

We can get the break-even $f(x)$ to be $\frac{L_{FP}}{L_{FN}+L_{FP}}$, so \hat{P} is $\frac{L_{FP}}{L_{FN}+L_{FP}}$

If we try to the probability of $Y=1$ is below \hat{P} , then we should predict it to be $Y=0$, otherwise we should predict it to be $Y=1$

Problem 3

In [1]:

```
# Train Logistic Regression model using training set data.
import numpy as np
import pandas as pd
from plotnine import ggplot

fram_train = pd.read_csv('framingham_train.csv')
fram_test = pd.read_csv('framingham_test.csv')
fram_train.head()
print(fram_train.info)
fram_no = fram_train[fram_train['TenYearCHD'] == 0]
fram_yes = fram_train[fram_train['TenYearCHD'] == 1]
print("No Disease: " + str(len(fram_no)))
print("Yes Disease: " + str(len(fram_yes)))

#Baseline training set prediction: Always No Disease
#Baseline accuracy:
base_accur = len(fram_no)/len(fram_train)
print(base_accur)
```

```
<bound method DataFrame.info of          male  age      education  currentSmoker  cigsPerDay  BPMeds  \
0          0  41  Some high school          1          43          0
1          0  38  High school/GED          1          15          0
2          1  42  High school/GED          1           9          0
3          0  42  High school/GED          0           0          0
4          0  53  High school/GED          0           0          0
...      ...  ...      ...      ...      ...      ...
2555         1  39  Some high school          1          40          0
2556         0  52  Some high school          0           0          0
2557         1  39  Some high school          0           0          0
2558         1  49  High school/GED          1          50          0
2559         0  64  Some high school          0           0          0

prevalentStroke  prevalentHyp  diabetes  totChol  sysBP  diaBP  BMI  \
0                0            1         0      306  199.0  106.0  38.75
1                0            0         0      176  110.0   80.0  24.03
```

2	0	0	0	205	110.0	73.0	22.40
3	0	1	0	263	150.0	88.0	23.68
4	0	1	0	272	146.0	89.0	25.50
...
2555	0	0	0	209	134.0	82.0	28.34
2556	0	0	0	292	125.0	87.0	31.92
2557	0	0	0	188	105.0	65.0	22.85
2558	0	0	0	252	156.0	91.0	25.35
2559	0	0	0	280	127.0	77.0	30.39

	heartRate	glucose	TenYearCHD
0	100	75	0
1	100	113	0
2	61	66	0
3	96	78	0
4	73	67	0
...
2555	70	75	0
2556	75	67	0
2557	63	76	0
2558	70	114	1
2559	56	78	1

```
[2560 rows x 16 columns]>
No Disease: 2178
Yes Disease: 382
0.85078125
```

i) The Logistic Regression Model on Training Set

The fitted logistic regression is

$$y = \frac{1}{1 + \exp(-(-9.274 - 0.1053 * eduHigh / GED - 0.1025 * eduCol + 0.061 * eduHigh + 0.56 * male + 0.07 * age + 0.15 * currSmoker + 0.0 * prevStroke + 0.2075 * prevHyp - 0.2975 * diabetes + 0.002 * totChol + 0.0181 * sysBP - 0.0045 * diaBP + 0.0136 * BMI - 0.0046 * h\epsilon$$

In [2]:

```
import statsmodels.formula.api as smf

logreg = smf.logit(formula = 'TenYearCHD ~ male + age + education + currentSmoker + cigsPerDay + BPMeds + prevalentStroke',
                    data = fram_train).fit()

print(logreg.summary())
```

```
corr = fram_train.corr()
print(corr)
import seaborn as sns
sns.heatmap(corr, xticklabels = corr.columns.values, yticklabels=corr.columns.values)
```

Optimization terminated successfully.

Current function value: 0.365281

Iterations 7

Logit Regression Results

```
=====
Dep. Variable:          TenYearCHD    No. Observations:          2560
Model:                  Logit         Df Residuals:              2542
Method:                 MLE          Df Model:                  17
Date:                  Tue, 05 Oct 2021    Pseudo R-squ.:           0.1331
Time:                  17:29:41          Log-Likelihood:          -935.12
converged:              True            LL-Null:                 -1078.7
Covariance Type:        nonrobust        LLR p-value:             5.181e-51
=====
```

	coef	std err	z	P> z	[0.025	0.975]
Intercept	-9.2740	0.882	-10.516	0.000	-11.002	-7.546
education[T.High school/GED]	-0.1053	0.217	-0.485	0.628	-0.531	0.321
education[T.Some college/vocational school]	-0.1025	0.241	-0.425	0.671	-0.575	0.370
education[T.Some high school]	0.0610	0.202	0.302	0.762	-0.334	0.456
male	0.5621	0.134	4.189	0.000	0.299	0.825
age	0.0689	0.008	8.303	0.000	0.053	0.085
currentSmoker	0.1539	0.189	0.816	0.415	-0.216	0.524
cigsPerDay	0.0155	0.007	2.077	0.038	0.001	0.030
BPMeds	0.1528	0.281	0.544	0.587	-0.398	0.704
prevalentStroke	0.8209	0.570	1.441	0.150	-0.296	1.938
prevalentHyp	0.2075	0.167	1.246	0.213	-0.119	0.534
diabetes	-0.2975	0.395	-0.753	0.452	-1.072	0.477
totChol	0.0020	0.001	1.445	0.148	-0.001	0.005
sysBP	0.0181	0.005	3.900	0.000	0.009	0.027
diaBP	-0.0045	0.008	-0.584	0.560	-0.020	0.011
BMI	0.0136	0.016	0.867	0.386	-0.017	0.044
heartRate	-0.0046	0.005	-0.888	0.374	-0.015	0.006
glucose	0.0096	0.003	3.439	0.001	0.004	0.015

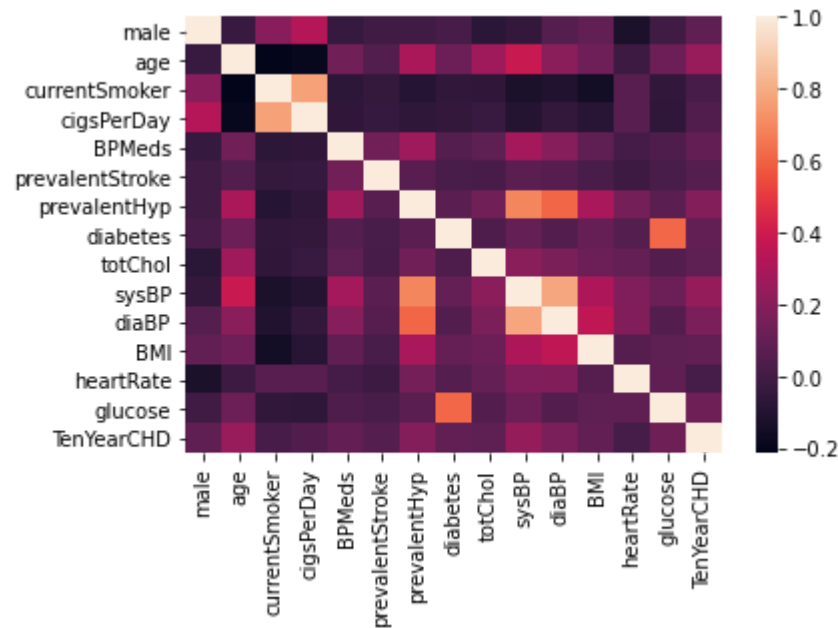
```
=====
                    male      age  currentSmoker  cigsPerDay  BPMeds  \
male              1.000000 -0.029909      0.204207   0.327397 -0.041037
age              -0.029909  1.000000     -0.212590  -0.189122  0.136687
currentSmoker    0.204207 -0.212590      1.000000   0.769880 -0.069135
cigsPerDay       0.327397 -0.189122      0.769880   1.000000 -0.057161
BPMeds          -0.041037  0.136687     -0.069135  -0.057161  1.000000
prevalentStroke -0.006701  0.046245     -0.043744  -0.040217  0.134277
prevalentHyp    -0.005309  0.302459     -0.093215  -0.062643  0.264434
=====
```

diabetes	0.018548	0.123349	-0.054224	-0.049984	0.056487
totChol	-0.076278	0.265056	-0.056851	-0.028286	0.081220
sysBP	-0.046821	0.388899	-0.123977	-0.097569	0.280192
diaBP	0.056319	0.208740	-0.105782	-0.051435	0.196694
BMI	0.094351	0.130222	-0.154004	-0.079980	0.086536
heartRate	-0.127251	-0.014015	0.063123	0.070557	0.008832
glucose	-0.005773	0.122493	-0.056205	-0.065092	0.037385
TenYearCHD	0.087473	0.249548	0.014960	0.042388	0.096434

	prevalentStroke	prevalentHyp	diabetes	totChol	sysBP	\
male	-0.006701	-0.005309	0.018548	-0.076278	-0.046821	
age	0.046245	0.302459	0.123349	0.265056	0.388899	
currentSmoker	-0.043744	-0.093215	-0.054224	-0.056851	-0.123977	
cigsPerDay	-0.040217	-0.062643	-0.049984	-0.028286	-0.097569	
BPMeds	0.134277	0.264434	0.056487	0.081220	0.280192	
prevalentStroke	1.000000	0.069614	0.019806	0.016979	0.071750	
prevalentHyp	0.069614	1.000000	0.076278	0.140809	0.695793	
diabetes	0.019806	0.076278	1.000000	0.037066	0.101584	
totChol	0.016979	0.140809	0.037066	1.000000	0.206660	
sysBP	0.071750	0.695793	0.101584	0.206660	1.000000	
diaBP	0.058465	0.607500	0.048151	0.169206	0.783140	
BMI	0.022100	0.292661	0.106759	0.119180	0.307599	
heartRate	-0.014379	0.148294	0.054499	0.104704	0.177102	
glucose	0.020678	0.073367	0.614267	0.049336	0.119062	
TenYearCHD	0.054035	0.190281	0.090978	0.085381	0.239592	

	diaBP	BMI	heartRate	glucose	TenYearCHD
male	0.056319	0.094351	-0.127251	-0.005773	0.087473
age	0.208740	0.130222	-0.014015	0.122493	0.249548
currentSmoker	-0.105782	-0.154004	0.063123	-0.056205	0.014960
cigsPerDay	-0.051435	-0.079980	0.070557	-0.065092	0.042388
BPMeds	0.196694	0.086536	0.008832	0.037385	0.096434
prevalentStroke	0.058465	0.022100	-0.014379	0.020678	0.054035
prevalentHyp	0.607500	0.292661	0.148294	0.073367	0.190281
diabetes	0.048151	0.106759	0.054499	0.614267	0.090978
totChol	0.169206	0.119180	0.104704	0.049336	0.085381
sysBP	0.783140	0.307599	0.177102	0.119062	0.239592
diaBP	1.000000	0.359543	0.183339	0.052196	0.162136
BMI	0.359543	1.000000	0.058991	0.083668	0.090559
heartRate	0.183339	0.058991	1.000000	0.088158	0.013079
glucose	0.052196	0.083668	0.088158	1.000000	0.126177
TenYearCHD	0.162136	0.090559	0.013079	0.126177	1.000000

Out[2]: <AxesSubplot:>



ii)

The variables that will have significant impact on the probability of getting CHD are 'male', 'age', 'cigsPerDay', 'sysBP' and 'glucose'.

The coefficient of 'male' is 0.5621, which means that holding other variables constant, if a person is male, his odds of getting CHD within 10 years would multiply by $e^{0.5621}$, which means the odds would increase by 75.44%.

iii)

The expected cost of prescribing medication is $775000 * 0.15p + 75000 * (1 - 0.15p)$

The expected cost of not prescribing medication is $700000 * p$

$$775000 * 0.15p + 75000 * (1 - 0.15p) = 700000 * p$$

We can achieve the break-even P should be 0.1261

So if the probability of a patient getting CHD is larger than 0.1261, we should give the medicine to the patient. If the probability is lower than 0.1261, then we shouldn't give the medication to the patient.

iv)

The accuracy of logistic regression model using threshold value of 0.126 is 0.627. This suggests that the possibility of our model correctly predicting whether the patient would get CHD within 10 years is 62.7%.

The TPR is 0.68, suggesting that given that a patient would get CHD in reality, there is a probability of 68% that our model would predict it correctly.

The FPR is 0.38, suggesting that given that a patient wouldn't get CHD in reality, there is a probability of 38% that our model would wrongfully classify him to be in the group that would get CHD.

In [3]:

```
print(fram_test)
print(fram_test['education'].dtypes)
fram_test.dtypes
y_predProb = logreg.predict(fram_test)

print(y_predProb)
y_predResult = pd.Series([1 if i > 0.1261 else 0 for i in y_predProb], index = y_predProb.index)
y_realResult = fram_test['TenYearCHD']

from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_realResult, y_predResult)
print(cm)
print(type(cm))
cm_result = cm.ravel()
print(cm_result)

#Accuracy
acc = (cm_result[0]+cm_result[3])/(sum(cm_result))
print(acc)

#TPR
tpr = cm_result[3]/(cm_result[2]+cm_result[3])
print(tpr)

#FPR
```

```
fpr = cm_result[1]/(cm_result[1]+cm_result[0])
print(fpr)
```

	male	age	education	currentSmoker	cigsPerDay	\
0	1	46	Some high school	1	20	
1	0	65	Some college/vocational school	0	0	
2	0	41	College	0	0	
3	0	39	Some high school	0	0	
4	0	51	Some high school	0	0	
...	
1093	0	64	Some high school	0	0	
1094	1	54	Some high school	0	0	
1095	0	55	Some college/vocational school	0	0	
1096	1	64	College	0	0	
1097	0	46	Some high school	1	40	

	BPMeds	prevalentStroke	prevalentHyp	diabetes	totChol	sysBP	diaBP	\
0	0	0	0	0	219	118.0	79.0	
1	0	0	1	0	216	163.0	102.0	
2	0	0	0	0	159	119.0	76.0	
3	0	0	0	0	229	125.0	80.0	
4	0	0	1	0	225	155.0	92.5	
...	
1093	0	0	1	0	273	155.0	86.0	
1094	0	0	0	0	250	123.0	75.0	
1095	0	0	0	0	260	136.5	87.5	
1096	0	0	0	0	210	123.0	81.0	
1097	0	0	0	0	253	118.0	74.0	

	BMI	heartRate	glucose	TenYearCHD
0	24.17	70	90	0
1	30.12	91	73	0
2	27.49	55	70	0
3	24.10	75	58	0
4	23.84	72	63	0
...
1093	27.53	100	91	0
1094	25.91	65	71	0
1095	25.41	75	60	0
1096	26.49	60	75	1
1097	26.42	75	64	1

```
[1098 rows x 16 columns]
```

```
object
```

```
0    0.126993
1    0.254170
2    0.028517
```

```

3      0.025976
4      0.113625
...
1093   0.301143
1094   0.141394
1095   0.082371
1096   0.228751
1097   0.088837
Length: 1098, dtype: float64
[[569 354]
 [ 56 119]]
<class 'numpy.ndarray'>
[569 354  56 119]
0.6265938069216758
0.68
0.38353196099674974

```

v)

If the outcomes in the test set aren't affected by the treatment decision, then the expected cost per patient would be:

$$\frac{775000*119+75000*354+700000*56+0*569}{569+354+56+119} = 143875.23$$

This assumption is not reasonable because the prescription of medicine is supposed to prevent the CHD and therefore less people would catch CHD after the prescription, which is expected to decrease the cost per person. But now that this is not taken into consideration, the expected cost per person would be estimated to be higher than it should be.

If the effect of treatment decision is taken into consideration, only 15% of those prescribed would be expected to catch CHD. The expected cost per patient would be calculated as below:

$$\frac{775000*119*0.15+75000*(119+354-119*0.15)+700000*56+0*569}{569+354+56+119} = 79389.80$$

vi)

The baseline model uses the principle of majority to do the prediction. Since most people in the training set doesn't have CHD, the baseline model predicts everyone in the testing set doesn't have CHD.

This model has high accuracy of 84%, which is higher than the logistic regression model produced. But its True Positive Rate is 0 and its False Positive Rate is also 0.

Although this model has higher accuracy, suggesting this model may be more accurate in predicting whether a person would have CHD or not

than the logistic regression. But the cost per person in the baseline model would be:

$$\frac{775000*0+75000*0+700000*175+0*569}{569+354+56+119} = 111566.48$$

This is significantly higher than cost per person in logistic regression, which means if our overall goal is to minimize cost per person, then we should choose the logistic regression although its prediction may be less accurate.

```
In [4]: fram_test2 = fram_test['TenYearCHD']

fram_basePredict = pd.Series([0 for i in fram_test2], index = fram_test2.index)
cm_base = confusion_matrix(fram_test2, fram_basePredict)
print(cm_base)
cm_resultB = cm_base.ravel()
#Acc
acc_b = (cm_resultB[0]+cm_resultB[3])/(sum(cm_resultB))
print(acc_b)
tpr_b = cm_resultB[3]/(cm_resultB[2]+cm_resultB[3])
print(tpr_b)
fpr_b = cm_resultB[1]/(cm_resultB[1]+cm_resultB[0])
print(fpr_b)
```

```
[[923    0]
 [175    0]]
0.8406193078324226
0.0
0.0
```

vii)

```
In [5]: print(fram_test.columns)
newPerson = {'male': [0], 'age': [45], 'education': ['College'], 'currentSmoker': [1], 'cigsPerDay': [9], 'BPMeds': [1],
             'prevalentStroke': [1], 'prevalentHyp': [0], 'diabetes': [1], 'totChol': [220], 'sysBP': [140.0],
             'diaBP': [100.0], 'BMI': [33.00], 'heartRate': [69], 'glucose': [74], 'TenYearCHD': [0]}

print(newPerson)
newData = pd.DataFrame(newPerson, index = [0])
print(type(newData.iloc[0]))
print(newData.iloc[0])
```

```
print(newData['male'])
print(type(newData['male']))
newPred = logreg.predict(newData)
print(newPred)

#newPredict = logreg.predict(df_new)
```

```
Index(['male', 'age', 'education', 'currentSmoker', 'cigsPerDay', 'BPMeds',
      'prevalentStroke', 'prevalentHyp', 'diabetes', 'totChol', 'sysBP',
      'diaBP', 'BMI', 'heartRate', 'glucose', 'TenYearCHD'],
      dtype='object')
{'male': [0], 'age': [45], 'education': ['College'], 'currentSmoker': [1], 'cigsPerDay': [9], 'BPMeds': [1], 'prevalentS
troke': [1], 'prevalentHyp': [0], 'diabetes': [1], 'totChol': [220], 'sysBP': [140.0], 'diaBP': [100.0], 'BMI': [33.0],
'heartRate': [69], 'glucose': [74], 'TenYearCHD': [0]}
<class 'pandas.core.series.Series'>
male          0
age           45
education     College
currentSmoker    1
cigsPerDay      9
BPMeds          1
prevalentStroke    1
prevalentHyp      0
diabetes         1
totChol         220
sysBP          140.0
diaBP          100.0
BMI            33.0
heartRate       69
glucose         74
TenYearCHD      0
Name: 0, dtype: object
0      0
Name: male, dtype: int64
<class 'pandas.core.series.Series'>
0      0.137213
dtype: float64
```

b)

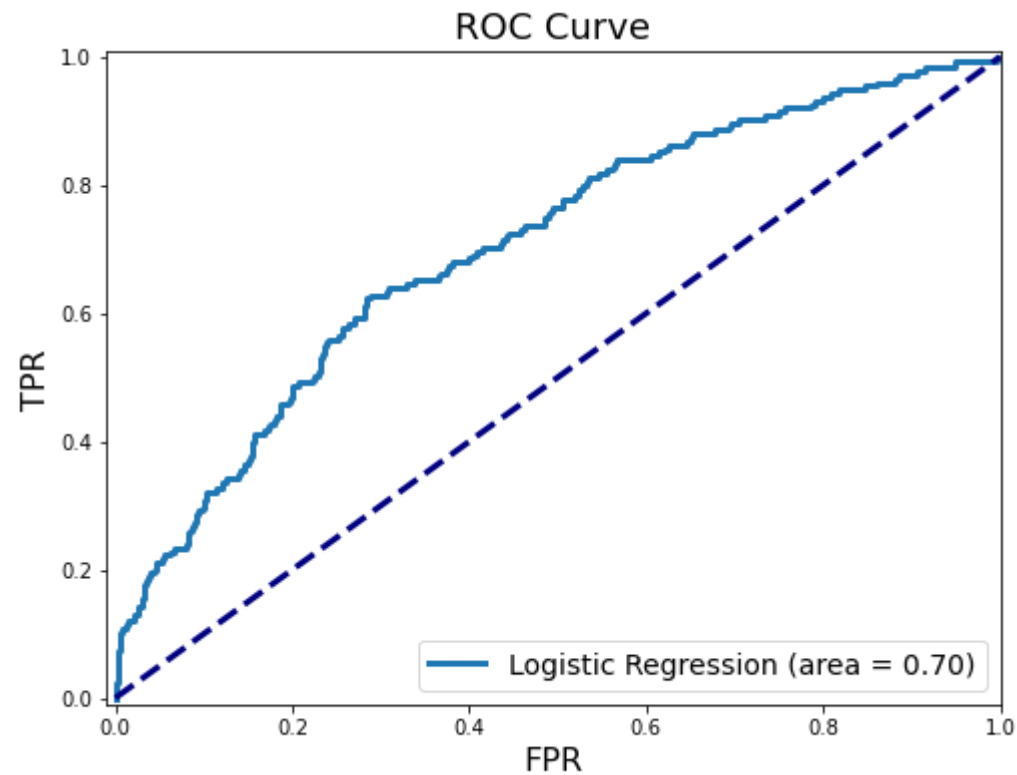
The AUC is 0.7 in the test set, which suggests that there is 70% probability that the model can successfully distinguish a person that will get CHD in 10 years from a person that won't catch CHD in 10 years.

The AUC in our model is larger than the naive bayes model and therefore our model is better than the Naive Bayes Model in the discriminative ability.

In [6]:

```
import matplotlib.pyplot as plt
from sklearn.metrics import roc_curve, auc

fpr, tpr, _ = roc_curve(y_realResult, y_predProb)
roc_auc = auc(fpr, tpr)
plt.figure(figsize = (8,6))
plt.title('ROC Curve', fontsize=18)
plt.xlabel('FPR', fontsize=16)
plt.ylabel('TPR', fontsize=16)
plt.xlim([-0.01, 1.00])
plt.ylim([-0.01, 1.01])
plt.plot(fpr, tpr, lw=3, label='Logistic Regression (area = {:.2f})'.format(roc_auc))
plt.plot([0, 1], [0, 1], color='navy', lw=3, linestyle='--')
plt.legend(loc='lower right', fontsize=14)
plt.show()
```



c)

For the scenario discussed in part a, if the probability of a patient getting CHD is larger than 0.1261, then we should prescribe the patient with the medication.

Suppose the co-payment amount is C .

If this threshold is still true, the probability of people who prescribe the medicine but still catch CHD would be $0.15 * 0.1261$. The cost incurred for the patient would be $500000 + C$.

The probability of person who prescribes the medicine but doesn't catch CHD would be $1 - 0.15 * 0.1261$. The cost incurred for the patient would be C .

The probability of person who doesn't prescribes the medicine but caught CHD would be 0.1261. The cost incurred for the patient would be 500000 since the treatment cost is covered by the insurance company.

The probability of person who doesn't prescribes the medicine and doesn't caught CHD would be $1 - 0.1261$. The cost incurred for the patient would be 0.

Set the expected cost incurred to person prescribed with medicine equal to the expected cost incurred to person that's not prescribed with medicine, we can solve the following equation:

$$(500000 + C) * 0.15 * 0.1261 + (1 - 0.15 * 0.1261) * C = 500000 * p$$

And we get that

$$C = 53592.5$$

When the co-payment cost is set to be 53592.5, then if the co-payment is larger than 53592.5, there would be a group of people not prescribing the medicine and the number of people in this group is the same as the number of people predicted by our logistic regression model to have lower than 0.1261 probability of catching CHD within future 10 years.

In this way, the co-payment cost of 53592.5 could match with the optimal strategy we have raised in part (a).

d)

I think one key issue here is that if we predict the patient to have the probability of less than 0.126 to catch CHD in 10 years, then we would not let the patient prescribe medication even though there is still a chance for them to catch the disease, which is unfair because everyone deserves the right to receive medication not to mention they have a chance to get infected.

So I think one way to solve this question is the approach discussed in part c, which is to give everyone the right to prescribe the medical prescription but setting a co-payment amount so that this cost would do self selection.