ISYE6740-HW5

pkubsad

March 2021

1. Part One (Divorce classification/prediction). (30 points)

   (a) (15 points) 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.

   Answer: I have written the code and based on the output, I have the following accuracy for different classifiers. Gaussian Naive Bayes and KNN classifiers perform slightly better than logistic regression. One reason KNN might be performing better than LR is because KNN is not factor dependent. There might not be co-linearity among the factors and the number of train data is not that high, this makes NB perform better than LR.

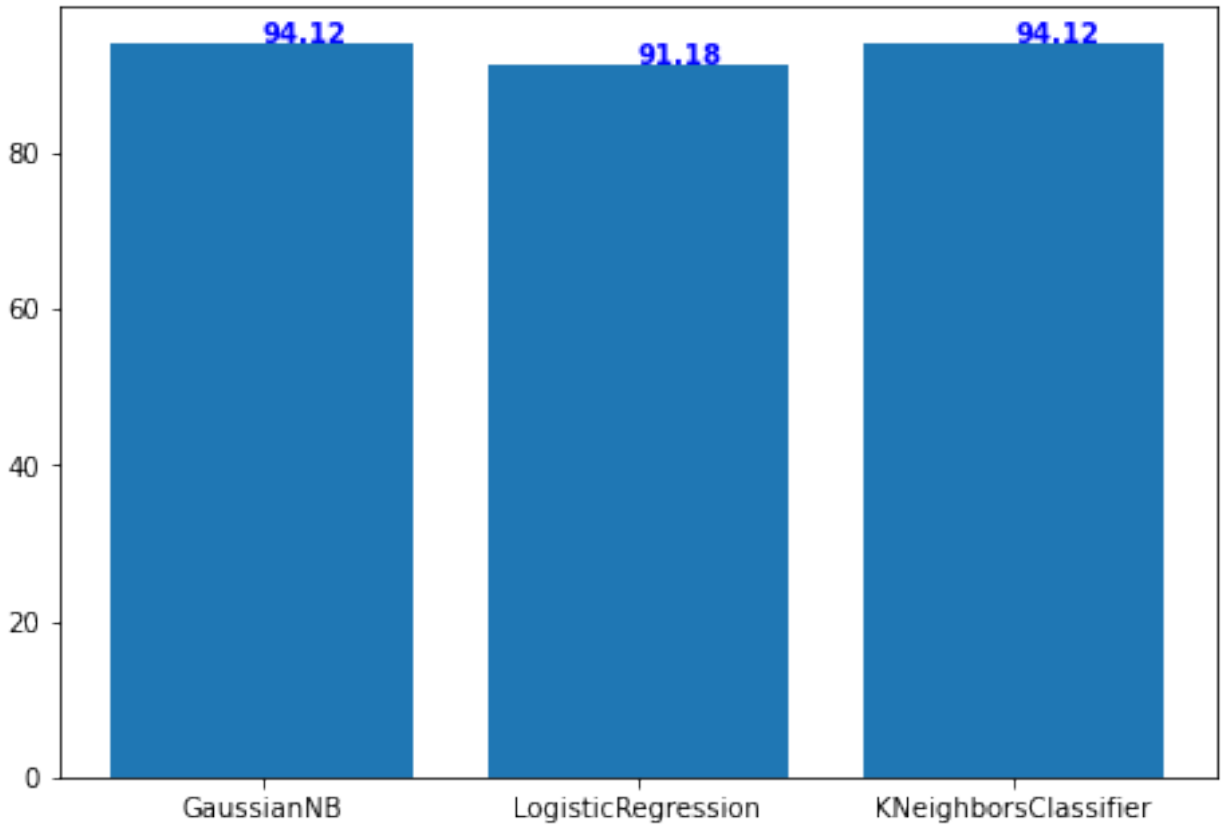

Figure 1: Accuracies of different classifiers

(b) (15 points) Now perform PCA to project the data into two-dimensional space. Build the classifiers (**Naive Bayes, Logistic Regression, and KNN**) using the two-dimensional PCA results. Plot the data points and decision boundary of each classifier in the two-dimensional space. Comment on the difference between the decision boundary for the three classifiers. Please clearly represent the data points with different labels using different colors.

Answer: Based on the output below, LR performs better on the reduced dataset. But, the classifier is in the proximity of 2 classes, it is prone to cause more classification errors on increased data sets.KNN classifier, even though it mis-classifies couple of data points, it is far from both classes and would perform even if data changes slightly.

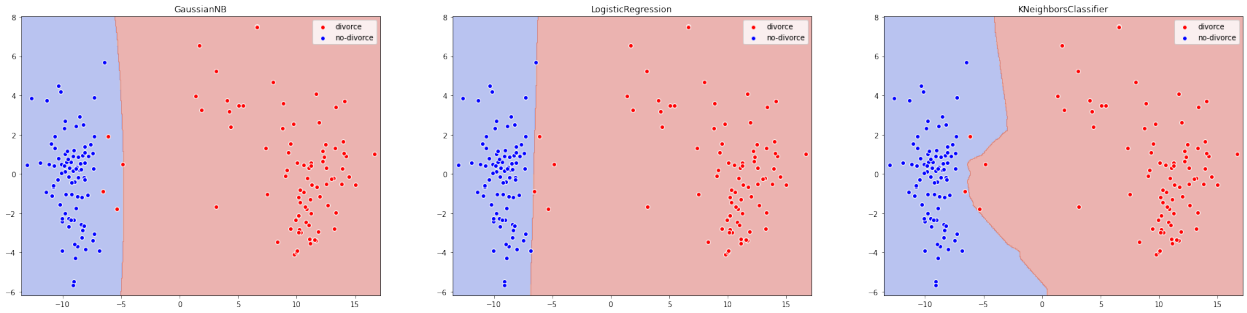

Figure 2: Classification using reduced features by PCA

2. Part Two (Handwritten digits classification). (35 points)

(a) (25 points) Report confusion matrix, precision, recall, and F-1 score for each of the classifiers. For precision, recall, and F-1 score of each classifier, we will need to report these for each of the digits. So you can create a table for this. For this question, each of the 5 classifier, **KNN, logistic regression, SVM, kernel SVM, and neural networks**, accounts for 10 points.

Answer: Please find below summary of classification report on the dataset by different classifier.
For KNN, I have run cross validation to determine optimal value of K. Best performance was observed at **k=6**. I have commented this part of code to avoid it running again and again.

For KSVM, I have used the following formula to start fine tuning the gamma parameter:

$$\gamma = \frac{1}{2\sigma^2}$$

src: https://stats.stackexchange.com/questions/317391/gamma-as-inverse-of-the-variance-of-rbf-kernelhyperref

	KNN			LOGISTIC REGRESSION			SVM			KSVM			NEURAL NETWORK		
	precision	recall	f1-score	precision	recall	f1-score	precision	recall	f1-score	precision	recall	f1-score	precision	recall	f1-score
0	0.94	0.99	0.96	0.94	0.92	0.93	0.94	0.97	0.95	1	0.9	0.94	0.98	0.99	0.98
1	0.88	0.99	0.93	0.96	0.97	0.97	0.96	0.99	0.97	1	0.96	0.98	1	0.98	0.99
2	0.98	0.89	0.93	0.88	0.87	0.87	0.9	0.92	0.91	0.48	0.99	0.65	0.97	0.97	0.97
3	0.92	0.94	0.93	0.83	0.86	0.85	0.88	0.9	0.89	0.93	0.88	0.91	0.94	0.97	0.95
4	0.96	0.91	0.93	0.88	0.91	0.89	0.89	0.95	0.92	0.96	0.82	0.88	0.98	0.96	0.97
5	0.93	0.91	0.92	0.8	0.83	0.82	0.88	0.85	0.86	0.93	0.89	0.91	0.97	0.96	0.97
6	0.96	0.97	0.97	0.91	0.92	0.91	0.94	0.94	0.94	0.99	0.76	0.86	0.97	0.98	0.98
7	0.92	0.93	0.92	0.91	0.88	0.89	0.93	0.9	0.91	0.99	0.81	0.89	0.97	0.97	0.97
8	0.97	0.86	0.91	0.82	0.79	0.8	0.92	0.85	0.88	0.94	0.81	0.87	0.98	0.94	0.96
9	0.9	0.92	0.91	0.87	0.85	0.86	0.9	0.88	0.89	0.96	0.81	0.88	0.94	0.98	0.96

Figure 3: Classification report for all classifiers

(b) (10 points) Comment on the performance of the classifier and give your explanation why some of them perform better than the others.

Answer: From the above table, we see the best performance is neural network classifier. We can attribute this to gradient descent at each node. With 10 nodes, and just 1 layer, we see such an improved performance.If we increase the number of layers, the accuracy will improve further. Also, KNN and SVM are not performing well with full data and requires us to down sample the data.

(2.1) (5 points) Calculate class prior $\mathbb{P}(y = 0)$ and $\mathbb{P}(y = 1)$ 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.

Answer:
There are 3 spam messages out of 7 so $\mathbb{P}(y = 0) = 3/7 = 0.429$
There are 4 non-spam messages out of 7 so $\mathbb{P}(y = 1) = 4/7 = 0.571$

Feature Vector for spam:

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

Feature Vector for non-spam:

	secret	offer	low	price	valued	customer	today	dollar	million	sports	is	for	play	healthy	pizza
4	0	0	1	1	1	1	0	0	0	0	0	1	0	0	0
5	1	0	0	0	0	0	1	0	0	1	0	0	1	0	0
6	0	0	0	0	0	0	0	0	0	1	1	0	0	1	0
7	0	0	1	1	0	0	0	0	0	0	0	0	0	0	1

(2.2) 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.

Answer: Source:https://www.cs.cornell.edu/courses/cs5740/2017sp/res/nb-prior.pdf

$$\ell(\theta_{0,1}, \ldots, \theta_{0,d}, \theta_{1,1}, \ldots, \theta_{1,d}) = \sum_{i=1}^{m} \sum_{k=1}^{d} x_k^{(i)} \log \theta_{y^{(i)},k}$$

We can use the method of Langrangian multipliers to solve the problem, which makes us to maximize the following langrangian function:

$$J = \sum_i \sum_k x_k^{(i)} log\theta_{y^{(i)},k} + \lambda(\sum_k \theta_{y^{(i)},k} - 1)$$

By taking first derivative wrt $\theta_{y^{(i)},k}$ and set it 0, we have

$$\nabla_{\theta_y^{(i)},k} J = \sum_i \frac{x_k^{(i)}}{\theta_y^{(i)}, k} + \lambda = 0$$

We can get then get $-\lambda = \sum_i \sum_k x_k^{(i)} = \sum_i = |D|$, the total number of objects in datapoints.By plugging in lambda we get:

$$\theta_y^{(i)}, k = \frac{\sum_i x^{(i)}}{|D|}$$

This denotes the total number of objects in D that belong to class C_k.
Using the above calculation, calculations for

$$\theta_{0,1} = \theta_{spam,secret} = \frac{\text{number of times 'secret' shows in spam}}{\text{number of words in spam messages}} = \frac{3}{9} = 0.333$$

$$\theta_{0,7} = \theta_{spam,today} = \frac{1}{9} = 0.111$$

$$\theta_{1,1} = \theta_{non-spam,secret} = \frac{1}{15} = 0.0667$$

$$\theta_{1,15} = \theta_{non-spam,pizza} = \frac{1}{15} = 0.0667$$

(2.3) (15 points) 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.

Answer: To figure out if the test string is spam or non-spam, we can run the probabilities calculated in the previous step.

$$\theta_{spam,\text{today is secret}} = \theta_{spam,today} * \theta_{spam,is} * \theta_{spam,secret}$$

$$\theta_{spam,\text{today is secret}} = 0.111 * 0.111 * 0.333 = 0.00410$$

$$\theta_{non\_spam,\text{today is secret}} = \theta_{non\_spam,today} * \theta_{non\_spam,is} * \theta_{non\_spam,secret}$$

$$\theta_{non\_spam,\text{today is secret}} = 0.0667 * 0.0667 * 0.0667 = 0.000297$$

The probability that the test message is spam is higher than non_span. We can conclude that this test message is more likely a spam.