Richard Albright
HW4
ISYE-6740
Fall 2020

1. Basic optimization.

a. Show step-by-step mathematical derivation for the gradient of the cost function $\ell(\theta)$.

$$\ell(\theta) = \sum_{i=1}^{m} \left\{ -\log(1 + \exp\{-\theta x^i\}) + (y^i - 1)\theta x^i \right\}$$

$$= \sum_{i=1}^{m} \left\{ (y^i - 1)\theta x^i - \log(1 + e^{-\theta x^i}) \right\}$$

$$= \sum_{i=1}^{m} \left\{ y^i \theta x^i - \theta x^i - \log(1 + e^{-\theta x^i}) \right\}$$

$$= \sum_{i=1}^{m} \left\{ y^i \theta x^i - (\log e^{\theta x^i} + \log(1 + e^{-\theta x^i}) \right\}$$

$$= \sum_{i=1}^{m} \left\{ y^i \theta x^i - \log(1 + e^{\theta x^i}) \right\}$$

$$\frac{\partial}{\partial \theta_j} y^i \theta x^i = y^i x^i$$

$$\frac{\partial}{\partial \theta_j} log(1 + e^{\theta x^i}) = \frac{x_j^i e^{\theta x^i}}{1 + e^{\theta x^i}}$$

$$\frac{\partial \ell(\theta)}{\partial \theta} = \sum_{i=1}^{m} \left\{ y^i x^i - \frac{x^i e^{\theta x^i}}{1 + e^{\theta x^i}} \right\} = \sum_{i=1}^{m} \left\{ (y^i - 1)x^i - \frac{x^i e^{-\theta x^i}}{1 + e^{-\theta x^i}} \right\}$$

b. Write a pseudo-code for performing gradient descent to find the optimizer $\theta*$. This is essentially what the training procedure does.

Initialize the error term ϵ
Initiliaze the learning rate α
Initiliaze θ randomly
whilte True:
    $\theta_t = \theta_{t-1} - \alpha\Delta(\theta_{t-1})$
    if $\| \theta_t - \theta_{t-1} \| < \epsilon$:
        break

c. Write the pseudo-code for performing the stochastic gradient descent algorithm to solve the training of logistic regression problem (1). Please explain the difference between gradient descent and stochastic gradient descent for training logistic regression.

Initialize the error term ϵ
Initiliaze the learning rate α
Initiliaze θ randomly
whilte True:
    i = random index of data points in the training set between 1 and m
    $\theta_t = \theta_{t-1} - \alpha\Delta(\theta_{t-1}, x_i, y_i)$
    if $\| \theta_t - \theta_{t-1} \| < \epsilon$:
        break

Gradient Descent has to calculate all gradients in the training set on each iteration, while Stochastic gradient descent takes a random sample of points until converging. Stochastic Gradient Descent will take less memory, but is a noisier algorithm, and can lead to noisier approximations that oscillate around the minimum until converging.

d. We will show that the training problem in basic logistic regression problem is concave. Derive the Hessian matrix of `(θ) and based on this, show the training problem (1) is concave (note that in this case, since we only have one feature, the Hessian matrix is just a scalar). Explain why the problem can be solved efficiently and gradient descent will achieve a unique global optimizer, as we discussed in class.

$$\frac{\partial \ell(\theta)}{\partial \theta} = \sum_{i=1}^{m} \left\{ y^i x^i - \frac{x^i e^{\theta x^i}}{1 + e^{\theta x^i}} \right\}$$

$$\frac{\partial}{\partial \theta} x^i y^i = 0$$

$$\frac{\partial}{\partial \theta} \left\{ -\frac{x^i e^{\theta x^i}}{1 + e^{\theta x^i}} \right\} = \frac{(x^i)^2 e^{\theta 2 x^i}}{\left(e^{\theta x^i} + 1\right)^2} - \frac{(x^i)^2 e^{\theta x^i}}{e^{\theta x^i} + 1} = -\frac{(x^i)^2 e^{\theta x^i}}{\left(e^{\theta x^i} + 1\right)^2}$$

$$\frac{\partial \partial \ell(\theta)}{\partial \partial \theta} = -\sum_{i=1}^{m} \frac{(x^i)^2 e^{\theta x^i}}{\left(e^{\theta x^i} + 1\right)^2}$$

The function is concave because of the squared terms in both the numerator and denominator. The resulting function is the 2nd order derivative of the sigmoid function which results in a bell curve. It has a single global minimum.

2. Comparing Bayes, logistic, and KNN classifiers.

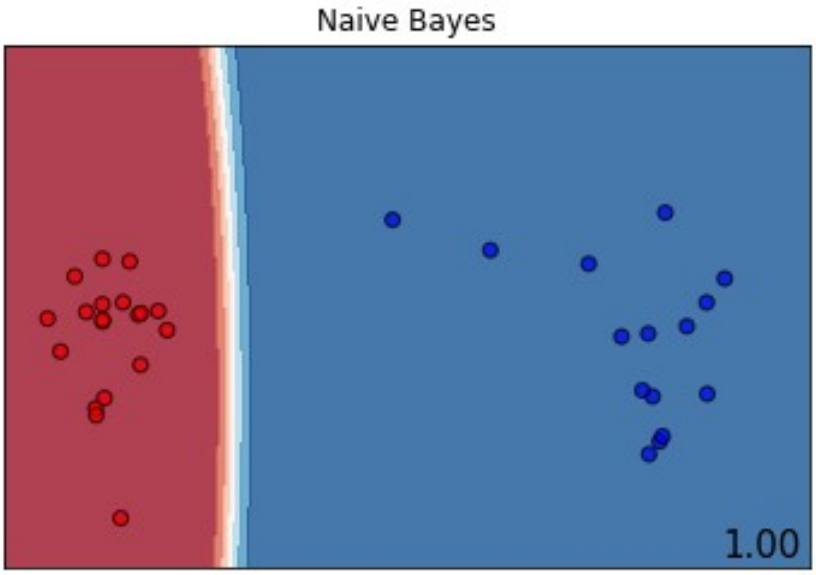
Part One (Divorce classification/prediction).

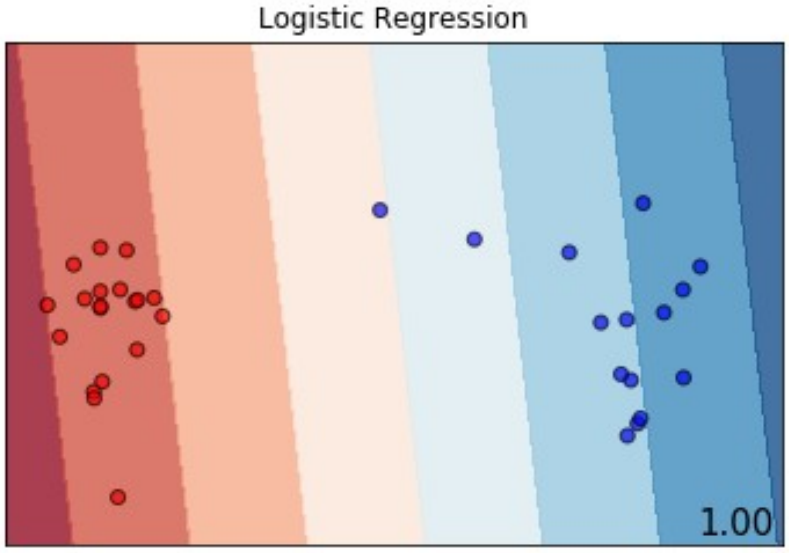
a. Report testing accuracy for each of the three classifiers. Comment on their performance: which performs the best and make a guess why they perform the best in this setting.

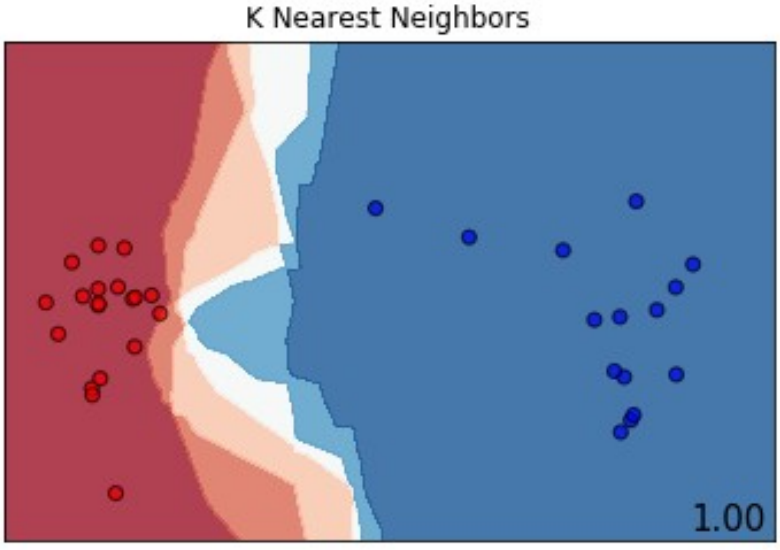
Classifier	Accuracy
Naive Bayes	100%
Logistic Regression	94.12%
K Nearest Neighbors	100%

Both Naive Bayes and K Nearest Neighbors perform the best in this setting, with a 100% Accuracy Rate. Logistic Regression does not perform as well. There is one point that is misclassified.

b. Now perform PCA to project the data into two-dimensional space. Plot the data points and decision boundary of each classifier. Comment on the difference between the decision boundary for the three classifiers. Please clearly represent the data points with different labels using different colors.







Using PCA to reduce to 2 dimensions results in all classifiers achieving 100% Accuracy. The decision boundary for Naive Bayes is smooth, slightly curved and also narrow. The decision boundary for Logistic Regression is Linear. The K nearest neighbor decision boundary is jagged with k=5 and has a wider boundary than Naive Bayes. The K nearest Neighbor decision boundary gets more smooth as K increases.

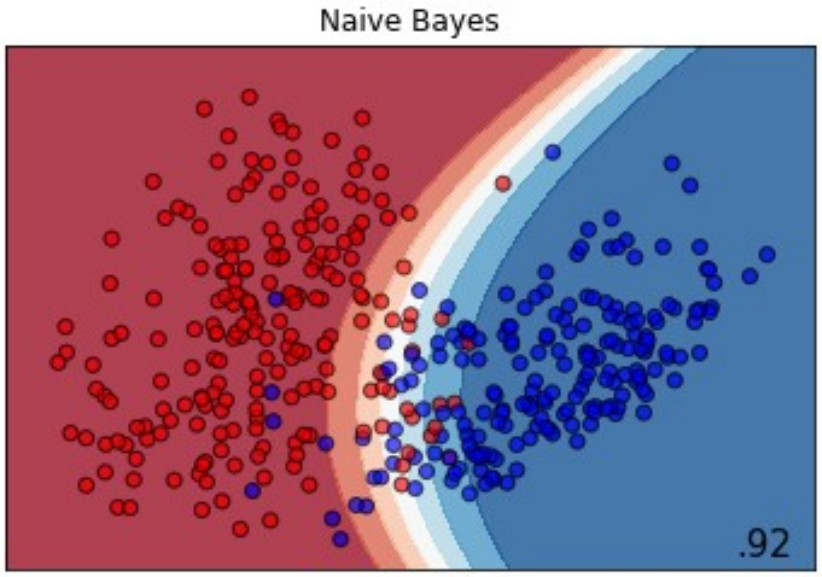
Part Two (Handwritten digits classification).

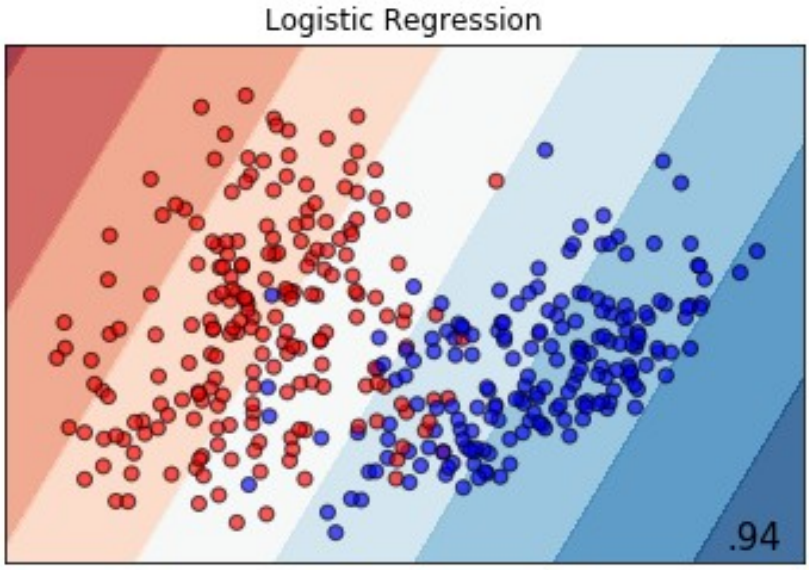
a. Report testing accuracy for each of the three classifiers. Comment on their performance: which performs the best and make a guess why they perform the best in this setting.

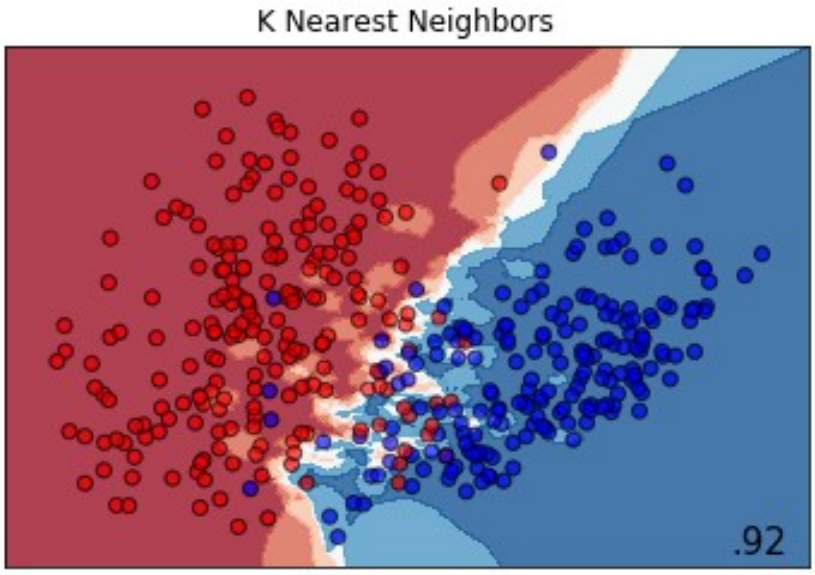
Classifier	Accuracy
Naive Bayes	91.96%
Logistic Regression	96.48%
K Nearest Neighbors	98.74%

The K Nearest Neighbor classifier performs the best with an accuracy rate of 98.74%. Since this problem uses image data, K nearest neighbors likely performs the best because it is clustering pixel data around each pixel. Naive Bayes performs much worse than both Logistic Regression and K Nearest Neighbors.

b. Now perform PCA to project the data into two-dimensional space. Plot the data points and decision boundary of each classifier. Comment on the difference between the decision boundary for the three classifiers. Please clearly represent the data points with different labels using different colors.







Projecting the MNIST data into a reduced 2 dimensional space results in lower classification accuracy for all classifiers. In the reduced space, Logistic Regression performs the best with a 94% accuracy rate. The decision boundary for Logistic Regression is linear, for Naive Bayes it is curved and smooth. K Nearest Neighbors decision boundary with k=5 is extremely jagged. Both Naive Bayes and K Nearest Neighbors have an accuracy rate of 92%.

3. Naive Bayes for spam filtering

a. Calculate class prior P(y = 0) and P(y = 1) from the training data, where y = 0 corresponds to spam messages, and y = 1 corresponds to non-spam messages. Note that these class prior essentially corresponds to the frequency of each class in the training sample. Write down the feature vectors for each spam and non-spam messages.

Priors:

```
P(y=0) Spam: 0.42857142857142855
P(y=1) Not Spam:  0.5714285714285714
```

Spam

['million dollar offer','secret offer today','secret is secret']

Feature Vectors:

	secret	offer	low	price	valued	customer	today	dollar	million	sports	is	for	play	healthy	pizza
0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	1.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0
1	1.0	1.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	2.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0

Not Spam

['low price for valued customer','play secret sports today','sports is healthy','low price pizza']

Feature Vectors:

	secret	offer	low	price	valued	customer	today	dollar	million	sports	is	for	play	healthy	pizza
0	0.0	0.0	1.0	1.0	1.0	1.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0
1	1.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	1.0	0.0	0.0	1.0	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	1.0	0.0	0.0	1.0	0.0
3	0.0	0.0	1.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0

b. (In this example, m = 7.) Calculate the maximum likelihood estimates of $\theta_{0,1}$, $\theta_{0,7}$, $\theta_{1,1}$, $\theta_{1,15}$ by maximizing the log-likelihood function above. (Hint: We are solving a constrained maximization problem: you can introduce Lagrangian multipliers, or directly substitute the $\theta_{0,k} = 1 - \theta_{1,k}$ into the objective function so you do not need to worry about the constraint.)

```
theta(0,1) = P(secret | spam) = 0.3333333333333333
theta(0,7) = P(today | spam) = 0.1111111111111111
theta(1,1) = P(secret | non-spam) = 0.06666666666666667
theta(1,15) = P(pizza | non-spam) = 0.06666666666666667
```

c. Given a test message “today is secret”, using the Naive Bayes classier that you have trained in Part (a)-(b), to calculate the posterior and decide whether it is spam or not spam.

```
Probability that 'today is secret' is spam: 0.004115226337448559
Probability that 'today is secret' is not spam 0.0002962962962962963

'today is secret' is categorized as spam since spam has the higher probability.
```