Homework 2

Gavin Monroe - ComS 474

1) **Suppose you are predicting feature y using feature x with logistic regression, and x is measured in kilometers. After fitting, you get coefficients β0 = 1.24 and β1 = −3.74.** *Thus, your model is:*

Prob(y = 1|x) = e 1.24−3.74x / 1 + e 1.24−3.74x .

Suppose our friend Sammie has an innate fear of the metric system, starts with the same data set, converts the x values to miles, does not change y values, and then fits.

What will Sammie's β0 and β1 be?

It would be the same, as the values would just be converted the ratio would equal out overall giving the same coefficients to the ones above:

β0: **1.24**

*β1:***−3.74**

This is for ofc the difference in the values doesn't matter when fitting the logistic regression in this case.

2) When the number of features p is large, there tends to be a deterioration in the performance of KNN and other local approaches that perform prediction using only observations that are near the test observation for which a prediction must be made. This phenomenon is known as the curse of dimensionality, and it ties into the fact that curse of dinon-parametric approaches often perform poorly when p is large. We mensionality will now investigate this curse.

(a) Suppose that we have a set of observations, *each with measurements on p = 1 feature, X.* **We assume that** *X is uniformly (evenly) distributed on [0, 1].*

Associated with each observation is a response value.
Suppose that we wish to predict a test observation's response using only observations that are **within 10 % of the range of X closest** to that test observation.

Example:
For instance, in order to predict the response for a test observation with X = 0.6, (4.7 Exercises 169) we will use observations in the range [0.55, 0.65].

On average, what fraction of the available observations will we use to make the prediction?

Innocently, on normal we would anticipate that 10% of perceptions should be accessible yet we should be cautious at the limits of the element. By integrating two straight capacities to

portray every limit we really get that on normal we have 9.75% of observations accessible. ∫[0.95, 0.05] 10dx + ∫[0.05, 0] 100x + 5dx + ∫[1, 0.95] 105 − 100x − dx => 9 + 0.375 + 0.375 = 9.75%

2 b) Now suppose that we have a set of observations, each with measurements on p = 2 features, X1 and X2. We assume that (X1, X2) are uniformly distributed on [0, 1] × [0, 1]. We wish to predict a test observation's response using only observations that are within 10 % of the range of X1 and within 10 % of the range of X2 closest to that test observation. For instance, in order to predict the response for a test observation with X1 = 0.6 and X2 = 0.35, we will use observations in the range [0.55, 0.65] for X1 and in the range [0.3, 0.4] for X2. On average, what fraction of the available observations will we use to make the prediction?

Since X1 and X2 are consistently and freely conveyed we can expect that the portion of accessible perceptions will simply be the result of two conditions indistinguishable from the one utilized to a limited extent Part 2a. Given this we can anticipate that (9.75×9.75)=0.950625 or about 1% of perceptions will be accessible to make the expectation.

2 c) Now suppose that we have a set of observations on p = 100 features. Again the observations are uniformly distributed on each feature, and again each feature ranges in value from 0 to 1. We wish to predict a test observation's response using observations within the 10 % of each feature's range that is closest to that test observation. What fraction of the available observations will we use to make the prediction? (d) Using your answers to parts (a)–(c), argue that a drawback of KNN when p is large is that there are very few training observations "near" any given test observation.

Roughly (9.75)^p =(9.75)^100≈0% of the observations will be available to make the prediction!

2 e) Now suppose that we wish to make a prediction for a test observation by creating a p-dimensional hypercube centered around the test observation that contains, on average, 10 % of the training observations. For p = 1, 2, and 100, what is the length of each side of the hypercube? Comment on your answer. Note: A hypercube is a generalization of a cube to an arbitrary number of dimensions. When p = 1, a hypercube is simply a line segment, when p = 2 it is a square, and when p = 100 it is a 100-dimensional cube.

For $p = 1$, we have $l = 0.1$, for $p = 2$, we have $l = 0.1^{1/2}$ and for $p = 100$, we have $l = 0.1^{1/100}$.

6. Suppose we collect data for a group of students in a statistics class with variables X1 = hours studied, X2 = undergrad GPA, and Y = receive an A. We fit a logistic regression and produce estimated coefficient, βˆ0 = −6, βˆ1 = 0.05, βˆ2 = 1.
6 a) Estimate the probability that a student who studies for 40 h and has an undergrad

GPA of 3.5 gets an A in the class.

The following equation satisfies the above question using regression.

$$p(X) = \frac{e^{\hat{\beta}_0+\hat{\beta}_1\cdot\text{Hours}+\hat{\beta}_2\cdot\text{GPA}}}{1+e^{\hat{\beta}_0+\hat{\beta}_1\cdot\text{Hours}+\hat{\beta}_2\cdot\text{GPA}}} = 0.3775$$

6 b) How many hours would the student in part (a) need to study to have a 50 % chance of getting an A in the class?

50 hours.

$$0.5 = \frac{e^{-6+0.05*X_1+3.5}}{1+e^{-6+0.05*X_1+3.5}} \Rightarrow X_1 = 50 \text{ Hours}$$

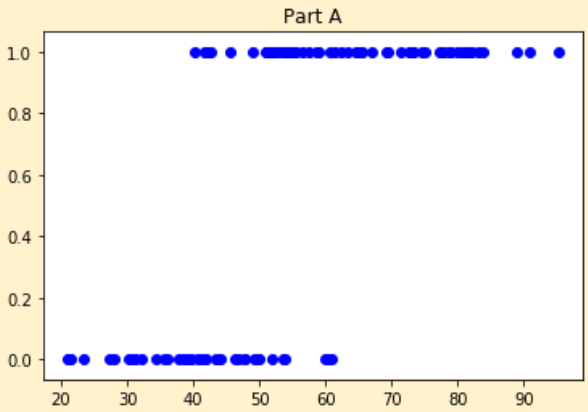
8) Suppose that we take a data set, divide it into equally-sized training and test sets, and then try out two different classification procedures. First we use logistic regression and get an error rate of 20 % on the training data and 30 % on the test data. Next we use 1-nearest neighbors (i.e. K = 1) and get an average error rate (averaged over both test and training data sets) of 18 %. Based on these results, which method should we prefer to use for classification of new observations? Why?

I would want to utilize strategic relapse. 1-closest neighbors will overfit the information and yield a preparation mistake pace of 0%. Consequently the test blunder rate is really 36% and more regrettable than what is found with the strategic relapse.

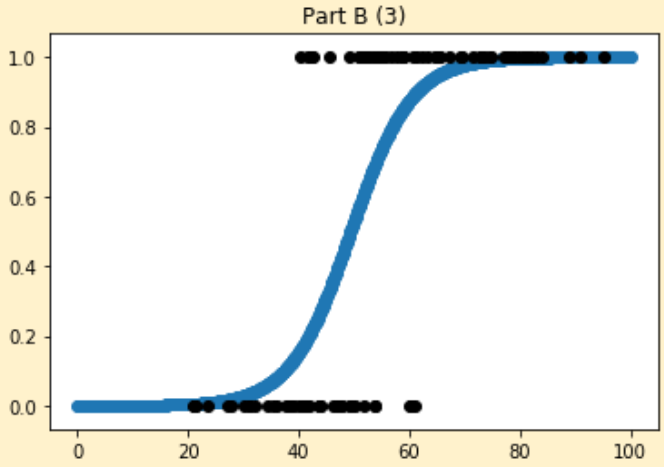
5 A & B) Charts and Data

The Data below is all the data for part 5.

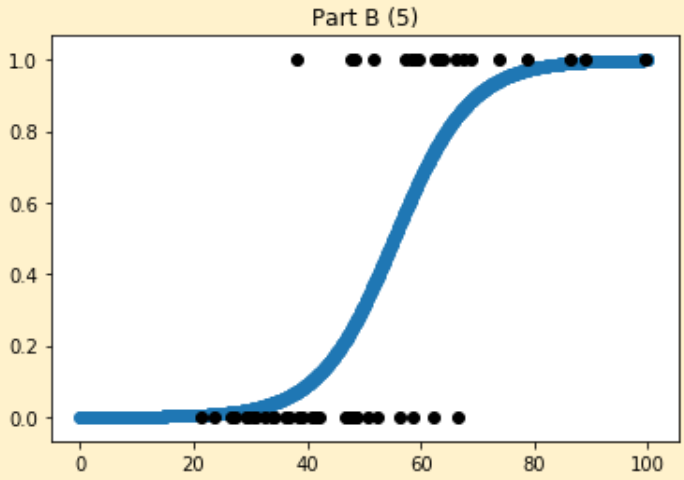
- Train Score: 0.8585858585858586
- Train B0: [0.18308215]
- Train B1: [-9.04016417]
- Test Score: 0.8367346938775511
- Test B0: [0.15080145]
- Test B1: [-8.33984437]



This is for part A where we just plot the csv data that we are asked to do.



This chart is for part 5 B 3) where we generate the 1000 different points evenly spaced out from 0 to 100.

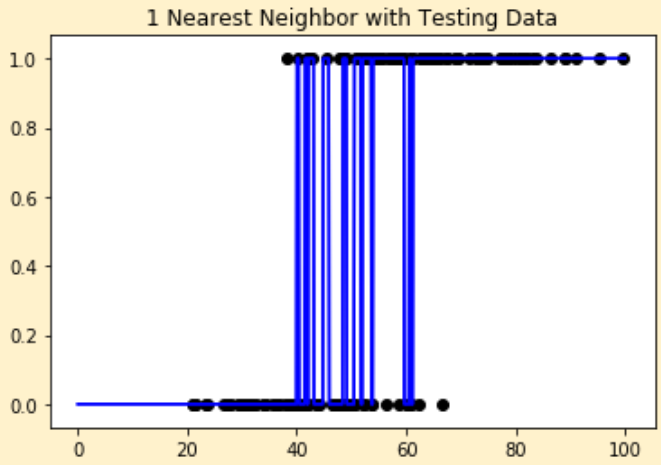
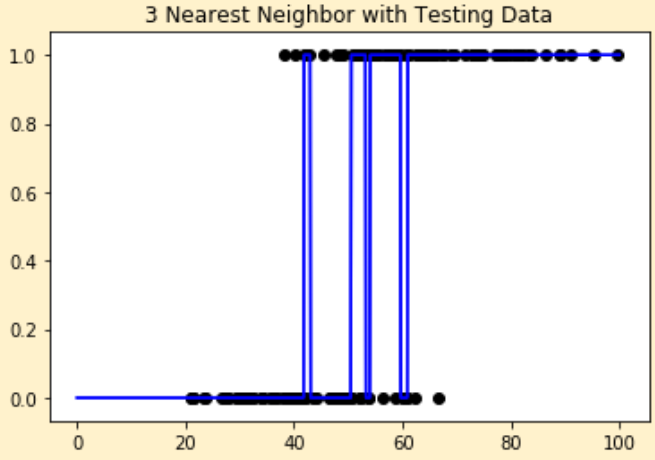


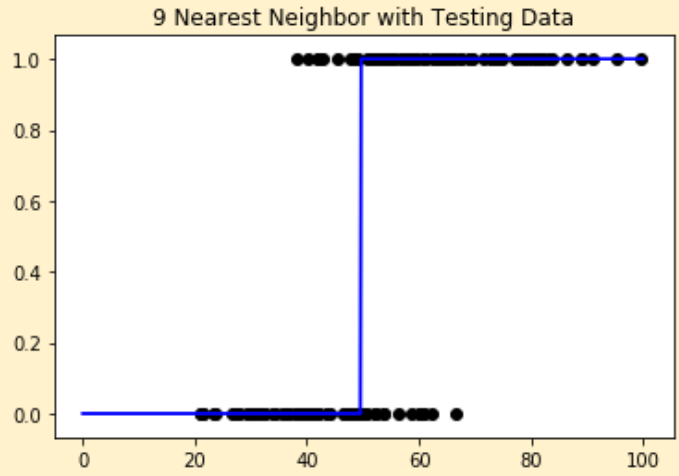
This is the same as above but we are doing it with the test data. The Data for all the charts can be found in the beginning.

5 C-1 & (2 & 3)) Charts and Data

The data below is data generated for the k 1, 3, & 9

- **k: 1 Score = 1.0 | Total accuracy = 0.6938775510204082**
- **k: 3 Score = 0.898989898989899 | Total accuracy = 0.7959183673469388**
- **k: 9 Score = 0.8585858585858586 | Total accuracy = 0.8163265306122449**

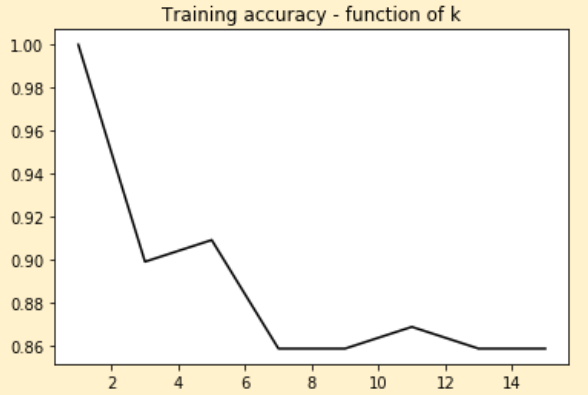
	**This is for the k = 1.**
	**This is for k = 3.**

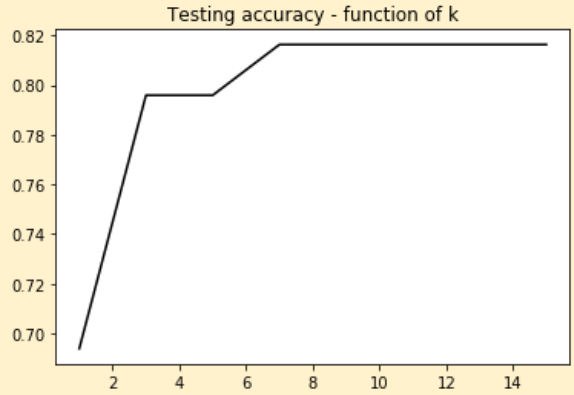


This is for k = 9.

5 C 2 & 3)

- **Train Score = [1., 0.8989899, 0.90909091, 0.85858586, 0.85858586, 0.86868687, 0.85858586, 0.85858586]**
- **Test Score = [0.69387755, 0.79591837, 0.79591837, 0.81632653, 0.81632653, 0.81632653, 0.81632653, 0.81632653]**





5 D) In about 4-6 sentences, comment on the performance of the different nearest neighbor classifiers for the different k values you used, including whether you see any evidence of over-fitting or under-fitting, and how they compare to the logistic regression classifier, and any other note-worthy aspects.

With the lower numbers the predictions are underfitting and you can see that correlation in the the testing graph. As the numbers continue to go up the fitting predictions become better. You can also see that in the two chars above in problem 5 C 2 & 3. So the answer to the question is both, underfitting and overfitting in certain predictions.