Neural Shrubs - Leaves

July 18, 2019

```
In [1]: import time

        from sklearn.tree import DecisionTreeRegressor

        from keras.models import Sequential
        from keras.layers import Dense

        from sklearn.preprocessing import  MinMaxScaler
        from sklearn import preprocessing

        from sklearn.metrics import mean_absolute_error
        from sklearn.metrics import mean_squared_error

        import numpy as np
        import csv
        import os
```

```
/home/shashwati/anaconda3/envs/py35/lib/python3.5/site-packages/h5py/__init__.py:36: FutureWar
  from ._conv import register_converters as _register_converters
Using TensorFlow backend.
```

```
In [2]: sc= MinMaxScaler()
```

```
In [3]: # function returns the data in the right format
        def get_data():
            # load training dataset
            dataset = np.genfromtxt("YearPredictionMSD.csv", dtype='float', delimiter=",")

            train_X = dataset[0:463715,1:91]
            train_Y = dataset[0:463715,0]

            test_X = dataset[463715:,1:91]
            test_Y = dataset[463715:,0]

            return train_X, train_Y, test_X, test_Y
```

```
In [4]: # builds the decision tree of depth 13
        def regression_tree(train, label):
            dt = DecisionTreeRegressor(max_depth=13, min_samples_leaf=5000)
            dt.fit(train, label)
            return dt
```

```
In [5]: # builds the neural network for a given class
        def neural_network(class_data):
            num_train = []
            num_label = []
            for x in class_data:
                num_train.append(x[0])
                num_label.append(x[1])

            num_train = np.array(num_train)
            num_label = np.array(num_label)


            # Scale the features so they have 0 mean
            num_train = preprocessing.scale(num_train)

            num_label = num_label.reshape(-1,1)
            num_label = sc.fit_transform(num_label)

            model = Sequential()
            model.add(Dense(90, input_dim=90, kernel_initializer='normal', activation='relu'))
            model.add(Dense(90, kernel_initializer='normal', activation='relu'))
            model.add(Dense(90, kernel_initializer='normal', activation='relu'))
            model.add(Dense(90, kernel_initializer='normal', activation='relu'))
            model.add(Dense(1, kernel_initializer='normal', activation='linear'))
            model.compile(loss='mean_squared_error', optimizer='adam', metrics=['mae','accurac
            model.fit(num_train, num_label, epochs=3, batch_size=32)

            return model
```

```
In [6]: # builds the neural shrub
        def neural_shrubs(tree, train, label):
            train = np.array(train)
            label = np.array(label)

            # leave_id: index of the leaf that cantains the instance
            leave_id = tree.apply(train)

            classes = dict()

            for x in range(len(train)):
                leaf = leave_id[x]

                # Gets the class for each leaf
                #.value: contains value of all the tree nodes
                #.value[leaf]: returns the value of the leaf
                #idx = tree.tree_.value[leaf][0][0]

                # insert the instance into the class
                if leaf in classes.keys():
                    classes[leaf].append([train[x], label[x]])
                else:
                    classes[leaf] = [[train[x], label[x]]]

            # stores the neural network for each class
            nn_models = dict()

            #stores the max time taken to build a neural network
            max_time = 0;

            for key in classes.keys():

                start = time.time()
                model = neural_network(classes[key])
                end = time.time()

                time_taken = end - start
                if max_time < time_taken:
                    max_time = time_taken

                nn_models[key] = model

            # returns a neural network for each class and the max
            # time taken to build the neural network
            return nn_models, max_time
```

```
In [7]: # The algorithm to build the decision tree
        train, train_label, test, test_label = get_data()

        dt_start = time.time()
        tree = regression_tree(train, train_label)
        dt_end = time.time()
        print("Decision tree made in: ", dt_end-dt_start)
```

```
Decision tree made in:  32.71749544143677
```

```
In [14]: # Neural shrub
         shrubs, max_time = neural_shrubs(tree, train, train_label)
```

```
Epoch 1/3
5023/5023 [==============================] - 5s 912us/step - loss: 0.0942 - mean_absolute_erro
Epoch 2/3
5023/5023 [==============================] - 1s 250us/step - loss: 0.0174 - mean_absolute_erro
```

```
Epoch 3/3
5023/5023 [==============================] - 1s 254us/step - loss: 0.0122 - mean_absolute_erro
Epoch 1/3
9213/9213 [==============================] - 6s 598us/step - loss: 0.0612 - mean_absolute_erro
Epoch 2/3
9213/9213 [==============================] - 2s 236us/step - loss: 0.0198 - mean_absolute_erro
Epoch 3/3
9213/9213 [==============================] - 2s 243us/step - loss: 0.0175 - mean_absolute_erro
Epoch 1/3
6667/6667 [==============================] - 5s 725us/step - loss: 0.0650 - mean_absolute_erro
Epoch 2/3
6667/6667 [==============================] - 2s 236us/step - loss: 0.0188 - mean_absolute_erro
Epoch 3/3
6667/6667 [==============================] - 2s 259us/step - loss: 0.0158 - mean_absolute_erro
Epoch 1/3
7565/7565 [==============================] - 5s 682us/step - loss: 0.0685 - mean_absolute_erro
Epoch 2/3
7565/7565 [==============================] - 2s 254us/step - loss: 0.0136 - mean_absolute_erro
Epoch 3/3
7565/7565 [==============================] - 2s 259us/step - loss: 0.0109 - mean_absolute_erro
Epoch 1/3
5001/5001 [==============================] - 5s 906us/step - loss: 0.0836 - mean_absolute_erro
Epoch 2/3
5001/5001 [==============================] - 1s 252us/step - loss: 0.0209 - mean_absolute_erro
Epoch 3/3
5001/5001 [==============================] - 1s 253us/step - loss: 0.0168 - mean_absolute_erro
Epoch 1/3
5006/5006 [==============================] - 4s 886us/step - loss: 0.0843 - mean_absolute_erro
Epoch 2/3
5006/5006 [==============================] - 1s 257us/step - loss: 0.0287 - mean_absolute_erro
Epoch 3/3
5006/5006 [==============================] - 1s 255us/step - loss: 0.0244 - mean_absolute_erro
Epoch 1/3
6770/6770 [==============================] - 5s 727us/step - loss: 0.0764 - mean_absolute_erro
Epoch 2/3
6770/6770 [==============================] - 2s 254us/step - loss: 0.0179 - mean_absolute_erro
Epoch 3/3
6770/6770 [==============================] - 2s 270us/step - loss: 0.0144 - mean_absolute_erro
Epoch 1/3
5493/5493 [==============================] - 5s 842us/step - loss: 0.0709 - mean_absolute_erro
Epoch 2/3
5493/5493 [==============================] - 1s 272us/step - loss: 0.0135 - mean_absolute_erro
Epoch 3/3
5493/5493 [==============================] - 1s 261us/step - loss: 0.0104 - mean_absolute_erro
Epoch 1/3
9233/9233 [==============================] - 6s 617us/step - loss: 0.0464 - mean_absolute_erro
Epoch 2/3
9233/9233 [==============================] - 2s 261us/step - loss: 0.0156 - mean_absolute_erro
```

```
Epoch 3/3
9233/9233 [==============================] - 2s 263us/step - loss: 0.0144 - mean_absolute_erro
Epoch 1/3
5223/5223 [==============================] - 5s 882us/step - loss: 0.0888 - mean_absolute_erro
Epoch 2/3
5223/5223 [==============================] - 1s 265us/step - loss: 0.0229 - mean_absolute_erro
Epoch 3/3
5223/5223 [==============================] - 1s 271us/step - loss: 0.0178 - mean_absolute_erro
Epoch 1/3
5082/5082 [==============================] - 5s 901us/step - loss: 0.0899 - mean_absolute_erro
Epoch 2/3
5082/5082 [==============================] - 1s 269us/step - loss: 0.0180 - mean_absolute_erro
Epoch 3/3
5082/5082 [==============================] - 1s 293us/step - loss: 0.0123 - mean_absolute_erro
Epoch 1/3
7866/7866 [==============================] - 5s 693us/step - loss: 0.0637 - mean_absolute_erro
Epoch 2/3
7866/7866 [==============================] - 2s 277us/step - loss: 0.0213 - mean_absolute_erro
Epoch 3/3
7866/7866 [==============================] - 2s 268us/step - loss: 0.0192 - mean_absolute_erro
Epoch 1/3
5132/5132 [==============================] - 5s 910us/step - loss: 0.0831 - mean_absolute_erro
Epoch 2/3
5132/5132 [==============================] - 1s 260us/step - loss: 0.0155 - mean_absolute_erro
Epoch 3/3
5132/5132 [==============================] - 1s 278us/step - loss: 0.0114 - mean_absolute_erro
Epoch 1/3
6828/6828 [==============================] - 5s 764us/step - loss: 0.0627 - mean_absolute_erro
Epoch 2/3
6828/6828 [==============================] - 2s 275us/step - loss: 0.0224 - mean_absolute_erro
Epoch 3/3
6828/6828 [==============================] - 2s 276us/step - loss: 0.0195 - mean_absolute_erro
Epoch 1/3
8527/8527 [==============================] - 6s 663us/step - loss: 0.0518 - mean_absolute_erro
Epoch 2/3
8527/8527 [==============================] - 2s 277us/step - loss: 0.0090 - mean_absolute_erro
Epoch 3/3
8527/8527 [==============================] - 2s 285us/step - loss: 0.0072 - mean_absolute_erro
Epoch 1/3
7779/7779 [==============================] - 6s 714us/step - loss: 0.0635 - mean_absolute_erro
Epoch 2/3
7779/7779 [==============================] - 2s 294us/step - loss: 0.0169 - mean_absolute_erro
Epoch 3/3
7779/7779 [==============================] - 2s 281us/step - loss: 0.0144 - mean_absolute_erro
Epoch 1/3
6502/6502 [==============================] - 5s 781us/step - loss: 0.0744 - mean_absolute_erro
Epoch 2/3
6502/6502 [==============================] - 2s 292us/step - loss: 0.0236 - mean_absolute_erro
```

```
Epoch 3/3
6502/6502 [==============================] - 2s 297us/step - loss: 0.0207 - mean_absolute_erro
Epoch 1/3
7886/7886 [==============================] - 6s 704us/step - loss: 0.0708 - mean_absolute_erro
Epoch 2/3
7886/7886 [==============================] - 2s 287us/step - loss: 0.0146 - mean_absolute_erro
Epoch 3/3
7886/7886 [==============================] - 2s 276us/step - loss: 0.0120 - mean_absolute_erro
Epoch 1/3
9248/9248 [==============================] - 6s 628us/step - loss: 0.0665 - mean_absolute_erro
Epoch 2/3
9248/9248 [==============================] - 3s 274us/step - loss: 0.0169 - mean_absolute_erro
Epoch 3/3
9248/9248 [==============================] - 3s 279us/step - loss: 0.0145 - mean_absolute_erro
Epoch 1/3
9210/9210 [==============================] - 6s 623us/step - loss: 0.0502 - mean_absolute_erro
Epoch 2/3
9210/9210 [==============================] - 2s 267us/step - loss: 0.0118 - mean_absolute_erro
Epoch 3/3
9210/9210 [==============================] - 3s 272us/step - loss: 0.0104 - mean_absolute_erro
Epoch 1/3
9910/9910 [==============================] - 6s 613us/step - loss: 0.0555 - mean_absolute_erro
Epoch 2/3
9910/9910 [==============================] - 3s 280us/step - loss: 0.0126 - mean_absolute_erro
Epoch 3/3
9910/9910 [==============================] - 3s 280us/step - loss: 0.0115 - mean_absolute_erro
Epoch 1/3
7138/7138 [==============================] - 5s 761us/step - loss: 0.0670 - mean_absolute_erro
Epoch 2/3
7138/7138 [==============================] - 2s 278us/step - loss: 0.0173 - mean_absolute_erro
Epoch 3/3
7138/7138 [==============================] - 2s 261us/step - loss: 0.0144 - mean_absolute_erro
Epoch 1/3
8951/8951 [==============================] - 6s 638us/step - loss: 0.0596 - mean_absolute_erro
Epoch 2/3
8951/8951 [==============================] - 2s 264us/step - loss: 0.0094 - mean_absolute_erro
Epoch 3/3
8951/8951 [==============================] - 2s 260us/step - loss: 0.0070 - mean_absolute_erro
Epoch 1/3
6685/6685 [==============================] - 5s 769us/step - loss: 0.0710 - mean_absolute_erro
Epoch 2/3
6685/6685 [==============================] - 2s 267us/step - loss: 0.0247 - mean_absolute_erro
Epoch 3/3
6685/6685 [==============================] - 2s 271us/step - loss: 0.0219 - mean_absolute_erro
Epoch 1/3
8791/8791 [==============================] - 6s 675us/step - loss: 0.0634 - mean_absolute_erro
Epoch 2/3
8791/8791 [==============================] - 2s 269us/step - loss: 0.0282 - mean_absolute_erro
```

```
Epoch 3/3
8791/8791 [==============================] - 3s 286us/step - loss: 0.0260 - mean_absolute_erro
Epoch 1/3
5000/5000 [==============================] - 5s 967us/step - loss: 0.0918 - mean_absolute_erro
Epoch 2/3
5000/5000 [==============================] - 1s 266us/step - loss: 0.0200 - mean_absolute_erro
Epoch 3/3
5000/5000 [==============================] - 1s 277us/step - loss: 0.0162 - mean_absolute_erro
Epoch 1/3
7889/7889 [==============================] - 6s 716us/step - loss: 0.0485 - mean_absolute_erro
Epoch 2/3
7889/7889 [==============================] - 2s 280us/step - loss: 0.0132 - mean_absolute_erro
Epoch 3/3
7889/7889 [==============================] - 2s 269us/step - loss: 0.0116 - mean_absolute_erro
Epoch 1/3
5061/5061 [==============================] - 5s 949us/step - loss: 0.1028 - mean_absolute_erro
Epoch 2/3
5061/5061 [==============================] - 1s 279us/step - loss: 0.0249 - mean_absolute_erro
Epoch 3/3
5061/5061 [==============================] - 1s 281us/step - loss: 0.0202 - mean_absolute_erro
Epoch 1/3
5547/5547 [==============================] - 5s 922us/step - loss: 0.0823 - mean_absolute_erro
Epoch 2/3
5547/5547 [==============================] - 2s 279us/step - loss: 0.0276 - mean_absolute_erro
Epoch 3/3
5547/5547 [==============================] - 1s 270us/step - loss: 0.0253 - mean_absolute_erro
Epoch 1/3
6704/6704 [==============================] - 5s 801us/step - loss: 0.0615 - mean_absolute_erro
Epoch 2/3
6704/6704 [==============================] - 2s 282us/step - loss: 0.0292 - mean_absolute_erro
Epoch 3/3
6704/6704 [==============================] - 2s 276us/step - loss: 0.0272 - mean_absolute_erro
Epoch 1/3
5691/5691 [==============================] - 5s 923us/step - loss: 0.0731 - mean_absolute_erro
Epoch 2/3
5691/5691 [==============================] - 2s 294us/step - loss: 0.0254 - mean_absolute_erro
Epoch 3/3
5691/5691 [==============================] - 2s 296us/step - loss: 0.0213 - mean_absolute_erro
Epoch 1/3
6611/6611 [==============================] - 6s 924us/step - loss: 0.0623 - mean_absolute_erro
Epoch 2/3
6611/6611 [==============================] - 2s 284us/step - loss: 0.0291 - mean_absolute_erro
Epoch 3/3
6611/6611 [==============================] - 2s 280us/step - loss: 0.0274 - mean_absolute_erro
Epoch 1/3
5052/5052 [==============================] - 5s 1ms/step - loss: 0.0711 - mean_absolute_error:
Epoch 2/3
5052/5052 [==============================] - 1s 290us/step - loss: 0.0272 - mean_absolute_erro
```

```
Epoch 3/3
5052/5052 [==============================] - 2s 300us/step - loss: 0.0240 - mean_absolute_error
Epoch 1/3
5033/5033 [==============================] - 5s 1ms/step - loss: 0.0871 - mean_absolute_error:
Epoch 2/3
5033/5033 [==============================] - 1s 291us/step - loss: 0.0308 - mean_absolute_error
Epoch 3/3
5033/5033 [==============================] - 2s 300us/step - loss: 0.0270 - mean_absolute_error
Epoch 1/3
5117/5117 [==============================] - 5s 1ms/step - loss: 0.0820 - mean_absolute_error:
Epoch 2/3
5117/5117 [==============================] - 2s 311us/step - loss: 0.0350 - mean_absolute_error
Epoch 3/3
5117/5117 [==============================] - 2s 295us/step - loss: 0.0305 - mean_absolute_error
Epoch 1/3
5464/5464 [==============================] - 6s 1ms/step - loss: 0.0755 - mean_absolute_error:
Epoch 2/3
5464/5464 [==============================] - 2s 298us/step - loss: 0.0232 - mean_absolute_error
Epoch 3/3
5464/5464 [==============================] - 2s 302us/step - loss: 0.0195 - mean_absolute_error
Epoch 1/3
9338/9338 [==============================] - 7s 709us/step - loss: 0.0679 - mean_absolute_error
Epoch 2/3
9338/9338 [==============================] - 3s 332us/step - loss: 0.0213 - mean_absolute_error
Epoch 3/3
9338/9338 [==============================] - 3s 305us/step - loss: 0.0198 - mean_absolute_error
Epoch 1/3
6629/6629 [==============================] - 6s 877us/step - loss: 0.0815 - mean_absolute_error
Epoch 2/3
6629/6629 [==============================] - 2s 297us/step - loss: 0.0201 - mean_absolute_error
Epoch 3/3
6629/6629 [==============================] - 2s 301us/step - loss: 0.0165 - mean_absolute_error
Epoch 1/3
5020/5020 [==============================] - 5s 1ms/step - loss: 0.1031 - mean_absolute_error:
Epoch 2/3
5020/5020 [==============================] - 1s 291us/step - loss: 0.0279 - mean_absolute_error
Epoch 3/3
5020/5020 [==============================] - 2s 313us/step - loss: 0.0232 - mean_absolute_error
Epoch 1/3
5200/5200 [==============================] - 6s 1ms/step - loss: 0.0855 - mean_absolute_error:
Epoch 2/3
5200/5200 [==============================] - 2s 346us/step - loss: 0.0367 - mean_absolute_error
Epoch 3/3
5200/5200 [==============================] - 2s 311us/step - loss: 0.0339 - mean_absolute_error
Epoch 1/3
5271/5271 [==============================] - 6s 1ms/step - loss: 0.0836 - mean_absolute_error:
Epoch 2/3
5271/5271 [==============================] - 2s 351us/step - loss: 0.0312 - mean_absolute_error
```

```
Epoch 3/3
5271/5271 [==============================] - 2s 387us/step - loss: 0.0267 - mean_absolute_erro
Epoch 1/3
5410/5410 [==============================] - 6s 1ms/step - loss: 0.0839 - mean_absolute_error:
Epoch 2/3
5410/5410 [==============================] - 2s 424us/step - loss: 0.0344 - mean_absolute_erro
Epoch 3/3
5410/5410 [==============================] - 2s 302us/step - loss: 0.0300 - mean_absolute_erro
Epoch 1/3
8932/8932 [==============================] - 7s 794us/step - loss: 0.0600 - mean_absolute_erro
Epoch 2/3
8932/8932 [==============================] - 3s 302us/step - loss: 0.0235 - mean_absolute_erro
Epoch 3/3
8932/8932 [==============================] - 3s 302us/step - loss: 0.0219 - mean_absolute_erro
Epoch 1/3
5022/5022 [==============================] - 6s 1ms/step - loss: 0.1030 - mean_absolute_error:
Epoch 2/3
5022/5022 [==============================] - 1s 299us/step - loss: 0.0418 - mean_absolute_erro
Epoch 3/3
5022/5022 [==============================] - 2s 345us/step - loss: 0.0372 - mean_absolute_erro
Epoch 1/3
5434/5434 [==============================] - 6s 1ms/step - loss: 0.0789 - mean_absolute_error:
Epoch 2/3
5434/5434 [==============================] - 2s 288us/step - loss: 0.0202 - mean_absolute_erro
Epoch 3/3
5434/5434 [==============================] - 2s 292us/step - loss: 0.0157 - mean_absolute_erro
Epoch 1/3
7998/7998 [==============================] - 7s 854us/step - loss: 0.0656 - mean_absolute_erro
Epoch 2/3
7998/7998 [==============================] - 2s 287us/step - loss: 0.0264 - mean_absolute_erro
Epoch 3/3
7998/7998 [==============================] - 2s 292us/step - loss: 0.0237 - mean_absolute_erro
Epoch 1/3
7653/7653 [==============================] - 7s 936us/step - loss: 0.0589 - mean_absolute_erro
Epoch 2/3
7653/7653 [==============================] - 3s 414us/step - loss: 0.0216 - mean_absolute_erro
Epoch 3/3
7653/7653 [==============================] - 2s 326us/step - loss: 0.0193 - mean_absolute_erro
Epoch 1/3
5014/5014 [==============================] - 6s 1ms/step - loss: 0.0833 - mean_absolute_error:
Epoch 2/3
5014/5014 [==============================] - 2s 316us/step - loss: 0.0264 - mean_absolute_erro
Epoch 3/3
5014/5014 [==============================] - 2s 323us/step - loss: 0.0219 - mean_absolute_erro
Epoch 1/3
5075/5075 [==============================] - 6s 1ms/step - loss: 0.0851 - mean_absolute_error:
Epoch 2/3
5075/5075 [==============================] - 1s 280us/step - loss: 0.0202 - mean_absolute_erro
```

```
Epoch 3/3
5075/5075 [==============================] - 1s 280us/step - loss: 0.0161 - mean_absolute_error
Epoch 1/3
9234/9234 [==============================] - 7s 776us/step - loss: 0.0582 - mean_absolute_error
Epoch 2/3
9234/9234 [==============================] - 3s 320us/step - loss: 0.0281 - mean_absolute_error
Epoch 3/3
9234/9234 [==============================] - 3s 312us/step - loss: 0.0266 - mean_absolute_error
Epoch 1/3
7722/7722 [==============================] - 7s 868us/step - loss: 0.0618 - mean_absolute_error
Epoch 2/3
7722/7722 [==============================] - 2s 315us/step - loss: 0.0208 - mean_absolute_error
Epoch 3/3
7722/7722 [==============================] - 2s 305us/step - loss: 0.0184 - mean_absolute_error
Epoch 1/3
8577/8577 [==============================] - 7s 850us/step - loss: 0.0671 - mean_absolute_error
Epoch 2/3
8577/8577 [==============================] - 3s 347us/step - loss: 0.0227 - mean_absolute_error
Epoch 3/3
8577/8577 [==============================] - 3s 358us/step - loss: 0.0205 - mean_absolute_error
Epoch 1/3
8522/8522 [==============================] - 7s 862us/step - loss: 0.0660 - mean_absolute_error
Epoch 2/3
8522/8522 [==============================] - 3s 322us/step - loss: 0.0153 - mean_absolute_error
Epoch 3/3
8522/8522 [==============================] - 3s 333us/step - loss: 0.0123 - mean_absolute_error
Epoch 1/3
5339/5339 [==============================] - 6s 1ms/step - loss: 0.0800 - mean_absolute_error:
Epoch 2/3
5339/5339 [==============================] - 2s 312us/step - loss: 0.0168 - mean_absolute_error
Epoch 3/3
5339/5339 [==============================] - 2s 314us/step - loss: 0.0128 - mean_absolute_error
Epoch 1/3
6955/6955 [==============================] - 7s 959us/step - loss: 0.0715 - mean_absolute_error
Epoch 2/3
6955/6955 [==============================] - 2s 322us/step - loss: 0.0236 - mean_absolute_error
Epoch 3/3
6955/6955 [==============================] - 2s 322us/step - loss: 0.0200 - mean_absolute_error
Epoch 1/3
7481/7481 [==============================] - 7s 912us/step - loss: 0.0640 - mean_absolute_error
Epoch 2/3
7481/7481 [==============================] - 2s 319us/step - loss: 0.0141 - mean_absolute_error
Epoch 3/3
7481/7481 [==============================] - 2s 324us/step - loss: 0.0107 - mean_absolute_error
Epoch 1/3
6672/6672 [==============================] - 7s 998us/step - loss: 0.0782 - mean_absolute_error
Epoch 2/3
6672/6672 [==============================] - 2s 329us/step - loss: 0.0108 - mean_absolute_error
```

```
Epoch 3/3
6672/6672 [==============================] - 2s 324us/step - loss: 0.0066 - mean_absolute_error
Epoch 1/3
5233/5233 [==============================] - 6s 1ms/step - loss: 0.0902 - mean_absolute_error:
Epoch 2/3
5233/5233 [==============================] - 2s 340us/step - loss: 0.0124 - mean_absolute_error
Epoch 3/3
5233/5233 [==============================] - 2s 327us/step - loss: 0.0065 - mean_absolute_error
Epoch 1/3
5255/5255 [==============================] - 6s 1ms/step - loss: 0.0942 - mean_absolute_error:
Epoch 2/3
5255/5255 [==============================] - 2s 338us/step - loss: 0.0225 - mean_absolute_error
Epoch 3/3
5255/5255 [==============================] - 2s 323us/step - loss: 0.0185 - mean_absolute_error
Epoch 1/3
5037/5037 [==============================] - 6s 1ms/step - loss: 0.0901 - mean_absolute_error:
Epoch 2/3
5037/5037 [==============================] - 2s 335us/step - loss: 0.0268 - mean_absolute_error
Epoch 3/3
5037/5037 [==============================] - 2s 326us/step - loss: 0.0227 - mean_absolute_error
Epoch 1/3
9772/9772 [==============================] - 8s 812us/step - loss: 0.0636 - mean_absolute_error
Epoch 2/3
9772/9772 [==============================] - 3s 344us/step - loss: 0.0297 - mean_absolute_error
Epoch 3/3
9772/9772 [==============================] - 4s 368us/step - loss: 0.0274 - mean_absolute_error
Epoch 1/3
6561/6561 [==============================] - 7s 1ms/step - loss: 0.0709 - mean_absolute_error:
Epoch 2/3
6561/6561 [==============================] - 3s 381us/step - loss: 0.0199 - mean_absolute_error
Epoch 3/3
6561/6561 [==============================] - 2s 375us/step - loss: 0.0169 - mean_absolute_error
Epoch 1/3
5005/5005 [==============================] - 7s 1ms/step - loss: 0.0904 - mean_absolute_error:
Epoch 2/3
5005/5005 [==============================] - 2s 368us/step - loss: 0.0192 - mean_absolute_error
Epoch 3/3
5005/5005 [==============================] - 2s 376us/step - loss: 0.0133 - mean_absolute_error
Epoch 1/3
7507/7507 [==============================] - 8s 1ms/step - loss: 0.0659 - mean_absolute_error:
Epoch 2/3
7507/7507 [==============================] - 3s 374us/step - loss: 0.0114 - mean_absolute_error
Epoch 3/3
7507/7507 [==============================] - 3s 352us/step - loss: 0.0087 - mean_absolute_error
Epoch 1/3
6954/6954 [==============================] - 7s 1ms/step - loss: 0.0678 - mean_absolute_error:
Epoch 2/3
6954/6954 [==============================] - 2s 334us/step - loss: 0.0164 - mean_absolute_error
```

```
Epoch 3/3
6954/6954 [==============================] - 3s 362us/step - loss: 0.0134 - mean_absolute_erro
Epoch 1/3
7894/7894 [==============================] - 7s 941us/step - loss: 0.0658 - mean_absolute_erro
Epoch 2/3
7894/7894 [==============================] - 3s 336us/step - loss: 0.0104 - mean_absolute_erro
Epoch 3/3
7894/7894 [==============================] - 3s 326us/step - loss: 0.0082 - mean_absolute_erro
Epoch 1/3
7006/7006 [==============================] - 7s 1ms/step - loss: 0.0784 - mean_absolute_error:
Epoch 2/3
7006/7006 [==============================] - 2s 336us/step - loss: 0.0157 - mean_absolute_erro
Epoch 3/3
7006/7006 [==============================] - 2s 333us/step - loss: 0.0125 - mean_absolute_erro
Epoch 1/3
5004/5004 [==============================] - 7s 1ms/step - loss: 0.0935 - mean_absolute_error:
Epoch 2/3
5004/5004 [==============================] - 2s 343us/step - loss: 0.0178 - mean_absolute_erro
Epoch 3/3
5004/5004 [==============================] - 2s 355us/step - loss: 0.0123 - mean_absolute_erro
Epoch 1/3
6091/6091 [==============================] - 7s 1ms/step - loss: 0.0796 - mean_absolute_error:
Epoch 2/3
6091/6091 [==============================] - 2s 361us/step - loss: 0.0153 - mean_absolute_erro
Epoch 3/3
6091/6091 [==============================] - 2s 341us/step - loss: 0.0124 - mean_absolute_erro
```

```
In [15]: # training time
         total_time = dt_end - dt_start + max_time
         print("Training time: ", round(total_time, 5))
```

```
Training time:  47.95906
```

```
In [16]: # predicts using the neural shrub
         def neural_shrub_predict(tree, nn_model, test):
             test = np.asarray(test)

             pred = []

             # getting the predicted value
             tree_pred = tree.apply(test)

             test = preprocessing.scale(test)
             for i in range(len(test)):

                 # gets the neural network for the tree predicted val
                 x = tree_pred[i]
                 nn_model_class = nn_model[x]

                 # preprocessing
                 l = np.asarray(test[i])
                 l = l.reshape(-1, 90)

                 # gets the val using the nn associated with
                 # the current predicted value
                 ans = nn_model_class.predict(l)
                 pred.append(ans)

             return pred
```

```
In [17]: # predicting for test data
         # results scaled between 0 - 1

         pred = neural_shrub_predict(tree, shrubs, test)
```

```
In [18]: pred = np.asarray(pred)
         pred = pred.ravel()
         pred = pred.reshape(-1,1)
         pred = sc.inverse_transform(pred)

         print(mean_absolute_error(test_label, pred), mean_squared_error(test_label, pred))
```

```
6.7961220344692705 99.47642204472457
```

```
In [19]: print(test_label[0:10])
         print(pred[0:10])
```

```
[2007. 2003. 2005. 2003. 2005. 2007. 2003. 2003. 2003. 2005.]
[[1996.5908]
 [1999.8171]
 [2002.4985]
 [2005.3513]
 [2005.3563]
 [1999.4575]
 [1998.3212]
 [1994.4529]
 [1998.7758]
 [1999.3042]]
```