basics

December 20, 2017

1 Crash Course Basics

Installation

```
In [90]: import tensorflow as tf
         hello = tf.constant("Hello World")
         sess = tf.Session()
         print(sess.run(hello))
```

```
b'Hello World'
```

Machine Learning Overview

Numpy

```
In [8]: import numpy as np
        my_list = [1,2,3]
        np.array(my_list)
```

```
Out[8]: array([1, 2, 3])
```

```
In [12]: np.arange(0,10,2)
```

```
Out[12]: array([0, 2, 4, 6, 8])
```

```
In [14]: np.zeros((5,5))
```

```
Out[14]: array([[ 0.,  0.,  0.,  0.,  0.],
                [ 0.,  0.,  0.,  0.,  0.],
                [ 0.,  0.,  0.,  0.,  0.],
                [ 0.,  0.,  0.,  0.,  0.],
                [ 0.,  0.,  0.,  0.,  0.]])
```

```
In [15]: np.ones((5,5))
```

```
Out[15]: array([[ 1.,  1.,  1.,  1.,  1.],
                [ 1.,  1.,  1.,  1.,  1.],
                [ 1.,  1.,  1.,  1.,  1.],
                [ 1.,  1.,  1.,  1.,  1.],
                [ 1.,  1.,  1.,  1.,  1.]])
```

```
In [16]: np.linspace(0,10,10)

Out[16]: array([  0.        ,   1.11111111,   2.22222222,   3.33333333,
                  4.44444444,   5.55555556,   6.66666667,   7.77777778,
                  8.88888889,  10.        ])

In [18]: np.random.randint(0,101,(3,3))

Out[18]: array([[41, 14, 48],
                [ 1, 14, 32],
                [43, 16, 59]])

In [21]: np.random.seed(101)
         arr = np.random.randint(0,100,10)

In [22]: arr.max()

Out[22]: 95

In [23]: arr.min()

Out[23]: 9

In [24]: arr.mean()

Out[24]: 60.799999999999997

In [25]: arr.argmax()

Out[25]: 0

In [26]: arr.argmin()

Out[26]: 7

In [27]: arr.reshape(2,5)

Out[27]: array([[95, 11, 81, 70, 63],
                [87, 75,  9, 77, 40]])

In [29]: mat = np.arange(0,100).reshape(10,10)

In [30]: mat[:,1]

Out[30]: array([ 1, 11, 21, 31, 41, 51, 61, 71, 81, 91])

In [32]: mat[mat > 50]

Out[32]: array([51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67,
                68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84,
                85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99])
```

Pandas

```
In [33]: import pandas as pd
```

```
In [40]: df = pd.read_csv('salaries.csv')
         df
```

```
Out[40]:      Name  Salary  Age
         0    John   50000   34
         1   Sally  120000   45
         2  Alyssa   80000   27
```

```
In [42]: df[['Salary', 'Age']]
```

```
Out[42]:    Salary  Age
         0   50000   34
         1  120000   45
         2   80000   27
```

```
In [43]: df.describe()
```

```
Out[43]:               Salary        Age
         count       3.000000   3.000000
         mean    83333.333333  35.333333
         std     35118.845843   9.073772
         min     50000.000000  27.000000
         25%     65000.000000  30.500000
         50%     80000.000000  34.000000
         75%    100000.000000  39.500000
         max    120000.000000  45.000000
```

```
In [46]: df[df['Salary']> 60000]
```

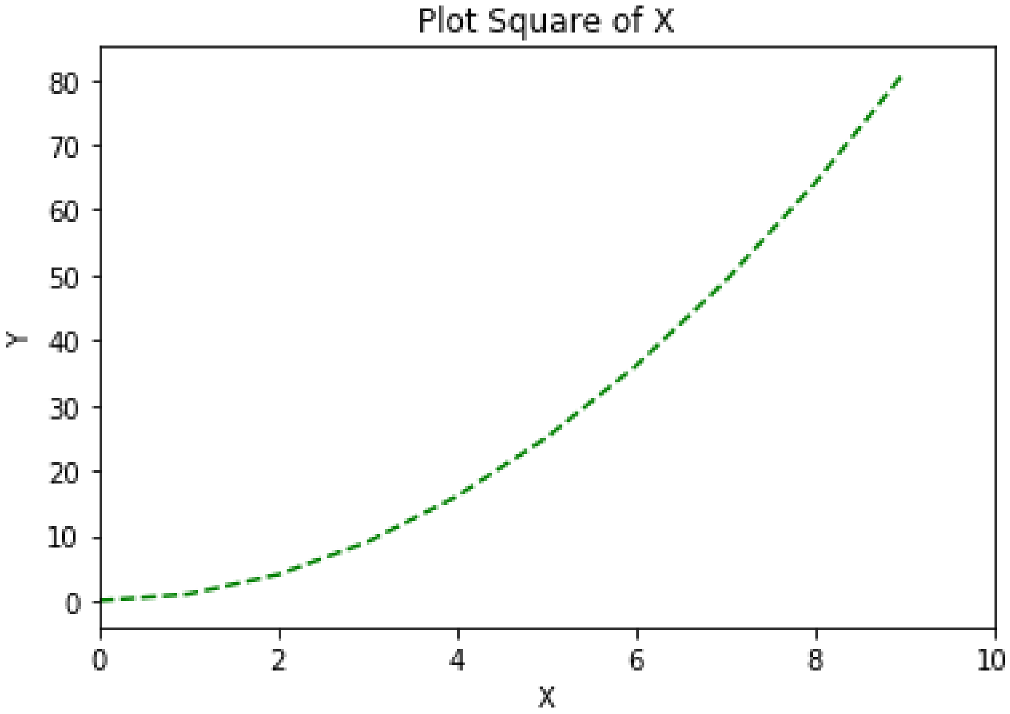
```
Out[46]:      Name  Salary  Age
         1   Sally  120000   45
         2  Alyssa   80000   27
```

Data Visualization

```
In [53]: import matplotlib.pyplot as plt
         %matplotlib inline
```

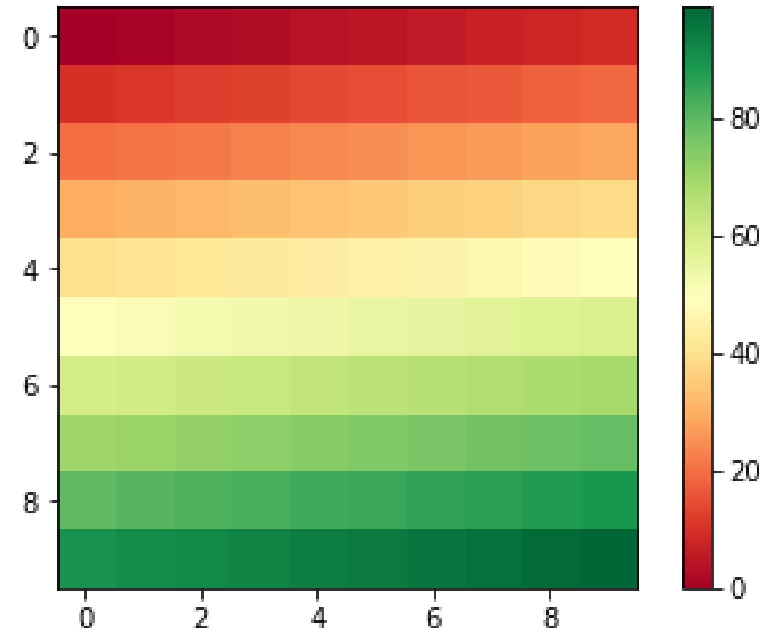
```
In [61]: x = np.arange(0,10)
         y = x**2
         plt.plot(x,y, 'g--')
         plt.xlim(0,10)
         plt.title('Plot Square of X')
         plt.xlabel('X')
         plt.ylabel('Y')
```

```
Out[61]: Text(0,0.5,'Y')
```



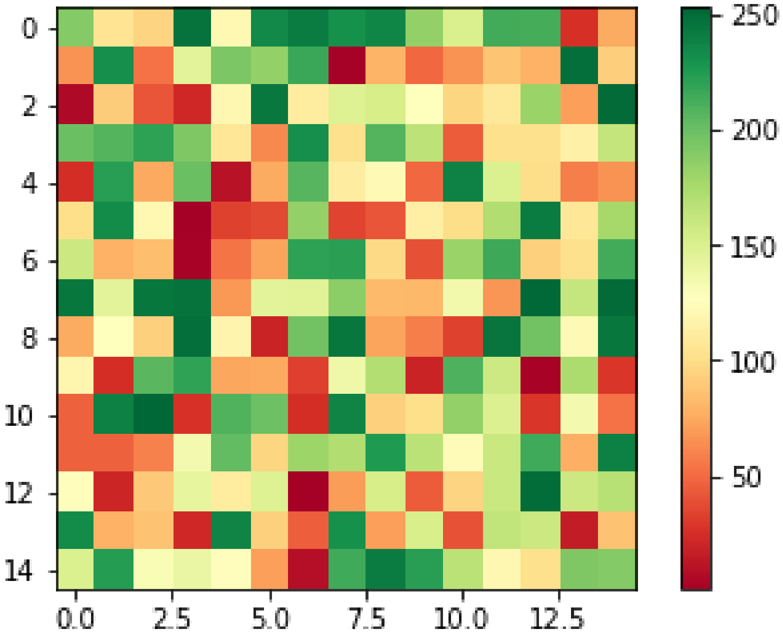
```
In [72]: mat = np.arange(0,100).reshape(10,10)
         plt.imshow(mat, cmap = 'RdYlGn')
         plt.colorbar()

Out[72]: <matplotlib.colorbar.Colorbar at 0x26470dc4898>
```



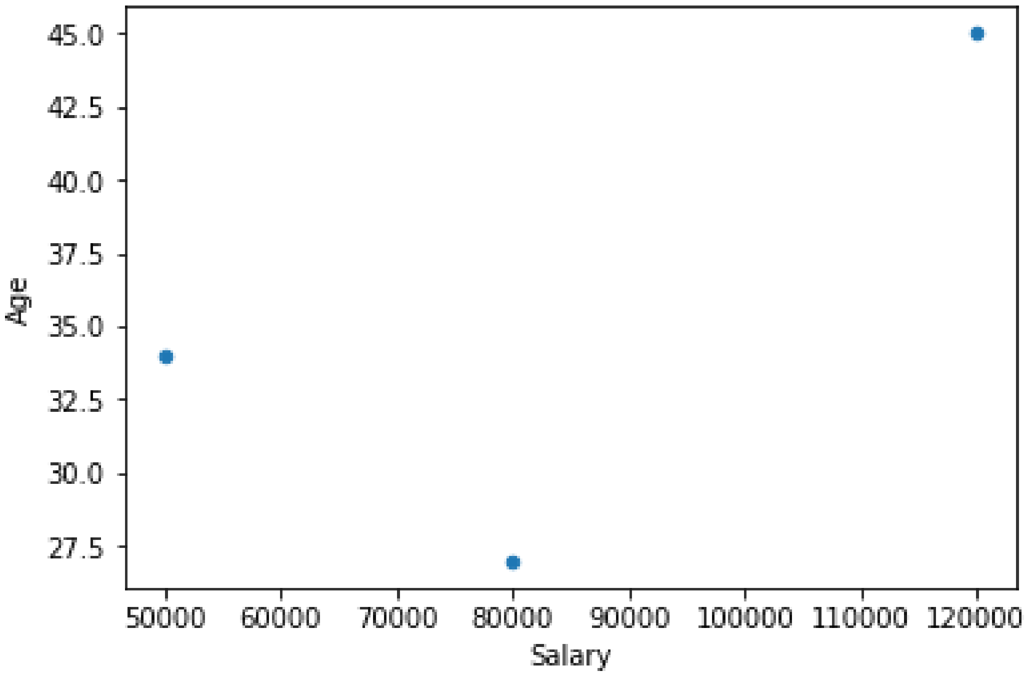
```
In [68]: mat = np.random.randint(0,255, (15,15))
         plt.imshow(mat,cmap = 'RdYlGn')
         plt.colorbar()

Out[68]: <matplotlib.colorbar.Colorbar at 0x26470bc3cf8>
```



```
In [73]: df.plot(x='Salary', y = 'Age', kind = 'scatter')

Out[73]: <matplotlib.axes._subplots.AxesSubplot at 0x26470d69470>
```



SciKit Learn

```
In [76]: from sklearn.preprocessing import MinMaxScaler

In [77]: data = np.random.randint(0,100,(10,2))

In [79]: scaler_model = MinMaxScaler()
         scaler_model.fit(data)

C:\Users\Ripti\Anaconda3\lib\site-packages\sklearn\utils\validation.py:475: DataConversionWarnin
  warnings.warn(msg, DataConversionWarning)


Out[79]: MinMaxScaler(copy=True, feature_range=(0, 1))

In [80]: scaler_model.transform(data)

Out[80]: array([[ 0.01162791,  1.        ],
               [ 0.6627907 ,  0.33333333],
               [ 0.31395349,  0.24      ],
              [ 1.        ,  0.58666667],
              [ 0.08139535,  0.65333333],
              [ 0.3372093 ,  0.        ],
              [ 0.38372093,  0.45333333],
              [ 0.        ,  0.05333333],
              [ 0.55813953,  0.12      ],
              [ 0.3255814 ,  0.30666667]])
```

```
In [86]: # Splitting variables
         df = pd.DataFrame(data = np.random.randint(0,101,(50,4)), columns = ['f1','f2','f3','la
         X = df[['f1', 'f2', 'f3']]
         y = df[['label']]
```

```
In [88]: # Test sets and training sets
         from sklearn.model_selection import train_test_split
         X_train,
         X_test,
         y_train,
         y_test = train_test_split(X,y, test_size = .2)
```

2 Crash Course Review Exercises

Import numpy,pandas,matplotlib,and sklearn. Also set visualizations to be shown inline in the notebook.

```
In [106]: import numpy as np
          import pandas as pd
          import matplotlib.pyplot as plt
          import sklearn
          %matplotlib inline
```

Set Numpy's Random Seed to 101

```
In [123]: np.random.seed(101)
```

Create a NumPy Matrix of 100 rows by 5 columns consisting of random integers from 1-100. (Keep in mind that the upper limit may be exclusive.

```
In [124]: mat = np.random.randint(1,101, (100,5))
          mat
```

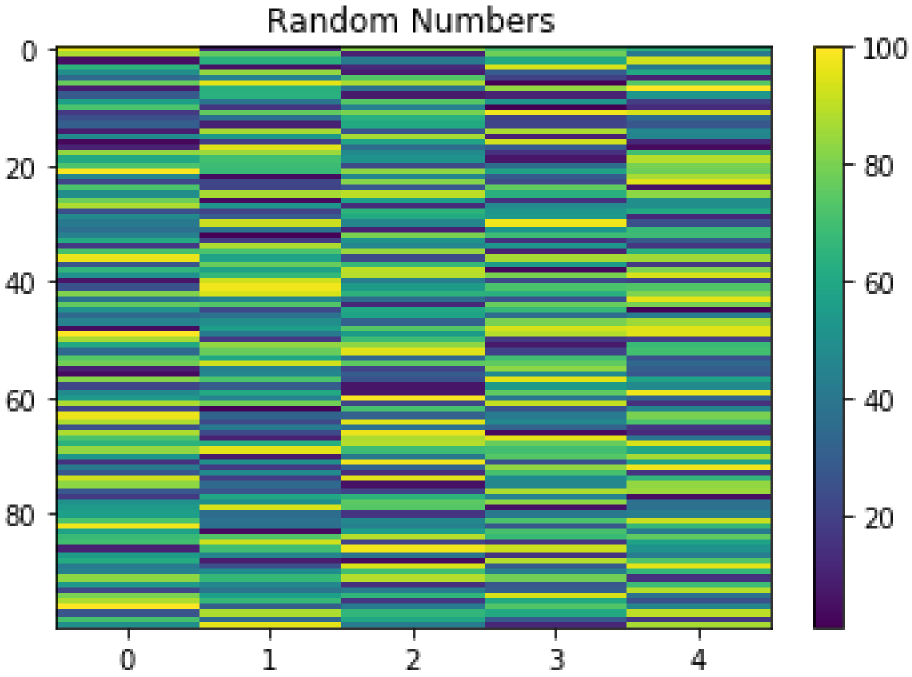
```
Out[124]: array([[ 96,  12,  82,  71,  64],
                 [ 88,  76,  10,  78,  41],
                 [  5,  64,  41,  61,  93],
                 [ 65,   6,  13,  94,  41],
                 [ 50,  84,   9,  30,  60],
                 [ 35,  45,  73,  20,  11],
                 [ 77,  96,  88,   1,  74],
                 [  9,  63,  37,  84, 100],
[ 29,  64,   8,  11,  53],
[ 57,  39,  74,  53,  19],
[ 72,  16,  45,   1,  13],
[ 18,  76,  80,  98,  94],
[ 25,  37,  64,  20,  36],
[ 31,  11,  61,  21,  28],
[  9,  87,  27,  88,  47],
[ 48,  55,  87,  10,  46],
[  3,  19,  59,  93,  12],
[ 11,  95,  36,  29,   4],
[ 84,  85,  48,  15,  70],
[ 61,  70,  52,   7,  89],
[ 72,  69,  24,  36,  80],
[ 99,  68,  83,  58,  78],
[ 47,   4,  47,  30,  87],
[ 22,  22,  82,  24,  95],
[ 72,  21,  28,  76,   6],
[ 50,  87,  90,  64,  83],
[ 78,   4,  57,  15,  50],
[ 88,  53,  14,  48,  50],
[ 25,  21,  65,  53,  61],
[ 48,  30,  61,  54,  12],
[ 41,  92,  46,  98,  25],
[ 37,  39,  10,  53,  68],
[ 44,   2,  80,  69,  69],
[ 62,  19,  52,  15,  29],
[ 18,  88,  47,  53,  17],
[ 71,  72,  85,  11,  63],
[ 97,  58,  24,  87,  86],
[ 27,  77,  67,  55,  18],
[ 66,  58,  90,   3,  81],
[ 51,  67,  89,  80,  94],
[  7,  93,  43,  23,  21],
[ 26,  98,  55,  72,  73],
[ 81,  94,  65,  64,  81],
[ 39,  46,  36,  26,  96],
[ 76,  73,  12,  77,  80],
[ 51,  23,  60,  67,   2],
[ 35,  38,  58,  36,  43],
[ 45,  50,  32,  80,  86],
[  4,  56,  74,  94,  95],
[100,  41,  55,  89,  95],
[ 87,  18,  69,  18,  19],
[ 61,  84,  83,   8,  68],
[ 35,  77,  95,  21,  70],
[ 74,  60,  35,  70,  26],
[ 79,  93,  75,  76,  34],
[ 10,  44,  21,  83,  31],
       [  4,  47,  30,  48,  28],
       [ 82,  72,  26,  95,  58],
       [ 22,  30,   7,  55,  48],
       [ 48,  61,   7,  76,  98],
       [ 54,  45,  99,  40,  33],
       [ 88,  79,  22,  91,  15],
       [ 21,   2,  71,  26,  46],
       [ 97,  33,  32,  42,  80],
       [ 88,  23,  95,  47,  72],
       [ 25,  42,  37,  32,  17],
       [ 88,  23,  97,   4,  13],
       [ 72,  10,  88,  96,  40],
       [ 65,  63,  89,  77,  94],
       [ 84,  96,  69,  70,  60],
       [ 53,   8,  41,  74,  87],
       [ 15,  50,  98,  26,  58],
       [ 41,  18,  33,  84,  98],
       [ 28,  48,  14,  71,  16],
       [ 93,  19,  95,  49,  66],
       [ 83,  35,   6,  47,  84],
       [ 28,  27,  21,  88,  85],
       [ 18,  60,  65,  45,   5],
       [ 52,  50,  75,  83,  38],
       [ 54,  94,  74,   6,  38],
       [ 57,  36,  16,  41,  43],
       [ 72,  38,  47,  72,  92],
       [ 98,  37,  44,  28,  67],
       [ 58,   4,  56,  71,  42],
       [ 68,  73,  89,  68,  76],
       [ 70,  93,  21,  16,  58],
       [ 10,  70,  98,  92,  52],
       [ 55,  46,  39,  16,  43],
       [ 62,   9,   4,  89,  73],
       [ 42,  25,  94,  29,  96],
       [ 44,  49,  70,  43,  67],
       [ 83,  67,  89,  79,  15],
       [ 54,  47,  15,  28,  69],
       [ 22,  39,  43,  31,  89],
       [ 80,  57,  66,  94,  38],
       [ 88,  67,  17,  61,  26],
       [100,  31,  42,  73,  46],
       [ 27,  88,  66,  61,  90],
       [ 71,  34,  60,  29,  17],
       [ 50,  96,  42,  12,  87]])
```

Create a 2-D visualization using plt.imshow of the numpy matrix with a colorbar. Add a title to your plot. Bonus: Figure out how to change the aspect of the imshow() plot.

```
In [125]: plt.imshow(mat, aspect = 'auto')
```

```
         plt.title('Random Numbers')
         plt.colorbar()
```

```
Out[125]: <matplotlib.colorbar.Colorbar at 0x26471e628d0>
```



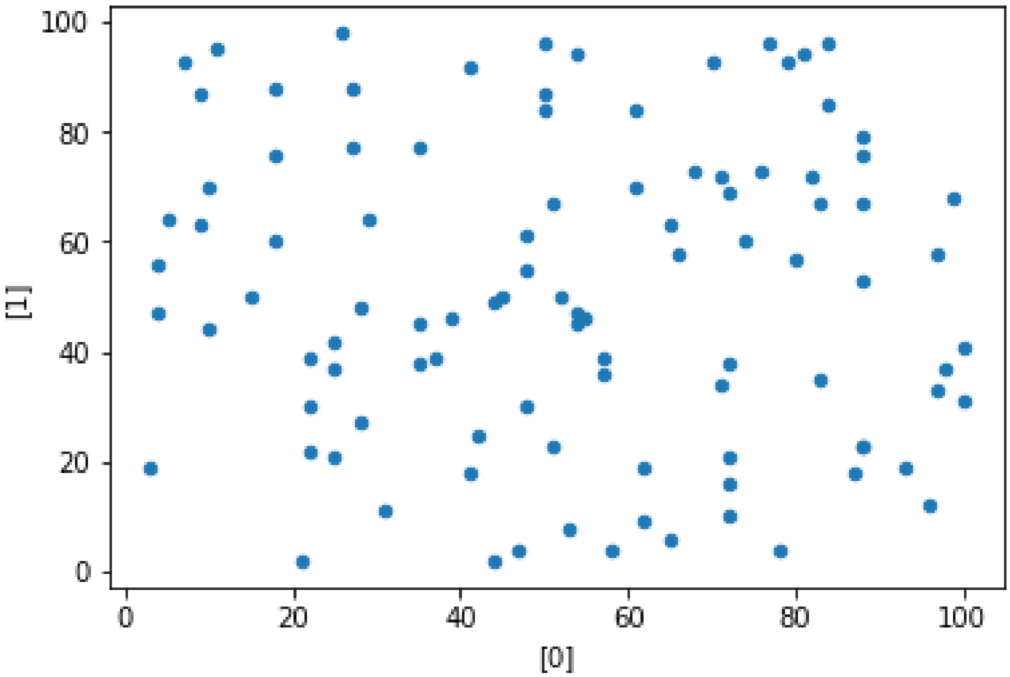
Now use pd.DataFrame() to read in this numpy array as a dataframe. Simple pass in the numpy array into that function to get back a dataframe. Pandas will auto label the columns to 0-4

```
In [126]: # Splitting variables
          df = pd.DataFrame(data = mat)
          X = df[[0, 1, 2]]
          y = df[[3]]
```

Now create a scatter plot using pandas of the 0 column vs the 1 column.

```
In [127]: df.plot(x=[0], y = [1], kind = 'scatter')
```

```
Out[127]: <matplotlib.axes._subplots.AxesSubplot at 0x26471e010f0>
```



Now scale the data to have a minimum of 0 and a maximum value of 1 using scikit-learn.

```
In [133]: from sklearn.preprocessing import MinMaxScaler
          scaler_model = MinMaxScaler()
          scaler_model.fit(df)
```

```
Out[133]: MinMaxScaler(copy=True, feature_range=(0, 1))
```

Using your previously created DataFrame, use df.columns = [...] to rename the pandas columns to be ['f1','f2','f3','f4','label']. Then perform a train/test split with scikitlearn.

```
In [132]: from sklearn.model_selection import train_test_split
          df.columns = ['f1','f2','f3','f4','label']
          X = df[['f1','f2','f3','f4']]
          y = df[['label']]
          X_train,
          X_test,
          y_train,
          y_test = train_test_split(X,y, test_size = .3)
          df.head()
```

```
Out[132]:    f1  f2  f3  f4  label
          0  96  12  82  71     64
          1  88  76  10  78     41
          2   5  64  41  61     93
          3  65   6  13  94     41
          4  50  84   9  30     60
```