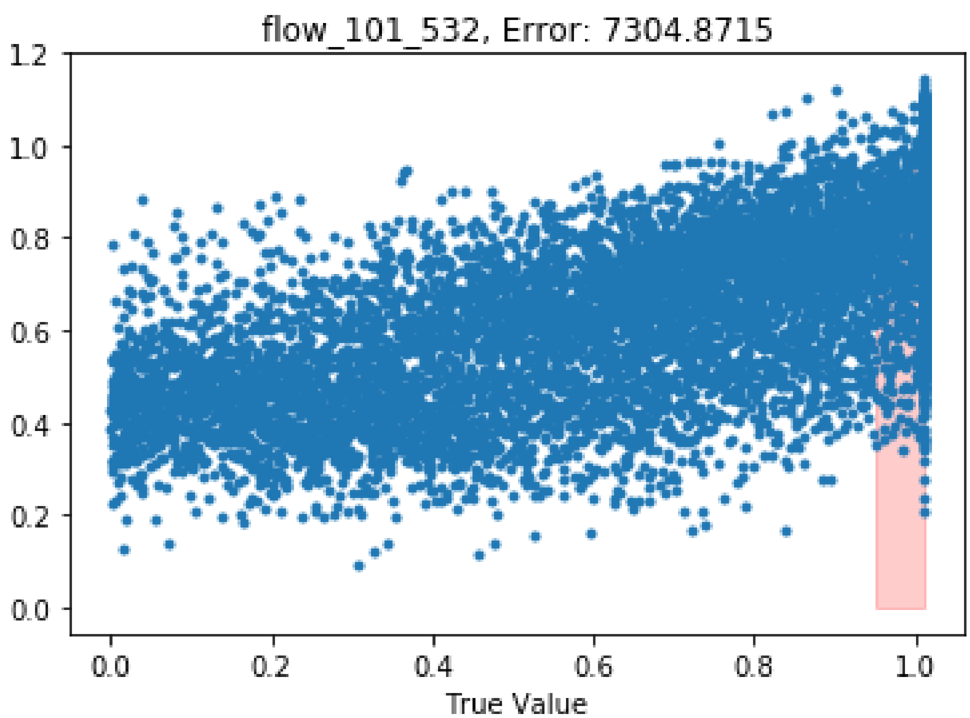
```
In [5]: # The code below implements a linear regression on your data and compares predicted an
        # On the plots there is a red square indicating the areas corresponding to false negat
        from sklearn.linear_model import LinearRegression
        lm = LinearRegression()
        lm.fit(field_data,crit_data)

        y_pred = lm.predict(field_data)
        for index in range(10):
            plt.figure()
            plt.fill_between([0.95, crit_data.iloc[:,index].max()], [0.95, 0.95], color='r', al
            plt.plot(crit_data.iloc[:,index],y_pred[:,index],'.')
            error = score_func(y_pred[:,index],crit_data.iloc[:,index])
            plt.title('{0}, Error: {1:.04f}'.format(crit_data.columns[index], error))
            plt.xlabel('True Value')

        plt.ylabel('Predicted Value')
        print('Overall error: {0:.04f}'.format(score_func(y_pred, crit_data)))
```

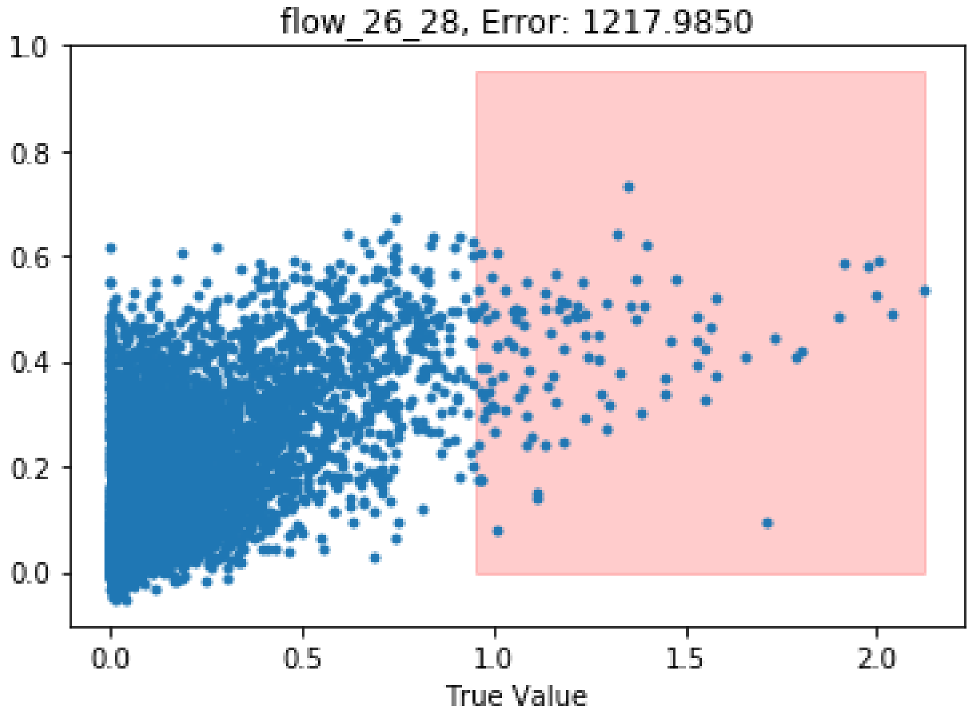
```
Overall error: 20005.8900
```

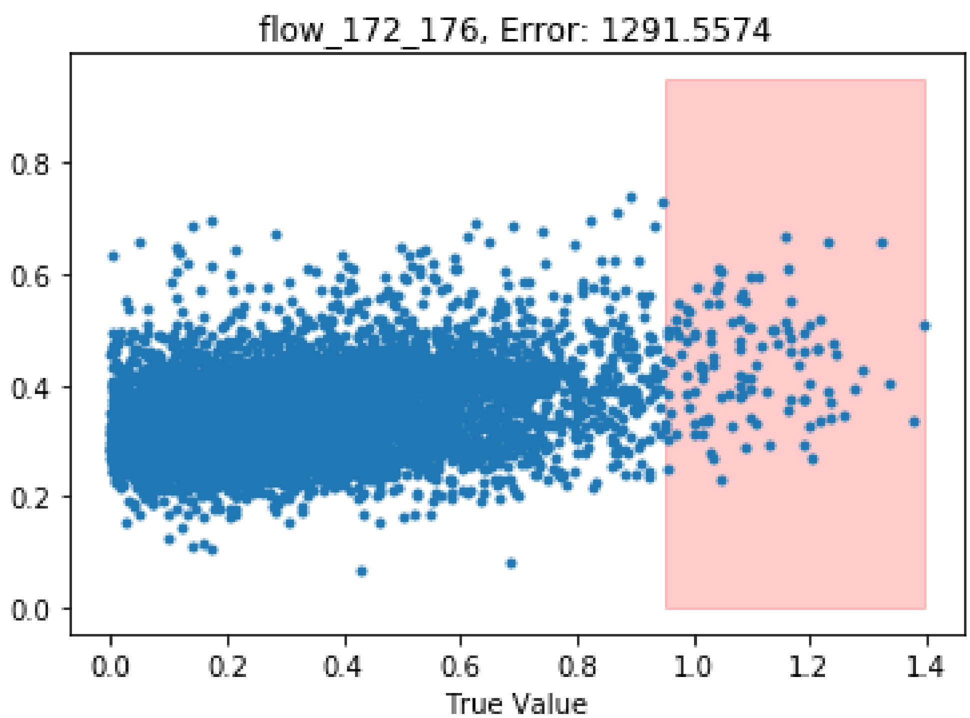

flow_89_94, Error: 1108.7157
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.5
1.0
1.5
2.0
2.5
True Value

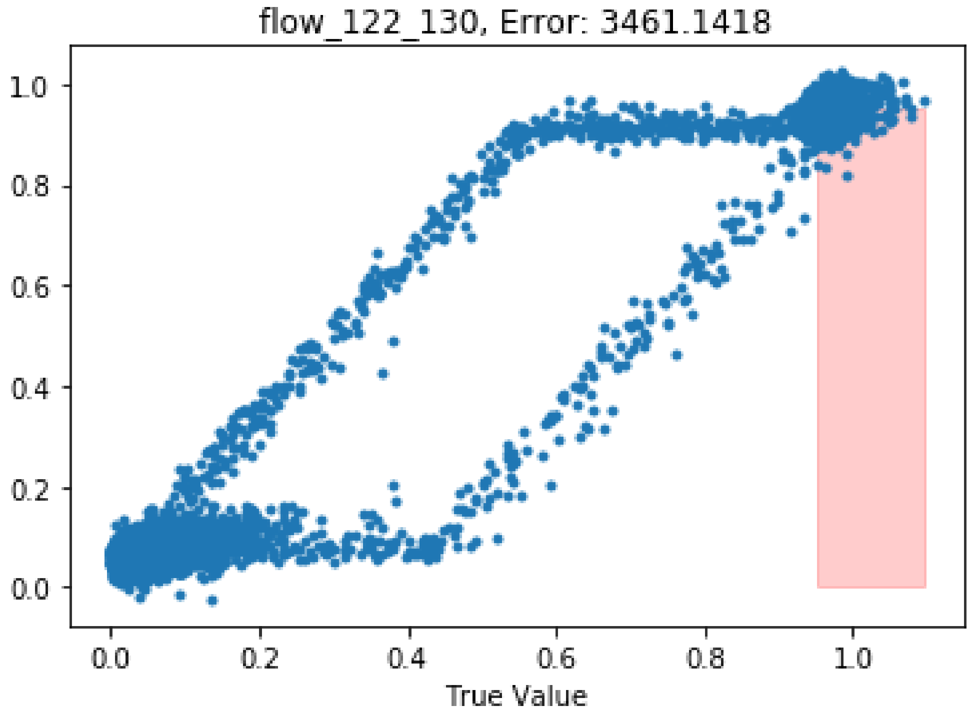

flow_20_23, Error: 470.7607
1.2
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0
1.2
1.4
1.6
True Value

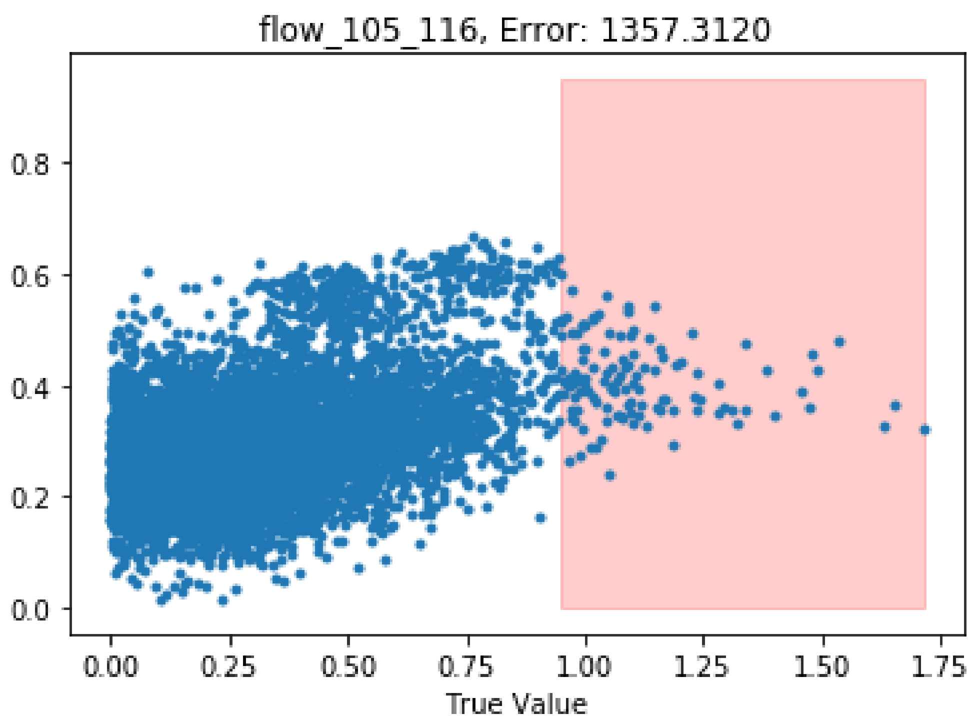

flow_49_64, Error: 1181.2641
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0
1.2
1.4
True Value

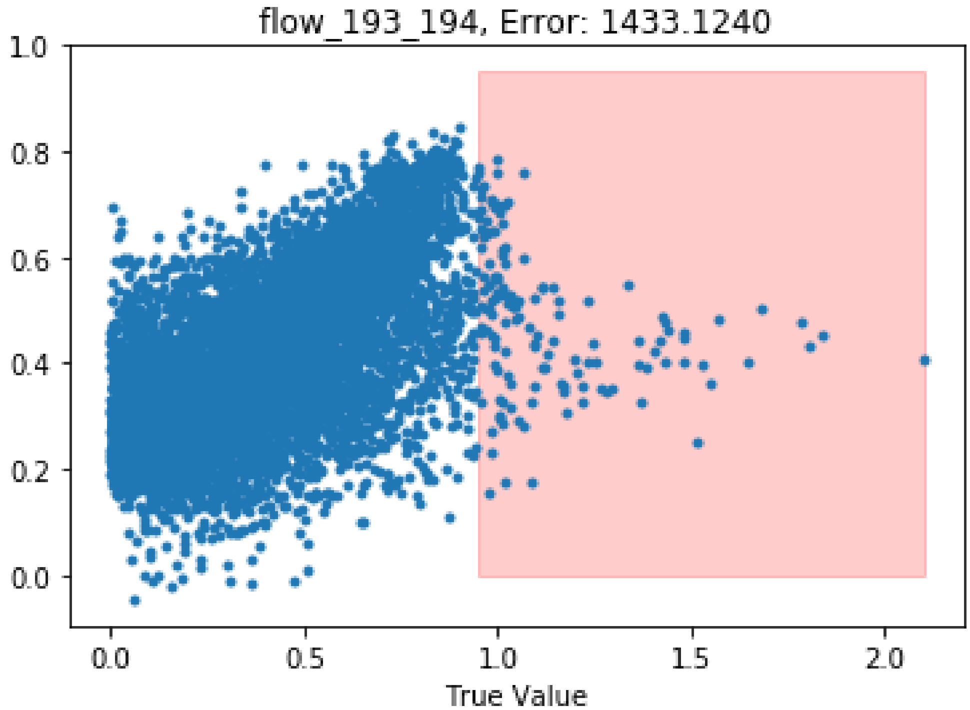

flow_101_532, Error: 7304.8715
1.2
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0
True Value

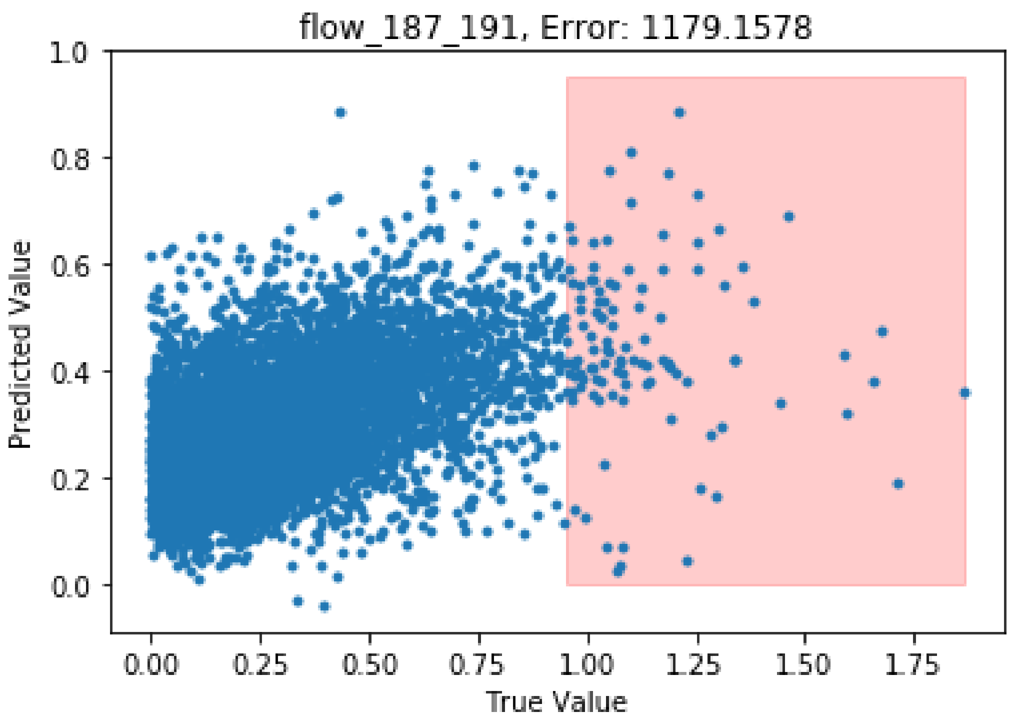

flow_26_28, Error: 1217.9850
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.5
1.0
1.5
2.0
True Value


flow_172_176, Error: 1291.5574
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0
1.2
1.4
True Value


flow_122_130, Error: 3461.1418
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0
True Value


flow_105_116, Error: 1357.3120
0.8
0.6
0.4
0.2
0.0
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
True Value


flow_193_194, Error: 1433.1240
True Value


flow_187_191, Error: 1179.1578
Predicted Value
True Value

Q#6 Would you recommend using Linear Regression for estimating line criticality indices? Why/why not?

2 Neural Network

You should now produce a neural network that can estimate criticality indices. Split your data into a training and test set, build your neural network below, and use the score_func defined above to estimate your model quality.

Q#7 Explain why you chose the layers, nodes and activation functions you did. (No wrong answers, we want to know your thought process!) What is the best score you can get with a single layer?

For the NN Linear regression We tried using the linear activation function. This gives a slow estimation, but gives a very good approximation. We ended up using the rectified linear activation function "relu", this gives a really quick and consistent estimation, but appears limited in the precission it can reach

We use the optimizer adam, as it is a memory light, computationally efficient algorithm From reading, the standard parameters of tensorflow, should be pretty good. However we tweaked hte learning rate, to be slightly faster 0.001 -> 0.05. As this gave quick results, but still seem to narrow in on the same result quickly.

We made the neural network, with a single layer, using 10 nodes. We try to keepit simple with small amount of layers. and we get a pretty good approximation using just one layer

using linear, the best score was(however it was very inconsistent) Epoch 50/50 6648/6648 [==============================] - 0s - loss: 0.1517
Score: 5386.79165989189

Using relu, the best score was Epoch 3/3 6648/6648 [==============================] - 0s - loss: 0.2178
Score: 15613.5309487

Both results are better than the linear regression function

```
In [6]: # Split data into training and test

        # Using the sklearn's builtin function, we split the data, so that 1/5 will be used fo
        # Given a random_state, to make the data reproducable
        from sklearn.model_selection import train_test_split

        x_train, x_test, y_train, y_test = train_test_split(field_data, crit_data, test_size=0
```

```
In [7]: #INSERT Neural Network Code here for a linear regression.
        random_state=100
        from keras.models import Sequential
        from keras.layers import Dense, Activation
        from keras import optimizers

        model = Sequential()
        model.add(Dense(10, input_shape=(20,)))
        model.add(Activation('relu'))
        adam = optimizers.Adam(lr=0.05, beta_1=0.95,
        beta_2=0.999, epsilon=1e-08, decay=0.0)
```

```
model.compile(loss='mean_squared_error', optimizer=adam)
model.fit(x_train.values,y_train.values,
epochs = 3, batch_size=250, shuffle=False)
y_pred = model.predict(x_test.values, batch_size = 1000)

print("Score: ", score_func(y_pred, y_test))
```

```
Using TensorFlow backend.


Epoch 1/3
6648/6648 [==============================] - 0s - loss: 12326.7091
Epoch 2/3
6648/6648 [==============================] - 0s - loss: 0.2178
Epoch 3/3
6648/6648 [==============================] - 0s - loss: 0.2178
Score:  15613.5309487
```

3 K-fold cross validation

To ensure that your neural network actually works when presented with new data, take the neural network you defined above, and perform a k-fold cross validation on it.

Q#8 Using a test window size of one tenth of your data (ten-fold cross validation), plot a histogram of the output of score_func for the validation. Tweak your neural network to achieve the lowest mean score.H

```
In [64]: # Your k-fold code goes here.

         from sklearn.model_selection import KFold
         import matplotlib.pyplot as plt
         import numpy as np
         from keras.models import Sequential
         from keras.layers import Dense, Activation
         from keras import optimizers

         listOfErrors2 = list()
         kf1 = KFold(n_splits=10,shuffle=True)
         for train_index, test_index in kf1.split(field_data):
             X_train, X_test = field_data.iloc[train_index], field_data.iloc[test_index]
             Y_train, Y_test = crit_data.iloc[train_index], crit_data.iloc[test_index]
             #print(X_test)
             model = Sequential()
             model.add(Dense(10, input_shape=(20,)))
             model.add(Activation('relu'))

             adam = optimizers.Adam(lr=0.05, beta_1=0.95,
                                    beta_2=0.999, epsilon=1e-08, decay=0.0)
```

```
    model.compile(loss='mean_squared_error', optimizer=adam)
    history1 = model.fit(x_train.values,y_train.values, epochs=3, batch_size=250, shu
    y_pred2 = model.predict(x_test.values, batch_size=1000)
    listOfErrors2.append(score_func(y_pred2, y_test))
```

```
Epoch 1/3
6648/6648 [==============================] - 1s - loss: 11234.1006
Epoch 2/3
6648/6648 [==============================] - 0s - loss: 0.2178
Epoch 3/3
6648/6648 [==============================] - 0s - loss: 0.2178
Epoch 1/3
6648/6648 [==============================] - 0s - loss: 10952.6376
Epoch 2/3
6648/6648 [==============================] - 0s - loss: 0.2178
Epoch 3/3
6648/6648 [==============================] - 0s - loss: 0.2178
Epoch 1/3
6648/6648 [==============================] - 0s - loss: 13270.1391
Epoch 2/3
6648/6648 [==============================] - 0s - loss: 0.2178
Epoch 3/3
6648/6648 [==============================] - 0s - loss: 0.2178
Epoch 1/3
6648/6648 [==============================] - 0s - loss: 38044.7376
Epoch 2/3
6648/6648 [==============================] - 0s - loss: 0.2178
Epoch 3/3
6648/6648 [==============================] - 0s - loss: 0.2178
Epoch 1/3
6648/6648 [==============================] - 1s - loss: 21656.4720
Epoch 2/3
6648/6648 [==============================] - 0s - loss: 0.2178
Epoch 3/3
6648/6648 [==============================] - 0s - loss: 0.2178
Epoch 1/3
6648/6648 [==============================] - 0s - loss: 21941.1040
Epoch 2/3
6648/6648 [==============================] - 0s - loss: 0.2178
Epoch 3/3
6648/6648 [==============================] - 0s - loss: 0.2178
Epoch 1/3
6648/6648 [==============================] - 1s - loss: 7655.4923
Epoch 2/3
6648/6648 [==============================] - 0s - loss: 0.2178
Epoch 3/3
6648/6648 [==============================] - 0s - loss: 0.2178
Epoch 1/3
```

```
6648/6648 [==============================] - 0s - loss: 27932.9444
Epoch 2/3
6648/6648 [==============================] - 0s - loss: 0.2178
Epoch 3/3
6648/6648 [==============================] - 0s - loss: 0.2178
Epoch 1/3
6648/6648 [==============================] - 0s - loss: 14246.5849
Epoch 2/3
6648/6648 [==============================] - 0s - loss: 0.2178
Epoch 3/3
6648/6648 [==============================] - 0s - loss: 0.2178     - ETA: 0s - loss: 0.2
Epoch 1/3
6648/6648 [==============================] - 1s - loss: 4481.1776
Epoch 2/3
6648/6648 [==============================] - 0s - loss: 0.2178
Epoch 3/3
6648/6648 [==============================] - 0s - loss: 0.2178
```

In [8]:
```
plt.hist(listOfErrors2)
plt.grid(True)
plt.show()

print("Mean score: " , np.mean(listOfErrors2))
print("KFold scores: ", listOfErrors2)
```

```
---------------------------------------------------------------------------

NameError                                 Traceback (most recent call last)

<ipython-input-8-4c6eb1c2ebfa> in <module>()
      1
      2
----> 3 plt.hist(listOfErrors2)
      4 plt.grid(True)
      5 plt.show()


NameError: name 'listOfErrors2' is not defined
```

In [19]:
```
# k-fold of the linear version

 from sklearn.model_selection import KFold
 import matplotlib.pyplot as plt
 import numpy as np
 from keras.models import Sequential
```

```
from keras.layers import Dense, Activation
from keras import optimizers

listOfErrors2 = list()
kf1 = KFold(n_splits=10,shuffle=True)
for train_index, test_index in kf1.split(field_data):
    X_train, X_test = field_data.iloc[train_index], field_data.iloc[test_index]
    Y_train, Y_test = crit_data.iloc[train_index], crit_data.iloc[test_index]
    #print(X_test)
    model = Sequential()
    model.add(Dense(10, input_shape=(20,)))
    model.add(Activation('linear'))
    model.add(Dense(10,))
    model.add(Activation('linear'))

    adam = optimizers.Adam(lr=0.009, beta_1=0.85,
                            beta_2=0.999, epsilon=1e-001, decay=0.0)
    model.compile(loss='mean_squared_error', optimizer=adam)
    history1 = model.fit(x_train.values,y_train.values, epochs=50, batch_size=100, shu
    y_pred2 = model.predict(x_test.values, batch_size=1000)
    listOfErrors2.append(score_func(y_pred2, y_test))
```

```
Epoch 1/50
6648/6648 [==============================] - 0s - loss: 34855.0576
Epoch 2/50
6648/6648 [==============================] - 0s - loss: 1011.0088
Epoch 3/50
6648/6648 [==============================] - 0s - loss: 313.9042
Epoch 4/50
6648/6648 [==============================] - 0s - loss: 148.0342
Epoch 5/50
6648/6648 [==============================] - 0s - loss: 85.5130
Epoch 6/50
6648/6648 [==============================] - 0s - loss: 54.7495
Epoch 7/50
6648/6648 [==============================] - 0s - loss: 37.1201
Epoch 8/50
6648/6648 [==============================] - 0s - loss: 26.0638
Epoch 9/50
6648/6648 [==============================] - 0s - loss: 18.6907
Epoch 10/50
6648/6648 [==============================] - 0s - loss: 13.6790
Epoch 11/50
6648/6648 [==============================] - 0s - loss: 10.2136
Epoch 12/50
6648/6648 [==============================] - 0s - loss: 7.7882
Epoch 13/50
6648/6648 [==============================] - 0s - loss: 6.0348
```

```
Epoch 14/50
6648/6648 [==============================] - 0s - loss: 4.7698
Epoch 15/50
6648/6648 [==============================] - 0s - loss: 3.8533
Epoch 16/50
6648/6648 [==============================] - 0s - loss: 3.1541
Epoch 17/50
6648/6648 [==============================] - 0s - loss: 2.6009
Epoch 18/50
6648/6648 [==============================] - 0s - loss: 2.1802
Epoch 19/50
6648/6648 [==============================] - 0s - loss: 1.8514
Epoch 20/50
6648/6648 [==============================] - 0s - loss: 1.5804
Epoch 21/50
6648/6648 [==============================] - 0s - loss: 1.3608
Epoch 22/50
6648/6648 [==============================] - 0s - loss: 1.1804
Epoch 23/50
6648/6648 [==============================] - 0s - loss: 1.0338
Epoch 24/50
6648/6648 [==============================] - 0s - loss: 0.9071
Epoch 25/50
6648/6648 [==============================] - 0s - loss: 0.8039
Epoch 26/50
6648/6648 [==============================] - 0s - loss: 0.7158
Epoch 27/50
6648/6648 [==============================] - 0s - loss: 0.6415
Epoch 28/50
6648/6648 [==============================] - 0s - loss: 0.5749
Epoch 29/50
6648/6648 [==============================] - 0s - loss: 0.5210
Epoch 30/50
6648/6648 [==============================] - 0s - loss: 0.4708
Epoch 31/50
6648/6648 [==============================] - 0s - loss: 0.4358
Epoch 32/50
6648/6648 [==============================] - 0s - loss: 0.3925
Epoch 33/50
6648/6648 [==============================] - 0s - loss: 0.3607
Epoch 34/50
6648/6648 [==============================] - 0s - loss: 0.3318
Epoch 35/50
6648/6648 [==============================] - 0s - loss: 0.3055
Epoch 36/50
6648/6648 [==============================] - 0s - loss: 0.2835
Epoch 37/50
6648/6648 [==============================] - 0s - loss: 0.2768
```

```
Epoch 38/50
6648/6648 [==============================] - 0s - loss: 0.2447
Epoch 39/50
6648/6648 [==============================] - 0s - loss: 0.2281
Epoch 40/50
6648/6648 [==============================] - 0s - loss: 0.2085
Epoch 41/50
6648/6648 [==============================] - 0s - loss: 0.2328
Epoch 42/50
6648/6648 [==============================] - 0s - loss: 0.2382
Epoch 43/50
6648/6648 [==============================] - 0s - loss: 0.1905
Epoch 44/50
6648/6648 [==============================] - 0s - loss: 1.1866
Epoch 45/50
6648/6648 [==============================] - 0s - loss: 28.1521
Epoch 46/50
6648/6648 [==============================] - 0s - loss: 1.1052
Epoch 47/50
6648/6648 [==============================] - 0s - loss: 0.1393
Epoch 48/50
6648/6648 [==============================] - 0s - loss: 0.1393
Epoch 49/50
6648/6648 [==============================] - 0s - loss: 0.1656
Epoch 50/50
6648/6648 [==============================] - 0s - loss: 0.3423
Epoch 1/50
6648/6648 [==============================] - 0s - loss: 30039.1577
Epoch 2/50
6648/6648 [==============================] - 0s - loss: 859.9691
Epoch 3/50
6648/6648 [==============================] - 0s - loss: 248.2900
Epoch 4/50
6648/6648 [==============================] - 0s - loss: 110.1715
Epoch 5/50
6648/6648 [==============================] - 0s - loss: 59.6159
Epoch 6/50
6648/6648 [==============================] - 0s - loss: 36.2383
Epoch 7/50
6648/6648 [==============================] - 0s - loss: 23.6255
Epoch 8/50
6648/6648 [==============================] - 0s - loss: 16.0242
Epoch 9/50
6648/6648 [==============================] - 0s - loss: 11.1583
Epoch 10/50
6648/6648 [==============================] - 0s - loss: 8.0774
Epoch 11/50
6648/6648 [==============================] - 0s - loss: 5.9419
```

```
Epoch 12/50
6648/6648 [==============================] - 0s - loss: 4.4580
Epoch 13/50
6648/6648 [==============================] - 0s - loss: 3.4687
Epoch 14/50
6648/6648 [==============================] - 0s - loss: 2.7326
Epoch 15/50
6648/6648 [==============================] - 0s - loss: 2.2012
Epoch 16/50
6648/6648 [==============================] - 0s - loss: 1.7869
Epoch 17/50
6648/6648 [==============================] - 0s - loss: 1.5013
Epoch 18/50
6648/6648 [==============================] - 0s - loss: 1.2724
Epoch 19/50
6648/6648 [==============================] - 0s - loss: 1.1119
Epoch 20/50
6648/6648 [==============================] - 0s - loss: 0.9839
Epoch 21/50
6648/6648 [==============================] - 0s - loss: 0.8706
Epoch 22/50
6648/6648 [==============================] - 0s - loss: 0.8037
Epoch 23/50
6648/6648 [==============================] - 0s - loss: 0.7303
Epoch 24/50
6648/6648 [==============================] - 0s - loss: 0.6732
Epoch 25/50
6648/6648 [==============================] - 0s - loss: 0.6443
Epoch 26/50
6648/6648 [==============================] - 0s - loss: 0.5930
Epoch 27/50
6648/6648 [==============================] - 0s - loss: 0.5756
Epoch 28/50
6648/6648 [==============================] - 0s - loss: 0.5329
Epoch 29/50
6648/6648 [==============================] - 0s - loss: 0.5034
Epoch 30/50
6648/6648 [==============================] - 0s - loss: 0.5012
Epoch 31/50
6648/6648 [==============================] - 0s - loss: 0.4868
Epoch 32/50
6648/6648 [==============================] - 0s - loss: 0.4684
Epoch 33/50
6648/6648 [==============================] - 0s - loss: 0.4564
Epoch 34/50
6648/6648 [==============================] - 0s - loss: 0.4279
Epoch 35/50
6648/6648 [==============================] - 0s - loss: 0.4176
```

```
Epoch 36/50
6648/6648 [==============================] - 0s - loss: 0.4959
Epoch 37/50
6648/6648 [==============================] - 0s - loss: 0.4416
Epoch 38/50
6648/6648 [==============================] - 0s - loss: 0.3989
Epoch 39/50
6648/6648 [==============================] - 0s - loss: 0.3684
Epoch 40/50
6648/6648 [==============================] - 0s - loss: 0.4419
Epoch 41/50
6648/6648 [==============================] - 0s - loss: 0.3977
Epoch 42/50
6648/6648 [==============================] - 0s - loss: 0.3879
Epoch 43/50
6648/6648 [==============================] - 0s - loss: 0.5043
Epoch 44/50
6648/6648 [==============================] - 0s - loss: 0.6009
Epoch 45/50
6648/6648 [==============================] - 0s - loss: 2.6699
Epoch 46/50
6648/6648 [==============================] - 0s - loss: 3.2851
Epoch 47/50
6648/6648 [==============================] - 0s - loss: 0.8010
Epoch 48/50
6648/6648 [==============================] - 0s - loss: 3.5110
Epoch 49/50
6648/6648 [==============================] - 0s - loss: 4.2906
Epoch 50/50
6648/6648 [==============================] - 0s - loss: 0.9152
Epoch 1/50
6648/6648 [==============================] - 0s - loss: 56337.7804
Epoch 2/50
6648/6648 [==============================] - 0s - loss: 2998.9747
Epoch 3/50
6648/6648 [==============================] - 0s - loss: 1094.9492
Epoch 4/50
6648/6648 [==============================] - 0s - loss: 518.9189
Epoch 5/50
6648/6648 [==============================] - 0s - loss: 272.4274
Epoch 6/50
6648/6648 [==============================] - 0s - loss: 151.6361
Epoch 7/50
6648/6648 [==============================] - 0s - loss: 88.2364
Epoch 8/50
6648/6648 [==============================] - 0s - loss: 52.7075
Epoch 9/50
6648/6648 [==============================] - 0s - loss: 32.3673
```

```
Epoch 10/50
6648/6648 [==============================] - 0s - loss: 20.6663
Epoch 11/50
6648/6648 [==============================] - 0s - loss: 13.7193
Epoch 12/50
6648/6648 [==============================] - 0s - loss: 9.3916
Epoch 13/50
6648/6648 [==============================] - 0s - loss: 6.7532
Epoch 14/50
6648/6648 [==============================] - 0s - loss: 5.0674
Epoch 15/50
6648/6648 [==============================] - 0s - loss: 3.7945
Epoch 16/50
6648/6648 [==============================] - 0s - loss: 2.9697
Epoch 17/50
6648/6648 [==============================] - 0s - loss: 2.4221
Epoch 18/50
6648/6648 [==============================] - 0s - loss: 1.9702
Epoch 19/50
6648/6648 [==============================] - 0s - loss: 1.6384
Epoch 20/50
6648/6648 [==============================] - 0s - loss: 1.3781
Epoch 21/50
6648/6648 [==============================] - 0s - loss: 1.1819
Epoch 22/50
6648/6648 [==============================] - 0s - loss: 1.0389
Epoch 23/50
6648/6648 [==============================] - 0s - loss: 0.9069
Epoch 24/50
6648/6648 [==============================] - 0s - loss: 0.8018
Epoch 25/50
6648/6648 [==============================] - 0s - loss: 0.7341
Epoch 26/50
6648/6648 [==============================] - 0s - loss: 0.6672
Epoch 27/50
6648/6648 [==============================] - 0s - loss: 0.6243
Epoch 28/50
6648/6648 [==============================] - 0s - loss: 0.5985
Epoch 29/50
6648/6648 [==============================] - 0s - loss: 0.5900
Epoch 30/50
6648/6648 [==============================] - 0s - loss: 0.5430
Epoch 31/50
6648/6648 [==============================] - 0s - loss: 0.4804
Epoch 32/50
6648/6648 [==============================] - 0s - loss: 0.6097
Epoch 33/50
6648/6648 [==============================] - 0s - loss: 0.7856
```

```
Epoch 34/50
6648/6648 [==============================] - 0s - loss: 0.5202
Epoch 35/50
6648/6648 [==============================] - 0s - loss: 1.5139
Epoch 36/50
6648/6648 [==============================] - 0s - loss: 2.2306
Epoch 37/50
6648/6648 [==============================] - 0s - loss: 2.1826
Epoch 38/50
6648/6648 [==============================] - 0s - loss: 10.4118
Epoch 39/50
6648/6648 [==============================] - 0s - loss: 1.5778
Epoch 40/50
6648/6648 [==============================] - 0s - loss: 1.2892
Epoch 41/50
6648/6648 [==============================] - 0s - loss: 2.5853
Epoch 42/50
6648/6648 [==============================] - 0s - loss: 7.9788
Epoch 43/50
6648/6648 [==============================] - 0s - loss: 3.6842
Epoch 44/50
6648/6648 [==============================] - 0s - loss: 1.5524
Epoch 45/50
6648/6648 [==============================] - 0s - loss: 9.6645
Epoch 46/50
6648/6648 [==============================] - 0s - loss: 3.5619
Epoch 47/50
6648/6648 [==============================] - 0s - loss: 4.0556
Epoch 48/50
6648/6648 [==============================] - 0s - loss: 8.8360
Epoch 49/50
6648/6648 [==============================] - 0s - loss: 2.9289
Epoch 50/50
6648/6648 [==============================] - 0s - loss: 16.2964
Epoch 1/50
6648/6648 [==============================] - 0s - loss: 75174.1706
Epoch 2/50
6648/6648 [==============================] - 0s - loss: 3390.7425
Epoch 3/50
6648/6648 [==============================] - 0s - loss: 1011.2583
Epoch 4/50
6648/6648 [==============================] - 0s - loss: 458.2169
Epoch 5/50
6648/6648 [==============================] - 0s - loss: 253.7832
Epoch 6/50
6648/6648 [==============================] - 0s - loss: 158.7919
Epoch 7/50
6648/6648 [==============================] - 0s - loss: 107.7765
```

```
Epoch 8/50
6648/6648 [==============================] - 0s - loss: 78.1121
Epoch 9/50
6648/6648 [==============================] - 0s - loss: 58.6356
Epoch 10/50
6648/6648 [==============================] - 0s - loss: 45.5865
Epoch 11/50
6648/6648 [==============================] - 0s - loss: 36.1042
Epoch 12/50
6648/6648 [==============================] - 0s - loss: 29.0643
Epoch 13/50
6648/6648 [==============================] - 0s - loss: 23.6964
Epoch 14/50
6648/6648 [==============================] - 0s - loss: 19.3840
Epoch 15/50
6648/6648 [==============================] - 0s - loss: 16.1155
Epoch 16/50
6648/6648 [==============================] - 0s - loss: 13.4751
Epoch 17/50
6648/6648 [==============================] - 0s - loss: 11.2227
Epoch 18/50
6648/6648 [==============================] - 0s - loss: 9.4224
Epoch 19/50
6648/6648 [==============================] - 0s - loss: 7.9972
Epoch 20/50
6648/6648 [==============================] - 0s - loss: 6.7382
Epoch 21/50
6648/6648 [==============================] - 0s - loss: 5.7587
Epoch 22/50
6648/6648 [==============================] - 0s - loss: 4.8153
Epoch 23/50
6648/6648 [==============================] - 0s - loss: 4.0937
Epoch 24/50
6648/6648 [==============================] - 0s - loss: 3.4365
Epoch 25/50
6648/6648 [==============================] - 0s - loss: 2.8622
Epoch 26/50
6648/6648 [==============================] - 0s - loss: 2.4482
Epoch 27/50
6648/6648 [==============================] - 0s - loss: 2.0255
Epoch 28/50
6648/6648 [==============================] - 0s - loss: 1.6858
Epoch 29/50
6648/6648 [==============================] - 0s - loss: 1.4096
Epoch 30/50
6648/6648 [==============================] - 0s - loss: 1.1979
Epoch 31/50
6648/6648 [==============================] - 0s - loss: 0.9754
```

```
Epoch 32/50
6648/6648 [==============================] - 0s - loss: 0.8226
Epoch 33/50
6648/6648 [==============================] - 0s - loss: 0.6836
Epoch 34/50
6648/6648 [==============================] - 0s - loss: 0.5738
Epoch 35/50
6648/6648 [==============================] - 0s - loss: 0.4829
Epoch 36/50
6648/6648 [==============================] - 0s - loss: 0.4111
Epoch 37/50
6648/6648 [==============================] - 0s - loss: 0.3580
Epoch 38/50
6648/6648 [==============================] - 0s - loss: 0.3111
Epoch 39/50
6648/6648 [==============================] - 0s - loss: 0.2715
Epoch 40/50
6648/6648 [==============================] - 0s - loss: 0.2459
Epoch 41/50
6648/6648 [==============================] - 0s - loss: 0.2219
Epoch 42/50
6648/6648 [==============================] - 0s - loss: 0.2041
Epoch 43/50
6648/6648 [==============================] - 0s - loss: 0.1894
Epoch 44/50
6648/6648 [==============================] - 0s - loss: 0.1777
Epoch 45/50
6648/6648 [==============================] - 0s - loss: 0.1702
Epoch 46/50
6648/6648 [==============================] - 0s - loss: 0.1631
Epoch 47/50
6648/6648 [==============================] - 0s - loss: 0.1549
Epoch 48/50
6648/6648 [==============================] - 0s - loss: 0.1533
Epoch 49/50
6648/6648 [==============================] - 0s - loss: 0.1535
Epoch 50/50
6648/6648 [==============================] - 0s - loss: 0.1462
Epoch 1/50
6648/6648 [==============================] - 0s - loss: 30898.4122
Epoch 2/50
6648/6648 [==============================] - 0s - loss: 836.1663
Epoch 3/50
6648/6648 [==============================] - 0s - loss: 290.3211
Epoch 4/50
6648/6648 [==============================] - 0s - loss: 135.8992
Epoch 5/50
6648/6648 [==============================] - 0s - loss: 75.1784
```

```
Epoch 6/50
6648/6648 [==============================] - 0s - loss: 47.1238
Epoch 7/50
6648/6648 [==============================] - 0s - loss: 31.9689      ETA: 0s - loss: 33.00
Epoch 8/50
6648/6648 [==============================] - 0s - loss: 22.5693
Epoch 9/50
6648/6648 [==============================] - 0s - loss: 16.2834
Epoch 10/50
6648/6648 [==============================] - 0s - loss: 12.0306
Epoch 11/50
6648/6648 [==============================] - 0s - loss: 8.9828
Epoch 12/50
6648/6648 [==============================] - 0s - loss: 6.7331
Epoch 13/50
6648/6648 [==============================] - 0s - loss: 5.1660
Epoch 14/50
6648/6648 [==============================] - 0s - loss: 3.9662
Epoch 15/50
6648/6648 [==============================] - 0s - loss: 3.0891
Epoch 16/50
6648/6648 [==============================] - 0s - loss: 2.4156
Epoch 17/50
6648/6648 [==============================] - 0s - loss: 1.9560
Epoch 18/50
6648/6648 [==============================] - 0s - loss: 1.5609
Epoch 19/50
6648/6648 [==============================] - 0s - loss: 1.2985
Epoch 20/50
6648/6648 [==============================] - 0s - loss: 1.0853
Epoch 21/50
6648/6648 [==============================] - 0s - loss: 0.9393
Epoch 22/50
6648/6648 [==============================] - 0s - loss: 0.8155
Epoch 23/50
6648/6648 [==============================] - 0s - loss: 0.7325
Epoch 24/50
6648/6648 [==============================] - 0s - loss: 0.6560
Epoch 25/50
6648/6648 [==============================] - 0s - loss: 0.6044
Epoch 26/50
6648/6648 [==============================] - 0s - loss: 0.5673
Epoch 27/50
6648/6648 [==============================] - 0s - loss: 0.5208
Epoch 28/50
6648/6648 [==============================] - 0s - loss: 0.4908
Epoch 29/50
6648/6648 [==============================] - 0s - loss: 0.4613
```

```
Epoch 30/50
6648/6648 [==============================] - 0s - loss: 0.4552
Epoch 31/50
6648/6648 [==============================] - 0s - loss: 0.4195
Epoch 32/50
6648/6648 [==============================] - 0s - loss: 1.6565
Epoch 33/50
6648/6648 [==============================] - 0s - loss: 2.0347
Epoch 34/50
6648/6648 [==============================] - 0s - loss: 5.4921
Epoch 35/50
6648/6648 [==============================] - 0s - loss: 0.8671
Epoch 36/50
6648/6648 [==============================] - 0s - loss: 0.4343
Epoch 37/50
6648/6648 [==============================] - 0s - loss: 6.0275
Epoch 38/50
6648/6648 [==============================] - 0s - loss: 0.4068
Epoch 39/50
6648/6648 [==============================] - 0s - loss: 0.3520
Epoch 40/50
6648/6648 [==============================] - 0s - loss: 8.1861
Epoch 41/50
6648/6648 [==============================] - 0s - loss: 0.3229
Epoch 42/50
6648/6648 [==============================] - 0s - loss: 0.3437
Epoch 43/50
6648/6648 [==============================] - 0s - loss: 10.7817
Epoch 44/50
6648/6648 [==============================] - 0s - loss: 0.2678
Epoch 45/50
6648/6648 [==============================] - 0s - loss: 0.2480
Epoch 46/50
6648/6648 [==============================] - 0s - loss: 0.4279
Epoch 47/50
6648/6648 [==============================] - 0s - loss: 7.4403
Epoch 48/50
6648/6648 [==============================] - 0s - loss: 0.3523
Epoch 49/50
6648/6648 [==============================] - 0s - loss: 10.7624
Epoch 50/50
6648/6648 [==============================] - 0s - loss: 0.4600
Epoch 1/50
6648/6648 [==============================] - 0s - loss: 41208.2412
Epoch 2/50
6648/6648 [==============================] - 0s - loss: 1242.4329
Epoch 3/50
6648/6648 [==============================] - 0s - loss: 316.8561
```

```
Epoch 4/50
6648/6648 [==============================] - 0s - loss: 137.1411
Epoch 5/50
6648/6648 [==============================] - 0s - loss: 79.3802
Epoch 6/50
6648/6648 [==============================] - 0s - loss: 52.1259
Epoch 7/50
6648/6648 [==============================] - 0s - loss: 35.5987
Epoch 8/50
6648/6648 [==============================] - 0s - loss: 25.3544
Epoch 9/50
6648/6648 [==============================] - 0s - loss: 18.3092
Epoch 10/50
6648/6648 [==============================] - 0s - loss: 13.5493
Epoch 11/50
6648/6648 [==============================] - 0s - loss: 10.2218
Epoch 12/50
6648/6648 [==============================] - 0s - loss: 7.9138
Epoch 13/50
6648/6648 [==============================] - 0s - loss: 6.1863
Epoch 14/50
6648/6648 [==============================] - 0s - loss: 4.9048
Epoch 15/50
6648/6648 [==============================] - 0s - loss: 3.9538
Epoch 16/50
6648/6648 [==============================] - 0s - loss: 3.2046
Epoch 17/50
6648/6648 [==============================] - 0s - loss: 2.6511
Epoch 18/50
6648/6648 [==============================] - 0s - loss: 2.1830
Epoch 19/50
6648/6648 [==============================] - 0s - loss: 1.8321
Epoch 20/50
6648/6648 [==============================] - 0s - loss: 1.5354
Epoch 21/50
6648/6648 [==============================] - 0s - loss: 1.3098
Epoch 22/50
6648/6648 [==============================] - 0s - loss: 1.1460
Epoch 23/50
6648/6648 [==============================] - 0s - loss: 0.9827
Epoch 24/50
6648/6648 [==============================] - 0s - loss: 0.8554
Epoch 25/50
6648/6648 [==============================] - 0s - loss: 0.7594
Epoch 26/50
6648/6648 [==============================] - 0s - loss: 0.6678
Epoch 27/50
6648/6648 [==============================] - 0s - loss: 0.5998
```

```
Epoch 28/50
6648/6648 [==============================] - 0s - loss: 0.5574
Epoch 29/50
6648/6648 [==============================] - 0s - loss: 0.4949
Epoch 30/50
6648/6648 [==============================] - 0s - loss: 0.4386
Epoch 31/50
6648/6648 [==============================] - 0s - loss: 0.4236
Epoch 32/50
6648/6648 [==============================] - 0s - loss: 0.3871
Epoch 33/50
6648/6648 [==============================] - 0s - loss: 0.4411
Epoch 34/50
6648/6648 [==============================] - 0s - loss: 0.4763
Epoch 35/50
6648/6648 [==============================] - 0s - loss: 0.4330
Epoch 36/50
6648/6648 [==============================] - 0s - loss: 13.7485
Epoch 37/50
6648/6648 [==============================] - 0s - loss: 1.4178
Epoch 38/50
6648/6648 [==============================] - 0s - loss: 0.3014
Epoch 39/50
6648/6648 [==============================] - 0s - loss: 0.3833
Epoch 40/50
6648/6648 [==============================] - 0s - loss: 1.5769
Epoch 41/50
6648/6648 [==============================] - 0s - loss: 5.9136
Epoch 42/50
6648/6648 [==============================] - 0s - loss: 4.1612
Epoch 43/50
6648/6648 [==============================] - 0s - loss: 41.5834
Epoch 44/50
6648/6648 [==============================] - 0s - loss: 0.6461
Epoch 45/50
6648/6648 [==============================] - 0s - loss: 0.4328
Epoch 46/50
6648/6648 [==============================] - 0s - loss: 7.1628
Epoch 47/50
6648/6648 [==============================] - 0s - loss: 4.8178
Epoch 48/50
6648/6648 [==============================] - 0s - loss: 15.2777
Epoch 49/50
6648/6648 [==============================] - 0s - loss: 1.1087
Epoch 50/50
6648/6648 [==============================] - 0s - loss: 1.0892
Epoch 1/50
6648/6648 [==============================] - 0s - loss: 30840.9321
```

```
Epoch 2/50
6648/6648 [==============================] - 0s - loss: 729.5566
Epoch 3/50
6648/6648 [==============================] - 0s - loss: 213.0054
Epoch 4/50
6648/6648 [==============================] - 0s - loss: 98.6354
Epoch 5/50
6648/6648 [==============================] - 0s - loss: 55.1363
Epoch 6/50
6648/6648 [==============================] - 0s - loss: 33.5962
Epoch 7/50
6648/6648 [==============================] - 0s - loss: 21.7004      ETA: 0s - loss: 2
Epoch 8/50
6648/6648 [==============================] - 0s - loss: 14.7286
Epoch 9/50
6648/6648 [==============================] - 0s - loss: 10.4593
Epoch 10/50
6648/6648 [==============================] - 0s - loss: 7.7516
Epoch 11/50
6648/6648 [==============================] - 0s - loss: 5.9424
Epoch 12/50
6648/6648 [==============================] - 0s - loss: 4.7227
Epoch 13/50
6648/6648 [==============================] - 0s - loss: 3.7353
Epoch 14/50
6648/6648 [==============================] - 0s - loss: 3.0602
Epoch 15/50
6648/6648 [==============================] - 0s - loss: 2.5337
Epoch 16/50
6648/6648 [==============================] - 0s - loss: 2.1456
Epoch 17/50
6648/6648 [==============================] - 0s - loss: 1.8442
Epoch 18/50
6648/6648 [==============================] - 0s - loss: 1.6600
Epoch 19/50
6648/6648 [==============================] - 0s - loss: 1.4585
Epoch 20/50
6648/6648 [==============================] - 0s - loss: 1.2599
Epoch 21/50
6648/6648 [==============================] - 0s - loss: 1.1802
Epoch 22/50
6648/6648 [==============================] - 0s - loss: 1.0950
Epoch 23/50
6648/6648 [==============================] - 0s - loss: 0.9522
Epoch 24/50
6648/6648 [==============================] - ETA: 0s - loss: 2.055 - 0s - loss: 8.6520
Epoch 25/50
6648/6648 [==============================] - 0s - loss: 1.1909
```

```
Epoch 26/50
6648/6648 [==============================] - 0s - loss: 1.0278
Epoch 27/50
6648/6648 [==============================] - 0s - loss: 9.4667
Epoch 28/50
6648/6648 [==============================] - 0s - loss: 7.3327
Epoch 29/50
6648/6648 [==============================] - 0s - loss: 0.6270
Epoch 30/50
6648/6648 [==============================] - 0s - loss: 6.6733
Epoch 31/50
6648/6648 [==============================] - 0s - loss: 8.1734
Epoch 32/50
6648/6648 [==============================] - 0s - loss: 2.8290
Epoch 33/50
6648/6648 [==============================] - 0s - loss: 0.6895
Epoch 34/50
6648/6648 [==============================] - 0s - loss: 13.0529
Epoch 35/50
6648/6648 [==============================] - 0s - loss: 22.2635
Epoch 36/50
6648/6648 [==============================] - 0s - loss: 18.6810
Epoch 37/50
6648/6648 [==============================] - 0s - loss: 0.3544
Epoch 38/50
6648/6648 [==============================] - 0s - loss: 0.3576
Epoch 39/50
6648/6648 [==============================] - 0s - loss: 0.6570
Epoch 40/50
6648/6648 [==============================] - 0s - loss: 10.9762
Epoch 41/50
6648/6648 [==============================] - 0s - loss: 2.8192
Epoch 42/50
6648/6648 [==============================] - 0s - loss: 23.3539
Epoch 43/50
6648/6648 [==============================] - 0s - loss: 22.3337
Epoch 44/50
6648/6648 [==============================] - 0s - loss: 0.8488
Epoch 45/50
6648/6648 [==============================] - 0s - loss: 11.3268
Epoch 46/50
6648/6648 [==============================] - 0s - loss: 3.8563
Epoch 47/50
6648/6648 [==============================] - 0s - loss: 58.1066
Epoch 48/50
6648/6648 [==============================] - 0s - loss: 1.8897
Epoch 49/50
6648/6648 [==============================] - 0s - loss: 8.5570
```

```
Epoch 50/50
6648/6648 [==============================] - 0s - loss: 0.3140
Epoch 1/50
6648/6648 [==============================] - 0s - loss: 31579.7100
Epoch 2/50
6648/6648 [==============================] - 0s - loss: 817.7438
Epoch 3/50
6648/6648 [==============================] - 0s - loss: 256.8814
Epoch 4/50
6648/6648 [==============================] - 0s - loss: 120.3248
Epoch 5/50
6648/6648 [==============================] - 0s - loss: 66.4461
Epoch 6/50
6648/6648 [==============================] - 0s - loss: 40.4838
Epoch 7/50
6648/6648 [==============================] - ETA: 0s - loss: 27.55 - 0s - loss: 26.3884
Epoch 8/50
6648/6648 [==============================] - 0s - loss: 18.1691
Epoch 9/50
6648/6648 [==============================] - 0s - loss: 13.0592
Epoch 10/50
6648/6648 [==============================] - 0s - loss: 9.7419
Epoch 11/50
6648/6648 [==============================] - 0s - loss: 7.4694
Epoch 12/50
6648/6648 [==============================] - 0s - loss: 5.9192
Epoch 13/50
6648/6648 [==============================] - 0s - loss: 4.7774
Epoch 14/50
6648/6648 [==============================] - 0s - loss: 3.9357
Epoch 15/50
6648/6648 [==============================] - 0s - loss: 3.2899
Epoch 16/50
6648/6648 [==============================] - 0s - loss: 2.8042
Epoch 17/50
6648/6648 [==============================] - 0s - loss: 2.4091
Epoch 18/50
6648/6648 [==============================] - 0s - loss: 2.0909
Epoch 19/50
6648/6648 [==============================] - 0s - loss: 1.8283
Epoch 20/50
6648/6648 [==============================] - 0s - loss: 1.6206
Epoch 21/50
6648/6648 [==============================] - 0s - loss: 1.4354
Epoch 22/50
6648/6648 [==============================] - 0s - loss: 1.2795
Epoch 23/50
6648/6648 [==============================] - 0s - loss: 1.1474
```

```
Epoch 24/50
6648/6648 [==============================] - 0s - loss: 1.0405
Epoch 25/50
6648/6648 [==============================] - 0s - loss: 0.9307
Epoch 26/50
6648/6648 [==============================] - 0s - loss: 0.8424
Epoch 27/50
6648/6648 [==============================] - 0s - loss: 0.7699
Epoch 28/50
6648/6648 [==============================] - 0s - loss: 0.7004
Epoch 29/50
6648/6648 [==============================] - 0s - loss: 0.6340
Epoch 30/50
6648/6648 [==============================] - 0s - loss: 0.5886
Epoch 31/50
6648/6648 [==============================] - 0s - loss: 0.5329
Epoch 32/50
6648/6648 [==============================] - 0s - loss: 0.4939
Epoch 33/50
6648/6648 [==============================] - 0s - loss: 0.4514
Epoch 34/50
6648/6648 [==============================] - 0s - loss: 0.4209
Epoch 35/50
6648/6648 [==============================] - 0s - loss: 0.3841
Epoch 36/50
6648/6648 [==============================] - 0s - loss: 0.3691
Epoch 37/50
6648/6648 [==============================] - 0s - loss: 0.3747
Epoch 38/50
6648/6648 [==============================] - 0s - loss: 7.9195
Epoch 39/50
6648/6648 [==============================] - 0s - loss: 0.3409
Epoch 40/50
6648/6648 [==============================] - 0s - loss: 0.3198
Epoch 41/50
6648/6648 [==============================] - 0s - loss: 0.3003
Epoch 42/50
6648/6648 [==============================] - 0s - loss: 31.6434
Epoch 43/50
6648/6648 [==============================] - 0s - loss: 2.9612
Epoch 44/50
6648/6648 [==============================] - 0s - loss: 0.2002
Epoch 45/50
6648/6648 [==============================] - 0s - loss: 0.1968
Epoch 46/50
6648/6648 [==============================] - 0s - loss: 0.1918
Epoch 47/50
6648/6648 [==============================] - 0s - loss: 0.1865
```

```
Epoch 48/50
6648/6648 [==============================] - 0s - loss: 0.1892
Epoch 49/50
6648/6648 [==============================] - 0s - loss: 0.1882
Epoch 50/50
6648/6648 [==============================] - 0s - loss: 0.3200
Epoch 1/50
6648/6648 [==============================] - 0s - loss: 37232.1088
Epoch 2/50
6648/6648 [==============================] - 0s - loss: 1391.2557
Epoch 3/50
6648/6648 [==============================] - 0s - loss: 357.0264
Epoch 4/50
6648/6648 [==============================] - 0s - loss: 147.6429
Epoch 5/50
6648/6648 [==============================] - 0s - loss: 74.1681
Epoch 6/50
6648/6648 [==============================] - 0s - loss: 42.4392
Epoch 7/50
6648/6648 [==============================] - 0s - loss: 26.2989
Epoch 8/50
6648/6648 [==============================] - 0s - loss: 17.5223
Epoch 9/50
6648/6648 [==============================] - 0s - loss: 12.1952
Epoch 10/50
6648/6648 [==============================] - 0s - loss: 8.8680
Epoch 11/50
6648/6648 [==============================] - 0s - loss: 6.5203
Epoch 12/50
6648/6648 [==============================] - 0s - loss: 4.8873
Epoch 13/50
6648/6648 [==============================] - 0s - loss: 3.6868
Epoch 14/50
6648/6648 [==============================] - 0s - loss: 2.8095
Epoch 15/50
6648/6648 [==============================] - 0s - loss: 2.1474
Epoch 16/50
6648/6648 [==============================] - 0s - loss: 1.6348
Epoch 17/50
6648/6648 [==============================] - 0s - loss: 1.2481
Epoch 18/50
6648/6648 [==============================] - 0s - loss: 0.9501
Epoch 19/50
6648/6648 [==============================] - 0s - loss: 0.7258
Epoch 20/50
6648/6648 [==============================] - 0s - loss: 0.5715
Epoch 21/50
6648/6648 [==============================] - 0s - loss: 0.4509
```

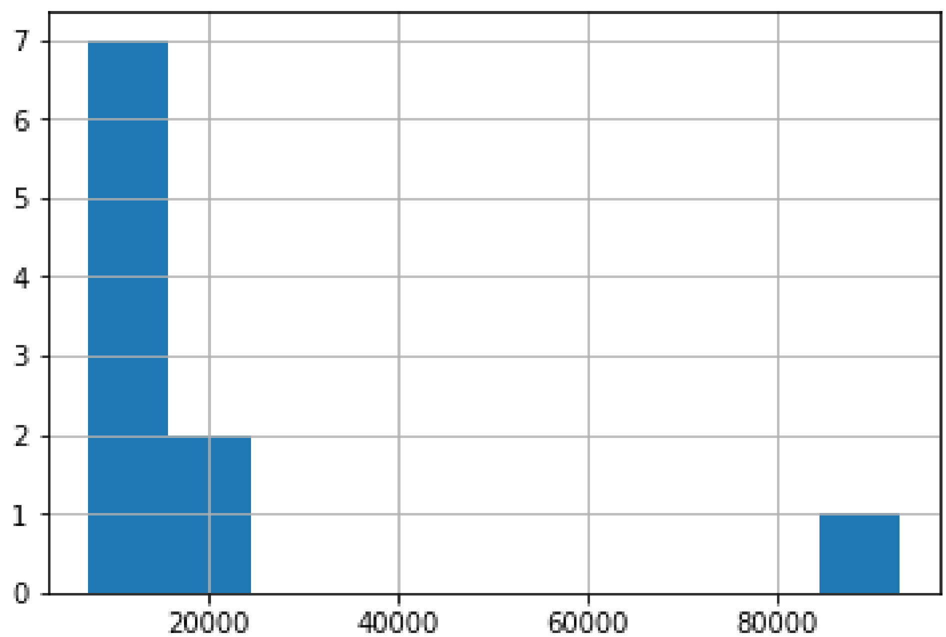
```
Epoch 22/50
6648/6648 [==============================] - 0s - loss: 0.3629
Epoch 23/50
6648/6648 [==============================] - 0s - loss: 0.2885
Epoch 24/50
6648/6648 [==============================] - 0s - loss: 0.2402
Epoch 25/50
6648/6648 [==============================] - 0s - loss: 0.2088
Epoch 26/50
6648/6648 [==============================] - 0s - loss: 0.1809
Epoch 27/50
6648/6648 [==============================] - 0s - loss: 0.1681
Epoch 28/50
6648/6648 [==============================] - 0s - loss: 0.1511
Epoch 29/50
6648/6648 [==============================] - 0s - loss: 0.1378
Epoch 30/50
6648/6648 [==============================] - 0s - loss: 0.1311
Epoch 31/50
6648/6648 [==============================] - 0s - loss: 0.1419
Epoch 32/50
6648/6648 [==============================] - 0s - loss: 0.1281
Epoch 33/50
6648/6648 [==============================] - 0s - loss: 0.1353
Epoch 34/50
6648/6648 [==============================] - 0s - loss: 0.2832
Epoch 35/50
6648/6648 [==============================] - 0s - loss: 0.4271
Epoch 36/50
6648/6648 [==============================] - 0s - loss: 6.5683
Epoch 37/50
6648/6648 [==============================] - 0s - loss: 19.5332
Epoch 38/50
6648/6648 [==============================] - 0s - loss: 0.1448
Epoch 39/50
6648/6648 [==============================] - 0s - loss: 0.2110
Epoch 40/50
6648/6648 [==============================] - 0s - loss: 2.4972
Epoch 41/50
6648/6648 [==============================] - 0s - loss: 55.7990
Epoch 42/50
6648/6648 [==============================] - 0s - loss: 3.0382
Epoch 43/50
6648/6648 [==============================] - 0s - loss: 1.5621
Epoch 44/50
6648/6648 [==============================] - 0s - loss: 24.1504
Epoch 45/50
6648/6648 [==============================] - 0s - loss: 0.2677
```

```
Epoch 46/50
6648/6648 [==============================] - 0s - loss: 9.0643
Epoch 47/50
6648/6648 [==============================] - 0s - loss: 13.9177
Epoch 48/50
6648/6648 [==============================] - 0s - loss: 55.4026
Epoch 49/50
6648/6648 [==============================] - 0s - loss: 0.1998
Epoch 50/50
6648/6648 [==============================] - 0s - loss: 0.1059
Epoch 1/50
6648/6648 [==============================] - 0s - loss: 24780.5021
Epoch 2/50
6648/6648 [==============================] - 0s - loss: 823.4628
Epoch 3/50
6648/6648 [==============================] - 0s - loss: 260.6005
Epoch 4/50
6648/6648 [==============================] - 0s - loss: 124.2759
Epoch 5/50
6648/6648 [==============================] - 0s - loss: 70.0041
Epoch 6/50
6648/6648 [==============================] - 0s - loss: 42.2772
Epoch 7/50
6648/6648 [==============================] - 0s - loss: 26.5562
Epoch 8/50
6648/6648 [==============================] - 0s - loss: 17.2113
Epoch 9/50
6648/6648 [==============================] - 0s - loss: 11.5134
Epoch 10/50
6648/6648 [==============================] - 0s - loss: 8.0024
Epoch 11/50
6648/6648 [==============================] - 0s - loss: 5.8063
Epoch 12/50
6648/6648 [==============================] - 0s - loss: 4.2741
Epoch 13/50
6648/6648 [==============================] - 0s - loss: 3.2988
Epoch 14/50
6648/6648 [==============================] - 0s - loss: 2.5922
Epoch 15/50
6648/6648 [==============================] - 0s - loss: 2.0840
Epoch 16/50
6648/6648 [==============================] - 0s - loss: 1.7104
Epoch 17/50
6648/6648 [==============================] - 0s - loss: 1.4480
Epoch 18/50
6648/6648 [==============================] - 0s - loss: 1.2377
Epoch 19/50
6648/6648 [==============================] - 0s - loss: 1.0683
```

```
Epoch 20/50
6648/6648 [==============================] - 0s - loss: 0.9503
Epoch 21/50
6648/6648 [==============================] - 0s - loss: 0.8319
Epoch 22/50
6648/6648 [==============================] - 0s - loss: 0.7261
Epoch 23/50
6648/6648 [==============================] - 0s - loss: 0.6530
Epoch 24/50
6648/6648 [==============================] - 0s - loss: 0.5976
Epoch 25/50
6648/6648 [==============================] - 0s - loss: 0.5383
Epoch 26/50
6648/6648 [==============================] - 0s - loss: 0.5061
Epoch 27/50
6648/6648 [==============================] - 0s - loss: 0.4747
Epoch 28/50
6648/6648 [==============================] - 0s - loss: 0.4311
Epoch 29/50
6648/6648 [==============================] - 0s - loss: 0.4291
Epoch 30/50
6648/6648 [==============================] - 0s - loss: 0.4156
Epoch 31/50
6648/6648 [==============================] - 0s - loss: 0.3955
Epoch 32/50
6648/6648 [==============================] - 0s - loss: 0.3931
Epoch 33/50
6648/6648 [==============================] - 0s - loss: 0.3117
Epoch 34/50
6648/6648 [==============================] - 0s - loss: 0.8605
Epoch 35/50
6648/6648 [==============================] - 0s - loss: 0.4854
Epoch 36/50
6648/6648 [==============================] - 0s - loss: 2.3523
Epoch 37/50
6648/6648 [==============================] - 0s - loss: 18.5607
Epoch 38/50
6648/6648 [==============================] - 0s - loss: 0.3433
Epoch 39/50
6648/6648 [==============================] - 0s - loss: 1.0557
Epoch 40/50
6648/6648 [==============================] - 0s - loss: 11.9116
Epoch 41/50
6648/6648 [==============================] - 0s - loss: 2.5800
Epoch 42/50
6648/6648 [==============================] - 0s - loss: 2.5337
Epoch 43/50
6648/6648 [==============================] - 0s - loss: 22.9881
```

```
Epoch 44/50
6648/6648 [==============================] - 0s - loss: 18.9495
Epoch 45/50
6648/6648 [==============================] - 0s - loss: 0.3280
Epoch 46/50
6648/6648 [==============================] - 0s - loss: 0.2617
Epoch 47/50
6648/6648 [==============================] - 0s - loss: 0.4279
Epoch 48/50
6648/6648 [==============================] - 0s - loss: 43.3865
Epoch 49/50
6648/6648 [==============================] - 0s - loss: 11.1643
Epoch 50/50
6648/6648 [==============================] - 0s - loss: 0.7446
```

```
In [20]: plt.hist(listOfErrors2)
         plt.grid(True)
         plt.show()

         print("Mean score: " , np.mean(listOfErrors2))
         print("KFold scores: ", listOfErrors2)
```




```
Mean score:  20928.5348776
KFold scores:  [10963.364549959259, 13219.340977444715, 93156.21636866506, 7302.5751914463171,
```

Q#9 Are you satisfied with the quality of the final estimator? What would be your recommendation for your boss on this issue?

Using the relu activation function, gives fast and consisten estimations, though they are limited in their precision.

However the linear activation function gives great precision, but requires many iterations, and therefore a lot more time, and is not very consistent. Possible you can tune the values, and get some consistent high-precision results.

The problem using relu, is that it is not differentiable in zero, and this data has a lot of zero's therefore it will not learn well.

4 Extra task for 3-person groups

Q#10 Repeat the process, once reducing to 15 sensors, and once reducing to 25 sensors. The costs to keep these sensors running is directly proportional to the number of sensors used. Does changing the number of sensors used change your conclusion in Q#9?

We are only 2 people