Advanced Machine Learning – Homework 3

Anna Berger, Ashwin Sadananda Bhat (AML 25)

December 22, 2017

Exercise 1

To investigate the effect of dimensionality reduction and PCA we apply the Bayesian approach to the predict whether a person has breast cancer or not. There are two possible classes: benign (labelled as 2) and malicious (labelled as 4). In the assignment we refer to them as B and M respectively.
From the data we can estimate the prior distributions of these classes by counting the number of B's and M's and normalising them by the total number of people:

$$priors \approx (0.65, 0.35)$$

Hereafter we define the error as the ratio of misclassified objects.

Part a

1. **Univariate classifiers**

   (a) **x2 (Clump Thickness) only**
   In this case, the feature is the second column of the dataset which represents clump thickness. First, we estimate the mean and the variance of this feature for each class:

$$\bar{x}_{2B} \approx 2.936, \quad var(x_{2B}) \approx 2.739,$$

$$\bar{x}_{2M} \approx 7.133, \quad var(x_{2M}) \approx 6.215.$$

   Then we fit the Gaussians using these estimations and apply the Bayes rule to classify objects. The statistics for this classifier can be seen from the Table 1:

	Train	**Validation**
Error	0.137	0.146
Precision (malicious)	0.907	0.868
Recall (malicious)	0.678	0.6875

Table 1: Univariate classifier (feature x2).

   The confusion matrices for train and validation sets are shown in the Figures 1 and 2.

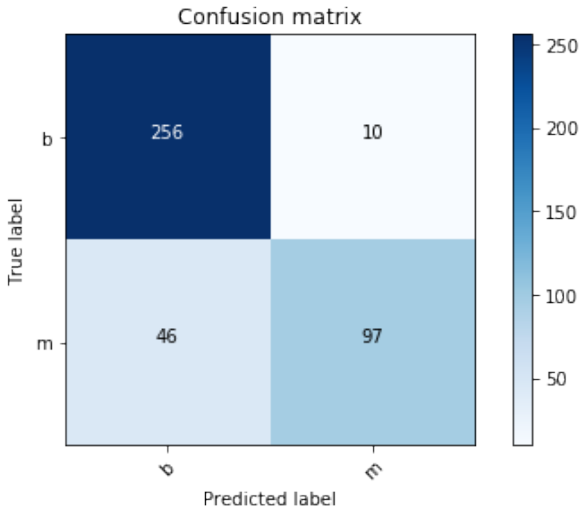

Figure 1: Feature x2, train

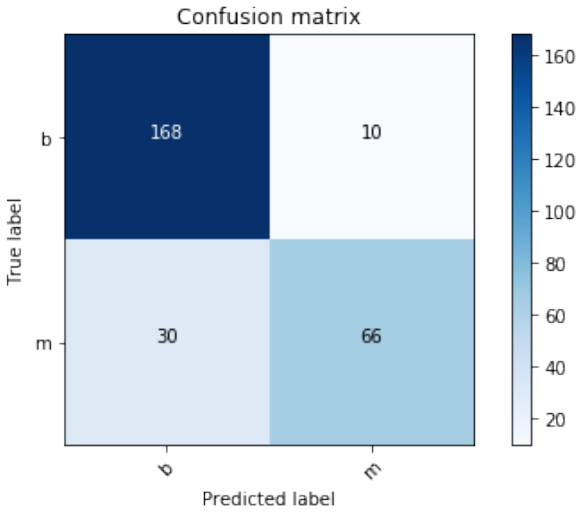

Figure 2: Feature x2, val

(b) **First principal component**

We perform SVD on the covariance matrix for the total dataset and obtain eigenvalues (diagonal matrix s) and eigenvectors for it (columns of matrix U). The first column of matrix U is the first principal component which covers 68,7% of the variance of the dataset and looks as follows:

$$[-0.3059, -0.4085, -0.3859, -0.3280, -0.2293, -0.4522, -0.2830, -0.3581, -0.1329]^\tau$$

Then, we project all the datapoints on the obtained vector.

After that, we estimate the mean and the variance of projected dataset:

$$\bar{x}_{pca1B} \approx -4.718, \quad var(x_{pca1B}) \approx 3.566,$$

$$\bar{x}_{pca1M} \approx -17.729, \quad var(x_{pca1M}) \approx 19.710.$$

Then we fit the Gaussians using these estimations and apply the Bayes rule to classify objects. The statistics for this classifier can be seen from the Table 2:

	Train	**Validation**
Error	0.037	0.014
Precision (malicious)	0.932	0.969
Recall (malicious)	0.965	0.989

Table 2: Univariate classifier (first principal component).

The confusion matrices for train and validation sets are shown in the Figures 3 and 4.

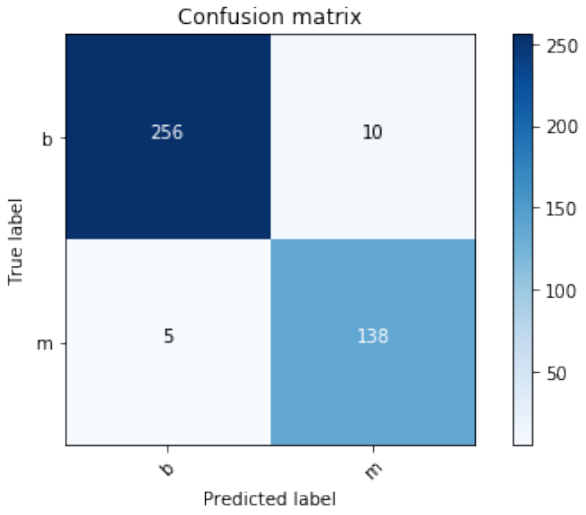

Figure 3: The first principal component, train set

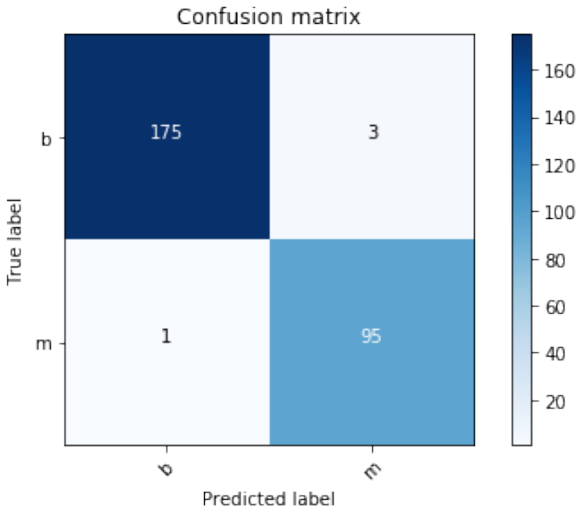

Figure 4: The first principal component, val set

2. **Bivariate classifiers**

(a) **x2 and x7 (Clump Thickness and Bare Nuclei)**
In this case, the features are the second and the seventh columns of the dataset which represent clump thickness and bare nuclei.
First, we estimate the mean and the covariance of this feature for each class:

$$\bar{x}_{27B} \approx (2.936, 1.365), \quad \Sigma_{27B} \approx \begin{bmatrix} 2.739 & 0.220 \\ 0.220 & 1.470 \end{bmatrix}.$$

$$\bar{x}_{27M} \approx (7.133, 7.706), \quad \Sigma_{27M} \approx \begin{bmatrix} 6.215 & 0.082 \\ 0.082 & 9.082 \end{bmatrix}.$$

Then we fit the Gaussians using these estimations and apply the Bayes rule to classify objects. The statistics for this classifier can be seen from the Table 3:

	Train	**Validation**
Error	0.059	0.054
Precision (malicious)	0.916	0.918
Recall (malicious)	0.916	0.927

Table 3: Bivariate classifier (features x2, x7).

The confusion matrices for train and validation sets are shown in the Figures 5 and 6.

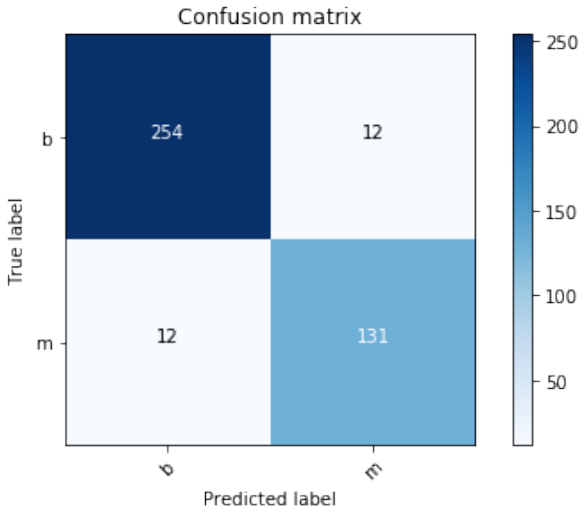

Figure 5: Features x2, x7, train

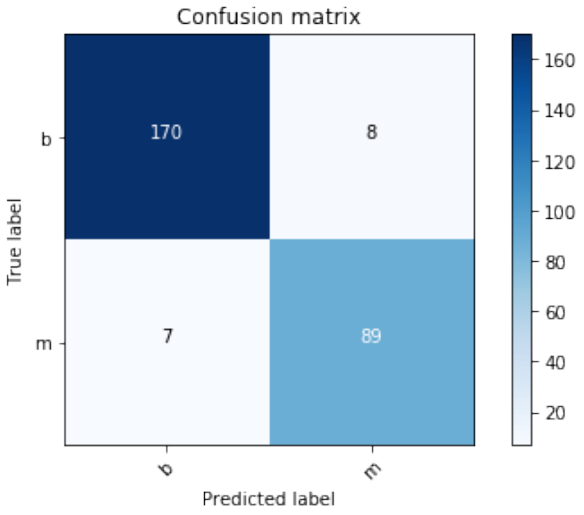

Figure 6: Features x2, x7, val

(b) **First two principal components**

Now we take first two columns of the matrix U — first two principal components. They cover 76.1% of the variance of the dataset. The first principal component is already shown in a previous part, the second principal component is given by:

$$[0.0756, 0.2158, 0.1710, -0.2682, 0.1916, -0.7441, 0.0371, 0.4670, 0.1917]^\tau$$

Then, we project all the datapoints on these vectors.

After that, we estimate the mean and the covariance of projected dataset:

$$\bar{x}_{pca12B} \approx (-4.718, 0.659), \quad \Sigma_{pca12B} \approx \begin{bmatrix} 3.567 & -0.129 \\ -0.129 & 0.932 \end{bmatrix}.$$

$$\bar{x}_{pca12M} \approx (-17.729, 0.210), \quad \Sigma_{pca12M} \approx \begin{bmatrix} 19.709 & -3.593 \\ -3.593 & 12.923 \end{bmatrix}.$$

Then we fit the Gaussians using these estimations and apply the Bayes rule to classify objects.

	Train	**Validation**
Error	0.037	0.021
Precision (malicious)	0.927	0.959
Recall (malicious)	0.972	0.979

Table 4: Bivariate classifier (first two principal components).

The confusion matrices for train and validation sets are shown in the Figures 7 and 8.

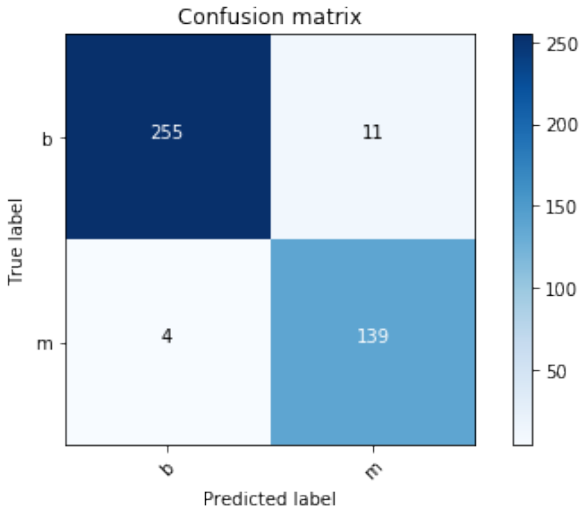

Figure 7: First two principal components, train set

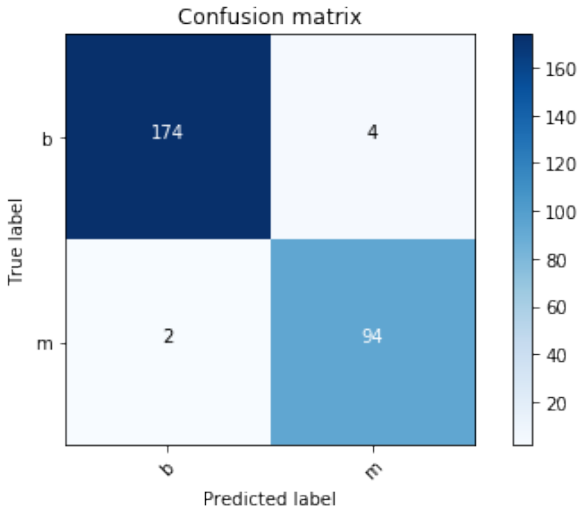

Figure 8: First two principal components, val set

3. **Multivariate classifiers**

(a) **The principal components which explain 80% of the variance**
We found out that the first three principal components cover 80% of the variance of the whole dataset, precisely 82.3%. The third principal component is given by:

$$[0.8107, -0.0523, 0.0440, -0.5190, -0.1229, 0.1079, -0.0952, -0.1489, -0.1033]^\tau$$

We repeat the described procedure: project all the data points on the first three eigenvectors and estimate the mean and the covariance of projected dataset for both classes:

$$\bar{x}_{pca123B} \approx (-4.718, 0.659, 1.059), \quad \Sigma_{pca123B} \approx \begin{bmatrix} 3.567 & -0.129 & -0.635 \\ -0.129 & 0.932 & 0.312 \\ -0.635 & 0.312 & 1.691 \end{bmatrix}.$$

$$\bar{x}_{pca123M} \approx (-17.729, 0.210, 1.437), \quad \Sigma_{pca123M} \approx \begin{bmatrix} 19.709 & -3.593 & 4.411 \\ -3.593 & 12.923 & -0.471 \\ 4.411 & -0.471 & 9.053 \end{bmatrix}.$$

The statistics for this classifier can be seen from the Table 5:

	Train	**Validation**
Error	0.037	0.022
Precision (malicious)	0.921	0.959
Recall (malicious)	0.979	0.979

Table 5: First three principal components.

The confusion matrices for train and validation sets are shown in the Figures 9 and 10.

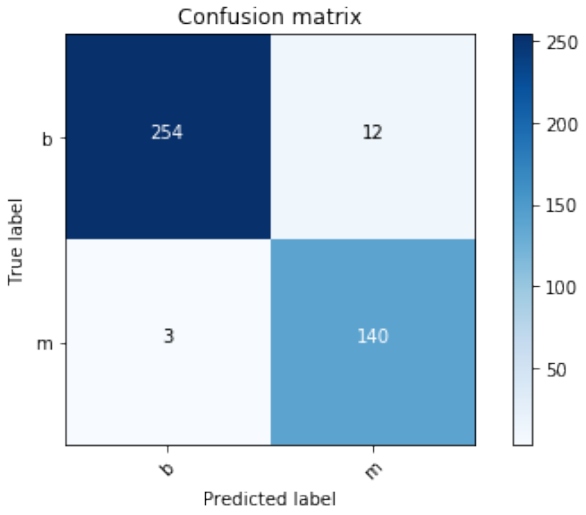

Figure 9: First two principal components, train set

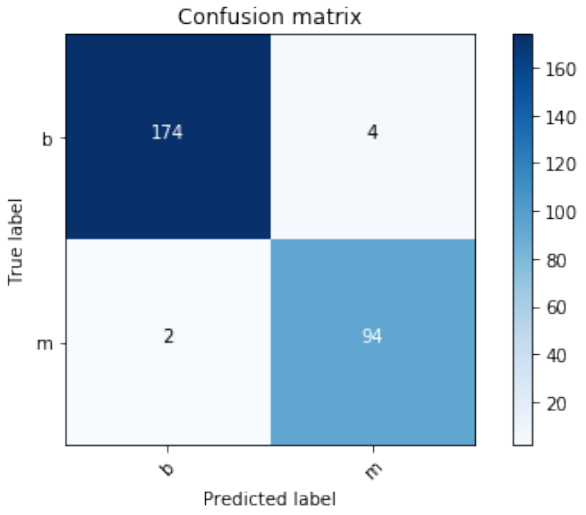

Figure 10: First two principal components, val set

(b) **All inputs**

In this case the features are all the columns from the second to the tenth. We again estimate the mean and the covariance of the dataset (the number of dimensions is equal to 9, that's why we decided not to overload the report with the matrix 9x9), then fit the Gaussians for both classes and apply the Bayes rule to classify the objects.

The statistics for this classifier can be seen from the Table 6:

	Train	**Validation**
Error	0.048	0.033
Precision (malicious)	0.897	0.931
Recall (malicious)	0.972	0.979

Table 6: Multivariate classifier (all inputs).

The confusion matrices for train and validation sets are shown in the Figures 11 and 12.

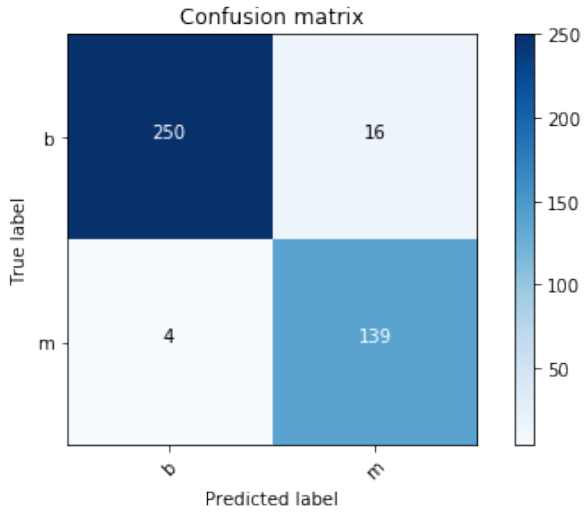

Figure 11: All inputs, train set

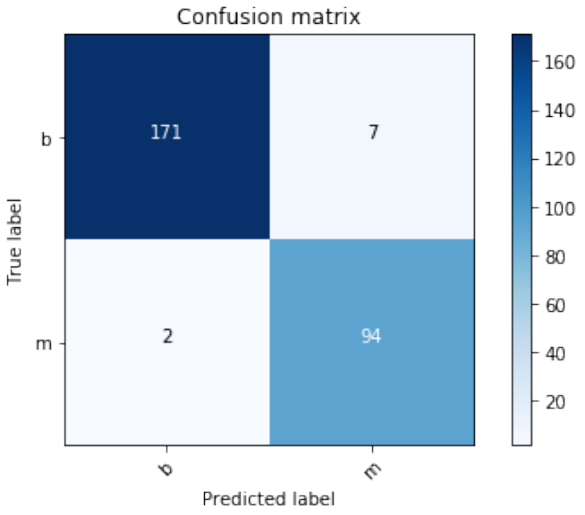

Figure 12: All inputs, val set

Part b

In this exercise we explored several versions of the classifier. The univariate classifier using one raw feature from the data showed the worst performance, as it didn't contain enough information for good prediction. Moreover, it's always hard to choose the most representative feature manually. The bivariate classifier using two features performed much better, but it wasn't in the top according to validation error (and the problem of choosing the most representative pair of feature is even harder).
We obtained the lowest validation error on the validation set with the approach using the first principal component. We think it is a reasonable balance between a number of parameters to estimate (it is already enough to capture the patterns in the data and small enough to avoid overfitting to the training data). We got slightly bigger validation error with the first two and three principal components, however, we don't need to add more features as they don't improve the results.
In the task of cancer detection, recall for malicious class is extremely important. The best approach with the first principal component showed the best recall for the malicious class as well as the best precision for this class again proving that the first principal component covers the majority of important information in this dataset.
In the last experiment with all inputs, we saw that even using all raw features from the dataset, we can't achieve the same performance as when applying PCA to extract information from the data.

Exercise 2

Part a

In this exercise we apply the logistic classifier (the implementation from scikit-learn python library, namely LogisticRegression) to this problem. The regularization parameter (for l2 regularization) is set to 1. Our training set consists of all provided features (9 columns). The adaptation of inputs (concatenation 1 as the first coordinate of inputs) and outputs (transformation from probabilities to labels 0 and 1) is done inside the algorithm provided by scikit-learn, so we don't have to do it manually.
The results for the logistic classifier can be seen from the Table 7:

	Train	Validation
Error	0.034	0.021
Precision (malicious)	0.957	0.978
Recall (malicious)	0.944	0.958

Table 7: Logistic classifier (all inputs).

The confusion matrices for train and validation sets are shown in the Figures 13 and 14.

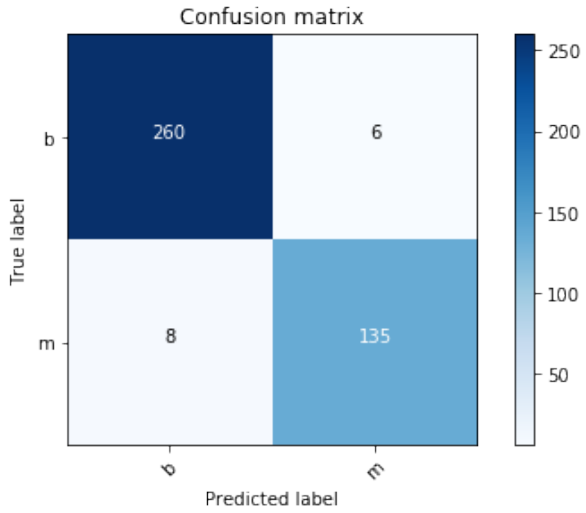

Figure 13: Logistic classifier, train set

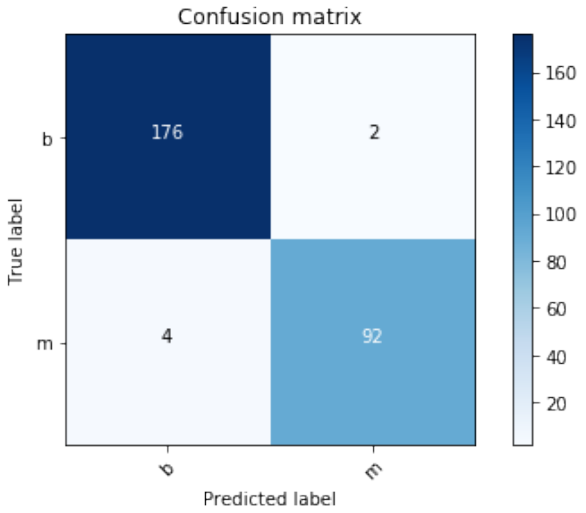

Figure 14: Logistic classifier, val set

Logistic classifier achieves the same level of accuracy as the best approach from the previous exercise — the approach with first two principal components obtained the lowest error 0.021 on the validation set. However, the recall on the validation set is slightly higher for the Bayesian approach with the first two principal components and we suppose that recall for malicious class is essential for the problem of cancer detection.

Part b

In Exercise 3 we used a generative approach to classify the objects: we explicitly modelled the distributions of classes $p(x|B)$ and $p(x|M)$ to determine class-conditional densities for each class individually. To get posterior probabilities, we applied Bayes theorem. Having found posterior probabilities, we used decision theory to determine class membership for each new input. It is rather demanding approach in which one need to estimate a lot of parameters $\left(n + \frac{n(n+1)}{2}\right)$ in this particular case, where n is the number of features), therefore, we need a large training set. However, it can be useful when we need to sample new datapoints from the distribution.

In Exercise 4 we used discriminative approach to modelling: we first determined the posterior class probabilities $p(B|x)$ and $p(M|x)$ and then used standard decision theory to assign each new input its category. The advantage of this approach is that it is less computationally expensive and requires less parameters to estimate, which may lead to a better predictive performance (especially under the lack of data conditions). On the other hand, by using this approach, we lose some information about structure of density probabilities.