Cogs 109: Modeling and Data Analysis

Homework 3

Due **Monday 10/23 in class (extended)**

1. ISLR chapter 4, exercise 4 (page 168-169)
    a. (1 point) 0.1 or 10%
    b. (1 point) 0.1^2 = 0.01 or 1%
    c. (1 point) 0.1^{100}
    d. (1 point) For any p, we would use "0.1^p * 100" percent of the points for our prediction. As our number of predictors increases, the number of observations used to make our prediction decreases exponentially.
    e. (2 point) Some p predictors, our sides of the hypercube will be the pth root of 0.1. Think of the 2D case: If you have a square with an area of 0.1, then the sides are √(0.1). This generalizes to the other dimensions.

2. ISLR chapter 4, exercise 6

$$p(X) = \frac{\exp(\beta_0 + \beta_1 X_1 + \beta_2 X_2)}{1 + \exp(\beta_0 + \beta_1 X_1 + \beta_2 X_2)}$$
$$X_1 = hoursstudied, X_2 = undergradGPA$$
$$\beta_0 = -6, \beta_1 = 0.05, \beta_2 = 1$$

a) (1 point)

$$p(X) = \frac{\exp(-6 + 0.05X_1 + X_2)}{1 + \exp(-6 + 0.05X_1 + X_2)}$$

$$= \frac{\exp(-6 + 0.05 \times 40 + 3.5)}{1 + \exp(-6 + 0.05 \times 40 + 3.5)} = 0.3775$$

b) (2 points)

$$X = [X_1 hours, 3.5GPA]$$

$$p(X) = \frac{\exp(-6 + 0.05X_1 + X_2)}{1 + \exp(-6 + 0.05X_1 + X_2)}$$

$$0.50 = \frac{\exp(-6 + 0.05X_1 + 3.5)}{1 + \exp(-6 + 0.05X_1 + 3.5)}$$

$$0.50(1 + \exp(-2.5 + 0.05X_1)) = \exp(-2.5 + 0.05X_1)$$

$$0.50 + 0.50\exp(-2.5 + 0.05X_1)) = \exp(-2.5 + 0.05X_1)$$

$$0.50 = 0.50\exp(-2.5 + 0.05X_1)$$

$$\log(1) = -2.5 + 0.05X_1$$

$$X_1 = 2.5/0.05 = 50 hours$$

3. (2 points)

   ISLR chapter 4, exercise 7

$$p_k(x) = \frac{\pi_k \frac{1}{\sqrt{2\pi}\sigma}\exp(-\frac{1}{2\sigma^2}(x-\mu_k)^2)}{\sum \pi_l \frac{1}{\sqrt{2\pi}\sigma}\exp(-\frac{1}{2\sigma^2}(x-\mu_l)^2)}$$

$$p_{yes}(x) = \frac{\pi_{yes}\exp(-\frac{1}{2\sigma^2}(x-\mu_{yes})^2)}{\sum \pi_l \exp(-\frac{1}{2\sigma^2}(x-\mu_l)^2)}$$

$$= \frac{\pi_{yes}\exp(-\frac{1}{2\sigma^2}(x-\mu_{yes})^2)}{\pi_{yes}\exp(-\frac{1}{2\sigma^2}(x-\mu_{yes})^2) + \pi_{no}\exp(-\frac{1}{2\sigma^2}(x-\mu_{no})^2)}$$

$$= \frac{0.80\exp(-\frac{1}{2*36}(x-10)^2)}{0.80\exp(-\frac{1}{2*36}(x-10)^2) + 0.20\exp(-\frac{1}{2*36}x^2)}$$

$$p_{yes}(4) = \frac{0.80\exp(-\frac{1}{2*36}(4-10)^2)}{0.80\exp(-\frac{1}{2*36}(4-10)^2) + 0.20\exp(-\frac{1}{2*36}4^2)} = 75.2\%$$

4. Most neurons in the brain develop before you are born and remain with you throughout your life. A small but important part of the brain called the dentate gyrus of the hippocampus continues to create new neurons past birth and into adulthood. These “adult newborn neurons” are thought to be important for creating distinct memories of similar events. In this problem, we will use a recently published data set containing gene expression measurements from single neurons to classify cells by their age. The study by Habib et al. is titled “*Div-Seq: Single-nucleus RNA-Seq reveals dynamics of rare adult*

newborn neurons" (Science, 2016: https://www.ncbi.nlm.nih.gov/pubmed/27471252).

Download the data set hw3_divseq_data.csv. There are 3 variables; the first 2 rows are shown here:

Lars2	*Malat1*	mature
9.95	6.69	1
10.54	8.53	1

- *Lars2* and *Malat1* are the expression levels[1] of two genes, both of which become highly expressed as neurons mature.
- Mature is a binary variable coding whether a cell is mature (1, corresponding to cell_age=14) or immature (0, corresponding to cell_age<14).
    a. (2 points) Create a box plot showing the expression level of *Lars2* for immature and mature neurons. Do the same for *Malat1*.
    b. (1 point) Based on these plots, comment on whether you expect that a classifier could perfectly predict a neuron's maturity based on *Lars2* expression alone.
    c. (3 points) Fit a logistic regression to predict mature based on *Lars2* alone; do not use *Malat1*. What is the p-value for coefficient (slope) of *Lars2*? What can you infer, i.e. what conclusion can you draw?
    d. (3 points) Using your model, calculate the predicted probability that each neuron is mature, i.e. $p = P(mature \mid Lars2)$. Make a plot showing *Lars2* on the x-axis vs. p on the y-axis. The plot should have a sigmoid shape. Based on this plot, what prediction would you make for the maturity of a cell with *Lars2* = 8?
    e. (1 point) Use a Bayesian classification criterion to predict, for each cell, whether or not it is mature. Recall that a Bayesian classifier chooses the most likely category; in this case, that means that it should predict "mature" whenever $P(mature \mid Lars2) > 0.5$. Using these predictions, compute the sensitivity of your classifier, i.e. the fraction of mature cells that are correctly classified as mature.
    f. (1 point) Compute the specificity of your classifier, i.e. the fraction of immature cells that are correctly classified as immature.
    g. (4 points) Try predicting the maturity level for each cell with a threshold of 20%, i.e. predict mature whenever $P(mature \mid Lars2) > 0.2$. What are the sensitivity and specificity? Explain why the sensitivity is increased, while the specificity is decreased. In what circumstance might you prefer to use this classification threshold (20%) instead of the Bayesian threshold (50%)?
    h. (4 points) Now we will incorporate data from both genes to try to improve our prediction. First, make a scatter plot showing *Lars2* expression (x-axis) vs. *Malat1* expression (y-axis). Use a

[1] Gene expression is measured in units called "log TPM", or log(transcripts per million).

different color and/or plot symbol for cells that are immature and mature. Make sure to label the axes of the plot and include a legend explaining which color/symbol corresponds to which condition.

i. (2 points) Fit a logistic regression using both *Lars2* and *Malat1* as predictors. Print the regression summary table showing the coefficients, SE, t-statistic and p-value for each term. Which predictors have a significant effect?

j. (3 points) Use your new model to predict whether each neuron is mature, using a Bayesian decision threshold, i.e. $P(mature \mid Malat1,\ Lars2) > 0.5$. What are the sensitivity and specificity for this new prediction? Compare these values to the sensitivity and specificity you calculated in part (e).

Solution to Question 4

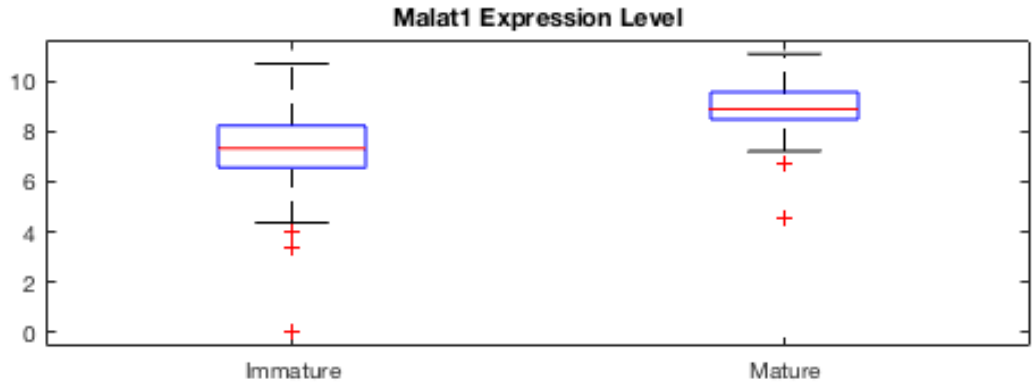
Contents

Read in

```
data = readtable('hw3_divseq_data.csv');
```

Part a - Create Boxplot of Mature vs Imature Neurons

```
subplot(2,1,1);
boxplot(data.Lars2, data.mature, 'Labels', {'Immature', 'Mature'});
title('Lars2 Expression Level');
subplot(2,1,2);
boxplot(data.Malat1, data.mature, 'Labels', {'Immature', 'Mature'});
title('Malat1 Expression Level');
```



Part b

Perfect prediction of mature vs non-mature neurons does not look possible for these two gene expression measures seperately, as the distibutions for gene expression are overlapping.

Part c - Fit Logistic Regression Using Lars2

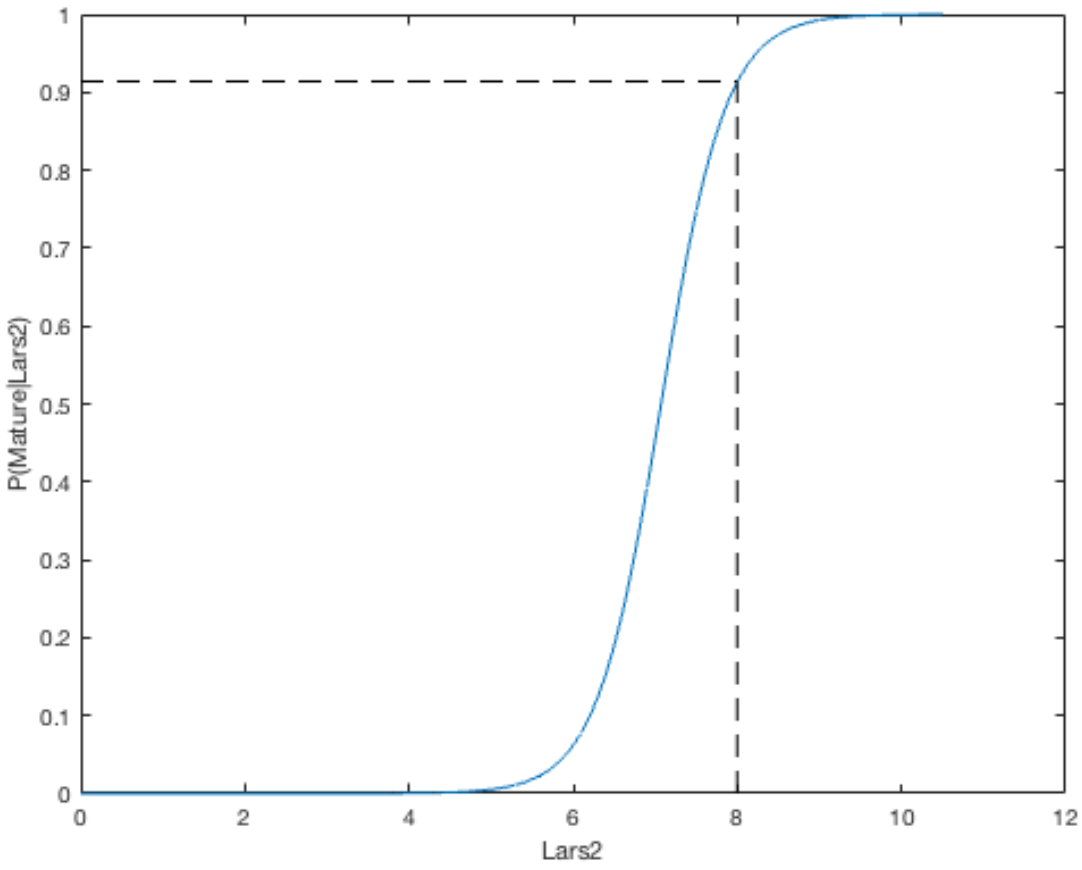
```
model = fitglm(data, 'mature~Lars2', 'distribution', 'binomial')
```

```
model =


Generalized linear regression model:
    logit(mature) ~ 1 + Lars2
    Distribution = Binomial

Estimated Coefficients:
                   Estimate      SE        tStat       pValue
                   ________    _______    _______    __________

    (Intercept)    -17.978      1.4318    -12.556    3.6794e-36
    Lars2           2.5422     0.20853     12.191    3.4558e-34


817 observations, 815 error degrees of freedom
Dispersion: 1
Chi^2-statistic vs. constant model: 436, p-value = 6.28e-97
```

P value for slope coefficient is very low indicating that there is a significant effect. The positive slope means that increasing Lars2 expression increases the probability of the neuron being classified as mature.

Part d - Plot Relationship Between Lars2 Expression and Maturity Probability

```
clf
x= array2table([0:0.1:max(data.Lars2)]', 'VariableNames', {'Lars2'});
y_hat = predict(model, x);
plot(x.Lars2, y_hat);
xlabel('Lars2');
ylabel('P(Mature|Lars2)');
hold on
% Predict probability for Lars2 = 8
cell8 = array2table(8, 'VariableNames', {'Lars2'});
prob8 = predict(model, cell8);
% Plot lines on graph
plot([0, 8], [prob8, prob8], 'LineStyle', '--', 'color', 'black');
plot([8, 8], [0, prob8], 'LineStyle', '--', 'color', 'black');
```



Investigating the graph, the predicted probability of a Lars2=8 cell being mature is approximatly 0.9

Part e - Calculate Predicted Probabilities of for each cell

```
data.Predict_Lars2 = predict(model, data);
% Use Bayesian classifier (0.5 threshold)
data.Predict_Lars2_bayes = (data.Predict_Lars2 > 0.5);
% Calculate sensitivity
TP = sum(data.Predict_Lars2_bayes(logical(data.mature)) == 1);
sensitivity = TP/sum(data.mature)
```

```
sensitivity =

    0.6545
```

Part f - Calculate Specificity

```
FP = sum(data.Predict_Lars2_bayes(~data.mature) == 0);
specificity = FP/sum(~data.mature)
```

```
specificity =

    0.9448
```

Part g - Use 20% Threshold Classifier

```
data.Pred_20perc = (data.Predict_Lars2 > 0.2);
% Calculate sensitivity
TP = sum(data.Pred_20perc(logical(data.mature)) == 1);
sensitivity = TP/sum(data.mature)

% Calculate Specificity
FP = sum(data.Pred_20perc(~data.mature) == 0);
specificity = FP/sum(~data.mature)
```

```
sensitivity =

    0.9091


specificity =

    0.8681
```

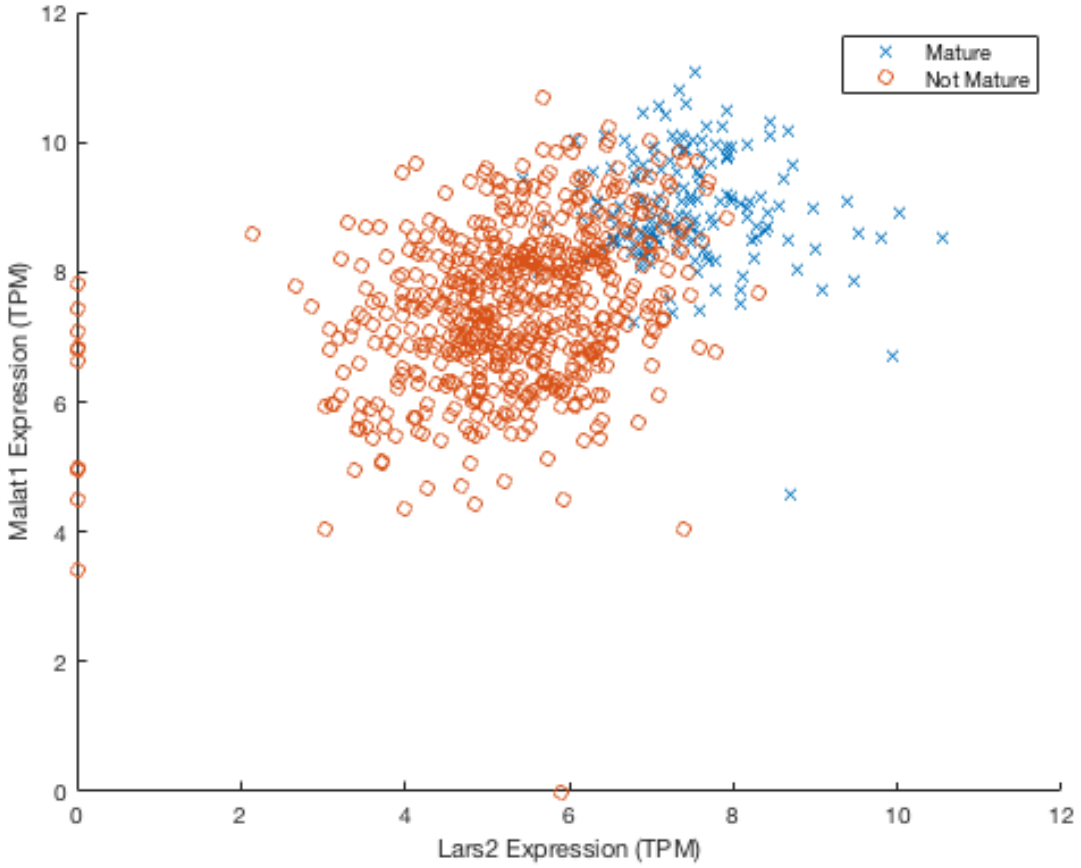
The sensitivity went up because with a lower threshold more cell are being classified as mature. The specificty went down as some of these cell were incorrectly classifed to be mature when in fact they were immature

You may want to use a 20% threshold when you are attempting to classify neurons as mature at an earlier timepoint.

You may also want a threshold of 20% when you care more about specificity than you do about sensitivity.

Part h - Make a 2D Scatter Plot of Expression for the Two Genes

```
clf
mature_ind = logical(data.mature);
scatter(data.Lars2(mature_ind), data.Malat1(mature_ind), 'Marker', 'x', 'DisplayName', 'Mat
ure');
hold on
scatter(data.Lars2(~mature_ind), data.Malat1(~mature_ind), 'Marker', 'o', 'DisplayName', 'N
ot Mature');
xlabel('Lars2 Expression (TPM)');
ylabel('Malat1 Expression (TPM)');
legend('show');
```



Part i - Fit Logistic Regression Using both Lars2 and Malat1 as Predictors

```
model = fitglm(data, 'mature ~ Lars2 + Malat1',  'distribution', 'binomial')
```

```
model =


Generalized linear regression model:
    logit(mature) ~ 1 + Lars2 + Malat1
    Distribution = Binomial

Estimated Coefficients:
                   Estimate      SE        tStat       pValue
                   ________    ________    ________    __________

    (Intercept)    -25.57       2.1775    -11.743    7.6914e-32
    Lars2          2.3119      0.22328     10.354    3.9955e-25
    Malat1         1.0836      0.15611     6.9413    3.8856e-12


817 observations, 814 error degrees of freedom
Dispersion: 1
Chi^2-statistic vs. constant model: 500, p-value = 2.12e-109
```

Given the very low P-values both the gene expression variables have a significant effect

Part j - Calculate sensitivity and specificity

```
data.Preict_Larst2_Malat1 = predict(model, data);
% Use Bayesian classifier (0.5 threshold)
```

```
y_hat = (data.Preict_Larst2_Malat1 > 0.5);
% Calculate sensitivity
TP = sum(y_hat(logical(data.mature)) == 1);
sensitivity = TP/sum(data.mature)

% Calculate Specificity
FP = sum(y_hat(~data.mature) == 0);
specificity = FP/sum(~data.mature)
```

```
sensitivity =

    0.7273


specificity =

    0.9479
```

Compared with the results of part (e), both sensitivity and specificity values here increases, indicating this new classifier with 2 predictors combined outperforms the one using only Lars2 as feature.

Published with MATLAB® R2016b

hw3_python_code

October 19, 2017

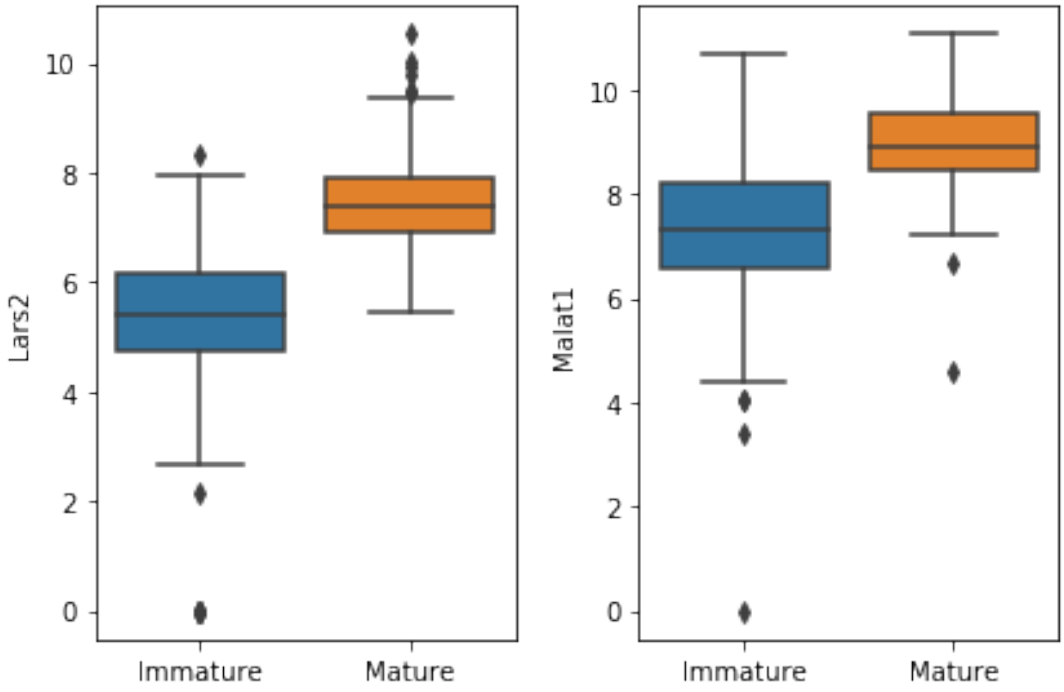
```
In [96]: import pandas as pd
         import numpy as np
         import matplotlib.pyplot as plt
         import seaborn as sns
         import statsmodels.formula.api as smf
```

```
In [97]: df = pd.read_csv('hw3_divseq_data.csv')
         df.head()
         # df.shape
```

```
Out[97]:    Lars2  Malat1  mature
         0   9.95    6.69       1
         1  10.54    8.53       1
         2   6.58    8.74       1
         3   7.49    9.09       1
         4   7.42    9.87       1
```

0.0.1 (a) boxplot

```
In [98]: fig, axs = plt.subplots(1,2)
         sns.boxplot(x='mature', y='Lars2', data=df, ax=axs[0]) # boxplot
         sns.boxplot(x='mature', y='Malat1', data=df, ax=axs[1])
         axs[0].set_xlabel('') # remove x_label
         axs[1].set_xlabel('')
         axs[0].set_xticklabels(['Immature', 'Mature']) # add x_ticklabels
         axs[1].set_xticklabels(['Immature', 'Mature'])

         fig.tight_layout()
         plt.show()
```



0.0.2 (b) comment

A classifier cannot perfectly predict a neuron's maturity based on Lars2, because Lar2 values of the 2 categories overlap with each other.

0.0.3 (c) logistic regression

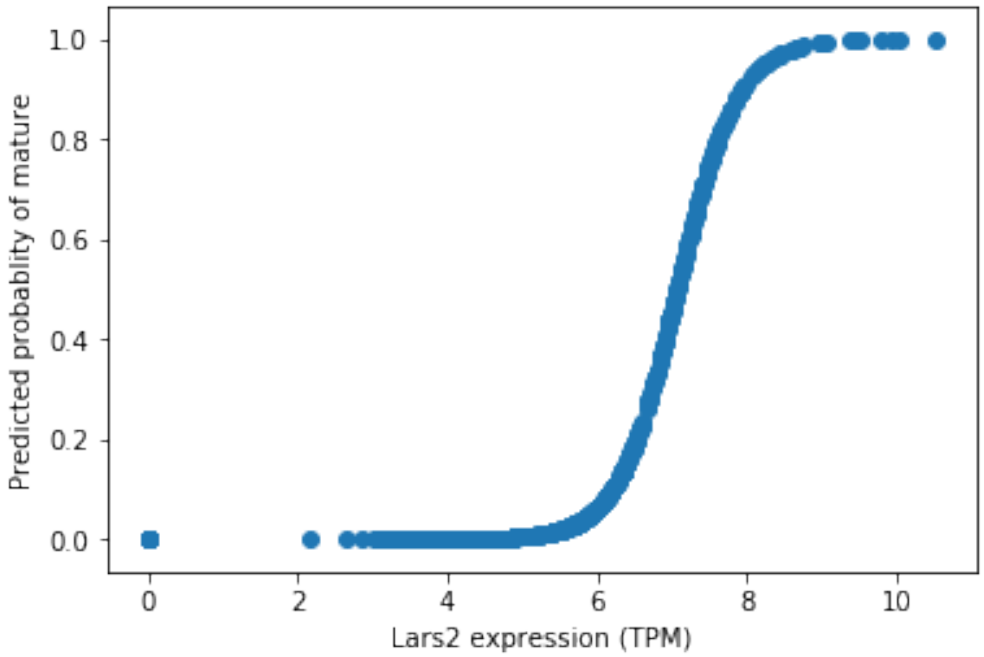
```
In [99]: res = smf.logit(formula='mature ~ Lars2', data=df).fit()
         print(res.summary())
         print(res.pvalues)

Optimization terminated successfully.
         Current function value: 0.235975
         Iterations 9
                           Logit Regression Results
==============================================================================
Dep. Variable:                 mature   No. Observations:                  817
Model:                          Logit   Df Residuals:                      815
Method:                           MLE   Df Model:                            1
Date:                Thu, 19 Oct 2017   Pseudo R-squ.:                  0.5310
Time:                        00:29:55   Log-Likelihood:                -192.79
converged:                       True   LL-Null:                       -411.04
                                        LLR p-value:                6.284e-97
==============================================================================
```

```
                coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept    -17.9775      1.432    -12.556      0.000     -20.784     -15.171
Lars2          2.5422      0.209     12.191      0.000       2.134       2.951
==============================================================================
Intercept    3.679364e-36
Lars2        3.455778e-34
dtype: float64
```

P-value is 3.46e-34, which is very small, which indicates Lars2 expression level significantly associates with maturity.

0.0.4 (d)

```
In [100]: p_mature = res.predict(df.Lars2)
          fig, ax = plt.subplots()
          ax.scatter(df.Lars2, p_mature)
          ax.set_xlabel('Lars2 expression (TPM)')
          ax.set_ylabel('Predicted probablity of mature')
          plt.show()
```



Based on the plot, Lars2=8 indicates there is about 80% chance that neuron is mature.

0.0.5 (e) and (f)

```
In [101]: yhat = 1*(res.predict(df.Lars2) > 0.5)
          TP = FP = TN = FN = 0
          for y_i, yhat_i in zip(y, yhat):
              if (y_i, yhat_i) == (0, 0):
                  TN += 1
              elif (y_i, yhat_i) == (0, 1):
                  FP += 1
              elif (y_i, yhat_i) == (1, 0):
                  FN += 1
              elif (y_i, yhat_i) == (1, 1):
                  TP += 1

          print(TP, FP, TN, FN)

          sensitivity = TP/(TP + FN)
          specificity = TN/(TN + FP)
          print('Sensitivity is %f, and specificity is %f' % (sensitivity, specificity))
```

```
108 36 616 57
Sensitivity is 0.654545, and specificity is 0.944785
```

0.0.6 (g)

```
In [102]: yhat_2 = 1*(res.predict(df.Lars2) > 0.2)
          # repeat the previous calculation for yhat_2
          TP = FP = TN = FN = 0
          for y_i, yhat_i in zip(y, yhat_2):
              if (y_i, yhat_i) == (0, 0):
                  TN += 1
              elif (y_i, yhat_i) == (0, 1):
                  FP += 1
              elif (y_i, yhat_i) == (1, 0):
                  FN += 1
              elif (y_i, yhat_i) == (1, 1):
                  TP += 1

          print(TP, FP, TN, FN)

          sensitivity = TP/(TP + FN)
          specificity = TN/(TN + FP)
          print('Sensitivity is %f, and specificity is %f' % (sensitivity, specificity))
```

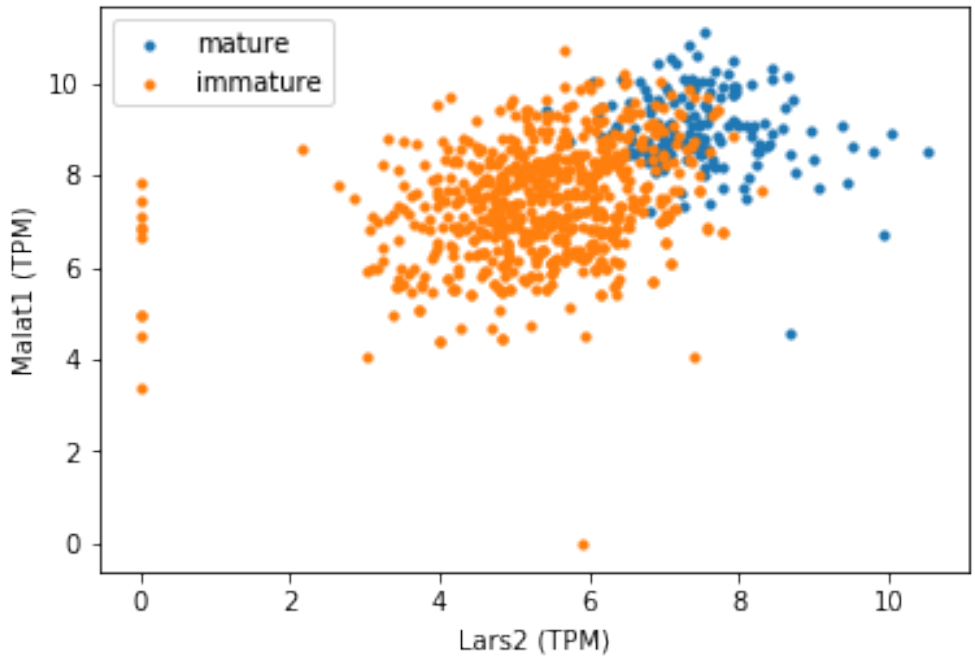
```
150 86 566 15
Sensitivity is 0.909091, and specificity is 0.868098
```

It's easier for a neuron to be classfied as mature for a threshold of 0.2 than for that of 0.5. So a mature cell should more likely to be (correctly) classified as mature, and immature cells are also more likely to be mislabelled as mature for a lower threshold.

Generally, in cases where we care sensitivity more than specificity, we might want to low down threshold. In this case, if we want to pick up neuron maturity in early stage, we could try using a low threshold.

0.0.7 (h)

```
In [103]: df_mature = df[df.mature==1]
          df_immature = df[df.mature==0]
          fig, ax = plt.subplots()
          ax.scatter(df_mature.Lars2, df_mature.Malat1, s=10, label='mature')
          ax.scatter(df_immature.Lars2, df_immature.Malat1, s=10, label='immature')
          ax.set_xlabel('Lars2 (TPM)')
          ax.set_ylabel('Malat1 (TPM)')
          ax.legend()
          plt.show()
```



0.0.8 (i)

```
In [104]: res_2 = smf.logit(formula='mature ~ Lars2 + Malat1', data=df).fit()
          res_2.summary()
```

```
Optimization terminated successfully.
         Current function value: 0.196827
         Iterations 9
```

Out[104]:
```
<class 'statsmodels.iolib.summary.Summary'>
"""
                           Logit Regression Results
==============================================================================
Dep. Variable:                 mature   No. Observations:                  817
Model:                          Logit   Df Residuals:                      814
Method:                           MLE   Df Model:                            2
Date:                Thu, 19 Oct 2017   Pseudo R-squ.:                 0.6088
Time:                        00:30:01   Log-Likelihood:                -160.81
converged:                       True   LL-Null:                       -411.04
                                        LLR p-value:                2.122e-109
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept    -25.5697      2.177    -11.743      0.000     -29.838     -21.302
Lars2          2.3119      0.223     10.354      0.000       1.874       2.750
Malat1         1.0836      0.156      6.941      0.000       0.778       1.390
==============================================================================
"""
```

Both predictors have significant effects.

0.0.9 (j)

In [105]:
```
yhat_3 = 1*(res_2.predict(df[['Lars2', 'Malat1']]) > 0.5)

# repeat the previous calculation for yhat_3
TP = FP = TN = FN = 0
for y_i, yhat_i in zip(y, yhat_3):
    if (y_i, yhat_i) == (0, 0):
        TN += 1
    elif (y_i, yhat_i) == (0, 1):
        FP += 1
    elif (y_i, yhat_i) == (1, 0):
        FN += 1
    elif (y_i, yhat_i) == (1, 1):
        TP += 1

print(TP, FP, TN, FN)

sensitivity = TP/(TP + FN)
specificity = TN/(TN + FP)
print('Sensitivity is %f, and specificity is %f' % (sensitivity, specificity))
```

```
120 34 618 45
Sensitivity is 0.727273, and specificity is 0.947853
```

Compared with the results of part (e), both sensitivity and specificity values here increases, indicating this new classifier with 2 predictors combined outperforms the one using only Lars2 as feature.