REPORT SUBMITTED BY:

KETAN WALIA

Data Description:

The dataset used is named "HW1Dataset.mat." This dataset has data from an accelerometer measuring 23,040 time periods with 8 features to predict an outcome. There are 8 features; the features are as follows:

• The first two features represent the "moving average" of the signal over the time period

• The next three features are wavelet analyses (specifically Daubechies level 5 wavelet transform) on the x, y and z axis of an accelerometer.

• The next three features are the results of a power spectrum density analysis on each of the three accelerometer axis.

The variable labels is 1 for the positive class (that we are trying to predict) and 0 for the negative class. The variable cvind is a random assignment of each data point into one of 10 buckets for a 10-fold cross-validation analysis. I will compare the performance of six modeling algorithms for classifying this data;

1. Logistic Regression (LR)

2. Gaussian Naïve Bayes (GNB)

3. Linear Discriminant (LD)

4. Support Vector Machine (SVM) with Linear Kernel

5. SVM with RBF (non linear) kernel

6. Artificial Neural Network (ANN)

The following steps have been taken in order:

For each algorithm, the performance is evaluated via 10-Fold cross validation. I have trained and tested 10 different models for each algorithm as suggested. At the end section the average total accuracy, sensitivity and specificity for both training and test sets for each algorithm is reported.

1.1 The data "HW1Dataset.mat" was loaded into Matlab.
1.2 The features were standardized so they have zero mean and unit variance.
1.3 In each of the 10 cross validation folds, a logistic regression model was built. The class labels were changed for logistic regression i.e 1 for negative class and 2 for the positive (hence the old label 1 becomes 2 and the old label 0 becomes 1). Once the model was built it was applied back on the training data to get training predictions. The accuracy, sensitivity and specificity were obtained for the training data. The results are presented at the end section of the document.

1.4 The trained logistic regression model was applied to the test data to get predictions and obtain accuracy, sensitivity and specificity.

1.5 Next the Gaussian Naïve Bayes Model was used. 10 different models were trained and tested across the 10-fold cross validation data splits and average training and test accuracy, sensitivity and specificity were obtained.

1.6 Next Linear Discriminant analysis was used. Again 10 different models were trained and tested across the 10-fold cross validation data splits and average training and test accuracy, sensitivity and specificity were obtained.

1.7 Next Support Vector Machine Model with a linear kernel was used. The optimization was done using SMO optimization method. The maximum number of iterations chosen were 50000. 10 different models were trained and tested across the 10-fold cross validation data splits and average training and test accuracy, sensitivity and specificity were obtained.

1.8 Then I Repeated step 1.7, except this time non-linear RBF SVM kernel was used.

1.9 Next an ANN model was applied to the data. A neural network was built with 3 layers and nodes running for 200 epochs. 10 different models were trained and tested across the 10-fold cross validation data splits and average training and test accuracy, sensitivity and specificity were obtained.

1.10 Next the ANN in 1.9 was re run, but this time on the original non-normalized data.

1.11 The methods were compared and contrasted not just in terms of overall accuracy, but also the sensitivity and the specificity. Also there is discussion about complexity of use and number of parameters to tune.

<u>APPROACH:</u>

Please find below the comparison of the algorithms as mentioned in the previous steps on the provided data set.

	Training Accuracy	**Training Senstivity**	**Training Specificity**	**Test Accuracy**	**Test Senstivity**	**Test Specificity**
Logistic Regression	0.8333	0.979	0.2919	0.8332	0.979	0.2913
Naïve Bayes	0.8243	0.8731	0.6428	0.8248	0.8736	0.6436
LDA	0.6586	0.6743	0.6003	0.6582	0.6740	0.5992
SVM(Linear Kernel)	0.2684	0.0718	0.9990	0.2685	0.0720	0.9990
SVM(Gaussian Kernel)	0.9884	0.9944	0.966	0.9868	0.9931	0.9638
ANN (with Normalized Data)	0.932	0.9777	0.762	0.9313	0.9779	0.759
ANN (without Normalized Data)	0.8663	0.9848	0.4259	0.8622	0.9843	0.4098

Note: It is mentioned in the question that the variable labels are 1 for the positive class (**that we are trying to predict**) and 0 for the negative class. Hence, the metrics are calculated treating

- True Positives as the instances which actually belongs to class with label 1 and are correctly classified as belonging to label 1.

- False Positives as the instances which actually belongs to label 0 but are incorrectly classified as belonging to label 1.

- True Negatives as the instances which actually belongs to class with label 0 and are correctly classified as belonging to label 0.

- False Negatives as the instances which actually belongs to label 1 but are incorrectly classified as belonging to label 0.

Comparison:

a) In terms of the above mentioned Metrics (Accuracy, Sensitivity, Specificity):

In terms of accuracy as we can see the best performance is given by non-linear SVM with Gaussian Kernel both for Training set and test set whereas for Logistic Regression the accuracy is 83% in both for training and test data. The accuracy for LDA is around 65% and for Linear SVM the accuracy is 26% which is lowest among the linear classifiers. This gives a clear intuition that the shape of the decision boundary separating the two classes is non-linear in the input space.
Even when SVM with Gaussian Kernel is compared to Naïve Bayes and Artifical Neural Network it is observed that it still leads in terms of accuracy. ANN with Normalized data gives the second best accuracy.

However, since we are trying to predict the positive class i.e instances with label 1 we need to focus on sensitivity for evaluation. In terms of Sensitivity, SVM with Gaussian Kernel gives pretty good results and almost correctly classifies 99% of the instances belonging to class 1. Also, it gives approximately the same results both for training set and test set. It is very closely followed by logistic regression and ANN (both with and without Normalized data). Gaussian Naïve Bayes is lacking behind the other classifiers with respect to sensitivity at 87.3% except for LDA which is at 67.5% and Linear SVM which mere 7%

In terms of specificity, except for the non-linear SVM with Gaussian Kernel & Linear SVM it is interesting to see that all the classifiers perform pretty badly. This means that out of the total instances in a dataset which actually belonged to class 0 the proportion of correctly classified instances as class 0 by the classifiers is pretty low. We can note that Gaussian Naïve Bayes and Linear SVM which lacked behind in terms of sensitivity is doing comparatively better in terms of specificity here.

ANN with Normalized Data vs ANN without Normalized data

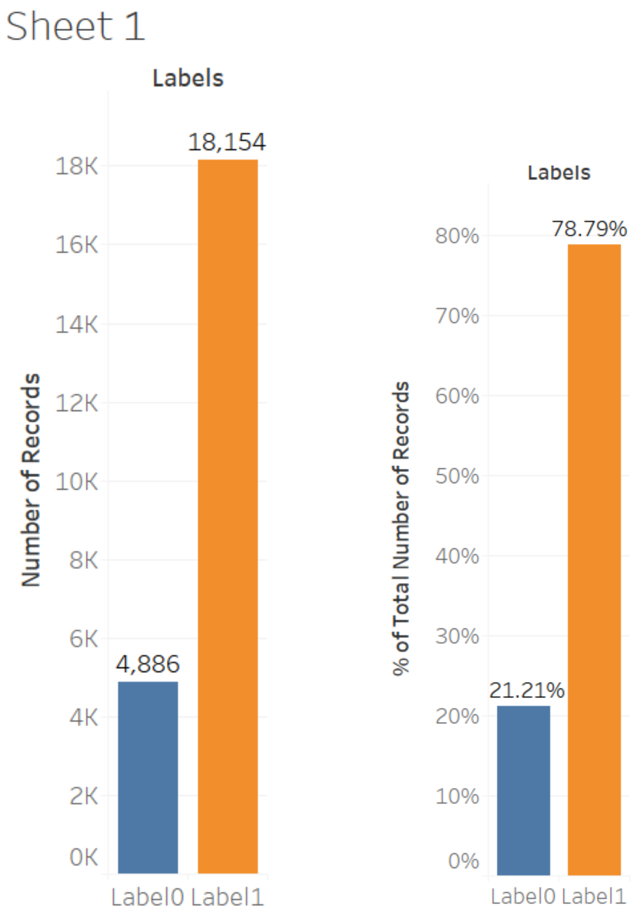
We can see that ANN with Normalized Data performs better than ANN without Normalized data
in almost all the three metrics considered to evaluate the performance. Although, ANN without Normalized data performs slightly better in terms of sensitivity but lacks far behind in terms of accuracy and specificity. We can see from the dataset that the 8 features are on different scale. It is imperative to bring them to common scale to build an unbiased model. It compensates for inflated or deflated variance that is caused by the value of a particular feature being apparently large/small just because it is measured on a different scale.

Conclusion:

We observed in the above section that all the classifiers, except for non-Linear SVM with Gaussian kernel and ANN with normalized data (to some extent), didn't do well in terms of specificity. The classifiers could not do well to classify instances belonging to class 0. In contrast, all of the classifiers did fairly good job in terms of sensitivity and accuracy. The conclusion we can draw from here is that the classifiers are not able to learn the concept for class 0 because of following reasons:

1) **Class Imbalance:**

It is observed from the class distribution that the number of instances with label 1 almost accounts for 79% of the data. This indicates that there is a significant class imbalance that we can see as well in the below figure.



Due to class imbalance classifiers are biased towards learning the concept for the majority class i.e class 1 labels at the cost of misclassifying the instances belonging to class with labels 0. We can say that the classifiers are overfitted to classify the class 1 labels correctly.

2) **Non-Linear Decision Boundary:**

As we can see that for non-linear SVM with Gaussian kernel for all the three metrics-accuracy, sensitivity & specificity the performance is very good. This indicates that the decision boundary separating the two classes in infact non- linear as is modelled by non-linear SVM. As such the maximum margin hyperplane is able to separate the two classes almost perfectly which is not the case with linear classifiers we have used. Also, one of the advantages of SVM is that the decision boundary is not dependent on the entire set of instances. Rather only the vectors for which the value of Lagrange Multiplier is greater than zero are used to form the decision boundary. These vectors which are called support vectors are the only ones responsible for classification. Hence, even when there is a significant amount of class imbalance due to the properties of non-linear SVM i.e

a) Non-Linear decision boundary

b) Decision boundary based on only support vectors

it is able to handle the dataset aptly and give reliable results. The effect of class imbalance is nullified as all the instances are participating in classification and hence biasing the classifier towards the majority class.

b) <u>Comparison: In terms of complexity of use and number of parameters to tune</u>

In terms of complexity to use, number of parameters and computational complexity the classifiers-ANN and non-Linear/Linear SVM came out to be among the most complex ones. Although we restricted the epoch value to 200 iterations for ANN and 50000 iterations for SVM still the time taken to train the model was much higher as compared to other classifiers. One of the problems faced was that for ANN the optimization is done through Gradient descent which at times gets stuck in local minima as such we have to tune the seed for which it gives the optimal results as well. The same is not the case with SVM which is convex optimization problem and gives the global maxima. As such, the number of parameters to tune is maximum for ANN where for the given problem we had to tune parameters like-Size of the ith layer for N1 layers, Transfer function of the ith layer, epochs. For SVM the parameters tuned were- Maximum Iteration, Type of Kernel, Optimization Method.

Also, ANN was a bit tricky to understand and implement as compared to other classifiers. The parameters to tune were pretty less for classifiers like- Logistic regression, Naïve Bayes, LDA and so were the time taken to train the model for these classifiers. Among all the classifiers non-linear SVM took the maximum training time followed by linear SVM and ANN.

****

<u>The analysis from 1.3-1.11 was done two more times, each time with a different cross-validation split of the data. Following are the results with original split provided initially</u>

<u>Approach:</u>

I have ran the analysis two more times with different cross validation indices as generated using Matlab function crossvalind().

Please find below the results of running algorithm for **two different cross validation indices** :

1)The MATLAB code file named is **Q1_Hw_1_optional_cvind1.m**

Algorithm	Training Accuracy	Training Senstivity	Training Specificity	Test Accuracy	Test Senstivity	Test Specificity
Logistic Regression	0.8325	0.9789	0.2884	0.8316	0.9786	0.2853
Naïve Bayes	0.8243	0.8734	0.6416	0.8237	0.873	0.6411
LDA	0.6592	0.6757	0.5977	0.6593	0.6759	0.5980
SVM(Linear Kernel)	0.2684	0.0718	0.9990	0.2685	0.0720	0.9990
SVM(Gaussian Kernel)	0.9887	0.9944	0.9674	0.9868	0.9933	0.9628
ANN (with Normalized Data)	0.9391	0.9797	0.7883	0.9381	0.9794	0.7836
ANN (without Normalized Data)	0.8484	0.978	0.3671	0.8477	0.9771	0.3615

2) The MATLAB code file named is **Q1_Hw_1_optional_cvind2.m**

Algorithm	Training Accuracy	Training Senstivity	Training Specificity	Test Accuracy	Test Senstivity	Test Specificity
Logistic Regression	0.8322	0.9788	0.2875	0.8315	0.9781	0.287
Naïve Bayes	0.8243	0.8734	0.6418	0.8237	0.8726	0.6425
LDA	0.6583	0.6742	0.5993	0.6584	0.6743	0.5997
SVM(Linear Kernel)	0.2564	0.0563	0.9997	0.2572	0.0574	0.9998
SVM(Gaussian Kernel)	0.9883	0.9943	0.9658	0.9871	0.9934	0.9632
ANN (with Normalized Data)	0.939	0.9754	0.8036	0.9379	0.975	0.8002
ANN (without Normalized Data)	0.8799	0.9767	0.5204	0.879	0.9767	0.5165

<u>Comparison with Original Split:</u>

Logistic regression results are pretty stable on different cross validation indices. The variation in results of the metrics is very small both for training and testing data.

For Gaussian Naïve Bayes and LDA as well the variation in the results across different cross validation indices is negligible. For Linear SVM we can see a little variation in accuracy across the three results. Specificity remains almost the same. The metric Sensitivity varies the most for Linear SVM. For non-linear SVM across the metrics the variation in the results across different cross validation indices is almost negligible both for training set and test set.

For ANN with normalized data there is only a small variation in accuracy and sensitivity however we can see variation in specificity which ranges from 76%-80% in the training set and similarly for the testing set as well, when run with different cross validation indices. This is due to one of the properties of ANN where the optimization is done through Gradient descent which at times gets stuck in local minima.

For ANN without normalized data the variation in the metrics is the maximum. The accuracy for training & test set varies between 84%-88%. There is a slight variation in sensitivity however there is significant variation in Specificity which ranges for 36%-52% both for training and test set.