Physical Activities and BMI

Anabel Hoang

Hypothesis

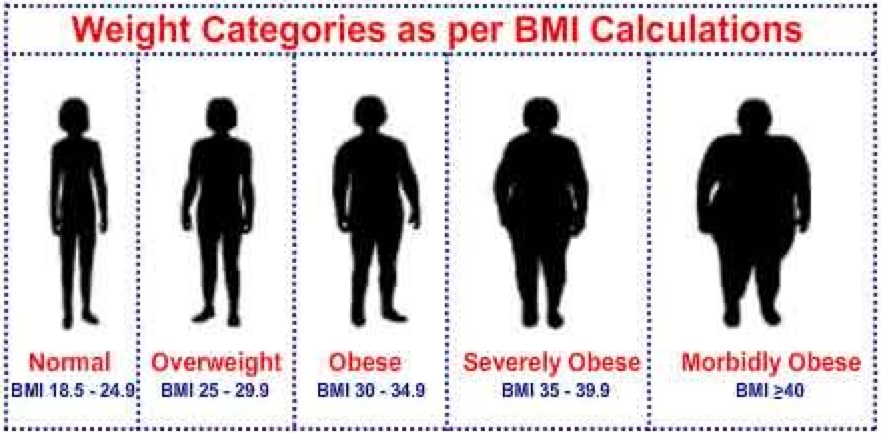
- Low physical activities level is correlated to higher BMI

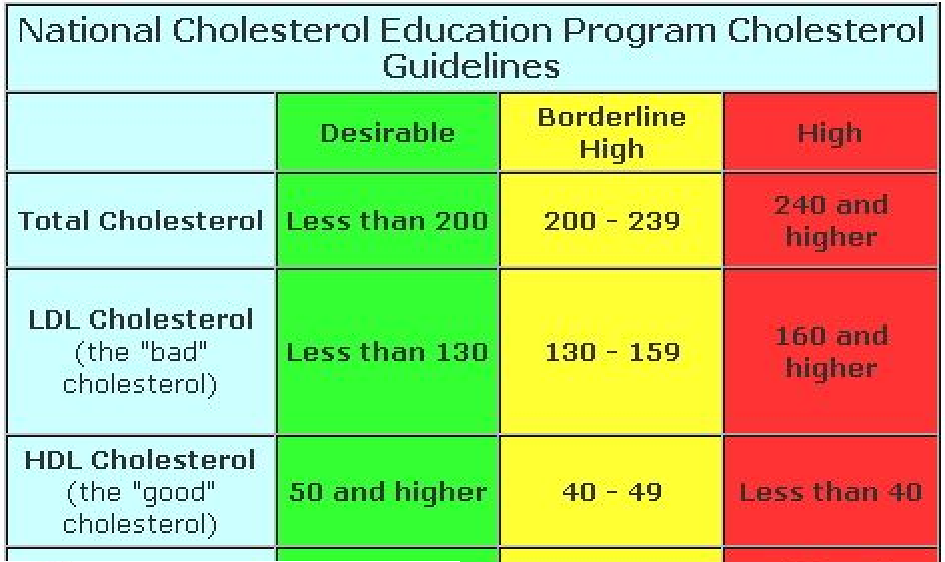
National Health and Nutrition Examination Survey (NHANES)

- Identification Code 1 – 6482 ID
- Gender 0 = Male, 1 = Female GENDER
- Age at Screening Years AGE
- Marital Status 1 = Married, 2 = Widowed, 3 = Divorced, 4= Separated, 5 = Never Married, 6 = Living Together MARSTAT
- Statistical Weight 4084.478 – 153810.3 SAMPLEWT
- Pseudo-PSU 1, 2 PSU
- Pseudo-Stratum 1 – 15 STRATA
- Total Cholesterol mg/dL TCHOL
- HDL-Cholesterol mg/dL HDL
- Systolic Blood Pressure mm Hg SYSBP
- Diastolic Blood Pressure mm Hg DBP
- Weight kg WT
- Standing Height cm HT
- Body mass Index Kg/m^2 BMI
- Vigorous Work Activity 0 = Yes, 1 = No VIGWRK
- Moderate Work Activity 0 = Yes, 1 = No MODWRK
- Walk or Bicycle 0 = Yes, 1 = No WLKBIK
- Vigorous Recreational Activities 0 = Yes, 1 = No VIGRECEXR
- Moderate Recreational Activities 0 = Yes, 1 = No MODRECEXR
- Minutes of Sedentary Activity per Week Minutes SEDMIN
- BMI>35 0 = No, 1 = Yes OBESE

6,482 X 21

Background

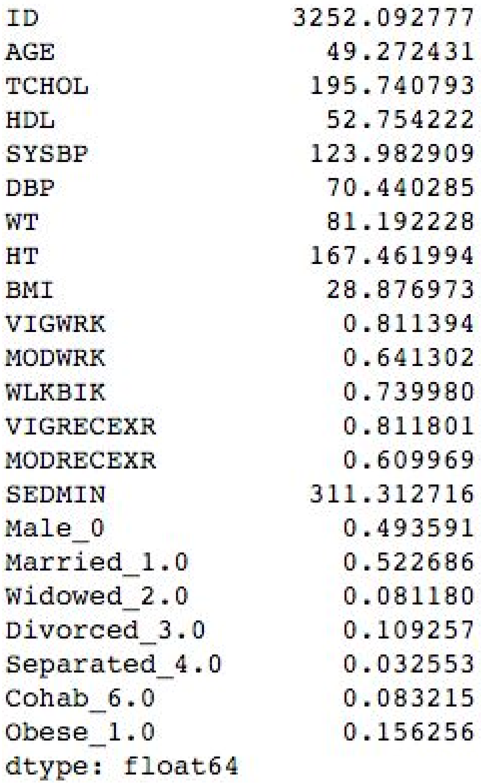




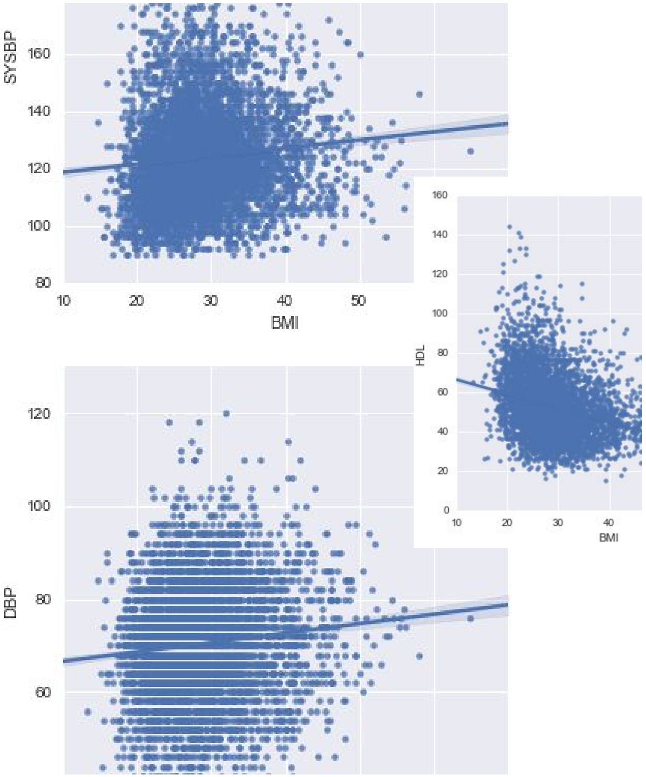
National Cholesterol Education Program Cholesterol Guidelines

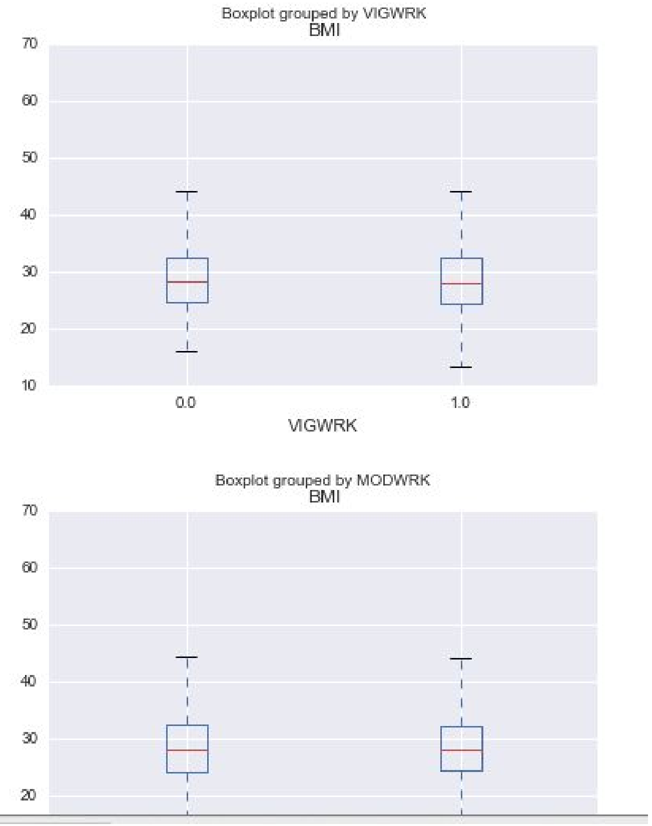
	Desirable	Borderline High	High
Total Cholesterol	Less than 200	200 - 239	240 and higher
LDL Cholesterol (the "bad" cholesterol)	Less than 130	130 - 159	160 and higher
HDL Cholesterol (the "good" cholesterol)	50 and higher	40 - 49	Less than 40
	00	200 - 399	400 and higher

Exploratory Data Analysis



```
ID              3252.092777
AGE               49.272431
TCHOL            195.740793
HDL               52.754222
SYSBP            123.982909
DBP               70.440285
WT                81.192228
HT               167.461994
BMI               28.876973
VIGWRK             0.811394
MODWRK             0.641302
WLKBIK             0.739980
VIGRECEXR          0.811801
MODRECEXR          0.609969
SEDMIN           311.312716
Male_0             0.493591
Married_1.0        0.522686
Widowed_2.0        0.081180
Divorced_3.0       0.109257
Separated_4.0      0.032553
Cohab_6.0          0.083215
Obese_1.0          0.156256
dtype: float64
```

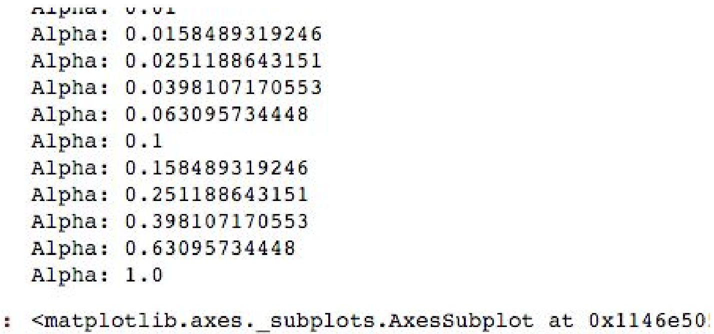




Models

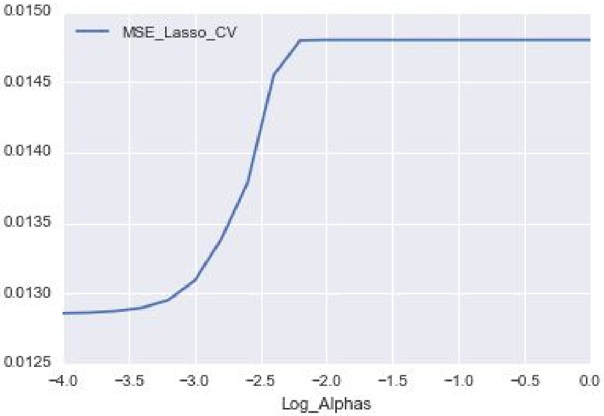
- Lasso Regression
- Logistic Regression
- Voting Classifier

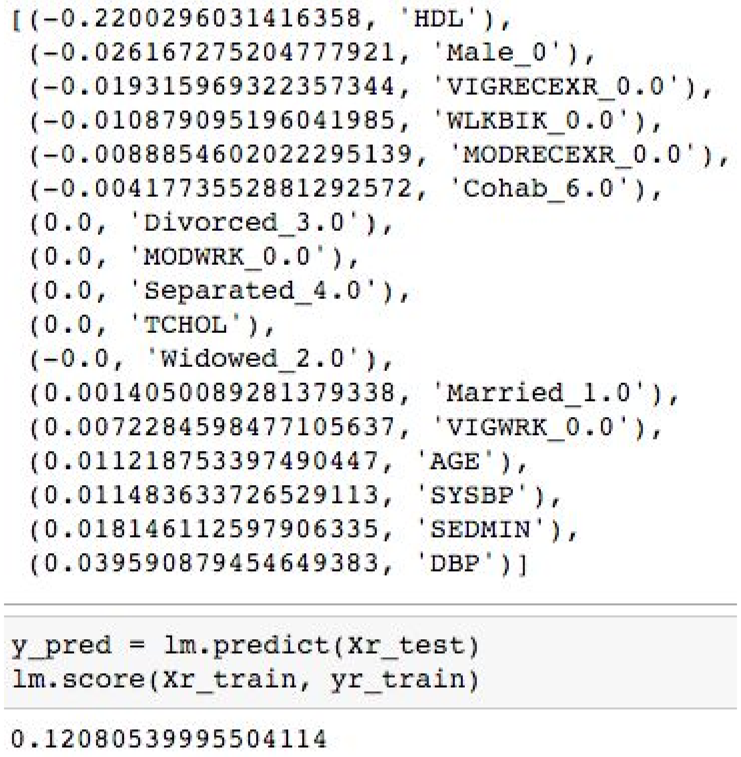
Lasso Regression



```
Alpha: 0.01
Alpha: 0.0158489319246
Alpha: 0.0251188643151
Alpha: 0.0398107170553
Alpha: 0.063095734448
Alpha: 0.1
Alpha: 0.158489319246
Alpha: 0.251188643151
Alpha: 0.398107170553
Alpha: 0.63095734448
Alpha: 1.0
```

```
: <matplotlib.axes._subplots.AxesSubplot at 0x1146e50
```





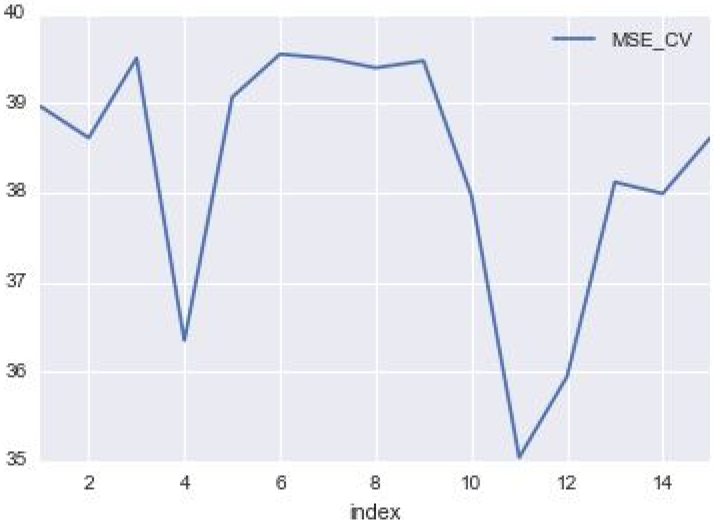
```
[(-0.2200296031416358, 'HDL'),
 (-0.026167275204777921, 'Male_0'),
 (-0.019315969322357344, 'VIGRECEXR_0.0'),
 (-0.010879095196041985, 'WLKBIK_0.0'),
 (-0.0088854602022295139, 'MODRECEXR_0.0'),
 (-0.0041773552881292572, 'Cohab_6.0'),
 (0.0, 'Divorced_3.0'),
 (0.0, 'MODWRK_0.0'),
 (0.0, 'Separated_4.0'),
 (0.0, 'TCHOL'),
 (-0.0, 'Widowed_2.0'),
 (0.0014050089281379338, 'Married_1.0'),
 (0.0072284598477105637, 'VIGWRK_0.0'),
 (0.011218753397490447, 'AGE'),
 (0.011483633726529113, 'SYSBP'),
 (0.018146112597906335, 'SEDMIN'),
 (0.039590879454649383, 'DBP')]
```

```
y_pred = lm.predict(Xr_test)
lm.score(Xr_train, yr_train)
```

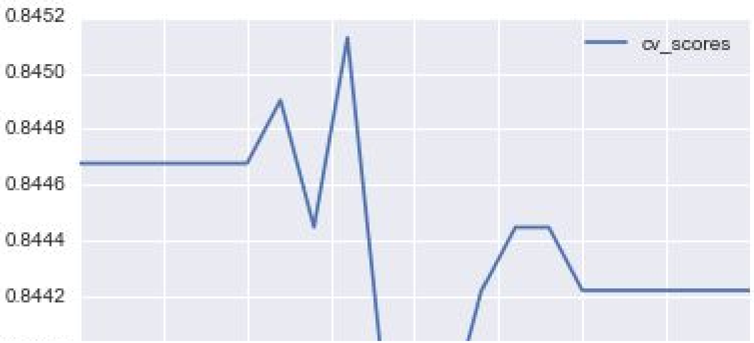
```
0.12080539995504114
```

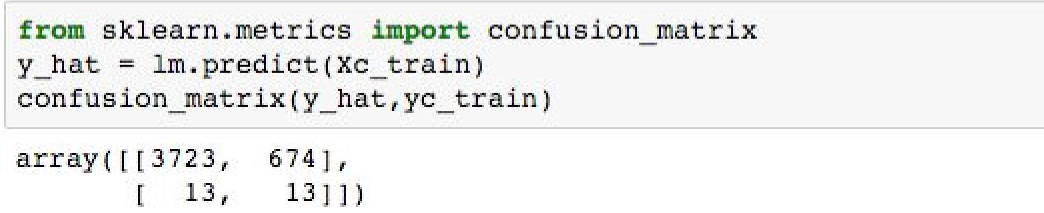
Lasso Regression

X11 = NewNew[['TCHOL','HDL','SYSBP','DBP','Male_0']]



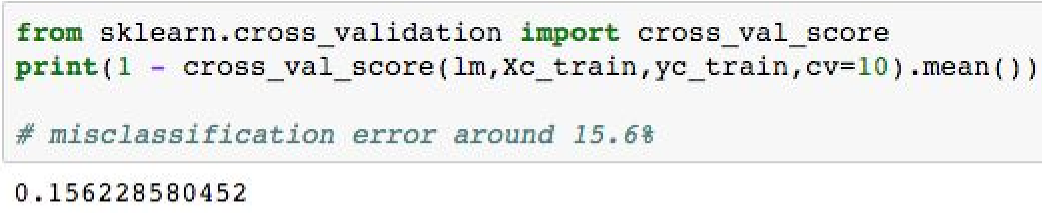
Logistic Regression





```
from sklearn.metrics import confusion_matrix
y_hat = lm.predict(Xc_train)
confusion_matrix(y_hat,yc_train)
```

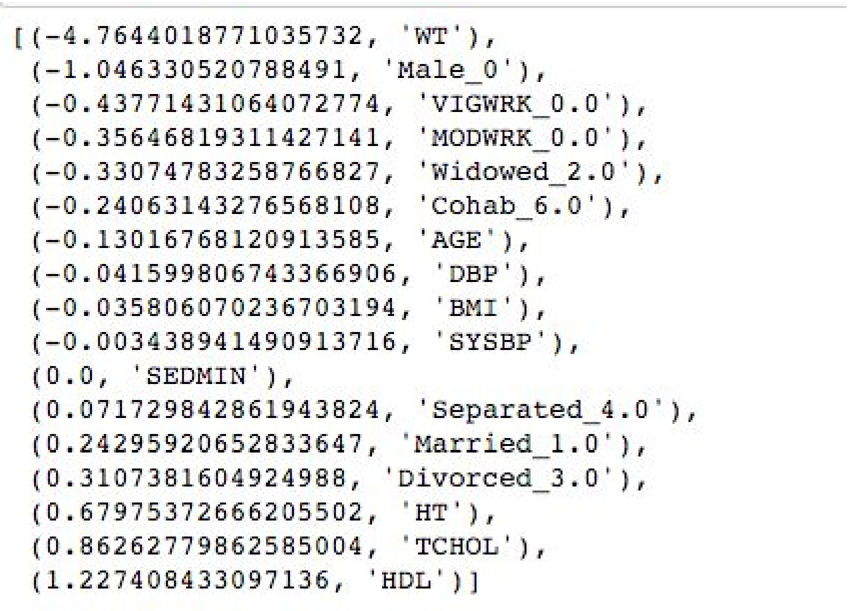
```
array([[3723,  674],
       [  13,   13]])
```



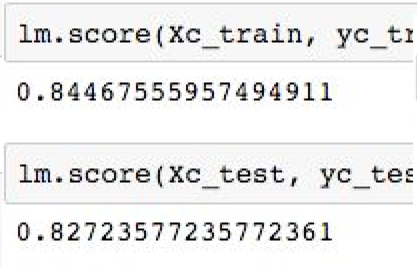
```
from sklearn.cross_validation import cross_val_score
print(1 - cross_val_score(lm,Xc_train,yc_train,cv=10).mean())

# misclassification error around 15.6%
```

```
0.156228580452
```



```
[(-4.7644018771035732, 'WT'),
 (-1.046330520788491, 'Male_0'),
 (-0.43771431064072774, 'VIGWRK_0.0'),
 (-0.35646819311427141, 'MODWRK_0.0'),
 (-0.33074783258766827, 'Widowed_2.0'),
 (-0.24063143276568108, 'Cohab_6.0'),
 (-0.13016768120913585, 'AGE'),
 (-0.041599806743366906, 'DBP'),
 (-0.035806070236703194, 'BMI'),
 (-0.0034389414909137716, 'SYSBP'),
 (0.0, 'SEDMIN'),
 (0.071729842861943824, 'Separated_4.0'),
 (0.24295920652833647, 'Married_1.0'),
 (0.31073816049249888, 'Divorced_3.0'),
 (0.67975372666205502, 'HT'),
 (0.86262779862585004, 'TCHOL'),
 (1.227408433097136, 'HDL')]
```

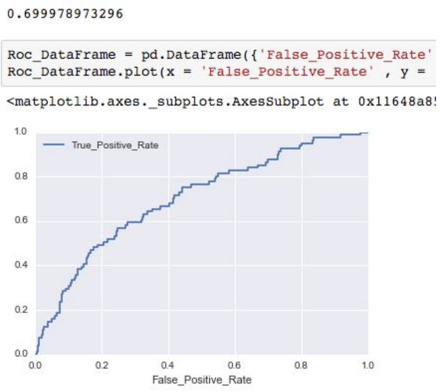


```
lm.score(Xc_train, yc_tr
```

```
0.84467555957494911
```

```
lm.score(Xc_test, yc_tes
```

```
0.82723577235772361
```

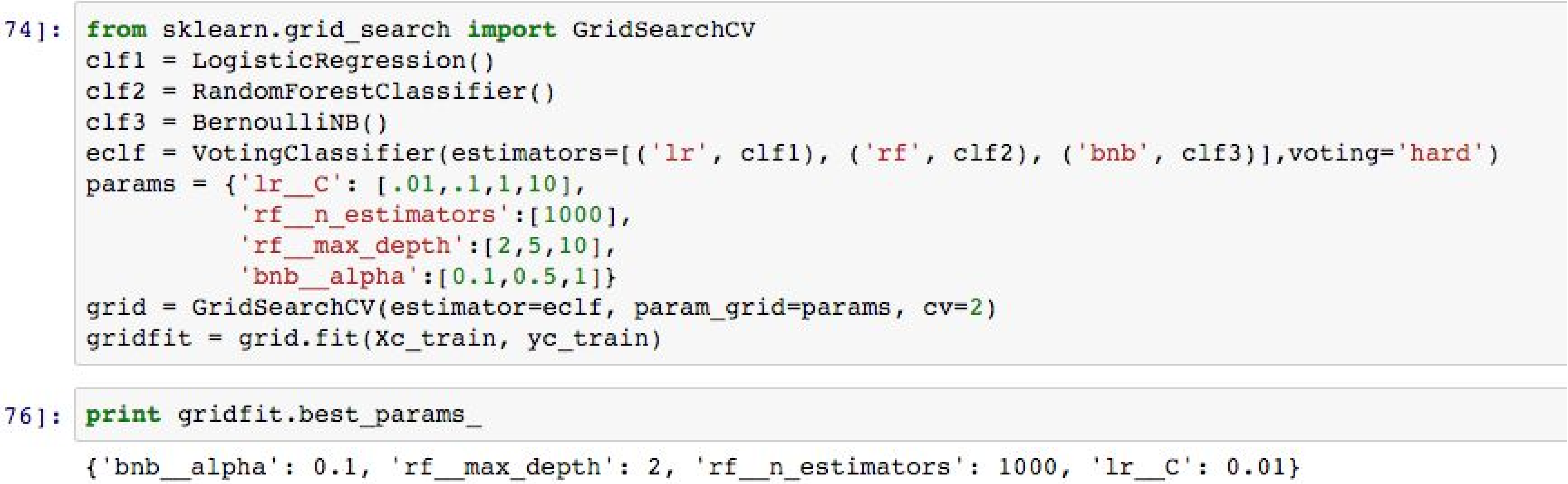


```
0.699978973296
```

```
Roc_DataFrame = pd.DataFrame({'False_Positive_Rate'
Roc_DataFrame.plot(x = 'False_Positive_Rate' , y =
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x11648a8!
```

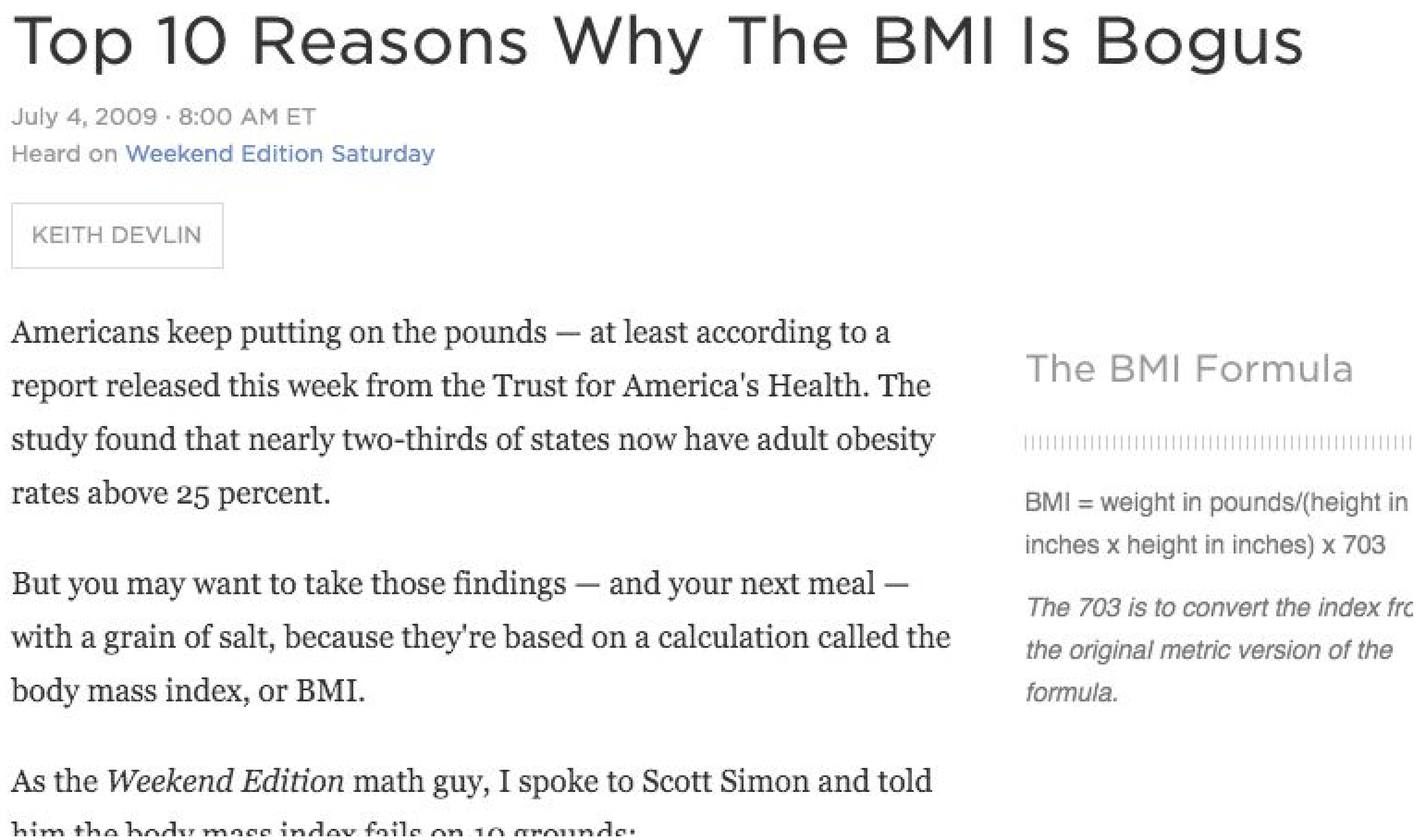
Ensemble Method



```
74]: from sklearn.grid_search import GridSearchCV
     clf1 = LogisticRegression()
     clf2 = RandomForestClassifier()
     clf3 = BernoulliNB()
     eclf = VotingClassifier(estimators=[('lr', clf1), ('rf', clf2), ('bnb', clf3)],voting='hard')
     params = {'lr__C': [.01,.1,1,10],
               'rf__n_estimators':[1000],
               'rf__max_depth':[2,5,10],
               'bnb__alpha':[0.1,0.5,1]}
     grid = GridSearchCV(estimator=eclf, param_grid=params, cv=2)
     gridfit = grid.fit(Xc_train, yc_train)
```

```
76]: print gridfit.best_params_

     {'bnb__alpha': 0.1, 'rf__max_depth': 2, 'rf__n_estimators': 1000, 'lr__C': 0.01}
```



Top 10 Reasons Why The BMI Is Bogus

July 4, 2009 · 8:00 AM ET
Heard on Weekend Edition Saturday

KEITH DEVLIN

Americans keep putting on the pounds — at least according to a report released this week from the Trust for America's Health. The study found that nearly two-thirds of states now have adult obesity rates above 25 percent.

But you may want to take those findings — and your next meal — with a grain of salt, because they're based on a calculation called the body mass index, or BMI.

As the *Weekend Edition* math guy, I spoke to Scott Simon and told him the body mass index fails on 10 grounds:

The BMI Formula

BMI = weight in pounds/(height in inches x height in inches) x 703

The 703 is to convert the index fro the original metric version of the formula.

Next Steps

- Add additional features/ data
- Find different measures of obesity and compare models

Thank you!