Machine Learning	**Homework 1**
CSCI 5622 Fall 2017	**Due Time Sep 15, 2017**
Name: Payoj Jain	**CU identitykey: paja3104**

1 K-nearest Neighbor (40pts)

1.1 Programming questions

1.2 Analysis

1.2.1 What is the role of the number of training instances with accuracy?

Solution. The accuracy increases with the number of training instances and becomes almost constant (97.27%) after 50,000 training samples.

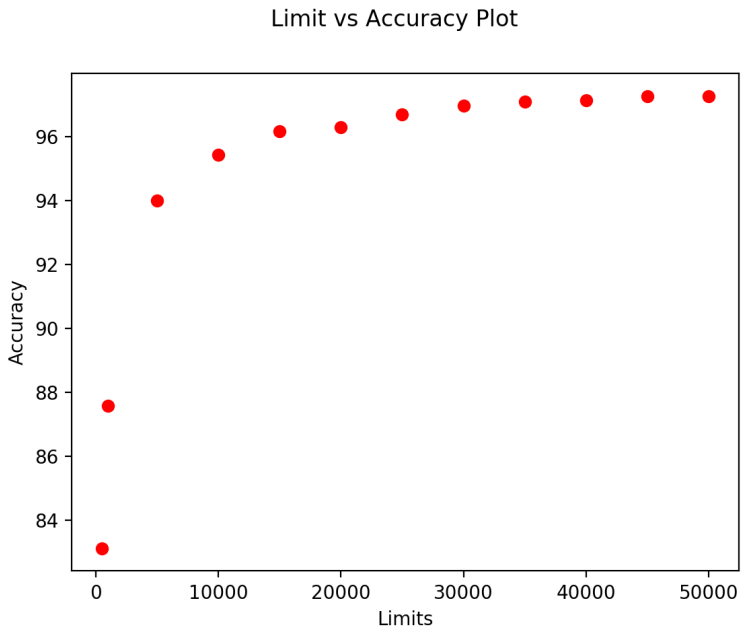

Figure 1: Graph shows that accuracy increases as number of training instances increases and becomes almost constant

1.2.2 What numbers get confused with each other most easily?

Solution. Pairs below get easily confused with each other (for 3 Nearest Neighbours and threshold 15):
(2,7); (4,9); (5,3); (5,6); (8,5)

```
    0   1   2   3   4   5   6   7   8   9
-------------------------------------------------
0:  982 0   3   0   0   0   2   1   1   2
1:  0   1060    1   0   1   0   1   1   0   0
2:  3   6   955 3   1   1   1   18  2   0
3:  0   0   4   1004    0   9   1   3   6   3
4:  0   9   0   0   951 0   0   4   0   19
5:  2   0   1   16  2   871 16  3   1   3
6:  1   0   0   0   0   2   964 0   0   0
7:  0   9   0   0   2   0   0   1073    0   6
8:  2   6   1   12  4   18  6   5   948 7
9:  2   2   0   8   11  5   0   11  3   919
Accuracy: 0.972700

Process finished with exit code 0
```

Figure 2: Confusion matrix created with 50,000 training instances and 3 Nearest Neighbours

1.2.3 What is the role of k with training accuracy?

Solution. The model overfits when k=1 as the training accuracy is 100%. Also, as k increases training accuracy decreases.

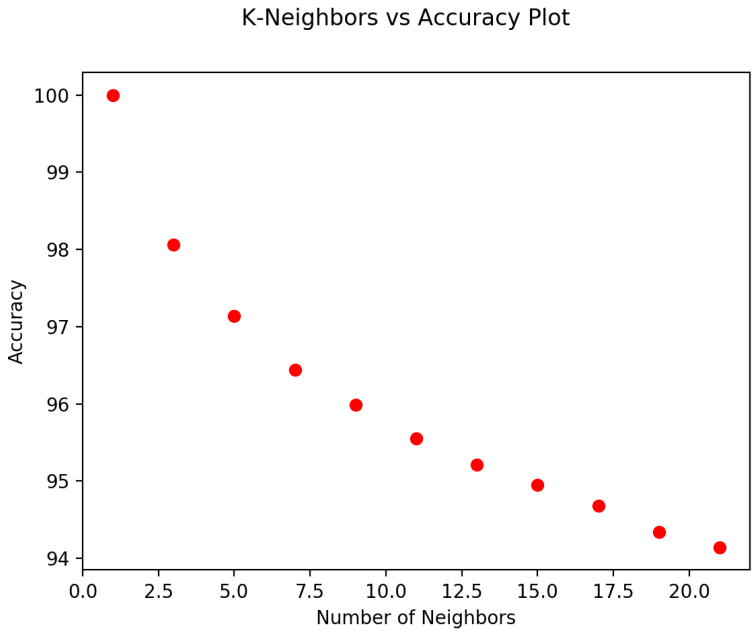

Figure 3: Graph shows that training accuracy decreases as number of nearest neighbours increases. Also, at k=1 training accuracy is 100%

1.2.4 In general, does a small value for k cause "overfitting" or "underfitting"?

Solution. From the above question, the model overfits when k=1 as the training accuracy is 100% and training accuracy decreases as we increase k.

2 Cross Validation (30pts)

2.1 Programming questions

2.2 Analysis

2.2.1 What is the best k chosen from 5-fold cross validation with "–limit 500"?

Solution. With "–limit 500" the best k chosen from 5-fold cross validation is 3. For 3 Nearest Neighbours, the test accuracy is 83.11%.

```
Working with 500 examples
1-nearest neighbor accuracy: 0.836000
3-nearest neighbor accuracy: 0.858000
5-nearest neighbor accuracy: 0.826000
7-nearest neighbor accuracy: 0.826000
9-nearest neighbor accuracy: 0.800000
Accuracy for chosen best k= 3: 0.831100


Process finished with exit code 0
```

Figure 4: K-cross validation result with limit = 500 training instances.

2.2.2 What is the best k chosen from 5-fold cross validation with "–limit 5000"?

Solution. With "–limit 5000" the best k chosen from 5-fold cross validation is 1. For 1 Nearest Neighbours, the test accuracy is 93.88%.

```
Working with 5000 examples
1-nearest neighbor accuracy: 0.941800
3-nearest neighbor accuracy: 0.937600
5-nearest neighbor accuracy: 0.930800
7-nearest neighbor accuracy: 0.927600
9-nearest neighbor accuracy: 0.925800
Accuracy for chosen best k= 1: 0.938800


Process finished with exit code 0
```

Figure 5: K-cross validation result with limit = 5000 training instances.

2.2.3 Is the best k consistent with the best performance k in problem 1?

Solution. No. Best k chosen in question 1 is k=3 while Best k chosen in question 2 is k=1. So, best k is not consistent with best performance k.

3 Bias-variance tradeoff (20pts)

Solution.

$$Err(x_0) = E[((y - h_s(x_0))^2)] = E[(f(x_0) + \epsilon - h_s(x_0))^2]$$
$$= E[(f(x_0) - h_s(x_0))^2 + \epsilon^2 + 2\epsilon(f(x_0) - h_s(x_0))]$$
$$= E[(f(x_0) - h_s(x_0))^2] + E[\epsilon^2] + E[2\epsilon(f(x_0) - h_s(x_0))]$$
$$= E[(f(x_0) - h_s(x_0))^2] + \sigma_\epsilon^2 + 0$$

Now, we know that $Var(X) = E[X^2] - E^2[X]$
Therefore, $E[X^2] = Var(X) + E^2[X]$
Hence, $E[(f(x_0) - h_s(x_0))^2] = Var(f(x_0) - h_s(x_0) + E^2[f(x_0) - h_s(x_0)]$
So, the Error equation becomes,

$$Err(x_0) = Var(f(x_0) - h_s(x_0)) + E^2[f(x_0) - h_s(x_0)] + \sigma_\epsilon^2$$

$$Var(f(x_0) - h_s(x_0)) = Var(h_s(x_0))$$

Therefore, $Err(x_0) = Var(h_s(x_0)) + E^2[f(x_0) - h_s(x_0)] + \sigma_\epsilon^2$
Substituting $h_s(x_0)$ *with* $\frac{1}{k}\sum_{l=1}^{k} y_{(l)}$ *we get,*

$$Err(x_0) = \sigma_\epsilon^2 + Var(\frac{1}{k}\sum_{l=1}^{k} y_{(l)}) + E^2[f(x_0) - \frac{1}{k}\sum_{l=1}^{k} y_{(l)}]$$

Now, $Var(\frac{1}{k}\sum_{l=1}^{k} y_{(l)}) = \frac{1}{k^2}Var(\sum_{l=1}^{k} y_{(l)}) == \frac{1}{k^2}Var(\sum_{l=1}^{k} f(x_{(l)}) + \epsilon_{(l)})$
Since, $\sum_{l=1}^{k} f(x_{(l)})$ *and* $\epsilon_{(l)}$ *are uncorrelated*
Therefore, $\frac{1}{k^2}Var(\sum_{l=1}^{k} f(x_{(l)}) + \epsilon_{(l)}) = \frac{1}{k^2}Var(\sum_{l=1}^{k} f(x_{(l)}) + Var(\epsilon_{(l)})$
Now, variance of all $\epsilon_{(l)}$ *is equal to variance of* σ_ϵ
Therefore, $\frac{1}{k^2}Var(\sum_{l=1}^{k} f(x_{(l)}) + Var(\epsilon_{(l)}) = \frac{1}{k^2}k\sigma_\epsilon^2$ *Now,*

$$E^2[f(x_0) - \frac{1}{k}\sum_{l=1}^{k} y_{(l)}] = (f(x_0) - E(\frac{1}{k}\sum_{l=1}^{k} y_{(l)}))^2$$

$$= (f(x_0) - E(\frac{1}{k}\sum_{l=1}^{k}(f(x_{(l)}) + \epsilon_{(l)})))^2$$

$$= (f(x_0) - \frac{1}{k}E(\sum_{l=1}^{k}(f(x_{(l)}) + \epsilon_{(l)})))^2 = (f(x_0) - \frac{1}{k}E[\sum_{l=1}^{k} f(x_{(l)})] + E[\epsilon_{(l)}])^2$$

Since, $E[\epsilon_{(l)}] = 0$ *and* $E[\sum_{l=1}^{k} f(x_{(l)})] = \sum_{l=1}^{k} f(x_{(l)})$
Therefore,

$$E^2[f(x_0) - \frac{1}{k}\sum_{l=1}^{k} y_{(l)}] = (f(x_0) - \frac{1}{k}\sum_{l=1}^{k} f(x_{(l)}))^2$$

Hence,

$$Err(x_0) = \frac{1}{k}\sigma_\epsilon^2 + (f(x_0) - \frac{1}{k}\sum_{l=1}^{k} f(x_{(l)}))^2 + \sigma_\epsilon^2$$