University_Building_ANN_Model

November 23, 2020

[34]:
```
# import the libraries
import numpy as np
import matplotlib.pyplot as plt
import datetime
import seaborn as sns
import tensorflow as tf
import pandas as pd
from datetime import datetime
print("TensorFlow version: ",tf.__version__)  #print the version of tensorflow
```

```
TensorFlow version:  2.3.0
```

[35]:
```
from tensorflow.python.keras.layers import Dense
from tensorflow.keras.layers import Dropout
from tensorflow.python.keras.models import Sequential
from tensorflow.python.keras.wrappers.scikit_learn import KerasRegressor
from tensorflow import keras
from tensorflow.keras import layers
from tensorflow.keras import Sequential
from tensorflow.keras.layers import Dense, Activation, Dropout
from tensorflow.keras.callbacks import EarlyStopping
from tensorflow.keras import regularizers
from tensorflow.keras.callbacks import EarlyStopping, ReduceLROnPlateau
from tensorflow.keras import regularizers
```

[36]:
```
#Helper Functions
def get_weekday2(year, month, day):
    dates = pd.DataFrame()
    dates['y'] = year
    dates['m'] = month
    dates['d'] = day
    dates['dates'] = dates['y'].astype('str') + '-' + dates['m'].astype('str')␣
 ↪+ '-' + dates['d'].astype('str')
    return get_weekday(dates['dates'])

#Get day of week based on date
def get_weekday(dates):
```

```
    return [1 if (datetime.strptime(d,"%Y-%m-%d").weekday() >= 5) else 0 for d␣
↪in dates]
```

1 Exploratory Data Analysis

[37]:
```
data = pd.read_csv('https://raw.githubusercontent.com/A-Wadhwani/ME597-Project/
↪main/Datasets/Combined_PowerWeatherData2.csv')
copy = data
data
```

[37]:
```
       Year  Month  …  Square Feet       Type
0      2016      8  …       113866  Classroom
1      2016      8  …       113866  Classroom
2      2016      8  …       113866  Classroom
3      2016      8  …       113866  Classroom
4      2016      8  …       113866  Classroom
…       …     …   …          …          …
29987  2019     12  …        59548  Classroom
29988  2019     12  …        59548  Classroom
29989  2019     12  …        59548  Classroom
29990  2019     12  …        59548  Classroom
29991  2019     12  …        59548  Classroom

[29992 rows x 18 columns]
```

[38]:
```
data.describe()
```

[38]:
```
              Year         Month  …       Weekday    Square Feet
count  29992.000000  29992.000000  …  29992.000000   29992.000000
mean    2018.172446      6.771239  …      0.286977   95477.812017
std        1.048141      3.404085  …      0.452358   41527.248371
min     2016.000000      1.000000  …      0.000000   10932.000000
25%     2017.000000      4.000000  …      0.000000   59548.000000
50%     2019.000000      7.000000  …      0.000000  105545.000000
75%     2019.000000     10.000000  …      1.000000  121074.000000
max     2019.000000     12.000000  …      1.000000  238270.000000

[8 rows x 15 columns]
```

[39]:
```
#Creating column to denote each building type
from sklearn.preprocessing import OrdinalEncoder
encoder = OrdinalEncoder()
data['Type'] = encoder.fit_transform(np.reshape(data['Type'].values, (-1,1)))
data['Type'].describe()
```

[39]:
```
count    29992.000000
mean         0.519405
std          0.499632
min          0.000000
25%          0.000000
50%          1.000000
75%          1.000000
max          1.000000
Name: Type, dtype: float64
```

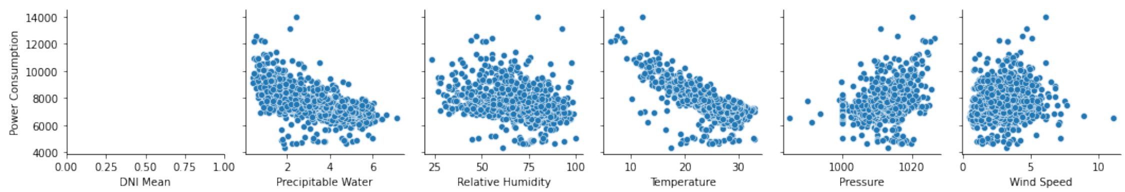
[40]:
```
encoder.inverse_transform(np.reshape([0, 1], (-1,1)))
```

[40]:
```
array([['Classroom'],
       ['Laboratory']], dtype=object)
```

[41]:
```
#Removing unnecessary columns
data = data.drop(['Year'], axis=1)
data = data.drop(['University Name', 'Building Name'], axis=1)
data.head()
```

[41]:
```
   Month  Day    DNI Mean  …  Weekday  Square Feet  Type
0      8   10  310.769231  …        0       113866   0.0
1      8   11  458.363636  …        0       113866   0.0
2      8   12  419.000000  …        0       113866   0.0
3      8   13  498.666667  …        1       113866   0.0
4      8   14   47.666667  …        1       113866   0.0

[5 rows x 15 columns]
```

[42]:
```
#Select one building's data
view = data[data['Square Feet'] == 113866]
#See graphs for data vs Power Consumption
sns.pairplot(view, x_vars = ['DNI Mean', 'Precipitable Water', 'Relative␣
↪Humidity', 'Temperature', 'Pressure', 'Wind Speed'], y_vars=['Power␣
↪Consumption'])
plt.show()
```



[43]:
```
#Splitting into X and Y
X = data.drop(["Power Consumption"],axis=1)
y = data["Power Consumption"]
```

[44]:
```
y = np.reshape(y.values, (-1,1))
```

[45]:
```
# scaling inputs using RobustScaler
from sklearn.preprocessing import RobustScaler
x_scaler = RobustScaler()
y_scaler = RobustScaler()

x_f = x_scaler.fit_transform(X)
y_f = y_scaler.fit_transform(y)

x_f = pd.DataFrame(x_f)
```

[46]:
```
x_f
```

[46]:
```
              0         1         2         3  …        10   11        12   13
0      0.166667 -0.400000 -0.498333 -0.033019  …  0.000000  0.0  0.135244 -1.0
1      0.166667 -0.333333 -0.095546 -0.051887  …  0.000000  0.0  0.135244 -1.0
2      0.166667 -0.266667 -0.202970  0.117925  …  0.000000  0.0  0.135244 -1.0
3      0.166667 -0.200000  0.014441  0.221698  …  0.000000  1.0  0.135244 -1.0
4      0.166667 -0.133333 -1.216342 -3.584906  …  0.000000  1.0  0.135244 -1.0
…           …       …        …       …     …  …       …   …
29987  0.833333  0.733333  0.293846  0.146226  … -0.725350  0.0 -0.747603 -1.0
29988  0.833333  0.800000  0.458542  0.216981  … -0.568777  1.0 -0.747603 -1.0
29989  0.833333  0.866667 -1.241553 -3.570755  … -0.689288  1.0 -0.747603 -1.0
29990  0.833333  0.933333  0.472050  0.122642  … -0.626293  0.0 -0.747603 -1.0
29991  0.833333  1.000000 -0.041686 -0.485849  … -0.578363  0.0 -0.747603 -1.0

[29992 rows x 14 columns]
```

[47]:
```
x_f = x_f.values
```

[48]:
```
x_f.dtype
```

[48]:
```
dtype('float64')
```

[49]:
```
# split the data into train and test sets
from sklearn.model_selection import train_test_split
x_f_train, x_f_test, y_f_train, y_f_test = train_test_split(x_f,y_f, test_size
↪= 0.25, shuffle=True,random_state=24)
```

[50]:
```
# print the number of training and test damples
print("Number of training samples: ",len(x_f_train))
print("Number of testing samples: ",len(x_f_test))
```

```
Number of training samples:  22494
Number of testing samples:  7498
```

2 Building the Model

[68]:
```
model = Sequential()

model.add(Dense(512, input_shape=(14, ), activation='relu', name='dense_1'))
model.add(Dense(256, activation='relu', name='dense_2'))
model.add(Dense(128, activation='relu', name='dense_3'))
model.add(Dense(64, activation='relu', name='dense_4'))
model.add(Dense(32, activation='relu', name='dense_5'))
model.add(Dense(16, activation='relu', name='dense_6'))
model.add(Dense(1, activation='linear', name='dense_output'))

model.summary()
```

```
Model: "sequential_7"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense_1 (Dense)              (None, 512)               7680
_________________________________________________________________
dense_2 (Dense)              (None, 256)               131328
_________________________________________________________________
dense_3 (Dense)              (None, 128)               32896
_________________________________________________________________
dense_4 (Dense)              (None, 64)                8256
_________________________________________________________________
dense_5 (Dense)              (None, 32)                2080
_________________________________________________________________
dense_6 (Dense)              (None, 16)                528
_________________________________________________________________
dense_output (Dense)         (None, 1)                 17
=================================================================
Total params: 182,785
Trainable params: 182,785
Non-trainable params: 0
_________________________________________________________________
```

[69]:
```
opt = keras.optimizers.Adam(learning_rate = 0.001)
model.compile(loss='mae', optimizer=opt, metrics=['mse', 'mae'])

#Tensorboard tool callback
log_dir = ''
#log_dir = "logs\\fit3\\" + datetime.now().strftime("%M")
```

```
tensorboard_callback = tf.keras.callbacks.TensorBoard(log_dir=log_dir,␣
 ↪histogram_freq=1, profile_batch = 100000000)

#Reduce Learning rate on Plateau
reduce_lr = ReduceLROnPlateau(monitor='val_loss', factor=0.2, patience=10,␣
 ↪verbose = 1)

#Earlystopping callback
early_stop = EarlyStopping(monitor ='val_loss', min_delta= 1e-3, patience = 40,␣
 ↪verbose = 1, restore_best_weights=True)

history = model.fit(x_f_train, y_f_train, callbacks = [tensorboard_callback,␣
 ↪early_stop, reduce_lr],
                    validation_data=(x_f_test, y_f_test), epochs=400,␣
 ↪batch_size=90, verbose=1)
```

```
Epoch 1/400
250/250 [==============================] - 1s 4ms/step - loss: 0.3636 - mse:
0.2813 - mae: 0.3636 - val_loss: 0.2983 - val_mse: 0.2170 - val_mae: 0.2983
Epoch 2/400
250/250 [==============================] - 1s 4ms/step - loss: 0.2681 - mse:
0.1755 - mae: 0.2681 - val_loss: 0.2552 - val_mse: 0.1753 - val_mae: 0.2552
Epoch 3/400
250/250 [==============================] - 1s 3ms/step - loss: 0.2330 - mse:
0.1489 - mae: 0.2330 - val_loss: 0.2101 - val_mse: 0.1455 - val_mae: 0.2101
Epoch 4/400
250/250 [==============================] - 1s 4ms/step - loss: 0.2122 - mse:
0.1328 - mae: 0.2122 - val_loss: 0.2174 - val_mse: 0.1459 - val_mae: 0.2174
Epoch 5/400
250/250 [==============================] - 1s 3ms/step - loss: 0.1778 - mse:
0.1061 - mae: 0.1778 - val_loss: 0.1797 - val_mse: 0.1103 - val_mae: 0.1797
Epoch 6/400
250/250 [==============================] - 1s 4ms/step - loss: 0.1678 - mse:
0.1002 - mae: 0.1678 - val_loss: 0.1640 - val_mse: 0.1036 - val_mae: 0.1640
Epoch 7/400
250/250 [==============================] - 1s 4ms/step - loss: 0.1628 - mse:
0.0978 - mae: 0.1628 - val_loss: 0.1580 - val_mse: 0.1035 - val_mae: 0.1580
Epoch 8/400
250/250 [==============================] - 1s 4ms/step - loss: 0.1585 - mse:
0.0969 - mae: 0.1585 - val_loss: 0.1616 - val_mse: 0.1040 - val_mae: 0.1616
Epoch 9/400
250/250 [==============================] - 1s 4ms/step - loss: 0.1540 - mse:
0.0942 - mae: 0.1540 - val_loss: 0.1524 - val_mse: 0.0983 - val_mae: 0.1524
Epoch 10/400
250/250 [==============================] - 1s 4ms/step - loss: 0.1521 - mse:
0.0948 - mae: 0.1521 - val_loss: 0.1574 - val_mse: 0.1050 - val_mae: 0.1574
Epoch 11/400
```

```
250/250 [==============================] - 1s 4ms/step - loss: 0.1511 - mse:
0.0932 - mae: 0.1511 - val_loss: 0.1610 - val_mse: 0.1026 - val_mae: 0.1610
Epoch 12/400
250/250 [==============================] - 1s 4ms/step - loss: 0.1503 - mse:
0.0928 - mae: 0.1503 - val_loss: 0.1492 - val_mse: 0.0976 - val_mae: 0.1492
Epoch 13/400
250/250 [==============================] - 1s 4ms/step - loss: 0.1491 - mse:
0.0924 - mae: 0.1491 - val_loss: 0.1530 - val_mse: 0.0993 - val_mae: 0.1530
Epoch 14/400
250/250 [==============================] - 1s 4ms/step - loss: 0.1482 - mse:
0.0915 - mae: 0.1482 - val_loss: 0.1446 - val_mse: 0.0983 - val_mae: 0.1446
Epoch 15/400
250/250 [==============================] - 1s 4ms/step - loss: 0.1431 - mse:
0.0895 - mae: 0.1431 - val_loss: 0.1442 - val_mse: 0.0921 - val_mae: 0.1442
Epoch 16/400
250/250 [==============================] - 1s 4ms/step - loss: 0.1425 - mse:
0.0890 - mae: 0.1425 - val_loss: 0.1451 - val_mse: 0.0959 - val_mae: 0.1451
Epoch 17/400
250/250 [==============================] - 1s 4ms/step - loss: 0.1425 - mse:
0.0885 - mae: 0.1425 - val_loss: 0.1521 - val_mse: 0.0956 - val_mae: 0.1521
Epoch 18/400
250/250 [==============================] - 1s 4ms/step - loss: 0.1392 - mse:
0.0864 - mae: 0.1392 - val_loss: 0.1430 - val_mse: 0.0942 - val_mae: 0.1430
Epoch 19/400
250/250 [==============================] - 1s 4ms/step - loss: 0.1366 - mse:
0.0830 - mae: 0.1366 - val_loss: 0.1476 - val_mse: 0.0941 - val_mae: 0.1476
Epoch 20/400
250/250 [==============================] - 1s 4ms/step - loss: 0.1351 - mse:
0.0785 - mae: 0.1351 - val_loss: 0.1454 - val_mse: 0.0860 - val_mae: 0.1454
Epoch 21/400
250/250 [==============================] - 1s 4ms/step - loss: 0.1368 - mse:
0.0773 - mae: 0.1368 - val_loss: 0.1364 - val_mse: 0.0797 - val_mae: 0.1364
Epoch 22/400
250/250 [==============================] - 1s 4ms/step - loss: 0.1283 - mse:
0.0726 - mae: 0.1283 - val_loss: 0.1287 - val_mse: 0.0754 - val_mae: 0.1287
Epoch 23/400
250/250 [==============================] - 1s 4ms/step - loss: 0.1258 - mse:
0.0705 - mae: 0.1258 - val_loss: 0.1370 - val_mse: 0.0802 - val_mae: 0.1370
Epoch 24/400
250/250 [==============================] - 1s 3ms/step - loss: 0.1237 - mse:
0.0704 - mae: 0.1237 - val_loss: 0.1253 - val_mse: 0.0775 - val_mae: 0.1253
Epoch 25/400
250/250 [==============================] - 1s 4ms/step - loss: 0.1216 - mse:
0.0686 - mae: 0.1216 - val_loss: 0.1236 - val_mse: 0.0742 - val_mae: 0.1236
Epoch 26/400
250/250 [==============================] - 1s 4ms/step - loss: 0.1204 - mse:
0.0679 - mae: 0.1204 - val_loss: 0.1297 - val_mse: 0.0775 - val_mae: 0.1297
Epoch 27/400
```

```
250/250 [==============================] - 1s 4ms/step - loss: 0.1208 - mse:
0.0690 - mae: 0.1208 - val_loss: 0.1287 - val_mse: 0.0790 - val_mae: 0.1287
Epoch 28/400
250/250 [==============================] - 1s 4ms/step - loss: 0.1179 - mse:
0.0672 - mae: 0.1179 - val_loss: 0.1310 - val_mse: 0.0806 - val_mae: 0.1310
Epoch 29/400
250/250 [==============================] - 1s 4ms/step - loss: 0.1185 - mse:
0.0691 - mae: 0.1185 - val_loss: 0.1242 - val_mse: 0.0761 - val_mae: 0.1242
Epoch 30/400
250/250 [==============================] - 1s 4ms/step - loss: 0.1179 - mse:
0.0665 - mae: 0.1179 - val_loss: 0.1277 - val_mse: 0.0766 - val_mae: 0.1277
Epoch 31/400
250/250 [==============================] - 1s 4ms/step - loss: 0.1141 - mse:
0.0654 - mae: 0.1141 - val_loss: 0.1213 - val_mse: 0.0737 - val_mae: 0.1213
Epoch 32/400
250/250 [==============================] - 1s 4ms/step - loss: 0.1146 - mse:
0.0658 - mae: 0.1146 - val_loss: 0.1270 - val_mse: 0.0751 - val_mae: 0.1270
Epoch 33/400
250/250 [==============================] - 1s 4ms/step - loss: 0.1175 - mse:
0.0687 - mae: 0.1175 - val_loss: 0.1277 - val_mse: 0.0745 - val_mae: 0.1277
Epoch 34/400
250/250 [==============================] - 1s 4ms/step - loss: 0.1139 - mse:
0.0643 - mae: 0.1139 - val_loss: 0.1255 - val_mse: 0.0773 - val_mae: 0.1255
Epoch 35/400
250/250 [==============================] - 1s 4ms/step - loss: 0.1152 - mse:
0.0661 - mae: 0.1152 - val_loss: 0.1269 - val_mse: 0.0786 - val_mae: 0.1269
Epoch 36/400
250/250 [==============================] - 1s 4ms/step - loss: 0.1126 - mse:
0.0642 - mae: 0.1126 - val_loss: 0.1251 - val_mse: 0.0765 - val_mae: 0.1251
Epoch 37/400
250/250 [==============================] - 1s 4ms/step - loss: 0.1151 - mse:
0.0668 - mae: 0.1151 - val_loss: 0.1251 - val_mse: 0.0778 - val_mae: 0.1251
Epoch 38/400
250/250 [==============================] - 1s 3ms/step - loss: 0.1130 - mse:
0.0673 - mae: 0.1130 - val_loss: 0.1299 - val_mse: 0.0785 - val_mae: 0.1299
Epoch 39/400
250/250 [==============================] - 1s 4ms/step - loss: 0.1103 - mse:
0.0640 - mae: 0.1103 - val_loss: 0.1234 - val_mse: 0.0722 - val_mae: 0.1234
Epoch 40/400
250/250 [==============================] - 1s 4ms/step - loss: 0.1092 - mse:
0.0639 - mae: 0.1092 - val_loss: 0.1237 - val_mse: 0.0752 - val_mae: 0.1237
Epoch 41/400
250/250 [==============================] - 1s 4ms/step - loss: 0.1125 - mse:
0.0650 - mae: 0.1125 - val_loss: 0.1193 - val_mse: 0.0739 - val_mae: 0.1193
Epoch 42/400
250/250 [==============================] - 1s 4ms/step - loss: 0.1100 - mse:
0.0631 - mae: 0.1100 - val_loss: 0.1247 - val_mse: 0.0769 - val_mae: 0.1247
Epoch 43/400
```

```
250/250 [==============================] - 1s 4ms/step - loss: 0.1085 - mse:
0.0630 - mae: 0.1085 - val_loss: 0.1205 - val_mse: 0.0709 - val_mae: 0.1205
Epoch 44/400
250/250 [==============================] - 1s 4ms/step - loss: 0.1046 - mse:
0.0602 - mae: 0.1046 - val_loss: 0.1156 - val_mse: 0.0685 - val_mae: 0.1156
Epoch 45/400
250/250 [==============================] - 1s 4ms/step - loss: 0.1082 - mse:
0.0634 - mae: 0.1082 - val_loss: 0.1174 - val_mse: 0.0729 - val_mae: 0.1174
Epoch 46/400
250/250 [==============================] - 1s 4ms/step - loss: 0.1043 - mse:
0.0612 - mae: 0.1043 - val_loss: 0.1219 - val_mse: 0.0708 - val_mae: 0.1219
Epoch 47/400
250/250 [==============================] - 1s 4ms/step - loss: 0.1030 - mse:
0.0603 - mae: 0.1030 - val_loss: 0.1145 - val_mse: 0.0717 - val_mae: 0.1145
Epoch 48/400
250/250 [==============================] - 1s 4ms/step - loss: 0.1056 - mse:
0.0628 - mae: 0.1056 - val_loss: 0.1186 - val_mse: 0.0723 - val_mae: 0.1186
Epoch 49/400
250/250 [==============================] - 1s 4ms/step - loss: 0.1007 - mse:
0.0597 - mae: 0.1007 - val_loss: 0.1206 - val_mse: 0.0771 - val_mae: 0.1206
Epoch 50/400
250/250 [==============================] - 1s 4ms/step - loss: 0.0977 - mse:
0.0581 - mae: 0.0977 - val_loss: 0.1225 - val_mse: 0.0750 - val_mae: 0.1225
Epoch 51/400
250/250 [==============================] - 1s 4ms/step - loss: 0.1028 - mse:
0.0615 - mae: 0.1028 - val_loss: 0.1213 - val_mse: 0.0776 - val_mae: 0.1213
Epoch 52/400
250/250 [==============================] - 1s 3ms/step - loss: 0.1012 - mse:
0.0621 - mae: 0.1012 - val_loss: 0.1198 - val_mse: 0.0765 - val_mae: 0.1198
Epoch 53/400
250/250 [==============================] - 1s 3ms/step - loss: 0.1007 - mse:
0.0618 - mae: 0.1007 - val_loss: 0.1156 - val_mse: 0.0769 - val_mae: 0.1156
Epoch 54/400
250/250 [==============================] - 1s 4ms/step - loss: 0.0972 - mse:
0.0582 - mae: 0.0972 - val_loss: 0.1184 - val_mse: 0.0715 - val_mae: 0.1184
Epoch 55/400
250/250 [==============================] - 1s 4ms/step - loss: 0.0976 - mse:
0.0597 - mae: 0.0976 - val_loss: 0.1112 - val_mse: 0.0678 - val_mae: 0.1112
Epoch 56/400
250/250 [==============================] - 1s 4ms/step - loss: 0.1002 - mse:
0.0617 - mae: 0.1002 - val_loss: 0.1126 - val_mse: 0.0737 - val_mae: 0.1126
Epoch 57/400
250/250 [==============================] - 1s 4ms/step - loss: 0.0991 - mse:
0.0612 - mae: 0.0991 - val_loss: 0.1177 - val_mse: 0.0769 - val_mae: 0.1177
Epoch 58/400
250/250 [==============================] - 1s 4ms/step - loss: 0.0964 - mse:
0.0585 - mae: 0.0964 - val_loss: 0.1160 - val_mse: 0.0789 - val_mae: 0.1160
Epoch 59/400
```

```
250/250 [==============================] - 1s 4ms/step - loss: 0.0920 - mse:
0.0565 - mae: 0.0920 - val_loss: 0.1123 - val_mse: 0.0698 - val_mae: 0.1123
Epoch 60/400
250/250 [==============================] - 1s 3ms/step - loss: 0.0937 - mse:
0.0584 - mae: 0.0937 - val_loss: 0.1181 - val_mse: 0.0756 - val_mae: 0.1181
Epoch 61/400
250/250 [==============================] - 1s 4ms/step - loss: 0.0941 - mse:
0.0586 - mae: 0.0941 - val_loss: 0.1135 - val_mse: 0.0753 - val_mae: 0.1135
Epoch 62/400
250/250 [==============================] - 1s 4ms/step - loss: 0.0917 - mse:
0.0569 - mae: 0.0917 - val_loss: 0.1198 - val_mse: 0.0756 - val_mae: 0.1198
Epoch 63/400
250/250 [==============================] - 1s 4ms/step - loss: 0.0928 - mse:
0.0591 - mae: 0.0928 - val_loss: 0.1184 - val_mse: 0.0814 - val_mae: 0.1184
Epoch 64/400
250/250 [==============================] - 1s 4ms/step - loss: 0.0935 - mse:
0.0598 - mae: 0.0935 - val_loss: 0.1099 - val_mse: 0.0718 - val_mae: 0.1099
Epoch 65/400
250/250 [==============================] - 1s 4ms/step - loss: 0.0915 - mse:
0.0590 - mae: 0.0915 - val_loss: 0.1180 - val_mse: 0.0769 - val_mae: 0.1180
Epoch 66/400
250/250 [==============================] - 1s 4ms/step - loss: 0.0913 - mse:
0.0590 - mae: 0.0913 - val_loss: 0.1117 - val_mse: 0.0690 - val_mae: 0.1117
Epoch 67/400
250/250 [==============================] - 1s 4ms/step - loss: 0.0887 - mse:
0.0570 - mae: 0.0887 - val_loss: 0.1155 - val_mse: 0.0776 - val_mae: 0.1155
Epoch 68/400
250/250 [==============================] - 1s 4ms/step - loss: 0.0880 - mse:
0.0556 - mae: 0.0880 - val_loss: 0.1110 - val_mse: 0.0791 - val_mae: 0.1110
Epoch 69/400
250/250 [==============================] - 1s 4ms/step - loss: 0.0891 - mse:
0.0573 - mae: 0.0891 - val_loss: 0.1116 - val_mse: 0.0725 - val_mae: 0.1116
Epoch 70/400
250/250 [==============================] - 1s 4ms/step - loss: 0.0874 - mse:
0.0574 - mae: 0.0874 - val_loss: 0.1141 - val_mse: 0.0804 - val_mae: 0.1141
Epoch 71/400
250/250 [==============================] - 1s 4ms/step - loss: 0.0843 - mse:
0.0551 - mae: 0.0843 - val_loss: 0.1086 - val_mse: 0.0690 - val_mae: 0.1086
Epoch 72/400
250/250 [==============================] - 1s 4ms/step - loss: 0.0820 - mse:
0.0523 - mae: 0.0820 - val_loss: 0.1147 - val_mse: 0.0789 - val_mae: 0.1147
Epoch 73/400
250/250 [==============================] - 1s 4ms/step - loss: 0.0867 - mse:
0.0561 - mae: 0.0867 - val_loss: 0.1135 - val_mse: 0.0754 - val_mae: 0.1135
Epoch 74/400
250/250 [==============================] - 1s 4ms/step - loss: 0.0867 - mse:
0.0558 - mae: 0.0867 - val_loss: 0.1094 - val_mse: 0.0815 - val_mae: 0.1094
Epoch 75/400
```

```
250/250 [==============================] - 1s 4ms/step - loss: 0.0856 - mse:
0.0582 - mae: 0.0856 - val_loss: 0.1058 - val_mse: 0.0662 - val_mae: 0.1058
Epoch 76/400
250/250 [==============================] - 1s 4ms/step - loss: 0.0838 - mse:
0.0550 - mae: 0.0838 - val_loss: 0.1146 - val_mse: 0.0811 - val_mae: 0.1146
Epoch 77/400
250/250 [==============================] - 1s 4ms/step - loss: 0.0834 - mse:
0.0554 - mae: 0.0834 - val_loss: 0.1104 - val_mse: 0.0740 - val_mae: 0.1104
Epoch 78/400
250/250 [==============================] - 1s 4ms/step - loss: 0.0842 - mse:
0.0555 - mae: 0.0842 - val_loss: 0.1052 - val_mse: 0.0670 - val_mae: 0.1052
Epoch 79/400
250/250 [==============================] - 1s 4ms/step - loss: 0.0838 - mse:
0.0560 - mae: 0.0838 - val_loss: 0.1091 - val_mse: 0.0739 - val_mae: 0.1091
Epoch 80/400
250/250 [==============================] - 1s 3ms/step - loss: 0.0777 - mse:
0.0516 - mae: 0.0777 - val_loss: 0.1075 - val_mse: 0.0783 - val_mae: 0.1075
Epoch 81/400
250/250 [==============================] - 1s 4ms/step - loss: 0.0861 - mse:
0.0585 - mae: 0.0861 - val_loss: 0.1055 - val_mse: 0.0712 - val_mae: 0.1055
Epoch 82/400
250/250 [==============================] - 1s 3ms/step - loss: 0.0856 - mse:
0.0585 - mae: 0.0856 - val_loss: 0.1076 - val_mse: 0.0693 - val_mae: 0.1076
Epoch 83/400
250/250 [==============================] - 1s 4ms/step - loss: 0.0845 - mse:
0.0568 - mae: 0.0845 - val_loss: 0.1148 - val_mse: 0.0891 - val_mae: 0.1148
Epoch 84/400
250/250 [==============================] - 1s 4ms/step - loss: 0.0795 - mse:
0.0547 - mae: 0.0795 - val_loss: 0.1049 - val_mse: 0.0747 - val_mae: 0.1049
Epoch 85/400
250/250 [==============================] - 1s 3ms/step - loss: 0.0784 - mse:
0.0525 - mae: 0.0784 - val_loss: 0.1132 - val_mse: 0.0827 - val_mae: 0.1132
Epoch 86/400
250/250 [==============================] - 1s 4ms/step - loss: 0.0803 - mse:
0.0539 - mae: 0.0803 - val_loss: 0.1086 - val_mse: 0.0708 - val_mae: 0.1086
Epoch 87/400
250/250 [==============================] - 1s 4ms/step - loss: 0.0805 - mse:
0.0530 - mae: 0.0805 - val_loss: 0.1112 - val_mse: 0.0842 - val_mae: 0.1112
Epoch 88/400
250/250 [==============================] - 1s 4ms/step - loss: 0.0798 - mse:
0.0538 - mae: 0.0798 - val_loss: 0.1130 - val_mse: 0.0820 - val_mae: 0.1130
Epoch 89/400
250/250 [==============================] - 1s 4ms/step - loss: 0.0829 - mse:
0.0566 - mae: 0.0829 - val_loss: 0.1111 - val_mse: 0.0728 - val_mae: 0.1111
Epoch 90/400
250/250 [==============================] - 1s 4ms/step - loss: 0.0825 - mse:
0.0552 - mae: 0.0825 - val_loss: 0.1067 - val_mse: 0.0819 - val_mae: 0.1067
Epoch 91/400
```

```
250/250 [==============================] - 1s 4ms/step - loss: 0.0794 - mse:
0.0550 - mae: 0.0794 - val_loss: 0.1056 - val_mse: 0.0794 - val_mae: 0.1056
Epoch 92/400
250/250 [==============================] - 1s 4ms/step - loss: 0.0773 - mse:
0.0526 - mae: 0.0773 - val_loss: 0.1151 - val_mse: 0.0845 - val_mae: 0.1151
Epoch 93/400
250/250 [==============================] - 1s 4ms/step - loss: 0.0762 - mse:
0.0524 - mae: 0.0762 - val_loss: 0.1047 - val_mse: 0.0711 - val_mae: 0.1047
Epoch 94/400
250/250 [==============================] - 1s 4ms/step - loss: 0.0734 - mse:
0.0500 - mae: 0.0734 - val_loss: 0.1018 - val_mse: 0.0651 - val_mae: 0.1018
Epoch 95/400
250/250 [==============================] - 1s 4ms/step - loss: 0.0785 - mse:
0.0538 - mae: 0.0785 - val_loss: 0.1071 - val_mse: 0.0764 - val_mae: 0.1071
Epoch 96/400
250/250 [==============================] - 1s 4ms/step - loss: 0.0788 - mse:
0.0528 - mae: 0.0788 - val_loss: 0.1126 - val_mse: 0.0870 - val_mae: 0.1126
Epoch 97/400
250/250 [==============================] - 1s 3ms/step - loss: 0.0797 - mse:
0.0554 - mae: 0.0797 - val_loss: 0.1088 - val_mse: 0.0769 - val_mae: 0.1088
Epoch 98/400
250/250 [==============================] - 1s 4ms/step - loss: 0.0761 - mse:
0.0506 - mae: 0.0761 - val_loss: 0.1052 - val_mse: 0.0786 - val_mae: 0.1052
Epoch 99/400
250/250 [==============================] - 1s 4ms/step - loss: 0.0762 - mse:
0.0539 - mae: 0.0762 - val_loss: 0.1117 - val_mse: 0.0703 - val_mae: 0.1117
Epoch 100/400
250/250 [==============================] - 1s 4ms/step - loss: 0.0747 - mse:
0.0490 - mae: 0.0747 - val_loss: 0.1076 - val_mse: 0.0696 - val_mae: 0.1076
Epoch 101/400
250/250 [==============================] - 1s 4ms/step - loss: 0.0793 - mse:
0.0552 - mae: 0.0793 - val_loss: 0.1317 - val_mse: 0.1118 - val_mae: 0.1317
Epoch 102/400
250/250 [==============================] - 1s 4ms/step - loss: 0.0796 - mse:
0.0540 - mae: 0.0796 - val_loss: 0.1029 - val_mse: 0.0766 - val_mae: 0.1029
Epoch 103/400
250/250 [==============================] - 1s 4ms/step - loss: 0.0737 - mse:
0.0515 - mae: 0.0737 - val_loss: 0.1075 - val_mse: 0.0731 - val_mae: 0.1075
Epoch 104/400
250/250 [==============================] - ETA: 0s - loss: 0.0742 - mse: 0.0507
- mae: 0.0742
Epoch 00104: ReduceLROnPlateau reducing learning rate to 0.00020000000949949026.
250/250 [==============================] - 1s 4ms/step - loss: 0.0742 - mse:
0.0507 - mae: 0.0742 - val_loss: 0.1054 - val_mse: 0.0741 - val_mae: 0.1054
Epoch 105/400
250/250 [==============================] - 1s 4ms/step - loss: 0.0610 - mse:
0.0440 - mae: 0.0610 - val_loss: 0.0983 - val_mse: 0.0733 - val_mae: 0.0983
Epoch 106/400
```

```
250/250 [==============================] - 1s 4ms/step - loss: 0.0575 - mse:
0.0435 - mae: 0.0575 - val_loss: 0.0971 - val_mse: 0.0739 - val_mae: 0.0971
Epoch 107/400
250/250 [==============================] - 1s 4ms/step - loss: 0.0562 - mse:
0.0424 - mae: 0.0562 - val_loss: 0.0977 - val_mse: 0.0753 - val_mae: 0.0977
Epoch 108/400
250/250 [==============================] - 1s 4ms/step - loss: 0.0554 - mse:
0.0419 - mae: 0.0554 - val_loss: 0.0978 - val_mse: 0.0766 - val_mae: 0.0978
Epoch 109/400
250/250 [==============================] - 1s 4ms/step - loss: 0.0549 - mse:
0.0434 - mae: 0.0549 - val_loss: 0.0979 - val_mse: 0.0724 - val_mae: 0.0979
Epoch 110/400
250/250 [==============================] - 1s 4ms/step - loss: 0.0543 - mse:
0.0418 - mae: 0.0543 - val_loss: 0.0973 - val_mse: 0.0751 - val_mae: 0.0973
Epoch 111/400
250/250 [==============================] - 1s 4ms/step - loss: 0.0539 - mse:
0.0416 - mae: 0.0539 - val_loss: 0.0978 - val_mse: 0.0762 - val_mae: 0.0978
Epoch 112/400
250/250 [==============================] - 1s 4ms/step - loss: 0.0537 - mse:
0.0419 - mae: 0.0537 - val_loss: 0.0969 - val_mse: 0.0726 - val_mae: 0.0969
Epoch 113/400
250/250 [==============================] - 1s 4ms/step - loss: 0.0533 - mse:
0.0419 - mae: 0.0533 - val_loss: 0.0972 - val_mse: 0.0707 - val_mae: 0.0972
Epoch 114/400
250/250 [==============================] - 1s 4ms/step - loss: 0.0530 - mse:
0.0410 - mae: 0.0530 - val_loss: 0.0978 - val_mse: 0.0748 - val_mae: 0.0978
Epoch 115/400
250/250 [==============================] - 1s 4ms/step - loss: 0.0529 - mse:
0.0410 - mae: 0.0529 - val_loss: 0.0965 - val_mse: 0.0674 - val_mae: 0.0965
Epoch 116/400
250/250 [==============================] - 1s 4ms/step - loss: 0.0527 - mse:
0.0400 - mae: 0.0527 - val_loss: 0.0983 - val_mse: 0.0742 - val_mae: 0.0983
Epoch 117/400
250/250 [==============================] - 1s 4ms/step - loss: 0.0525 - mse:
0.0403 - mae: 0.0525 - val_loss: 0.0980 - val_mse: 0.0795 - val_mae: 0.0980
Epoch 118/400
250/250 [==============================] - 1s 4ms/step - loss: 0.0520 - mse:
0.0406 - mae: 0.0520 - val_loss: 0.0978 - val_mse: 0.0720 - val_mae: 0.0978
Epoch 119/400
250/250 [==============================] - 1s 4ms/step - loss: 0.0516 - mse:
0.0398 - mae: 0.0516 - val_loss: 0.0973 - val_mse: 0.0712 - val_mae: 0.0973
Epoch 120/400
250/250 [==============================] - 1s 4ms/step - loss: 0.0513 - mse:
0.0393 - mae: 0.0513 - val_loss: 0.0968 - val_mse: 0.0695 - val_mae: 0.0968
Epoch 121/400
250/250 [==============================] - 1s 4ms/step - loss: 0.0513 - mse:
0.0397 - mae: 0.0513 - val_loss: 0.0983 - val_mse: 0.0739 - val_mae: 0.0983
Epoch 122/400
```

```
250/250 [==============================] - 1s 4ms/step - loss: 0.0510 - mse:
0.0391 - mae: 0.0510 - val_loss: 0.0981 - val_mse: 0.0736 - val_mae: 0.0981
Epoch 123/400
250/250 [==============================] - 1s 4ms/step - loss: 0.0512 - mse:
0.0388 - mae: 0.0512 - val_loss: 0.0995 - val_mse: 0.0769 - val_mae: 0.0995
Epoch 124/400
250/250 [==============================] - 1s 4ms/step - loss: 0.0502 - mse:
0.0376 - mae: 0.0502 - val_loss: 0.0985 - val_mse: 0.0743 - val_mae: 0.0985
Epoch 125/400
233/250 [==========================>…] - ETA: 0s - loss: 0.0500 - mse: 0.0370
- mae: 0.0500
Epoch 00125: ReduceLROnPlateau reducing learning rate to 4.0000001899898055e-05.
250/250 [==============================] - 1s 4ms/step - loss: 0.0502 - mse:
0.0375 - mae: 0.0502 - val_loss: 0.0974 - val_mse: 0.0697 - val_mae: 0.0974
Epoch 126/400
250/250 [==============================] - 1s 4ms/step - loss: 0.0477 - mse:
0.0359 - mae: 0.0477 - val_loss: 0.0974 - val_mse: 0.0726 - val_mae: 0.0974
Epoch 127/400
250/250 [==============================] - 1s 3ms/step - loss: 0.0470 - mse:
0.0362 - mae: 0.0470 - val_loss: 0.0975 - val_mse: 0.0725 - val_mae: 0.0975
Epoch 128/400
250/250 [==============================] - 1s 4ms/step - loss: 0.0467 - mse:
0.0363 - mae: 0.0467 - val_loss: 0.0974 - val_mse: 0.0723 - val_mae: 0.0974
Epoch 129/400
250/250 [==============================] - 1s 4ms/step - loss: 0.0465 - mse:
0.0359 - mae: 0.0465 - val_loss: 0.0975 - val_mse: 0.0728 - val_mae: 0.0975
Epoch 130/400
250/250 [==============================] - 1s 4ms/step - loss: 0.0463 - mse:
0.0358 - mae: 0.0463 - val_loss: 0.0977 - val_mse: 0.0736 - val_mae: 0.0977
Epoch 131/400
250/250 [==============================] - 1s 4ms/step - loss: 0.0461 - mse:
0.0360 - mae: 0.0461 - val_loss: 0.0973 - val_mse: 0.0704 - val_mae: 0.0973
Epoch 132/400
250/250 [==============================] - 1s 4ms/step - loss: 0.0460 - mse:
0.0355 - mae: 0.0460 - val_loss: 0.0976 - val_mse: 0.0718 - val_mae: 0.0976
Epoch 133/400
250/250 [==============================] - 1s 4ms/step - loss: 0.0458 - mse:
0.0352 - mae: 0.0458 - val_loss: 0.0975 - val_mse: 0.0707 - val_mae: 0.0975
Epoch 134/400
250/250 [==============================] - 1s 4ms/step - loss: 0.0458 - mse:
0.0354 - mae: 0.0458 - val_loss: 0.0978 - val_mse: 0.0722 - val_mae: 0.0978
Epoch 135/400
246/250 [============================>.] - ETA: 0s - loss: 0.0458 - mse: 0.0353
- mae: 0.0458
Epoch 00135: ReduceLROnPlateau reducing learning rate to 8.000000525498762e-06.
250/250 [==============================] - 1s 4ms/step - loss: 0.0457 - mse:
0.0350 - mae: 0.0457 - val_loss: 0.0979 - val_mse: 0.0741 - val_mae: 0.0979
Epoch 136/400
```

```
250/250 [==============================] - 1s 4ms/step - loss: 0.0452 - mse:
0.0353 - mae: 0.0452 - val_loss: 0.0976 - val_mse: 0.0725 - val_mae: 0.0976
Epoch 137/400
250/250 [==============================] - 1s 4ms/step - loss: 0.0450 - mse:
0.0351 - mae: 0.0450 - val_loss: 0.0976 - val_mse: 0.0722 - val_mae: 0.0976
Epoch 138/400
250/250 [==============================] - 1s 4ms/step - loss: 0.0449 - mse:
0.0349 - mae: 0.0449 - val_loss: 0.0976 - val_mse: 0.0726 - val_mae: 0.0976
Epoch 139/400
250/250 [==============================] - 1s 4ms/step - loss: 0.0449 - mse:
0.0350 - mae: 0.0449 - val_loss: 0.0976 - val_mse: 0.0725 - val_mae: 0.0976
Epoch 140/400
250/250 [==============================] - 1s 4ms/step - loss: 0.0448 - mse:
0.0349 - mae: 0.0448 - val_loss: 0.0975 - val_mse: 0.0721 - val_mae: 0.0975
Epoch 141/400
250/250 [==============================] - 1s 4ms/step - loss: 0.0448 - mse:
0.0349 - mae: 0.0448 - val_loss: 0.0976 - val_mse: 0.0721 - val_mae: 0.0976
Epoch 142/400
250/250 [==============================] - 1s 4ms/step - loss: 0.0448 - mse:
0.0348 - mae: 0.0448 - val_loss: 0.0976 - val_mse: 0.0724 - val_mae: 0.0976
Epoch 143/400
250/250 [==============================] - 1s 4ms/step - loss: 0.0448 - mse:
0.0347 - mae: 0.0448 - val_loss: 0.0976 - val_mse: 0.0723 - val_mae: 0.0976
Epoch 144/400
250/250 [==============================] - 1s 4ms/step - loss: 0.0447 - mse:
0.0348 - mae: 0.0447 - val_loss: 0.0976 - val_mse: 0.0719 - val_mae: 0.0976
Epoch 145/400
234/250 [===========================>..] - ETA: 0s - loss: 0.0450 - mse: 0.0349
- mae: 0.0450
Epoch 00145: ReduceLROnPlateau reducing learning rate to 1.6000001778593287e-06.
250/250 [==============================] - 1s 4ms/step - loss: 0.0447 - mse:
0.0347 - mae: 0.0447 - val_loss: 0.0976 - val_mse: 0.0724 - val_mae: 0.0976
Epoch 146/400
247/250 [============================>.] - ETA: 0s - loss: 0.0444 - mse: 0.0346
- mae: 0.0444Restoring model weights from the end of the best epoch.
250/250 [==============================] - 1s 4ms/step - loss: 0.0446 - mse:
0.0347 - mae: 0.0446 - val_loss: 0.0976 - val_mse: 0.0723 - val_mae: 0.0976
Epoch 00146: early stopping
```

3 Testing accuracy of Model with validation data

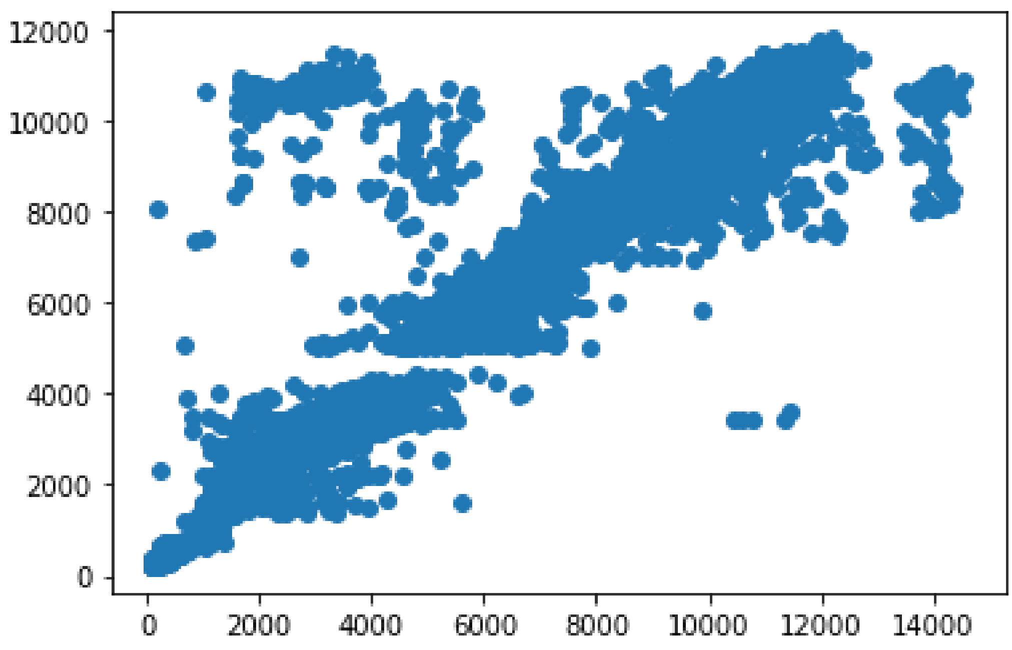
[62]:

```
y_f_result = model.predict(x_f_test)
y_result = y_scaler.inverse_transform(y_f_result)
y_actual = y_scaler.inverse_transform(y_f_test)

compare = pd.DataFrame()
compare['Expected'] = y_actual.reshape(1,-1)[0]
```

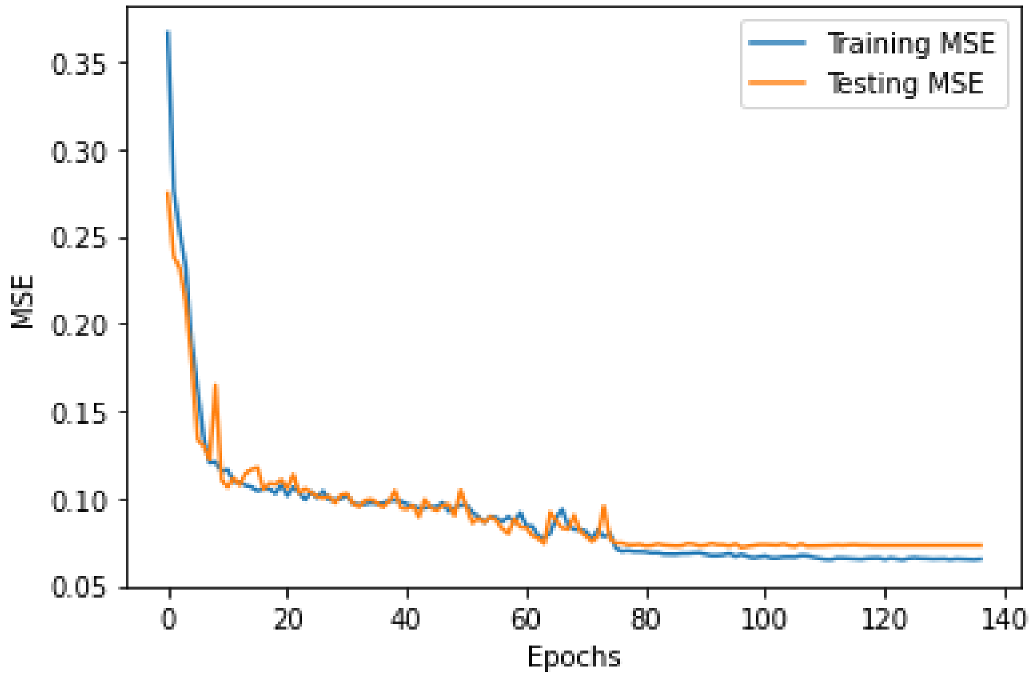
```
compare['Result'] = y_result.reshape(1,-1)[0]
compare['Difference'] = compare['Expected'] - compare['Result']
compare['Percentage Error'] = 100 * compare['Difference']/compare['Expected']

#Print out percentile descriptions of model accuracy
compare['Percentage Error'].describe(percentiles=[0.001, 0.01, 0.05, 0.25, 0.5,␣
↪0.75, 0.95, 0.99, 0.999])
```

[62]:
```
count     7498.000000
mean       -10.916233
std         67.227989
min      -3923.573845
0.1%      -545.837426
1%        -238.477236
5%         -73.881641
25%         -9.409581
50%         -0.973565
75%          5.432740
95%         21.019523
99%         35.267476
99.9%       60.016162
max         71.264463
Name: Percentage Error, dtype: float64
```

[63]:
```
plt.scatter(compare['Expected'], compare['Result'])
plt.show()
```



[59]:
```
plt.plot(history.history['mse'],label='Training MSE')
plt.plot(history.history['val_mse'],label='Testing MSE')
plt.xlabel('Epochs')
plt.ylabel('MSE')
plt.legend()
plt.show()
```



[70]:
```
model.save('Trained_Models/Building_Model' + datetime.now().
↪strftime("%Y%m%d-%H%M%S") + '.h5')
```

4 Building Model Analysis

[71]:
```
def clean_data(location, skiprows = 0):
    df_weather = pd.read_csv(location, skiprows=skiprows)
    df_weather = df_weather.drop(columns=['Hour', 'Minute'])
    df_weather = df_weather[df_weather.DNI != 0]

    #Take mean, max and min for each DNI in DataFrame and mean for everything␣
↪else

    max_dni = df_weather.groupby(['Year', 'Month', 'Day']).max().
↪reset_index()['DNI']
```

```
    min_dni = df_weather.groupby(['Year', 'Month', 'Day']).min().
↪reset_index()['DNI']

    df_weather = pd.DataFrame(df_weather.groupby(['Year', 'Month', 'Day']).
↪mean().reset_index())

    df_weather.insert(4, 'DNI Max', max_dni)
    df_weather.insert(5, 'DNI Min', min_dni)

    return df_weather
```

[72]:
```
def training_prep(data, square_feet, building_type):
    data = data.loc[:, ['Year', 'Month', 'Day', 'DNI', 'DNI Max', 'DNI Min',␣
↪'Wind Speed', 'Precipitable Water', 'Wind Direction', 'Relative Humidity',␣
↪'Temperature', 'Pressure']]
    data.loc[:, 'Weekday'] = get_weekday2(data['Year'], data['Month'],␣
↪data['Day'])
    data = data.drop(['Year'], axis=1)
    data.loc[:, 'Square Feet'] = square_feet
    data.loc[:, 'Type'] = building_type
    return data
```

[73]:
```
#Taking data for University of Michigan Research Building
michigan_data = training_prep(clean_data('https://raw.githubusercontent.com/
↪A-Wadhwani/ME597-Project/main/Datasets/AnnArbor_Weather.csv'), 100000, 2)
michigan_data.head()
```

[73]:

	Month	Day	DNI	DNI Max	…	Pressure	Weekday	Square Feet	Type
0	1	1	52.111111	178	…	993.888889	0	100000	2
1	1	2	396.000000	751	…	1011.400000	0	100000	2
2	1	3	299.700000	752	…	997.400000	0	100000	2
3	1	4	281.900000	576	…	997.300000	0	100000	2
4	1	5	10.000000	21	…	986.000000	1	100000	2

[5 rows x 14 columns]

[74]:
```
#Applying transform to data
mi_test = x_scaler.transform(michigan_data)
```

[75]:
```
#Testing model:
mi_result = y_scaler.inverse_transform(model.predict(mi_test))
compare = pd.DataFrame()
compare['Month'] = michigan_data['Month']
compare['Result'] = mi_result.reshape(1,-1)[0]
compare.head()
```

[75]:

	Month	Result
0	1	10769.524414
1	1	9488.891602
2	1	10889.340820
3	1	9792.707031
4	1	7874.032227

[76]:

```
compare = pd.DataFrame(compare.groupby(['Month']).sum().reset_index())
actual = [212259, 240083, 218423, 233777, 240106, 272017, 300123, 295701,␣
↪288447, 254802, 228097, 220258]
compare['Actual'] = actual
compare['Difference'] = compare['Actual'] - compare['Result']
compare['Percentage Error'] = 100 * compare['Difference']/compare['Actual']
compare.head(12)
```

[76]:

	Month	Result	Actual	Difference	Percentage Error
0	1	329494.250000	212259	-117235.250000	-55.232169
1	2	277154.531250	240083	-37071.531250	-15.441131
2	3	274379.218750	218423	-55956.218750	-25.618281
3	4	226625.281250	233777	7151.718750	3.059205
4	5	196259.890625	240106	43846.109375	18.261147
5	6	153253.062500	272017	118763.937500	43.660484
6	7	142121.968750	300123	158001.031250	52.645426
7	8	141248.828125	295701	154452.171875	52.232550
8	9	141370.921875	288447	147076.078125	50.988944
9	10	164804.406250	254802	89997.593750	35.320599
10	11	222006.390625	228097	6090.609375	2.670184
11	12	237003.234375	220258	-16745.234375	-7.602554

[77]:

```
compare.describe()
```

[77]:

	Month	Result	Actual	Difference	Percentage Error
count	12.000000	12.000000	12.000000	12.000000	12.000000
mean	6.500000	208810.171875	250341.083333	41530.917969	12.912034
std	3.605551	62929.816406	31378.512646	91700.318958	35.169540
min	1.000000	141248.828125	212259.000000	-117235.250000	-55.232169
25%	3.750000	150470.289062	226137.250000	-21826.808594	-9.562199
50%	6.500000	209133.140625	240094.500000	25498.914062	10.660176
75%	9.250000	246347.230469	276124.500000	125841.972656	45.492599
max	12.000000	329494.250000	300123.000000	158001.031250	52.645426

5 Comparison against UIUC data

[113]:
```
uiuc_test = pd.read_csv('https://raw.githubusercontent.com/A-Wadhwani/
↪ME597-Project/main/Datasets/UIUC_PowerWeatherData.csv')
#Adding other relevant data, including square feet and type
uiuc_test['Square Feet'] = 100000
#Laboratory
uiuc_test['Type'] = 1
uiuc_actual = uiuc_test['Power Consumption']
uiuc_test = uiuc_test.drop(['Building Name', 'University Name', 'Year','Power␣
↪Consumption'], axis=1)
uiuc_test.head()
```

[113]:

	Month	Day	DNI Mean	DNI Max	…	Pressure	Weekday	Square Feet	Type
0	10	1	597.916667	898	…	988.333333	0	100000	1
1	10	2	332.250000	772	…	986.250000	0	100000	1
2	10	3	408.000000	923	…	991.555556	0	100000	1
3	10	4	617.250000	931	…	999.416667	0	100000	1
4	10	5	309.400000	810	…	990.500000	1	100000	1

[5 rows x 14 columns]

[114]:
```
#Applying transform to data
uiuc_test = x_scaler.transform(uiuc_test)
```

[115]:
```
#Testing model:
uiuc_result = y_scaler.inverse_transform(model.predict(uiuc_test))

compare = pd.DataFrame()
compare['Actual'] = uiuc_actual
compare['Result'] = uiuc_result.reshape(1,-1)[0]
compare['Difference'] = compare['Actual'] - compare['Result']
compare['Percentage Error'] = 100 * compare['Difference']/compare['Actual']

compare.describe()
```

[115]:

	Actual	Result	Difference	Percentage Error
count	51.000000	51.000000	51.000000	51.000000
mean	6189.156863	7997.752441	-1808.595081	-29.337230
std	228.978372	682.941345	680.797346	11.350610
min	5069.000000	5545.342773	-3601.576172	-57.893846
25%	6142.000000	7724.354980	-2041.887695	-33.842293
50%	6243.000000	7895.011230	-1771.775879	-28.423119
75%	6301.000000	8197.769531	-1541.850098	-24.538141
max	6518.000000	9866.675781	506.657227	8.371732

[116]:
```
plt.hist(compare['Percentage Error'])
plt.show()
```