Ben Roth Week 6 - Jupyter Notebook (Trees Regression)

March 6, 2020

- https://scikit-learn.org/stable/modules/tree.html
- http://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeRegressor.html

In [1]:
```
import sklearn

sklearn.__version__
```

Out[1]: '0.21.2'

In [2]:
```
!pip3 install scikit-learn --upgrade
```

```
'pip3' is not recognized as an internal or external command,
operable program or batch file.
```

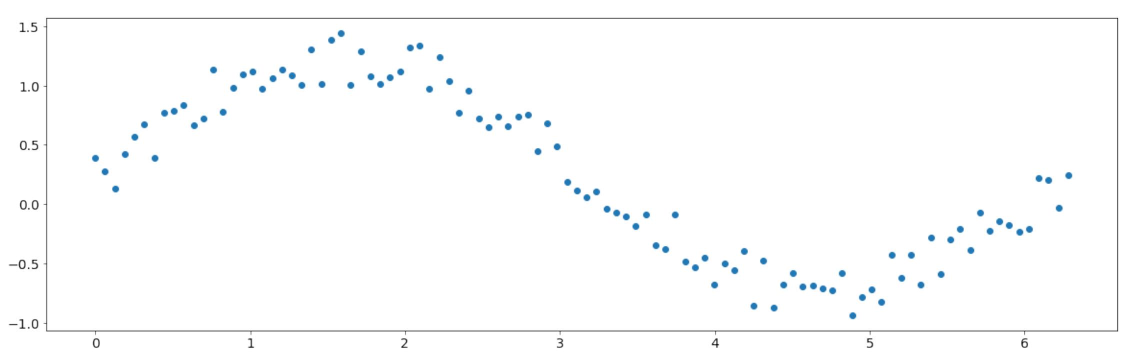
In [3]:
```
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
import pandas as pd
import numpy as np
```

In [4]:
```
plt.rcParams['figure.figsize'] = (20, 6)
plt.rcParams['font.size'] = 14
```

In [5]:
```
x = np.linspace(0, 2* np.pi, 100)
y = np.sin(x) + .5*np.random.random(100)
```

In [6]:
```
plt.scatter(x, y)
```

Out[6]: <matplotlib.collections.PathCollection at 0x1fa0bec1c88>



In [7]: from sklearn import tree

In [8]: 2**16

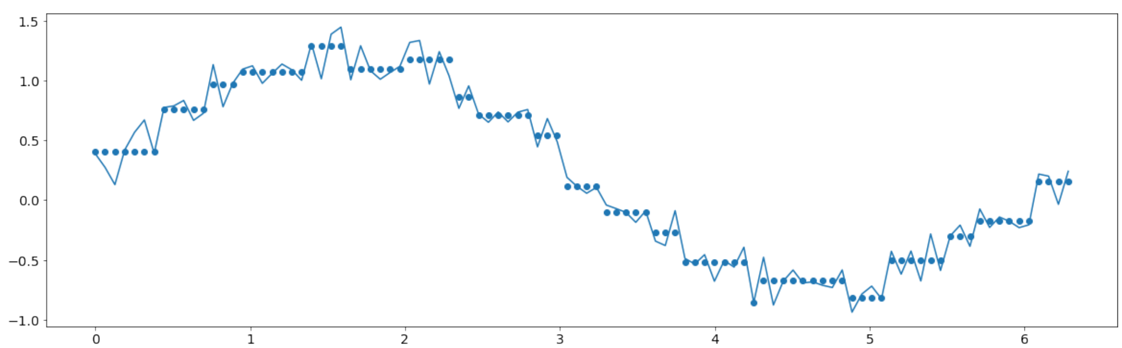
Out[8]: 65536

In [9]:
```
regression = tree.DecisionTreeRegressor(max_depth=8, min_samples_split=8)
regression.fit(x.reshape(-1, 1), y)

yp = regression.predict(x.reshape(-1,1))

plt.scatter(x, yp)
plt.plot(x, y)
```

Out[9]: [<matplotlib.lines.Line2D at 0x1fa0dae73c8>]



In [10]: regression.predict([[2]])

Out[10]: array([1.17958191])

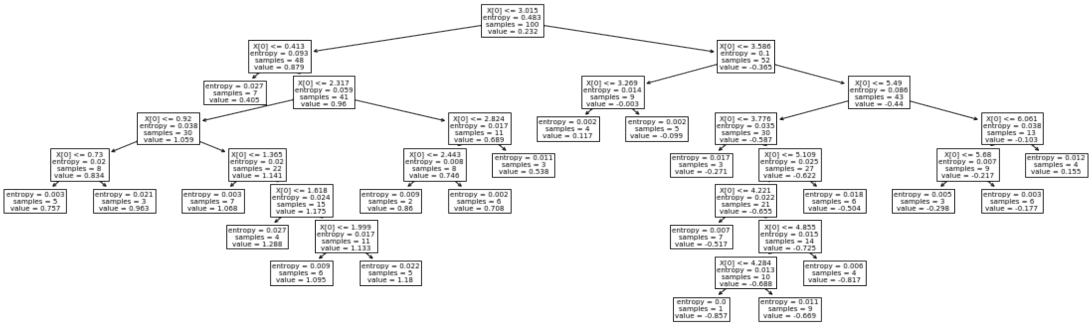
In [11]: path = regression.decision_path(x.reshape(-1, 1))

In [12]: path.todense()

Out[12]:
```
matrix([[1, 1, 1, ..., 0, 0, 0],
        [1, 1, 1, ..., 0, 0, 0],
        [1, 1, 1, ..., 0, 0, 0],
        ...,
        [1, 0, 0, ..., 0, 0, 1],
        [1, 0, 0, ..., 0, 0, 1],
        [1, 0, 0, ..., 0, 0, 1]], dtype=int64)
```

In [13]: tree.plot_tree(regression)

```
Out[13]: [Text(524.52, 308.04, 'X[0] <= 3.015\nentropy = 0.483\nsamples = 100\nvalue = 0.232')
          Text(290.16, 271.8, 'X[0] <= 0.413\nentropy = 0.093\nsamples = 48\nvalue = 0.879'),
          Text(245.52, 235.56, 'entropy = 0.027\nsamples = 7\nvalue = 0.405'),
          Text(334.8, 235.56, 'X[0] <= 2.317\nentropy = 0.059\nsamples = 41\nvalue = 0.96'),
          Text(178.56, 199.32000000000002, 'X[0] <= 0.92\nentropy = 0.038\nsamples = 30\nvalue
          Text(89.28, 163.08, 'X[0] <= 0.73\nentropy = 0.02\nsamples = 8\nvalue = 0.834'),
          Text(44.64, 126.84, 'entropy = 0.003\nsamples = 5\nvalue = 0.757'),
          Text(133.92000000000002, 126.84, 'entropy = 0.021\nsamples = 3\nvalue = 0.963'),
          Text(267.84000000000003, 163.08, 'X[0] <= 1.365\nentropy = 0.02\nsamples = 22\nvalue
          Text(223.2, 126.84, 'entropy = 0.003\nsamples = 7\nvalue = 1.068'),
          Text(312.48, 126.84, 'X[0] <= 1.618\nentropy = 0.024\nsamples = 15\nvalue = 1.175'),
          Text(267.84000000000003, 90.60000000000002, 'entropy = 0.027\nsamples = 4\nvalue = 1
          Text(357.12, 90.60000000000002, 'X[0] <= 1.999\nentropy = 0.017\nsamples = 11\nvalue
          Text(312.48, 54.360000000000014, 'entropy = 0.009\nsamples = 6\nvalue = 1.095'),
          Text(401.76, 54.360000000000014, 'entropy = 0.022\nsamples = 5\nvalue = 1.18'),
          Text(491.04, 199.32000000000002, 'X[0] <= 2.824\nentropy = 0.017\nsamples = 11\nvalu
          Text(446.4, 163.08, 'X[0] <= 2.443\nentropy = 0.008\nsamples = 8\nvalue = 0.746'),
          Text(401.76, 126.84, 'entropy = 0.009\nsamples = 2\nvalue = 0.86'),
          Text(491.04, 126.84, 'entropy = 0.002\nsamples = 6\nvalue = 0.708'),
          Text(535.6800000000001, 163.08, 'entropy = 0.011\nsamples = 3\nvalue = 0.538'),
          Text(758.88, 271.8, 'X[0] <= 3.586\nentropy = 0.1\nsamples = 52\nvalue = -0.365'),
          Text(624.96, 235.56, 'X[0] <= 3.269\nentropy = 0.014\nsamples = 9\nvalue = -0.003'),
          Text(580.32, 199.32000000000002, 'entropy = 0.002\nsamples = 4\nvalue = 0.117'),
          Text(669.6, 199.32000000000002, 'entropy = 0.002\nsamples = 5\nvalue = -0.099'),
          Text(892.8, 235.56, 'X[0] <= 5.49\nentropy = 0.086\nsamples = 43\nvalue = -0.44'),
          Text(758.88, 199.32000000000002, 'X[0] <= 3.776\nentropy = 0.035\nsamples = 30\nvalu
          Text(714.24, 163.08, 'entropy = 0.017\nsamples = 3\nvalue = -0.271'),
          Text(803.52, 163.08, 'X[0] <= 5.109\nentropy = 0.025\nsamples = 27\nvalue = -0.622')
          Text(758.88, 126.84, 'X[0] <= 4.221\nentropy = 0.022\nsamples = 21\nvalue = -0.655')
          Text(714.24, 90.60000000000002, 'entropy = 0.007\nsamples = 7\nvalue = -0.517'),
          Text(803.52, 90.60000000000002, 'X[0] <= 4.855\nentropy = 0.015\nsamples = 14\nvalue
          Text(758.88, 54.360000000000014, 'X[0] <= 4.284\nentropy = 0.013\nsamples = 10\nvalu
          Text(714.24, 18.120000000000005, 'entropy = 0.0\nsamples = 1\nvalue = -0.857'),
          Text(803.52, 18.120000000000005, 'entropy = 0.011\nsamples = 9\nvalue = -0.669'),
          Text(848.16, 54.360000000000014, 'entropy = 0.006\nsamples = 4\nvalue = -0.817'),
          Text(848.16, 126.84, 'entropy = 0.018\nsamples = 6\nvalue = -0.504'),
          Text(1026.72, 199.32000000000002, 'X[0] <= 6.061\nentropy = 0.038\nsamples = 13\nval
          Text(982.08, 163.08, 'X[0] <= 5.68\nentropy = 0.007\nsamples = 9\nvalue = -0.217'),
          Text(937.44, 126.84, 'entropy = 0.005\nsamples = 3\nvalue = -0.298'),
          Text(1026.72, 126.84, 'entropy = 0.003\nsamples = 6\nvalue = -0.177'),
          Text(1071.3600000000001, 163.08, 'entropy = 0.012\nsamples = 4\nvalue = 0.155')]
```



```
In [14]: bikeshare = pd.read_csv('../data/bikeshare_daily_agg.csv', index_col='hour_of_day')

In [15]: bikeshare
```

Out[15]:

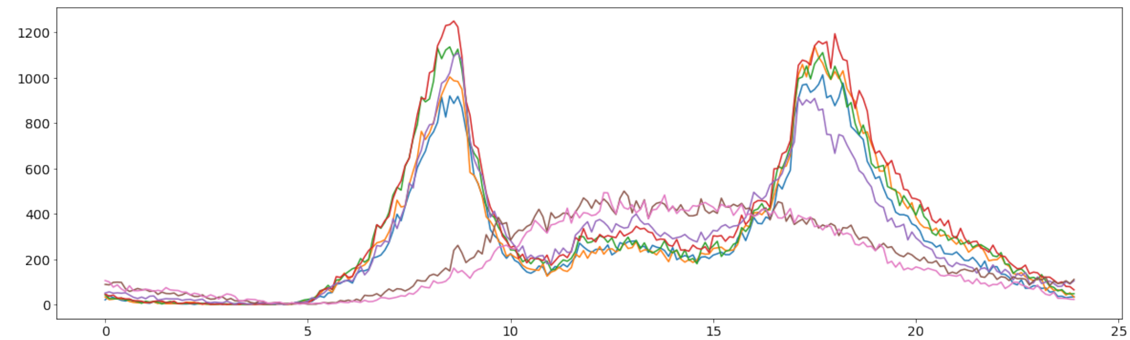
hour_of_day	0	1	2	3	4	5	6
0.0	21.0	34.0	43.0	47.0	51.0	89.0	106.0
0.1	39.0	22.0	27.0	37.0	56.0	87.0	100.0
0.2	31.0	24.0	26.0	42.0	50.0	98.0	77.0
0.3	26.0	27.0	25.0	29.0	52.0	99.0	87.0
0.4	19.0	24.0	29.0	29.0	50.0	98.0	69.0
0.5	16.0	23.0	20.0	25.0	52.0	70.0	58.0
0.6	21.0	14.0	22.0	25.0	42.0	58.0	83.0
0.7	16.0	10.0	17.0	17.0	32.0	64.0	61.0
0.8	11.0	8.0	13.0	17.0	27.0	71.0	61.0
0.9	11.0	8.0	6.0	15.0	33.0	66.0	63.0
1.0	8.0	9.0	15.0	9.0	26.0	66.0	68.0
1.1	6.0	11.0	11.0	9.0	37.0	58.0	65.0
1.2	14.0	12.0	13.0	14.0	38.0	61.0	59.0
1.3	8.0	5.0	10.0	17.0	20.0	45.0	63.0
1.4	10.0	10.0	3.0	16.0	23.0	55.0	59.0
1.5	6.0	4.0	9.0	9.0	26.0	49.0	75.0
1.6	4.0	4.0	14.0	10.0	25.0	48.0	65.0
1.7	5.0	5.0	8.0	15.0	25.0	51.0	63.0
1.8	4.0	4.0	5.0	4.0	25.0	39.0	62.0
1.9	3.0	4.0	6.0	6.0	19.0	43.0	55.0
2.0	3.0	10.0	5.0	9.0	22.0	52.0	55.0
2.1	4.0	10.0	6.0	5.0	23.0	43.0	43.0
2.2	1.0	10.0	5.0	7.0	18.0	36.0	50.0
2.3	4.0	3.0	5.0	8.0	21.0	36.0	46.0
2.4	4.0	3.0	5.0	2.0	24.0	43.0	36.0
2.5	5.0	2.0	3.0	7.0	11.0	30.0	34.0
2.6	6.0	2.0	2.0	1.0	8.0	24.0	43.0
2.7	3.0	2.0	2.0	2.0	11.0	25.0	36.0
2.8	5.0	2.0	2.0	3.0	9.0	33.0	25.0

```
2.9          2.0    NaN    4.0    1.0    8.0   33.0   38.0
...          ...    ...    ...    ...    ...    ...    ...
21.0       232.0  266.0  300.0  292.0  185.0  137.0  125.0
21.1       214.0  286.0  266.0  336.0  176.0  151.0  132.0
21.2       195.0  263.0  271.0  293.0  175.0  156.0  102.0
21.3       194.0  231.0  282.0  268.0  167.0  150.0  107.0
21.4       208.0  246.0  252.0  277.0  162.0  146.0   92.0
21.5       189.0  257.0  262.0  279.0  139.0  161.0  114.0
21.6       164.0  260.0  264.0  249.0  158.0  123.0   92.0
21.7       192.0  229.0  253.0  244.0  162.0  120.0   95.0
21.8       142.0  193.0  242.0  231.0  148.0  139.0   81.0
21.9       167.0  197.0  215.0  262.0  175.0  135.0   89.0
22.0       167.0  198.0  211.0  228.0  147.0  111.0   88.0
22.1       135.0  214.0  229.0  215.0  138.0  123.0   99.0
22.2       133.0  180.0  179.0  211.0  155.0  120.0   71.0
22.3       120.0  169.0  177.0  187.0  141.0  106.0   82.0
22.4        94.0  137.0  138.0  172.0  109.0  123.0   98.0
22.5        87.0  132.0  152.0  158.0  137.0  114.0   99.0
22.6       109.0  141.0  157.0  160.0  127.0  115.0   87.0
22.7        72.0  123.0  140.0  174.0  141.0  112.0   81.0
22.8        73.0  114.0  117.0  136.0   92.0   86.0   72.0
22.9        83.0  106.0  120.0  143.0   96.0  105.0   68.0
23.0        82.0  111.0  111.0  140.0  122.0  117.0   86.0
23.1        71.0  115.0  132.0  122.0   93.0  108.0   61.0
23.2        74.0   62.0  105.0  121.0   99.0  121.0   46.0
23.3        55.0   80.0   82.0  121.0  104.0   84.0   41.0
23.4        57.0   63.0   76.0  108.0   93.0  103.0   54.0
23.5        36.0   65.0   60.0   94.0   80.0   93.0   28.0
23.6        37.0   61.0   66.0  100.0   81.0   95.0   28.0
23.7        30.0   42.0   49.0   80.0  101.0  105.0   27.0
23.8        33.0   52.0   47.0   79.0   91.0   93.0   24.0
23.9        34.0   33.0   48.0   65.0  105.0  111.0   23.0

[240 rows x 7 columns]
```

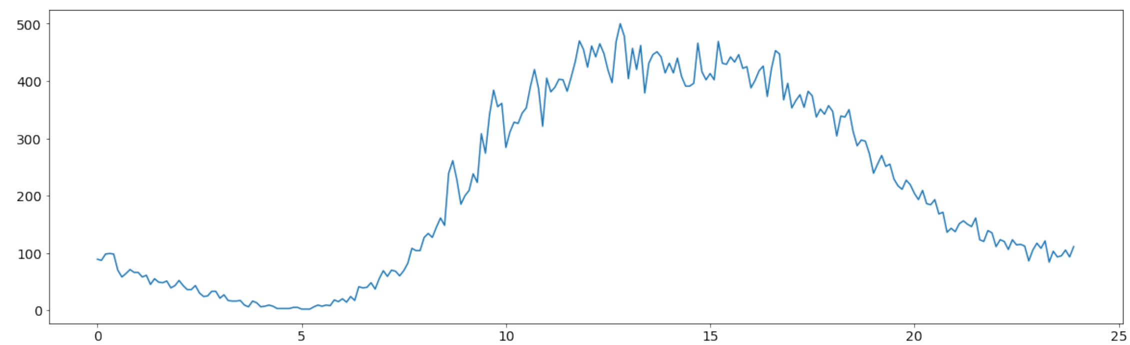
```
In [16]: plt.plot(bikeshare)
```

```
Out[16]: [<matplotlib.lines.Line2D at 0x1fa0dfcd860>,
          <matplotlib.lines.Line2D at 0x1fa0dfcd9b0>,
          <matplotlib.lines.Line2D at 0x1fa0dfcdb00>,
          <matplotlib.lines.Line2D at 0x1fa0dfcdc50>,
          <matplotlib.lines.Line2D at 0x1fa0dfcdda0>,
          <matplotlib.lines.Line2D at 0x1fa0dfcdef0>,
          <matplotlib.lines.Line2D at 0x1fa0dfd9080>]
```




```
In [17]: plt.plot(bikeshare['5'])
```

```
Out[17]: [<matplotlib.lines.Line2D at 0x1fa0dbc76d8>]
```



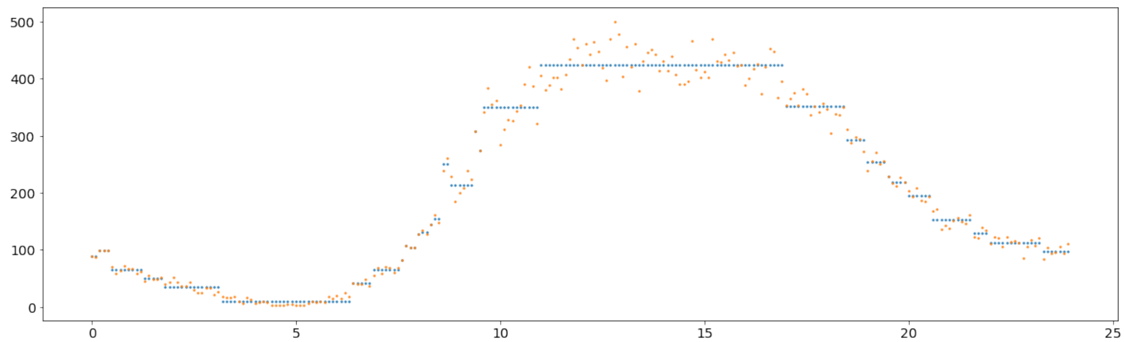

```
In [18]: hours = bikeshare.index.values.reshape(-1,1)

         bike_reg = tree.DecisionTreeRegressor(max_depth=5)
         bike_reg.fit(hours, bikeshare['5'].fillna(0))

         bike_pred = bike_reg.predict(hours)

         plt.scatter(hours, bike_pred, s=2)
         plt.scatter(hours, bikeshare['5'], s=2)
```

```
Out[18]: <matplotlib.collections.PathCollection at 0x1fa0e055630>
```



1 Use the bikeshare dataset (see above) and choose a weekday (0,1,2,3,4).

2 1. Create 5 Decision Tree Regressors using `max_depth=4,5,6,7,8`. For each one of these models, calculate the MSE between the predicted values from the model (`bike_pred`) and the actual values (`bikeshare['n']`). Create a plot showing the predictions along with the actuals. You may also show the `print_tree()` for a sanity check as well.

In [19]:
```
bikeshare['2'].index
```

Out[19]:
```
Float64Index([ 0.0,  0.1,  0.2,  0.3,  0.4,  0.5,  0.6,  0.7,  0.8,  0.9,
              ...
              23.0, 23.1, 23.2, 23.3, 23.4, 23.5, 23.6, 23.7, 23.8, 23.9],
             dtype='float64', name='hour_of_day', length=240)
```

In [27]:
```
from sklearn.metrics import mean_squared_error

max_depth = [4, 5, 6, 7, 8]

wed = bikeshare['2']

wed = wed.fillna(0)

mse = []
preds = []
for depth in max_depth:
    dt = tree.DecisionTreeRegressor(max_depth=depth)
    dt.fit(np.array(wed.index).reshape(-1, 1), wed.values)
    pred = dt.predict(np.array(wed.index).reshape(-1, 1))
    preds.append(pred)
```

```
        mse.append(mean_squared_error(y_true = wed.values, y_pred = pred))

    for i in range(len(mse)):
        print('Max Depth:', max_depth[i], ', MSE: ', mse[i])
```

```
Max Depth: 4 , MSE:  12346.81080131674
Max Depth: 5 , MSE:  6101.514951765633
Max Depth: 6 , MSE:  1669.1621186591121
Max Depth: 7 , MSE:  596.2443307249763
Max Depth: 8 , MSE:  300.7029178072744
```

In [29]:
```
preds
```

Out[29]:
```
[array([ 26.125     ,  26.125     ,  26.125     ,  26.125     ,
         26.125     ,  26.125     ,  26.125     ,  26.125     ,
          7.13333333,   7.13333333,   7.13333333,   7.13333333,
          7.13333333,   7.13333333,   7.13333333,   7.13333333,
          7.13333333,   7.13333333,   7.13333333,   7.13333333,
          7.13333333,   7.13333333,   7.13333333,   7.13333333,
          7.13333333,   7.13333333,   7.13333333,   7.13333333,
          7.13333333,   7.13333333,   7.13333333,   7.13333333,
          7.13333333,   7.13333333,   7.13333333,   7.13333333,
          7.13333333,   7.13333333,   7.13333333,   7.13333333,
          7.13333333,   7.13333333,   7.13333333,   7.13333333,
          7.13333333,   7.13333333,   7.13333333,   7.13333333,
          7.13333333,   7.13333333,   7.13333333,   7.13333333,
          7.13333333,  63.5       ,  63.5       ,  75.5       ,
         75.5       , 109.        , 109.        , 138.5       ,
        138.5       , 161.        , 161.        , 196.66666667,
        196.66666667, 196.66666667, 349.2       , 349.2       ,
        349.2       , 349.2       , 349.2       , 551.6       ,
        551.6       , 551.6       , 551.6       , 551.6       ,
        989.14285714, 989.14285714, 989.14285714, 989.14285714,
        989.14285714, 989.14285714, 989.14285714, 989.14285714,
        989.14285714, 989.14285714, 989.14285714, 989.14285714,
        989.14285714, 989.14285714, 672.        , 672.        ,
        672.        , 416.        , 416.        , 416.        ,
        416.        , 416.        , 245.        , 245.        ,
        245.        , 245.        , 245.        , 245.        ,
        245.        , 245.        , 245.        , 245.        ,
        245.        , 245.        , 245.        , 245.        ,
        245.        , 245.        , 245.        , 245.        ,
        245.        , 245.        , 245.        , 245.        ,
        245.        , 245.        , 245.        , 245.        ,
        245.        , 245.        , 245.        , 245.        ,
        245.        , 245.        , 245.        , 245.        ,
        245.        , 245.        , 245.        , 245.        ,
```

```
       245.        , 245.        , 245.        , 245.        ,
       245.        , 245.        , 245.        , 245.        ,
       245.        , 245.        , 245.        , 245.        ,
       245.        , 245.        , 245.        , 245.        ,
       245.        , 245.        , 245.        , 245.        ,
       245.        , 245.        , 245.        , 245.        ,
       756.41666667, 756.41666667, 756.41666667, 756.41666667,
       756.41666667, 756.41666667, 756.41666667, 756.41666667,
       756.41666667, 756.41666667, 756.41666667, 756.41666667,
       756.41666667, 756.41666667, 756.41666667, 756.41666667,
       756.41666667, 756.41666667, 756.41666667, 756.41666667,
       756.41666667, 756.41666667, 756.41666667, 756.41666667,
       756.41666667, 756.41666667, 756.41666667, 756.41666667,
       756.41666667, 756.41666667, 756.41666667, 756.41666667,
       756.41666667, 756.41666667, 756.41666667, 756.41666667,
       232.65909091, 232.65909091, 232.65909091, 232.65909091,
       232.65909091, 232.65909091, 232.65909091, 232.65909091,
       232.65909091, 232.65909091, 232.65909091, 232.65909091,
       232.65909091, 232.65909091, 232.65909091, 232.65909091,
       232.65909091, 232.65909091, 232.65909091, 232.65909091,
       232.65909091, 232.65909091, 232.65909091, 232.65909091,
       232.65909091, 232.65909091, 232.65909091, 232.65909091,
       232.65909091, 232.65909091, 232.65909091, 232.65909091,
       232.65909091, 232.65909091, 232.65909091, 232.65909091,
       232.65909091, 232.65909091, 232.65909091, 232.65909091,
       232.65909091, 232.65909091, 232.65909091, 232.65909091]),
array([  43.        ,   23.71428571,   23.71428571,   23.71428571,
         23.71428571,   23.71428571,   23.71428571,   23.71428571,
          5.31707317,    5.31707317,    5.31707317,    5.31707317,
          5.31707317,    5.31707317,    5.31707317,    5.31707317,
          5.31707317,    5.31707317,    5.31707317,    5.31707317,
          5.31707317,    5.31707317,    5.31707317,    5.31707317,
          5.31707317,    5.31707317,    5.31707317,    5.31707317,
          5.31707317,    5.31707317,    5.31707317,    5.31707317,
          5.31707317,    5.31707317,    5.31707317,    5.31707317,
          5.31707317,    5.31707317,    5.31707317,    5.31707317,
          5.31707317,    5.31707317,    5.31707317,    5.31707317,
          5.31707317,    5.31707317,    5.31707317,    5.31707317,
          5.31707317,   25.75      ,   25.75      ,   25.75      ,
         25.75      ,   61.        ,   66.        ,   85.        ,
         66.        ,  108.        ,  110.        ,  132.        ,
        145.        ,  158.        ,  164.        ,  203.        ,
        193.5       ,  193.5       ,  290.        ,  364.        ,
        364.        ,  364.        ,  364.        ,  501.        ,
        501.        ,  501.        ,  627.5       ,  627.5       ,
        848.        ,  848.        ,  848.        ,  848.        ,
        848.        , 1067.55555556, 1067.55555556, 1067.55555556,
       1067.55555556, 1067.55555556, 1067.55555556, 1067.55555556,
```

```
        1067.55555556, 1067.55555556,  709.        ,  653.5       ,
         653.5       ,  521.        ,  389.75      ,  389.75      ,
         389.75      ,  389.75      ,  207.11111111,  207.11111111,
         207.11111111,  207.11111111,  207.11111111,  207.11111111,
         207.11111111,  207.11111111,  207.11111111,  207.11111111,
         207.11111111,  207.11111111,  207.11111111,  207.11111111,
         207.11111111,  207.11111111,  207.11111111,  207.11111111,
         260.5       ,  260.5       ,  260.5       ,  260.5       ,
         260.5       ,  260.5       ,  260.5       ,  260.5       ,
         260.5       ,  260.5       ,  260.5       ,  260.5       ,
         260.5       ,  260.5       ,  260.5       ,  260.5       ,
         260.5       ,  260.5       ,  260.5       ,  260.5       ,
         260.5       ,  260.5       ,  260.5       ,  260.5       ,
         260.5       ,  260.5       ,  260.5       ,  260.5       ,
         260.5       ,  260.5       ,  260.5       ,  260.5       ,
         260.5       ,  260.5       ,  260.5       ,  260.5       ,
         260.5       ,  260.5       ,  260.5       ,  260.5       ,
         260.5       ,  260.5       ,  260.5       ,  260.5       ,
         499.11111111,  499.11111111,  499.11111111,  499.11111111,
         499.11111111,  499.11111111,  499.11111111,  499.11111111,
         499.11111111,  842.18518519,  842.18518519,  842.18518519,
         842.18518519,  842.18518519,  842.18518519,  842.18518519,
         842.18518519,  842.18518519,  842.18518519,  842.18518519,
         842.18518519,  842.18518519,  842.18518519,  842.18518519,
         842.18518519,  842.18518519,  842.18518519,  842.18518519,
         842.18518519,  842.18518519,  842.18518519,  842.18518519,
         842.18518519,  842.18518519,  842.18518519,  842.18518519,
         331.56521739,  331.56521739,  331.56521739,  331.56521739,
         331.56521739,  331.56521739,  331.56521739,  331.56521739,
         331.56521739,  331.56521739,  331.56521739,  331.56521739,
         331.56521739,  331.56521739,  331.56521739,  331.56521739,
         331.56521739,  331.56521739,  331.56521739,  331.56521739,
         331.56521739,  331.56521739,  331.56521739,  124.33333333,
         124.33333333,  124.33333333,  124.33333333,  124.33333333,
         124.33333333,  124.33333333,  124.33333333,  124.33333333,
         124.33333333,  124.33333333,  124.33333333,  124.33333333,
         124.33333333,  124.33333333,  124.33333333,  124.33333333,
         124.33333333,  124.33333333,  124.33333333,  124.33333333]),
 array([  43.        ,   26.75      ,   26.75      ,   26.75      ,
          26.75      ,   19.66666667,   19.66666667,   19.66666667,
          10.2       ,   10.2       ,   10.2       ,   10.2       ,
          10.2       ,   10.2       ,   10.2       ,   10.2       ,
          10.2       ,   10.2       ,    3.74193548,    3.74193548,
           3.74193548,    3.74193548,    3.74193548,    3.74193548,
           3.74193548,    3.74193548,    3.74193548,    3.74193548,
           3.74193548,    3.74193548,    3.74193548,    3.74193548,
           3.74193548,    3.74193548,    3.74193548,    3.74193548,
           3.74193548,    3.74193548,    3.74193548,    3.74193548,
```

```
   3.74193548,    3.74193548,    3.74193548,    3.74193548,
   3.74193548,    3.74193548,    3.74193548,    3.74193548,
   3.74193548,   23.        ,   23.        ,   23.        ,
  34.        ,   61.        ,   66.        ,   85.        ,
  66.        ,  108.        ,  110.        ,  132.        ,
 145.        ,  158.        ,  164.        ,  203.        ,
 187.        ,  200.        ,  290.        ,  350.        ,
 350.        ,  350.        ,  406.        ,  482.        ,
 510.5       ,  510.5       ,  608.        ,  647.        ,
 763.        ,  763.        ,  904.66666667,  904.66666667,
 904.66666667, 1088.125     , 1088.125     , 1088.125     ,
1088.125     , 1088.125     , 1088.125     , 1088.125     ,
1088.125     ,  903.        ,  709.        ,  667.        ,
 640.        ,  521.        ,  406.        ,  406.        ,
 406.        ,  341.        ,  270.25      ,  270.25      ,
 270.25      ,  270.25      ,  189.07142857,  189.07142857,
 189.07142857,  189.07142857,  189.07142857,  189.07142857,
 189.07142857,  189.07142857,  189.07142857,  189.07142857,
 189.07142857,  189.07142857,  189.07142857,  189.07142857,
 255.02439024,  255.02439024,  255.02439024,  255.02439024,
 255.02439024,  255.02439024,  255.02439024,  255.02439024,
 255.02439024,  255.02439024,  255.02439024,  255.02439024,
 255.02439024,  255.02439024,  255.02439024,  255.02439024,
 255.02439024,  255.02439024,  255.02439024,  255.02439024,
 255.02439024,  255.02439024,  255.02439024,  255.02439024,
 255.02439024,  255.02439024,  255.02439024,  255.02439024,
 255.02439024,  255.02439024,  255.02439024,  255.02439024,
 255.02439024,  255.02439024,  255.02439024,  255.02439024,
 255.02439024,  255.02439024,  255.02439024,  255.02439024,
 255.02439024,  335.33333333,  335.33333333,  335.33333333,
 423.2       ,  423.2       ,  423.2       ,  423.2       ,
 423.2       ,  594.        ,  594.        ,  594.        ,
 594.        ,  980.4375    ,  980.4375    ,  980.4375    ,
 980.4375    ,  980.4375    ,  980.4375    ,  980.4375    ,
 980.4375    ,  980.4375    ,  980.4375    ,  980.4375    ,
 980.4375    ,  980.4375    ,  980.4375    ,  980.4375    ,
 980.4375    ,  641.09090909,  641.09090909,  641.09090909,
 641.09090909,  641.09090909,  641.09090909,  641.09090909,
 641.09090909,  641.09090909,  641.09090909,  641.09090909,
 423.125     ,  423.125     ,  423.125     ,  423.125     ,
 423.125     ,  423.125     ,  423.125     ,  423.125     ,
 282.73333333,  282.73333333,  282.73333333,  282.73333333,
 282.73333333,  282.73333333,  282.73333333,  282.73333333,
 282.73333333,  282.73333333,  282.73333333,  282.73333333,
 282.73333333,  282.73333333,  282.73333333,  177.55555556,
 177.55555556,  177.55555556,  177.55555556,  177.55555556,
 177.55555556,  177.55555556,  177.55555556,  177.55555556,
  84.41666667,   84.41666667,   84.41666667,   84.41666667,
```

```
          84.41666667,   84.41666667,   84.41666667,   84.41666667,
          84.41666667,   84.41666667,   84.41666667,   84.41666667]),
 array([  43.        ,   26.        ,   26.        ,   26.        ,
          29.        ,   21.        ,   21.        ,   17.        ,
          11.6       ,   11.6       ,   11.6       ,   11.6       ,
          11.6       ,    8.8       ,    8.8       ,    8.8       ,
           8.8       ,    8.8       ,    3.20689655,    3.20689655,
           3.20689655,    3.20689655,    3.20689655,    3.20689655,
           3.20689655,    3.20689655,    3.20689655,    3.20689655,
           3.20689655,    3.20689655,    3.20689655,    3.20689655,
           3.20689655,    3.20689655,    3.20689655,    3.20689655,
           3.20689655,    3.20689655,    3.20689655,    3.20689655,
           3.20689655,    3.20689655,    3.20689655,    3.20689655,
           3.20689655,    3.20689655,    3.20689655,   11.5       ,
          11.5       ,   20.        ,   24.5       ,   24.5       ,
          34.        ,   61.        ,   66.        ,   85.        ,
          66.        ,  108.        ,  110.        ,  132.        ,
         145.        ,  158.        ,  164.        ,  203.        ,
         187.        ,  200.        ,  290.        ,  343.5       ,
         343.5       ,  363.        ,  406.        ,  482.        ,
         517.        ,  504.        ,  608.        ,  647.        ,
         733.        ,  793.        ,  915.        ,  899.5       ,
         899.5       ,  984.        , 1103.        , 1103.        ,
        1103.        , 1103.        , 1103.        , 1103.        ,
        1103.        ,  903.        ,  709.        ,  667.        ,
         640.        ,  521.        ,  436.        ,  391.        ,
         391.        ,  341.        ,  300.        ,  300.        ,
         240.5       ,  240.5       ,  221.        ,  186.61538462,
         186.61538462,  186.61538462,  186.61538462,  186.61538462,
         186.61538462,  186.61538462,  186.61538462,  186.61538462,
         186.61538462,  186.61538462,  186.61538462,  186.61538462,
         275.83333333,  275.83333333,  275.83333333,  275.83333333,
         275.83333333,  275.83333333,  275.83333333,  275.83333333,
         275.83333333,  275.83333333,  275.83333333,  275.83333333,
         275.83333333,  275.83333333,  275.83333333,  275.83333333,
         275.83333333,  275.83333333,  238.73913043,  238.73913043,
         238.73913043,  238.73913043,  238.73913043,  238.73913043,
         238.73913043,  238.73913043,  238.73913043,  238.73913043,
         238.73913043,  238.73913043,  238.73913043,  238.73913043,
         238.73913043,  238.73913043,  238.73913043,  238.73913043,
         238.73913043,  238.73913043,  238.73913043,  238.73913043,
         238.73913043,  330.5       ,  330.5       ,  345.        ,
         417.5       ,  417.5       ,  427.        ,  427.        ,
         427.        ,  525.        ,  617.        ,  617.        ,
         617.        ,  697.        ,  999.33333333,  999.33333333,
         999.33333333,  999.33333333,  999.33333333,  999.33333333,
         999.33333333,  999.33333333,  999.33333333,  999.33333333,
         999.33333333,  999.33333333,  999.33333333,  999.33333333,
```

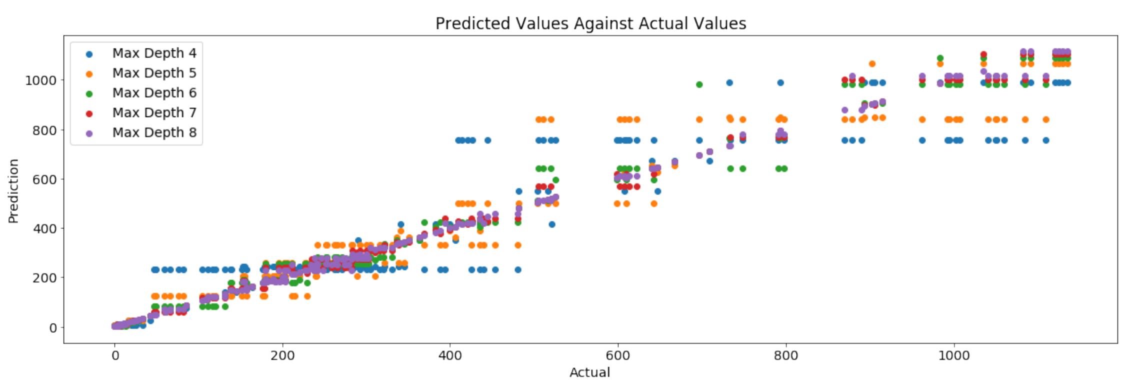
```
       999.33333333,  767.75      ,  767.75      ,  767.75      ,
       767.75      ,  568.71428571,  568.71428571,  568.71428571,
       568.71428571,  568.71428571,  568.71428571,  568.71428571,
       438.        ,  438.        ,  438.        ,  438.        ,
       438.        ,  438.        ,  378.5       ,  378.5       ,
       307.        ,  307.        ,  307.        ,  307.        ,
       307.        ,  307.        ,  307.        ,  261.5       ,
       261.5       ,  261.5       ,  261.5       ,  261.5       ,
       261.5       ,  261.5       ,  261.5       ,  218.33333333,
       218.33333333,  218.33333333,  157.16666667,  157.16666667,
       157.16666667,  157.16666667,  157.16666667,  157.16666667,
       117.        ,  117.        ,  117.        ,  117.        ,
       117.        ,   61.14285714,   61.14285714,   61.14285714,
        61.14285714,   61.14285714,   61.14285714,   61.14285714]),
array([  43.        ,   27.        ,   25.5       ,   25.5       ,
         29.        ,   20.        ,   22.        ,   17.        ,
          9.5       ,    9.5       ,   13.        ,   13.        ,
         13.        ,    6.5       ,    6.5       ,   10.33333333,
         10.33333333,   10.33333333,    5.28571429,    5.28571429,
          5.28571429,    5.28571429,    5.28571429,    5.28571429,
          5.28571429,    2.54545455,    2.54545455,    2.54545455,
          2.54545455,    2.54545455,    2.54545455,    2.54545455,
          2.54545455,    2.54545455,    2.54545455,    2.54545455,
          2.54545455,    2.54545455,    2.54545455,    2.54545455,
          2.54545455,    2.54545455,    2.54545455,    2.54545455,
          2.54545455,    2.54545455,    2.54545455,    9.        ,
         14.        ,   20.        ,   23.        ,   26.        ,
         34.        ,   61.        ,   66.        ,   85.        ,
         66.        ,  108.        ,  110.        ,  132.        ,
        145.        ,  158.        ,  164.        ,  203.        ,
        187.        ,  200.        ,  290.        ,  351.        ,
        336.        ,  363.        ,  406.        ,  482.        ,
        517.        ,  504.        ,  608.        ,  647.        ,
        733.        ,  793.        ,  915.        ,  893.        ,
        906.        ,  984.        , 1114.33333333, 1114.33333333,
       1114.33333333, 1114.33333333, 1114.33333333, 1114.33333333,
       1035.        ,  903.        ,  709.        ,  667.        ,
        640.        ,  521.        ,  436.        ,  399.        ,
        383.        ,  341.        ,  311.        ,  289.        ,
        241.        ,  240.        ,  221.        ,  194.6       ,
        194.6       ,  194.6       ,  194.6       ,  194.6       ,
        181.625     ,  181.625     ,  181.625     ,  181.625     ,
        181.625     ,  181.625     ,  181.625     ,  181.625     ,
        235.        ,  278.23529412,  278.23529412,  278.23529412,
        278.23529412,  278.23529412,  278.23529412,  278.23529412,
        278.23529412,  278.23529412,  278.23529412,  278.23529412,
        278.23529412,  278.23529412,  278.23529412,  278.23529412,
        278.23529412,  278.23529412,  228.46153846,  228.46153846,
```

```
        228.46153846,  228.46153846,  228.46153846,  228.46153846,
        228.46153846,  228.46153846,  228.46153846,  228.46153846,
        228.46153846,  228.46153846,  228.46153846,  252.1       ,
        252.1       ,  252.1       ,  252.1       ,  252.1       ,
        252.1       ,  252.1       ,  252.1       ,  252.1       ,
        252.1       ,  339.       ,  322.       ,  345.       ,
        414.       ,  421.       ,  445.       ,  418.       ,
        418.       ,  525.       ,  604.5      ,  604.5      ,
        642.       ,  697.       , 1017.69230769, 1017.69230769,
       1017.69230769, 1017.69230769, 1017.69230769, 1017.69230769,
       1017.69230769, 1017.69230769, 1017.69230769, 1017.69230769,
       1017.69230769, 1017.69230769, 1017.69230769,  880.       ,
        880.       ,  779.       ,  779.       ,  779.       ,
        734.       ,  611.25      ,  611.25      ,  611.25      ,
        611.25      ,  512.       ,  512.       ,  512.       ,
        456.33333333,  456.33333333,  456.33333333,  419.66666667,
        419.66666667,  419.66666667,  388.       ,  369.       ,
        318.25      ,  318.25      ,  318.25      ,  318.25      ,
        292.       ,  292.       ,  292.       ,  273.       ,
        273.       ,  273.       ,  254.6       ,  254.6       ,
        254.6       ,  254.6       ,  254.6       ,  213.       ,
        213.       ,  229.       ,  178.       ,  178.       ,
        146.75      ,  146.75      ,  146.75      ,  146.75      ,
        120.       ,  120.       ,  120.       ,  120.       ,
        105.       ,   71.       ,   71.       ,   71.       ,
         71.       ,   48.       ,   48.       ,   48.       ])]
```

In [35]:
```
colours = ['r', 'g', 'm', 'black', 'blue']

for i in range(len(preds)):
    plt.scatter(wed.values, preds[i], label = ('Max Depth ' + str(max_depth[i])))

plt.title('Predicted Values Against Actual Values')
plt.xlabel('Actual')
plt.ylabel('Prediction')
plt.legend(loc="upper left")
plt.show()
```



In [30]:
```python
'max depth' + str(1)
```

Out[30]:
```
'max depth1'
```

3 2. Using the 5 models created with various `max_depth` values, calculate the MSE between the predicted values (`bike_pred`) and values from all of the weekdays [`0,1,2,3,4`]. You should have 25 total MSE values, 5 values for each `max_depth`.

In [38]:
```python
weekday = ['0', '1', '2', '3', '4']

bikeshare = bikeshare.fillna(0)

for day in weekday:
    for i in range(len(preds)):
        mse = mean_squared_error(y_true = bikeshare[day].values, y_pred = preds[i])
        print('Max Depth:', max_depth[i], ', MSE: ', mse, ', Weekday: ', day)
```

```
Max Depth: 4 , MSE:  16200.667148719334 , Weekday:  0
Max Depth: 5 , MSE:  10579.826219781049 , Weekday:  0
Max Depth: 6 , MSE:  7088.172542190728 , Weekday:  0
Max Depth: 7 , MSE:  5955.647449694338 , Weekday:  0
Max Depth: 8 , MSE:  5792.606878882963 , Weekday:  0
Max Depth: 4 , MSE:  15114.876464195526 , Weekday:  1
Max Depth: 5 , MSE:  8084.399600813383 , Weekday:  1
Max Depth: 6 , MSE:  4200.2675531109235 , Weekday:  1
Max Depth: 7 , MSE:  3006.770403405211 , Weekday:  1
Max Depth: 8 , MSE:  2847.6663810025757 , Weekday:  1
Max Depth: 4 , MSE:  12346.81080131674 , Weekday:  2
Max Depth: 5 , MSE:  6101.514951765633 , Weekday:  2
Max Depth: 6 , MSE:  1669.1621186591121 , Weekday:  2
Max Depth: 7 , MSE:  596.2443307249763 , Weekday:  2
Max Depth: 8 , MSE:  300.7029178072744 , Weekday:  2
Max Depth: 4 , MSE:  16400.019857954547 , Weekday:  3
Max Depth: 5 , MSE:  8476.439264264578 , Weekday:  3
Max Depth: 6 , MSE:  4047.2954779416596 , Weekday:  3
Max Depth: 7 , MSE:  2648.647442047062 , Weekday:  3
Max Depth: 8 , MSE:  2474.806256531459 , Weekday:  3
Max Depth: 4 , MSE:  13739.871225198413 , Weekday:  4
Max Depth: 5 , MSE:  12681.66137906294 , Weekday:  4
Max Depth: 6 , MSE:  9487.773181941471 , Weekday:  4
Max Depth: 7 , MSE:  9019.185179517614 , Weekday:  4
Max Depth: 8 , MSE:  8853.029768277067 , Weekday:  4
```

4 3. (2 cont'd) Describe which max_depth you would recommend based on the groups of MSE values. Use the idea of generality of the model for your argument along with the MSE values as proof.

Looking at the above printed statement, I would argue that a max depth of at least 8 should be used for the model. In each of the above 25 models, a max depth of 8 has by far the lowest MSE for each weekday when compared to lower max depth values. Although the MSE values are better for a max depth of 8, I would hesitate to use a model trained on one day for a different; the only weekday values that produce a relatively low MSE with increased depth