CMPS 142 Machine Learning
Spring 2018, Homework #2

Aaron Steele, atsteele@ucsc.edu
Tommy Tran, ttran56@ucsc.edu

Problem 3: Experiments with Weka

1

(a)

Algorithm Name	Where to find?	Train Accuracy
Decision Trees	J48 under trees	84.1%
1NN	IBK under Lazy	100%
Naive Bayes	Naive Bayes under bayes	76.3%
Logistic Regression	Logistic under functions	78.25%
SVM	SMO under functions	77.5%

(b)

The highest accuracy classifier is 1NN. With 100% accuracy it is almost certainly overfitting.

2

(a)

Learned decision boundary:

$(-0.1232 * \text{preg}) + (-0.0352 * \text{plas}) + (0.0133 * \text{pres}) + (-0.0006 * \text{skin}) + (0.0012 * \text{insu}) + (-0.0897 * \text{mass}) + (-0.9452 * \text{pedi}) + (-0.0149 * \text{age}) + 8.4047$

(b)

The most important feature is pedi, with -0.9452 weight.

(c)

3

(a)

```
         1.3614 * (normalized) preg
+        4.8764 * (normalized) plas
+       -0.8118 * (normalized) pres
+       -0.1158 * (normalized) skin
+       -0.1776 * (normalized) insu
+        3.0745 * (normalized) mass
+        1.4242 * (normalized) pedi
+        0.2601 * (normalized) age
-        5.1761
```

(b)

The most important feature is plas with 4.8764 weight.

4

(a)

Algorithm Name	10-fold CV Accuracy
Decision Trees	73.8281%
1NN	70.1823%
Naive Bayes	76.3021%
Logistic Regression	77.2135%
SVM	77.3438%

(b)

1NN is the largest change, from 100% to 70%. The 1NN is probably being overfit on training, so it makes sense that it goes to having one of the lowest accuracies.

5

(a)

The feature weights of the data mostly stay the same, with the notable exceptions of mass and pedi.
Mass stays mostly constant until 100, when it drops sharply, then raises again at 1000.
Pedi starts out extremely low and has a slight increase at 10, then a sharp one at 100, where it joins the rest of the numbers, or very close to it.

(b)

(c)

```
=== Confusion Matrix ===

   a   b   <-- classified as
 500   0 |   a = tested_negative
 268   0 |   b = tested_positive
```

The odd thing here is that absolutely no b's are picked as a classifier. This is probably for a similar reason as the KNN when set to a large number, the relevant differences between the instances are destroyed with such a high ridge value.

6

(a)

1: 70.18%
3: 72.65%
5: 73.17%

The trend is increasing in accuracy as the neighbors go up. This happens because the data is grouped by neighbors well.

(b)

```
=== Confusion Matrix ===
a     b     <-- classified as
500   0 |   a = tested_negative
268   0 |   b = tested_positive
```

Nothing was classified as b, that's the strangeness. This is because there are more a's in the set than b's, and K is huge, so it makes sense.

7

(a)

Algorithm Name	10-fold CV Accuracy
1NN	67.3177%
Naive Bayes	71.7448
Logistic Regression	77.2135%
SVM	77.474%

(b)

1NN and Naive Bayes drop in accuracy.
1NN is very sensitive to duplicates, and apparently Naive Bayes is too.

(c)

Logistic regression and SVM stay the same, and apparently aren't affected by replication.