**Performance of different classifiers on heart disease detection dataset**

**Machine Learning Concepts Used**

**Problem 1**

We classify data using a Linear SVM and predict likelihood of disease belonging to a particular class of severity ranging from 1 to 4 i.e. least to most severe with value of C=0.001. Here, C is the penalty that the classifier incurs every time there is a misclassification that takes place so job of the classifier is to incur penalty as minimal as possible while classifying data in order to keep cost of classification at minimum. In order to check if other type of classifiers work better for this dataset than Linear SVM, we use a RBF i.e. non-linear kernel for the SVM classifier and classify data keeping value of C same. Similarly as last part of our problem 1, we use Stratified k-fold cross validation with 5 folds with a RBF kernel and keeping value of C same as for above classifiers in our search to find which classifier works better for this dataset. The results for this have been shown below in **Fig 1 below.**

**Problem 2**

For problem-2 of our experiment, we go a step further by predicting absence (zero) or presence (non-zero) of heart disease. This is possible because we group all severity classes (1 to 4) together which mean that a non-zero would indicate presence of heart disease and a zero would indicate absence of heart disease. Problem-2 of the experiment follows same procedure as that of problem-1. First step is dimensionality reduction for which we use PCA with 5 components that picks best 5 components out of 14 attributes. Now what we get is a vector representation as we obtained in problem-1, which basically implies 303 samples x 5 features. For problem-2, we use an 80/20 split where 80% of data is used to train classifier and 20% is used to test. Now, we follow the same procedure as we did for problem-1 we apply 3 classifiers i.e. Linear SVM, Non-Linear SVM with RBF kernel and Stratified k-means cross validation with 5 folds, all for a value of C=0.001. The results are shown in **FIG 2**

Table and figures below describe classifiers that we used and interprets results that they generated.

**3 Result**

Table 1: Problem-1 60/40 Data Split

|  |  |  |
| --- | --- | --- |
| **Classifier** | **Classification Accuracy (X\_new)** | **Classification Accuracy (X\_new- Split)** |
| Linear SVM | 80% | 55% |
| Non-Linear SVM (kernel ‘RBF’) | 100% | 49% |
| Stratified k-mean cross validation (kernel ‘RBF’) | 100% | 55% |

Table 2: Problem-1 80/20 Data Split

|  |  |  |
| --- | --- | --- |
| **Classifier** | **Classification Accuracy (X\_new)** | **Classification Accuracy (X\_new- Split)** |
| Linear SVM | 81% | 59% |
| Non-Linear SVM (kernel ‘RBF’) | 100% | 57.37% |
| Stratified k-mean cross validation (kernel ‘RBF’) | 100% | 54.23% |

Table 3: Problem-2 80/20 Data Split

|  |  |  |
| --- | --- | --- |
| **Classifier** | **Classification Accuracy (X\_new)** | **Classification Accuracy (X\_new- Split)** |
| Linear SVM | 100% | 73.77% |
| Non-Linear SVM (kernel ‘RBF’) | 100% | 57.37% |
| Stratified k-mean cross validation (kernel ‘RBF’) | 100% | 54.23% |

Figure 1: Problem-1 PCA Dimensionality Reduction (n\_components=5)

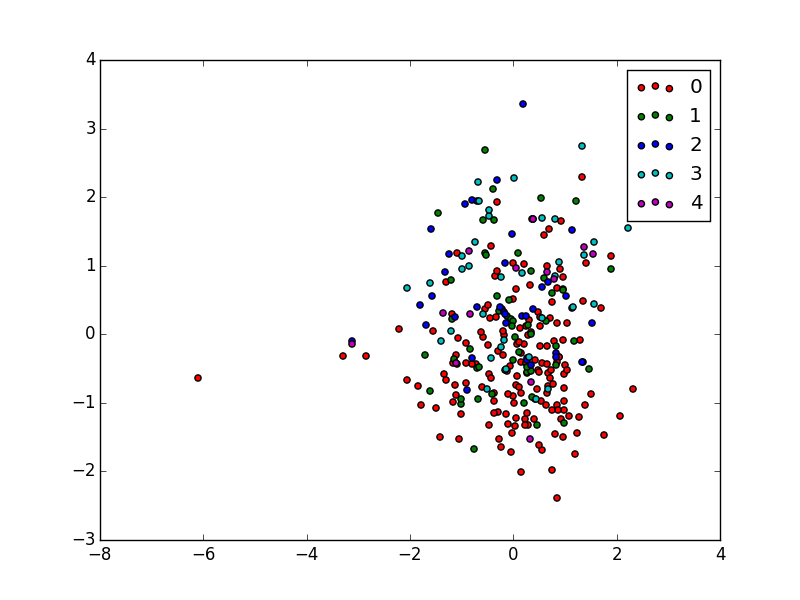
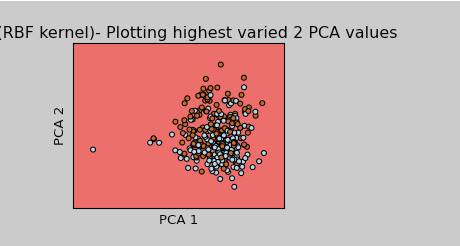


Figure 2: Problem-2 Integrating all severity classes into one i.e. presence (non-zero)



To our observation, with 80% of data for training and 20% data for testing and integrating all severity classes into one that would indicate presence of heart disease improved accuracy of our classifiers as compared to what we observed in problem-1.

**Experimental Setup**

We did our work entirely using the machine learning repositories provided by Scikit-Learn module, numpy module and matplotlib module of the Python programming language. The Python version used was 2.7.5. In order to run the programs, just verify if Python Programming language version 2.7.3 and above is installed with the mentioned Machine Learning module dependencies in your system.

**4 Conclusion**

Based on the results shown above and experiments performed, it is evident that input data plays an important role in prediction along with machine learning techniques. As is seen in the dataset, provided, we have labels from 0 to 4 where the labels of 4 are hardly 13 and when we split the data into train and test, the number become very less which is nothing but noise and can be totally removed from the dataset by using filtering techniques and hence the linear model will be available to predict the outcome much better with absence of noise. Moreover, PCA has again proven that we can get rid of similar feature set and still obtain predictions with great efficiency. Moreover, we have conducted tests using non linear RBF kernel which is a normal first choice and then validating against linear SVC kernel which outperformed RBF in split case. Most importantly, the above experiment not only helped us in predicting the outcome but also gave us valuable insights about the nature of data, which can be used in future to train our classifiers in a much better way.