LAB3_Part2_StudentUse_Hideki_v2

February 10, 2021

1 LAB3Part2: More Linear Regression with Health Datasets

Datasets: from kaggles https://www.kaggle.com/nareshbhat/health-care-data-set-on-heart-attack-possibility?select=heart.csv

Objective: Trying to figure out the correlation between choosing variables.

Plan: choose continuous values: age, trestbps, chol, thalach, oldpeak. Ignore other variables since they're binary

1.1 1. Import libraries

[1]:
```
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np
import seaborn as sns
```

1.2 2. Import excel data file into pandas data frame

Data Info:

`Attribute Information`

1) age
2) sex
3) cp = chest pain type (4 values)
4) trestbps = resting blood pressure
5) chol = serum cholestoral in mg/dl
6) fbs = fasting blood sugar > 120 mg/dl
7) restecg = resting electrocardiographic results (values 0,1,2)
8) thalach = maximum heart rate achieved

9) exang = exercise induced angina

10) oldpeak = ST depression induced by exercise relative to rest

11) slope = the slope of the peak exercise ST segment

12) ca = number of major vessels (0-3) colored by flourosopy

13) thal: 0 = normal; 1 = fixed defect; 2 = reversable defect

14) target: 0= less chance of heart attack 1= more chance of heart attack

[3]:
```
# df = pd.read_csv("health2.csv") # I recommand you using this one just␣
↪uncomment
```

[2]:
```
# to check dataframe, use display()
```

1.3 3. Data Cleaning

There might be a possiblity that the data is missing its values. use "print(df.isnull().sum))" to check if the data is ready to be processed.

[4]:
```
print(df.isnull().sum())
```

```
age         0
sex         0
cp          0
trestbps    0
chol        0
fbs         0
restecg     0
thalach     0
exang       0
oldpeak     0
slope       0
ca          0
thal        0
target      0
dtype: int64
```

1.4 4. Feature Selection

Now that the data is good to go, we are ready to move on to the next step of the process. As there are 14 features in the dataset, we do not want to use all of these features for training our model, because not all of them are relevant. Instead, we want to choose those features that directly influence the result (that is, prices of houses) to train the model. For this, we can use the corr() function. The corr() function computes the pairwise correlation of columns:

[5]:
```
corr = df.corr()
display(corr)
```

```
                age       sex        cp  trestbps      chol       fbs  \
age        1.000000 -0.098447 -0.068653  0.279351  0.213678  0.121308
sex       -0.098447  1.000000 -0.049353 -0.056769 -0.197912  0.045032
cp        -0.068653 -0.049353  1.000000  0.047608 -0.076904  0.094444
trestbps   0.279351 -0.056769  0.047608  1.000000  0.123174  0.177531
chol       0.213678 -0.197912 -0.076904  0.123174  1.000000  0.013294
fbs        0.121308  0.045032  0.094444  0.177531  0.013294  1.000000
restecg   -0.116211 -0.058196  0.044421 -0.114103 -0.151040 -0.084189
thalach   -0.398522 -0.044020  0.295762 -0.046698 -0.009940 -0.008567
exang      0.096801  0.141664 -0.394280  0.067616  0.067023  0.025665
oldpeak    0.210013  0.096093 -0.149230  0.193216  0.053952  0.005747
slope     -0.168814 -0.030711  0.119717 -0.121475 -0.004038 -0.059894
ca         0.276326  0.118261 -0.181053  0.101389  0.070511  0.137979
thal       0.068001  0.210041 -0.161736  0.062210  0.098803 -0.032019
target    -0.225439 -0.280937  0.433798 -0.144931 -0.085239 -0.028046

            restecg   thalach     exang   oldpeak     slope        ca  \
age       -0.116211 -0.398522  0.096801  0.210013 -0.168814  0.276326
sex       -0.058196 -0.044020  0.141664  0.096093 -0.030711  0.118261
cp         0.044421  0.295762 -0.394280 -0.149230  0.119717 -0.181053
trestbps  -0.114103 -0.046698  0.067616  0.193216 -0.121475  0.101389
chol      -0.151040 -0.009940  0.067023  0.053952 -0.004038  0.070511
fbs       -0.084189 -0.008567  0.025665  0.005747 -0.059894  0.137979
restecg    1.000000  0.044123 -0.070733 -0.058770  0.093045 -0.072042
thalach    0.044123  1.000000 -0.378812 -0.344187  0.386784 -0.213177
exang     -0.070733 -0.378812  1.000000  0.288223 -0.257748  0.115739
oldpeak   -0.058770 -0.344187  0.288223  1.000000 -0.577537  0.222682
slope      0.093045  0.386784 -0.257748 -0.577537  1.000000 -0.080155
ca        -0.072042 -0.213177  0.115739  0.222682 -0.080155  1.000000
thal      -0.011981 -0.096439  0.206754  0.210244 -0.104764  0.151832
target     0.137230  0.421741 -0.436757 -0.430696  0.345877 -0.391724

              thal    target
age       0.068001 -0.225439
sex       0.210041 -0.280937
cp       -0.161736  0.433798
trestbps  0.062210 -0.144931
chol      0.098803 -0.085239
fbs      -0.032019 -0.028046
restecg  -0.011981  0.137230
thalach  -0.096439  0.421741
exang     0.206754 -0.436757
oldpeak   0.210244 -0.430696
slope    -0.104764  0.345877
```

```
ca          0.151832 -0.391724
thal        1.000000 -0.344029
target     -0.344029  1.000000
```

1.5 Since "thalach" and "trestbps" have high correlation values, so we will still use these two features to train our model to predict the variable "age"

[6]:
```
#---get the top 3 features that has the highest correlation---
#select "Age" to see which variables has a strong correlation with age.

print(df.corr().abs().nlargest(3, 'age').index)

#---print the top 3 correlation values---
print(df.corr().abs().nlargest(3, 'age').values[:,13])
```

```
Index(['age', 'thalach', 'trestbps'], dtype='object')
[0.22543872 0.42174093 0.14493113]
```

2 Multiple Regression

2.1 5.1 plot a scatter plot showing the relationship between the "age" and "thalach" label:

hint: Figure6.4 from kvoval ch6

[ ]:
```

```

2.2 5.2 Let's also plot a scatter plot showing the relationship between the "age" feature and the "trestbps" label:

hint: Just change the variables

- Side Note: Using sns.regplot(x-value, y-value, ci = None) will give a better plotting with a line.

[9]:
```
#sns.regplot(df['age'],df['trestbps'], ci=None) # uncomment and see the result
```

fig2: Scatter plot showing the relationship between "age" and "trestbps"

[10]:
```
#sns.regplot(df['age'],df['chol'], ci=None) # uncomment and see the result
```

fig3: Scatter plot showing the relationship between "age" and "chol"

2.3 5.3 let's plot the two features and the label on a 3D chart:

hint: Figure6.6 from knovel ch6

[ ]:
```

```

3 Training the Model

We can now train the model. First, create two DataFrames: x and Y. The x DataFrame will contain the combination of the thalach and trestbps features, while the Y DataFrame will contain the age label:

[ ]:

3.1 6.1 Create DataFrames: x and Y

We will split the dataset into 70 percent for training and 30 percent for testing:

Once the model is trained, we will use the testing set to perform some predictions:

[ ]:

3.2 6.2 Find a R-Squared Value

To learn how well our model performed, we use the R-Squared method that you learned in the previous chapter. The R-Squared method lets you know how close the test data fits the regression line. A value of 1.0 means a perfect fit. So, you aim for a value of R-Squared that is close to 1:

[ ]:

3.3 6.3 Plot a scatter plot showing the

[ ]:

3.4 6.4 Getting the Intercept and Coefficients

[ ]:

4 Plotting the 3D Hyperplane

4.1 7. Plot the 3D graph

hint: Figure6.8

[ ]:

5 Polynomial Regression

In the previous section, you saw how to apply linear regression to predict the prices of houses in the Boston area. While the result is somewhat acceptable, it is not very accurate. This is because sometimes a linear regression line might not be the best solution to capture the relationships between the features and label accurately. In some cases, a curved line might do better.

[1]:
```
#display(df)
#df = pd.read_csv("health2.csv") # I'd recommend to reload the datasets since␣
↪variable has changed.
```

5.1 8.1 Plot the points of “age” and “thalach”

Using linear regression, you can try to plot a straight line cutting through most of the points:

[ ]:

5.2 8.2 Try to plot a straight line cutting through most of the points:

hint: Figure6.11

[ ]:

5.3 8.3 Plot Polynomial Regression in Scikit-learn

hint: https://www.w3schools.com/python/python_ml_polynomial_regression.asp
knovel did not work well in this dataset. So I recommend you using from diffrent resource I put it above

[ ]:

5.4 8.4 Find R-Squared Value

[ ]:

5.4.1 Did the result: indicates good or bad relationship? and put a possible reason you can think of.

6 Polynomial Multiple Regression

6.1 9.1 Find R-Squared, model intercept and coefficient

[ ]:

6.2 9.2 Plot the 3D Hyperplane

hint: Figure 6.20

[ ]:

7 GOOD JOB!! :)

[ ]: