Chapter 4 - Classification

1. $p(x) = \frac{e^{X^T\beta}}{1+e^{X^T\beta}}$

   $p(x)[1 + e^{X^T\beta}] = e^{X^T\beta}$

   $p(x) + p(x)e^{X^T\beta} = e^{X^T\beta}$

   $p(x) = [1 - p(x)]e^{X^T\beta}$

   $\frac{p(x)}{1-p(x)} = e^{X^T\beta}$

2. $p_k(x) = \frac{\pi_k \frac{1}{\sqrt{2\pi}\sigma} e^{\frac{-(x-\mu_k)^2}{2\sigma^2}}}{\sum_{l=1}^{K} \pi_l \frac{1}{\sqrt{2\pi}\sigma} e^{\frac{-(x-\mu_l)^2}{2\sigma^2}}}$

   Taking log on both sides we get,

   $\delta_k(x) = log(\pi_k) + log(\frac{1}{\sqrt{2\pi}}) - \frac{(x-\mu_k)^2}{2\sigma^2} - log(\sum_{l=1}^{K} \pi_l \frac{1}{\sqrt{2\pi}\sigma} e^{\frac{-(x-\mu_l)^2}{2\sigma^2}})$

   The above equation is got by applying the following log properties:

   - $log(a/b) = log(a) - log(b)$
   - $log(ab) = log(a) + log(b)$

   $\delta_k(x) = log(\pi_k) + log(\frac{1}{\sqrt{2\pi}}) - \frac{x^2}{2\sigma^2} - \frac{\mu_k^2}{2\sigma^2} + \frac{2x\mu_k}{2\sigma^2} - log(\frac{1}{\sqrt{2\pi}} \sum_{l=1}^{K} \pi_l e^{\frac{-(x-\mu_l)^2}{2\sigma^2}})$

   $\delta_k(x) = log(\pi_k) + log(\frac{1}{\sqrt{2\pi}}) - \frac{x^2}{2\sigma^2} - \frac{\mu_k^2}{2\sigma^2} + \frac{2x\mu_k}{2\sigma^2} - log(\frac{1}{\sqrt{2\pi}}) - log(\sum_{l=1}^{K} \pi_l e^{\frac{2x\mu_l - x^2 - \mu_l^2}{2\sigma^2}})$

   Inside the *log* term we see that $e^{\frac{x^2}{2\sigma^2}}$ is independent of l and can be got out of the summation.

$$\delta_k(x) = log(\pi_k) - \frac{x^2}{2\sigma^2} - \frac{\mu_k^2}{2\sigma^2} + \frac{2x\mu_k}{2\sigma^2} + \frac{x^2}{2\sigma^2} - log(\sum_{l=1}^{K} \pi_l e^{\frac{2x\mu_l - x^2 - \mu_l^2}{2\sigma^2}})$$

$$\delta_k(x) = log(\pi_k) - \frac{\mu_k^2}{2\sigma^2} + \frac{2x\mu_k}{2\sigma^2} - log(\sum_{l=1}^{K} \pi_l e^{\frac{2x\mu_l - x^2 - \mu_l^2}{2\sigma^2}})$$

Now the *log* term is a summation over all classes, and hence will have the same value for any class k. Thus the term can be ignored while looking for the largest $\delta_k(x)$. Thus,

$$\delta_k(x) = log(\pi_k) - \frac{\mu_k^2}{2\sigma^2} + \frac{2x\mu_k}{2\sigma^2}$$

3. $p(x) = \frac{\pi_k f_k(x)}{\sum_{l=1}^{K} \pi_l f_l(x)}$

$$p_k(x) = \frac{\pi_k \frac{1}{\sqrt{2\pi}\sigma_k} e^{\frac{-(x-\mu_k)^2}{2\sigma_k^2}}}{\sum_{l=1}^{K} \pi_l \frac{1}{\sqrt{2\pi}\sigma_l} e^{\frac{-(x-\mu_l)^2}{2\sigma_l^2}}}$$

Taking log on both sides we get,

$$\delta_k(x) = log(\pi_k) - log(\sqrt{2\pi}\sigma_k) - \frac{(x-\mu_k)^2}{2\sigma_k^2} - log(\sum_{l=1}^{K} \pi_l \frac{1}{\sqrt{2\pi}\sigma_l} e^{\frac{-(x-\mu_l)^2}{2\sigma_l^2}})$$

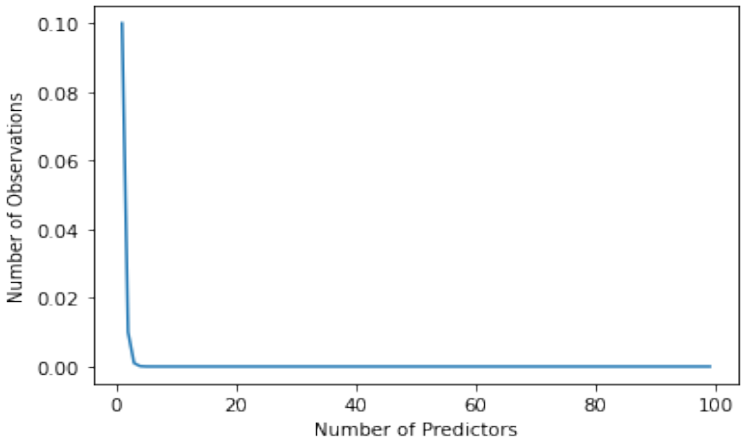
Now the denominator will be constant while comparing between classes, and hence we can treat it as a constant. Thus we can see that the equation is quadratic in x and hence the Bayes' decision boundary will be quadratic and not linear.

4. (a) For every test observation we use an interval of 10% of the total number of observations to carry out KNN. Thus on average we will be using 10% of the available observations to make a prediction.

   (b) If we had a unit square where every point of the square is an observation, then we would only be using a square of dimensions 0.1 X 0.1 to make predictions. Thus the average would be 1% of all observations.

(c) For a model with $p = 100$ predictors we would only be using 0.1^{100} of the total points in the hyper cube formed by the predictors (1^{100}). In general for p predictors, we would be using 0.1^p number of observations from a 1^p hyper-cube.

(d) From the above three example it is clear that with an increase in the number of predictors, KNN has lesser and lesser points in observation set to make a prediction, leading to very poor performance. The below graph shows how the number of observations falls exponentially with the increase in the number of predictors.



(e) Number of training observations $= x^p = 0.1$ where x is the length of the hyper cube which we use for making predictions.

$plog(x) = log(0.1)$

$x = e^{\frac{log(0.1)}{p}}$

Thus for $p = 1$, $x = 0.1$, for $p = 2$, $x = 0.31$ and for $p = 100$, $x = 0.98$. We see that as the number of predictors increases, the percentage of the hyper-cube required to make predictions increases drastically. The localized method uses 98% of the 100 dimensional hyper-cube.

5. (a) We expect QDA to perform similar to or better than LDA on the training set as it is more flexible. However, on the test set we expect LDA to do better as the Bayes' decision boundary is linear.

   (b) If the Bayes' decision boundary is non-linear, we have two situations that might arise. In the first case if the true relationship is closer to linear than non-linear than LDA is expected to do better. If the true relationship is very far from linear, then the QDA expected to perform better.

   (c) In general, with an increase in n, we will see that the test prediction accuracy of QDA improve as an increase in observations leads to a reduction in variance. In case of a linear Bayes' decision boundary, QDA will perform worse than LDA but in case of a non-linear Bayes' decision boundary, QDA will outperform LDA.

   (d) False. If the Bayes' decision boundary is linear, then the QDA will over fit the data by finding relationships amongst the noise present in the data. This over-fitting prohibits QDA from generalizing well on the test set.

6. $p(Y) = \frac{1}{1+e^{(6-0.05X_1-X_2)}}$

   (a) $X_1 = 40$ and $X_2 = 3.5$

   $$p(Y) = \frac{1}{1+e^{(6-0.05(40)-3.5)}}$$

   $$p(Y) = 0.37$$

   (b) $p(Y) = 0.5$ and $X_2 = 3.5$

   $$\frac{p(Y)}{1-p(Y)} = e^{(6-0.05X_1-X_2)}$$
   $$log(1) = 2.5 - 0.05X_1$$

   $$X_1 = 50$$

7. Given $\pi_Y = 0.8$, $\pi_N = 0.2$, $\mu_Y = 10$, $\mu_N = 0$ and $\sigma^2 = 36$. Probability that a company will issue dividend this year is given as follows:

   $p_Y(X) = \frac{\pi_Y f_Y(X)}{\pi_N f_Y(X) + \pi_Y f_Y(X)}$ where $f_k(X) = \frac{1}{\sqrt{2\pi}\sigma} e^{\frac{-(X-\mu_k)^2}{2\sigma^2}}$

   $f_Y(X = 4) = \frac{1}{6\sqrt{2\pi}} e^{\frac{-(4-10)^2}{2(36)}} = 0.04$

   $f_N(X = 4) = \frac{1}{6\sqrt{2\pi}} e^{\frac{-(4-0)^2}{2(36)}} = 0.053$

   $p_Y(X = 4) = \frac{0.8(0.04)}{0.8(0.04)+0.2(0.053)} = 0.7511$

8. I would use the logistic regression algorithm. The KNN classifier is trained using K=1. This leads to over-fitting due very high variance. As a result of this over-fitting the training error rate might be very low while the test error rate might be very high (an average of the two might be low).

9. (a) $\frac{p(X)}{1-p(X)} = 0.37$

   Thus $p(X) = 0.27$

   (b) $odds = \frac{p(X)}{1-p(X)}$

   Give $p(X) = 0.16$, $odds = \frac{0.16}{0.84} = 0.19$