C2_W1_lecture_ex_01

August 9, 2020

1 Course 2 week 1 lecture notebook 01

2 Create a Linear Model

2.1 Linear model using scikit-learn

We'll practice using a scikit-learn model for linear regression. You will do something similar in this week's assignment (but with a logistic regression model).

[sklearn.linear_model.LinearRegression()](#)

First, import `LinearRegression`, which is a Python 'class'.

```
[1]: # Import the module 'LinearRegression' from sklearn
     from sklearn.linear_model import LinearRegression
```

Next, use the class to create an object of type LinearRegression.

```
[2]: # Create an object of type LinearRegression
     model = LinearRegression()
     model
```

```
[2]: LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None, normalize=False)
```

Generate some data by importing a module 'load_data', which is implemented for you. The features in 'X' are:

- Age: (years)
- Systolic_BP: Systolic blood pressure (mmHg)
- Diastolic_BP: Diastolic blood pressure (mmHg)
- Cholesterol: (mg/DL)

The labels in `y` indicate whether the patient has a disease (diabetic retinopathy). - y = 1 : patient has retinopathy. - y = 0 : patient does not have retinopathy.

```
[3]: # Import the load_data function from the utils module
     from utils import load_data
```

```
[4]: # Generate features and labels using the imported function
     X, y = load_data(100)
```

Explore the data by viewing the features and the labels

```
[5]: # View the features
     X.head()
```

```
[5]:          Age  Systolic_BP  Diastolic_BP  Cholesterol
     0   77.196340    78.784208     87.026569    82.760275
     1   63.529850   105.171676     83.396113    80.923284
     2   69.003986   117.582259     91.161966    92.915422
     3   82.638210    94.131208     69.470423    95.766098
     4   78.346286   105.385186     87.250583   120.868124
```
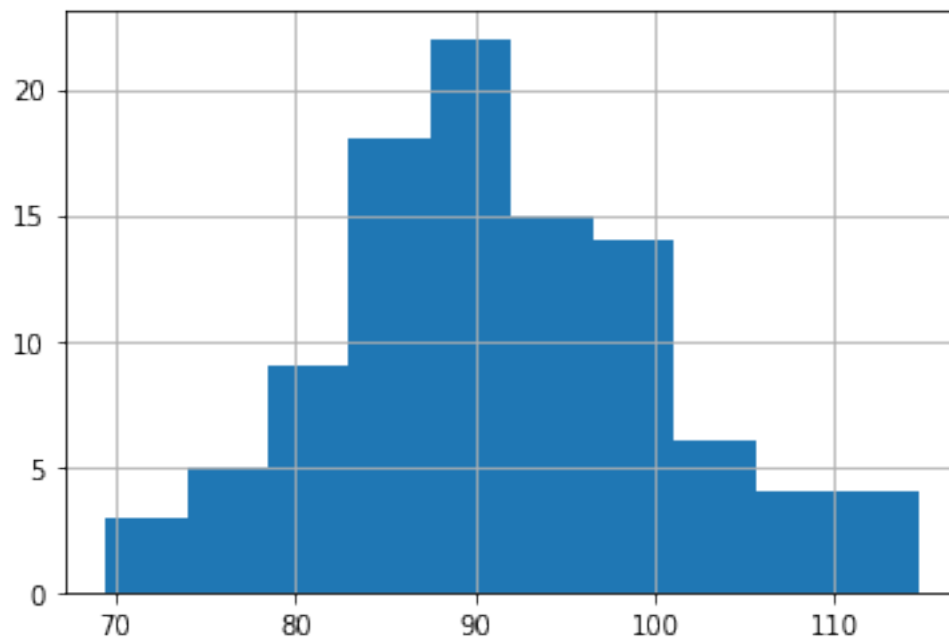
```
[6]: # Plot a histogram of the Age feature
     X['Age'].hist();
```
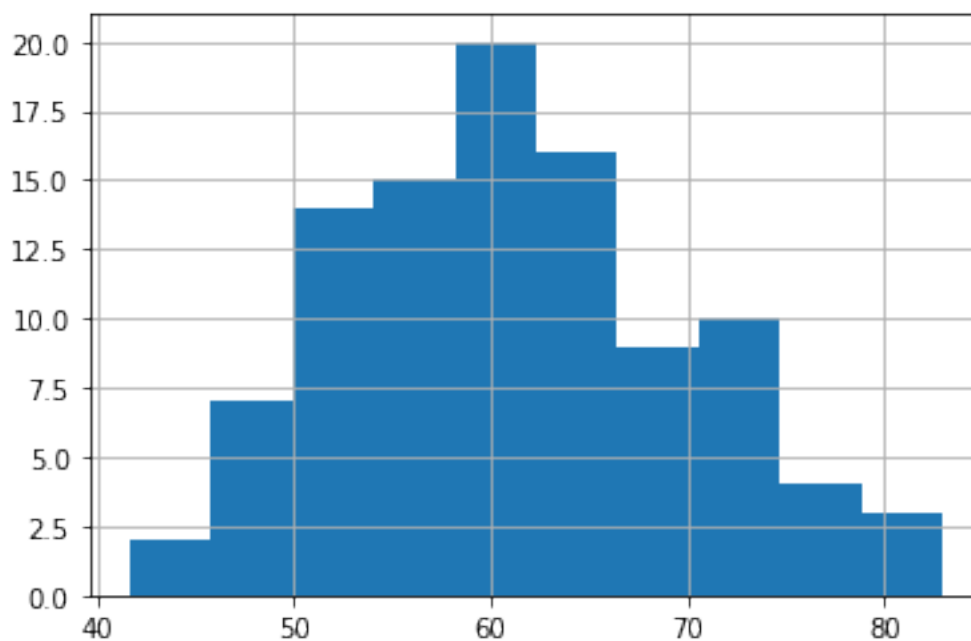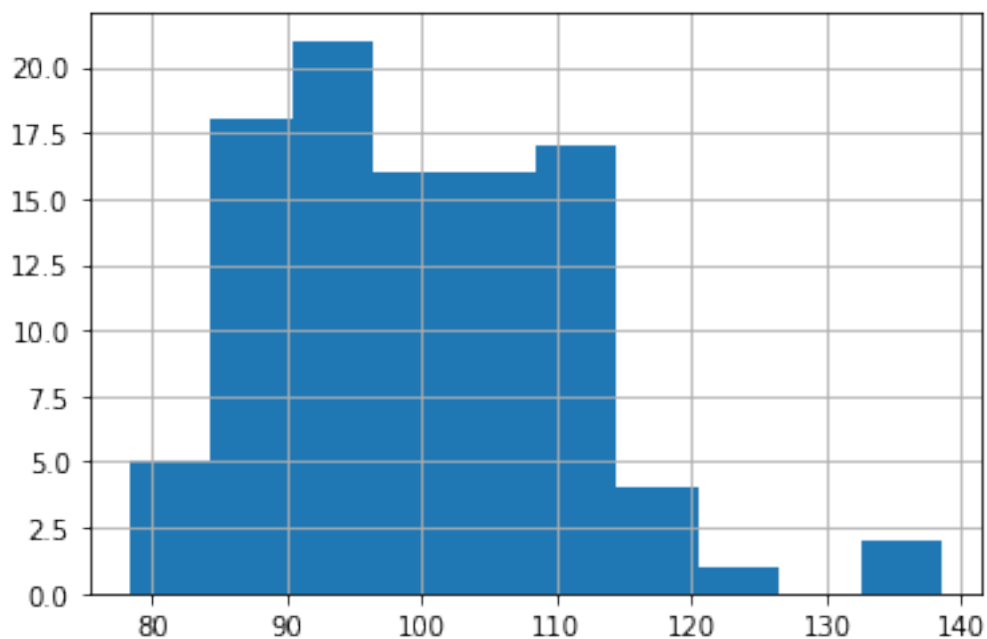


```
[7]: # Plot a histogram of the systolic blood pressure feature
     X['Systolic_BP'].hist();
```
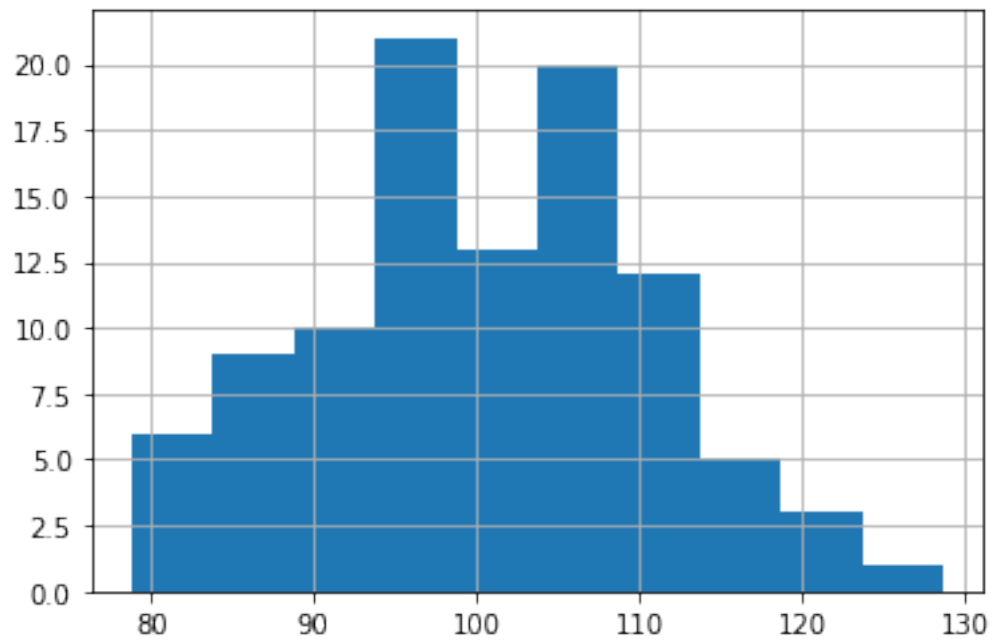
```
[8]:  # Plot a histogram of the diastolic blood pressure feature
      X['Diastolic_BP'].hist();
```

[9]:
```
# Plot a histogram of the cholesterol feature
X['Cholesterol'].hist();
```
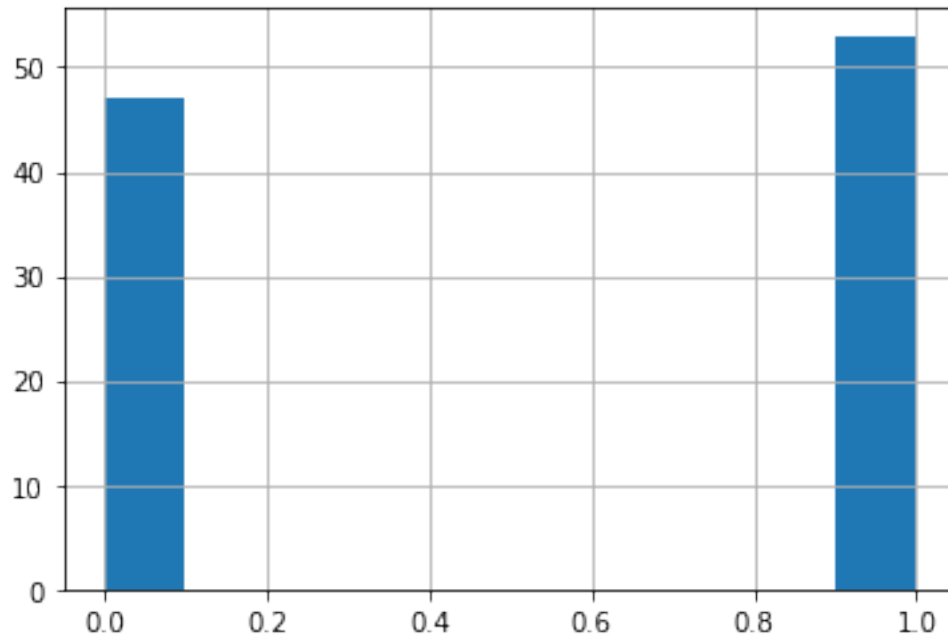

Also take a look at the labels

[10]:
```
# View a few values of the labels
y.head()
```

[10]:
```
0    0.0
1    0.0
2    1.0
3    1.0
4    1.0
Name: y, dtype: float64
```

[11]:
```
# Plot a histogram of the labels
y.hist();
```

Fit the LinearRegression using the features in X and the labels in y. To "fit" the model is another way of saying that we are training the model on the data.

```
[12]: # Fit the linear regression model
      model.fit(X, y)
      model
```

```
[12]: LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None, normalize=False)
```

- View the coefficients of the trained model.
- The coefficients are the 'weights' or βs associated with each feature
- You'll use the coefficients for making predictions.

$$\hat{y} = \beta_1 x_1 + \beta_2 x_2 + ...\beta_N x_N$$

```
[13]: # View the coefficients of the model
      model.coef_
```

```
[13]: array([0.00975155, 0.00835816, 0.00836864, 0.00971064])
```

In the assignment, you will do something similar, but using a logistic regression, so that the output of the prediction will be bounded between 0 and 1.

2.1.1 This is the end of this practice section.

Please continue on with the lecture videos!