IITM-CS5691 : Pattern Recognition and Machine Learning — Release Date: 23 Oct, 2021
New Method Writeup — Due Date: 12 Nov, 2021

Roll No: CS21M007 — Name: Anurag Mahendra Shukla
Roll No: CS21M013 — Name: Chandra Churh Chatterjee

- Dear Student, You may have tried different methods for predicting each of the clinical descriptors in the data contest. Submit a write-up of the methods chosen for the data contest in the template provided below. **You will have to add the details in your own words and submit it as a team in gradescope**.
- We will run plagiarism checks on codes/write-up, and any detected plagiarism in writing/code will be strictly penalized.

1. (points) [Name of the Method chosen for each descriptor, and its Paradigm: paradigm could be linear/non-linear models, etc.)]:

Solution: The following study for the Data Contest included testing various classifiers both linear and non-linear for each of the descriptors. After a lot of study and validaion and testing we finally settled with the following paradigm and classifier for the following descriptor.

- From **Dataset 1** we have the following.
  - **Descriptor C0:1**
    The classifier used for the classification task of C0:1 descriptor is **Logistic Regression** which turns out to be a linear model. Hyper parameter tuning was performed to obtain the ideal parameters.
  - **Descriptor C0:2**
    The classifier used for classification task of C0:2 descriptor includes the linear model paradigm of **Logistic Regression**, which was improved using some hyper parameter tuning.
- From **Dataset 2** we have the following.
  - **Descriptor C0:3**
    The classifier used for Descriptor C0:3 is again **Logistic Regression** (linear model) followed by tuning some of the hyper parameters for the task.
  - **Descriptor C0:4**
    The classifier used for the aforementioned descriptor is **Logistic regression** (linear model) even though AdaBoost classifier and the Naive Bayes classifier assuming Gaussian Prior was a close contender used using ideal hyper-parameters evaluated on small sample of the original data.

- **Descriptor C0:5**
  AdaBoost classifier with Decision tree as the base classifier which is a non-linear model is used for classification of the C0:5 descriptor. The model was improved using hyper-parameter tuning.
- **Descriptor C0:6**
  Again **AdaBoost classifier** (non-linear model) was chosen for the classification of the aforementioned descriptor, even though Logistic Regression and Naive Bayes assuming Gaussian prior was a close contender evaluated on a small sample of the original data.

The Comparison of the performance for few models for each of the descriptor is provided in the part 3 of this report, which forms somewhat of a base on why we chose the corresponding paradigm and classifier for the following tasks.

2. (points) [Brief description on the dataset: could show graphs illustrating data distribution or brief any additional analysis/data augmentation/data exploration performed]

Solution: Data that has been provided to us is divided into two Datasets that is Dataset 1 and Dataset 2. Now the descriptors are present as follows.

- Dataset 1 - It contains the two descriptors C0:1 and C0:2 corresponding to 22,283 feature vectors or gene ids' and 130 such data points.
- Dataset 2 - It contains the remaining 4 descriptors C0:3,C0:4,C0:5,C0:6 corresponding to 54,675 feature vectors or gene ids' and 340 such data points.

Prepossessing

We perform a little bit of prepossessing on the data, that is we standardize the data by removing the mean and scaling to the unit variance using sklearn's standard scaler.
We also performed re-sampling on the data using SMOTE procedure for the descriptors C0:3 and C0:4, which showed severe class imbalancing, which can be visualized below. Re-sampling did not improve the performance of the models and as a result we avoided re-sampling for the final model development.

Visualization

Visualizing such a big data is very cumbersome, hence we perform Dimensionality Reduction using singular value decomposition on both the datasets and try to visualize the correlation between two genes or feature vectors with the highest variance. For each of the corresponding descriptors, the plots thus obtained are as follows.

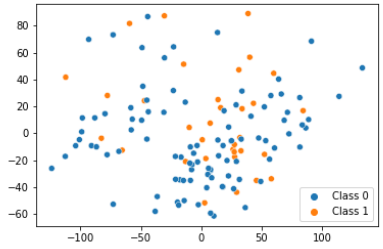

Figure 1: C0:1

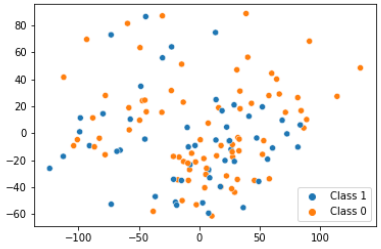

Figure 2: C0:2

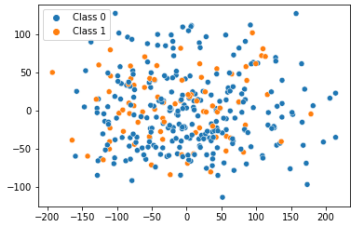

Figure 3: C0:3

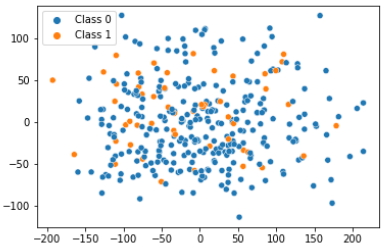

Figure 4: C0:4

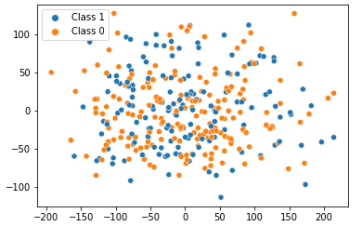

Figure 5: C0:5

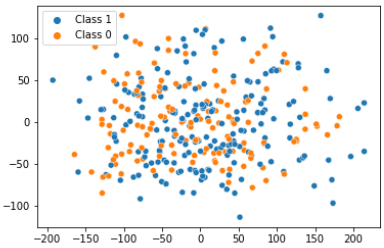

Figure 6: C0:6

From the figures it is clearly visible that the data points plotted against the 2 feature vectors with highest variance have a even distribution along the various values of the feature vectors. For the descriptor C0:4, we can see that the data points corresponding to class 1 are very less compared to that of class 0 and hence it is displaying class unbalancing. To some extent it is also displayed by the descriptor C0:3. This class unbalancing created some problems which affected the performance and we talk more about it in the part 3 of the report.
The distribution of the Highest variance feature vector or the gene id corresponding to the class can also be visualized as follows.

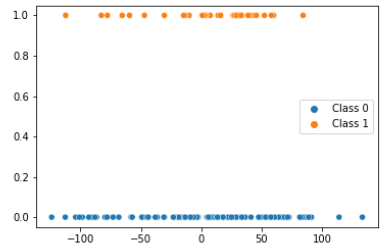

Figure 7: C0:1

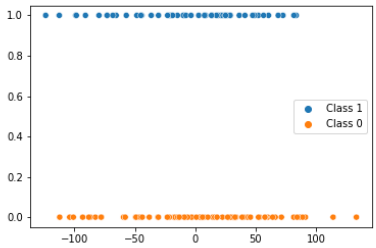

Figure 8: C0:2

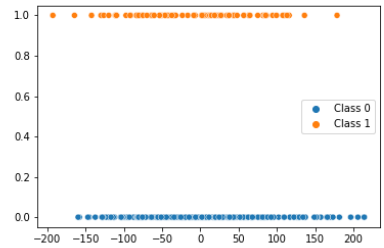

Figure 9: C0:3

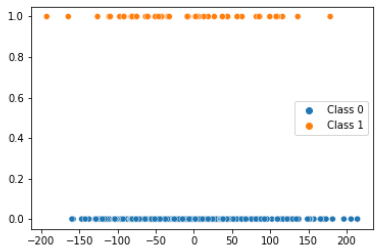

Figure 10: C0:4

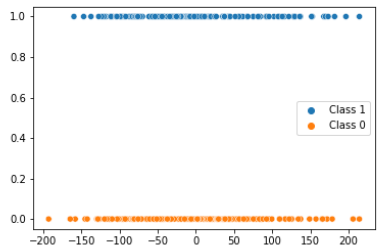

Figure 11: C0:5

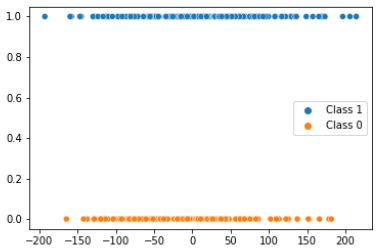

Figure 12: C0:6

From these images also it is relevant that the data shows class imbalancing for the C0:3 and C0:4 descriptor to some extent.
Thus we see various forms of visualization of the 2 datasets, understanding the correlation among the feature vectors or the gene ids' using singular value decomposition to make the data more interpret-able as it contained a lot of feature vectors to start with. The SVD or PCA was used only for data visualization and not for actual model development.

3. (points) [Brief introduction/motivation: Describe briefly the reason behind choosing a specific method for predicting each of the descriptors (could show plots or tables summarizing the scores of training different model)]

Solution: The performance comparison of various models on the Dataset 1 and 2 for the 6 genetic descriptors are provided below. The results provided in the table obtained after scoring the model based on the Matthews Correlation Coefficient score by evaluating only on a 30% sample of the original training data, where the remaining 70% was used for training.

Table:

MCC scores	C0:1	C0:2	C0:3	C0:4	C0:5	C0:6
Logistic Regression	0.332	0.529	0.231	-0.042	0.274	0.079
AdaBoost	0.117	0.383	-0.004	0.133	0.668	-0.009
SVM (linear)	0.331	0.529	0.231	-0.042	0.274	0.013
Decision Tree	0.037	-0.059	0.006	-0.067	0.602	-0.101
Random Forest	0.243	0.402	0.113	0	0.433	-0.034
Naive bayes (Gaussian)	0.331	0.523	0.225	0.158	0.162	0.075

From the above mentioned table it is clear that we used 6 classifiers some linear and some non-linear. The classifier here used for the performance comparison were trained using the ideal

hyper-parameters obtained using Grid Searching. For example the for the Logistic Regression we used 500 max iteration rather than default 100, for the AdaBoost classifier we used 20 estimators and 0.46 as the learning rate, for the SVM we used the linear kernel after grid searching as it turns out that we obtain the same result as logistic regression, and for Decision tree and Random forest the default parameters turned out to be the ideal parameters and finally for the Naive Bayes there are no hyper-parameters to tune. We evaluated the models based on the MCC scores as provided in the Kaggle to compare their performances.

Inferences And Conclusions:

We took 3 best performing models for each dataset and tried different combinations of each for each descriptor and observed the score produced in the Kaggle submissions for validation.

- For the **C0:1 Descriptor**
  We can see from the table that the 3 best classifiers for the same are the Logistic Regression, the Naive Bayes and the linear SVM. After validation form kaggle submissions we finalized the Logistic Regression classifier.

- For the **C0:2 Descriptor**
  From the table we can see that the 3 best classifiers for the C0:2 descriptor are the again the Logistic Regression AdaBoost whose parameters are tuned using Grid Searching and the Naive Bayes. After validation from Kaggle submission we finalized the Logistic Regression (linear paradigm) classifier.

- For the **C0:3 Descriptor**
  For the following descriptor the from the second dataset, we see that the 3 best classifier are the logistic regression the linear SVM and the Naive bayes, out of which we chose the the logistic regression after trying all combinations. This class showed some class imbalancing as well, hence the various models performed poorly on the data.

- For the **C0:4 Descriptor**
  The classifier and the paradigm chosen for the classification of the above was the logisitic regression even though the best seems to be the AdaBoost classifier and the Naive Bayes classifier. Due to some amounts of class imbalancing that we visualize in the data presented in part 2 of this report, the best that performed when trained on the overall data was the logistic regression, that we validated by the kaggle submissions.

- For the **C0:5 Descriptor**
  The 3 best classifiers for the same turns out to be the AdaBoost, Decision Tree and the Random Forest, but after validation using kaggle submissions, the chosen classifier tuned out to be the AdaBoost classifier again tuned using the Grid Searching method.

- For the **C0:6 Descriptor**
  The following Descriptor does not show much class imbalancing, still the best classifier from the aforementioned table seems to be Logistic regression and Naive Bayes but after training on the complete data and validating using the kaggle submissions we finalized the AdaBoost classifier with the following parameters as mentioned in the part 1 of this report.

The class imbalancing for the descriptors C0:3 and C0:4 was handled using the SMOTE procedure which we talk more about in the last part of the report. It did not improve the model performance and hence was avoided for the final model development.
Thus we came to the conclusion for the classifier and the paradigm to be used for classifying the descriptors on the aforementioned ways and the performance comparison table for some of the most widely used classifiers.

4. (points) [As the data comes from a clinical setting, model interpretation is also an important task along with prediction. Describe briefly your preferred choice of methods if you are asked to determine the significant genes behind the regulation of the endpoints or restrict the number of features/genes.]:

Solution: The Data comes from a clinical setting containing a lot of feature vectors without any information among the feature vectors, thus it is highly likely that the feature vectors are highly correlated. Determination of significant genes or feature vectors seems necessary or restricting the number of feature vectors also seems important. The correlation between the gene ids' or the feature vectors of the 2 datasets are visualized as follows in terms of the correlation heat map.

Feature Selection

First 4 feature vectors are used to visualize the correlation heat-map on the original datasets with all the feature vectors mentioned as "Before SVD". After performing SVD we restrict the no of feature vectors based on high variance and lower correlation among each other. After SVD we obtain the 4 most significant feature vectors and visualize the correlation among them in the figures mentioned as "After SVD". The images or the plots are as follows.

Plots:

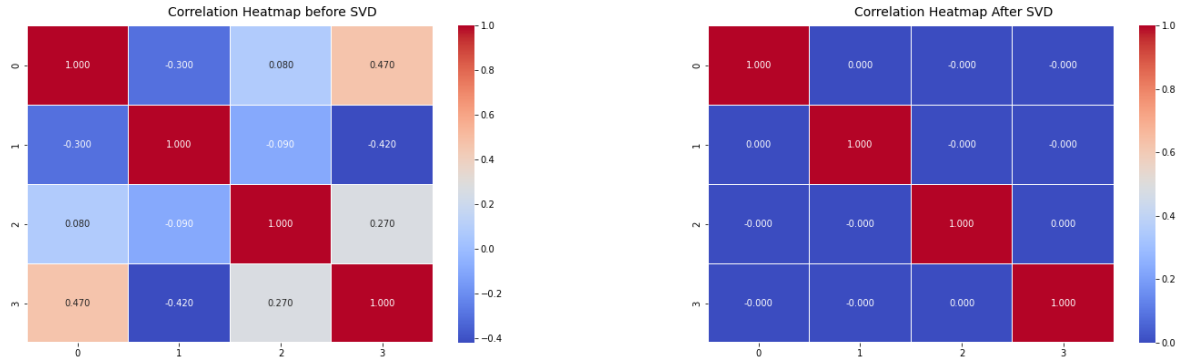

Figure 13: Dataset 1

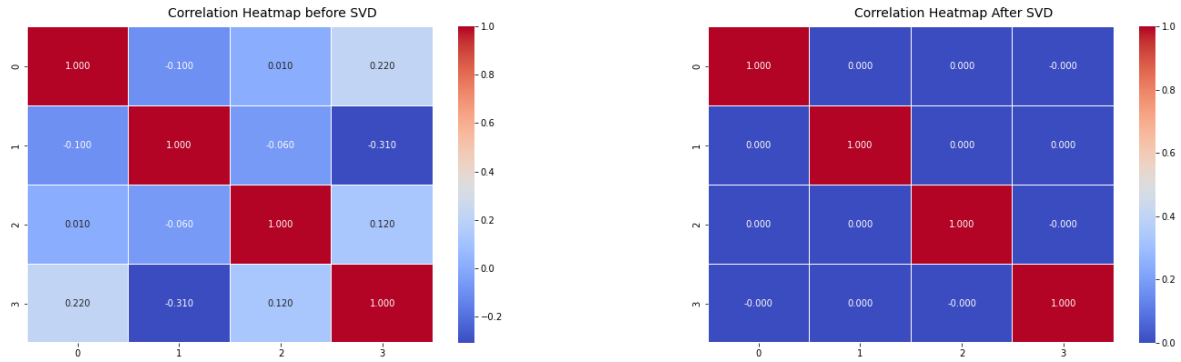

Figure 14: Dataset 2

From the images it is clear that the original dataset contained a lot of feature vectors with high correlation among them, hence SVD seemed a good choice. Now for visualization as we saw in part 2 performing SVD to make the data more interpret-able was help full.

Now performing SVD or PCA for training the model on the restricted no of feature vectors or the significant feature vectors actually generated poor results individually for each of the descriptors. PCA was performed, whch generated 130 feature vetors for the Dataset 1 and 340 feature vectors for the Dataset 2. Singular value decomposition was performed with the parameter "number of components" chosen to be 2200 for the Dataset 1 and 5400 for the Dataset 2, which generated the top k feature vectors with highest variance and least correlation. The SVD result can be visualized for 4 genes in the aforementioned correlation heatmaps. Other values were tried as well but they performed more poorly.

Due to the model performing poorly when trained on a sub sample of feature vectors selected using either SVD or PCA, we developed and trained the model on the entire dataset. It gave better results than training on the Restricted dataset or the k significant gene datasets. Thus extensive feature selection using SVD or PCA was preferred only for visualization and interpret-ability but not model development.

5. (points) [Share your thoughts on each of the endpoints: whether easy/difficult to predict]:

Solution: There were 6 endpoints 2 in one dataset 1 each containing 130 datapoints and 4 in dataset 2 containing 340 data points. From the data visualization it was clear that the some of the endpoints or the descriptors have similar number of datapoints for both the 0 and 1 class. But there were some descriptors which had a lot of class imbalancing making them very difficult to predict.
From the performance comparison table of the models present in the part 3 of this report we see that the some of the models perform well on some descriptors , which is natural but for some of the descriptor we see that none of the model performs well, when evaluated using the MCC(Matthews correlation coefficient). The descriptor C0:1, C0:2 and C0:5 are easily classified using most of the linear or non-linear models as can be seen from the table, whereas the descriptors C0:4 and C0:6, are the endpoints that are the most difficult to classify using vanilla models using base hyper parameters. The descriptor C0:3 showing some amounts of class imbalancing also was moderately difficult to classify. As it turns out that these are the descriptors which show class imbalancing as well except the C0:6 descriptor.

Conclusion:

- Descriptor C0:1 - Easy to predict using linear as well as non-linear classifiers
- Descriptor C0:2 - Easy to Predict using linear classifiers.
- Descriptor C0:3 - Moderately difficult to predict
- Descriptor C0:4 - Difficult to predict.
- Descriptor C0:5 - Easy to predict using non-linear classifiers.
- Descriptor C0:6 - Difficult to predict.

Thus the aforementioned following are the conclusion that we came about the endpoints from the 2 datasets containing a total of 6 descriptors or endpoints.

6. (points) [Add any additional information: like challenges faced or some details that would help us to better understand the strategies that you have utilized for model development]:

Solution: The major challenge that we faced in this Data Contest turns out to be the class imbalancing problem which could has been handled using re-sampling the data. Thus re-sampling involves either undersampling or oversampling the data, where oversampling turns out to be

the preferred choice. One of the most common ways of oversampling is the SMOTE procedure. The SMOTE stands for synthetic minority oversampling technique where synthetic samples are generated for the minority class by focusing on the feature space. New instances are generated using interpolation between the positive instances that lie together.
Our study involved implementing the SMOTE procedure for the descriptor C0:3 and C0:4, but after implementing it, we observed no improvement in the performance of the models on the datasets. Hence we avoided using it for the actual model development.
The data after re-sampling can be visualized as follows.

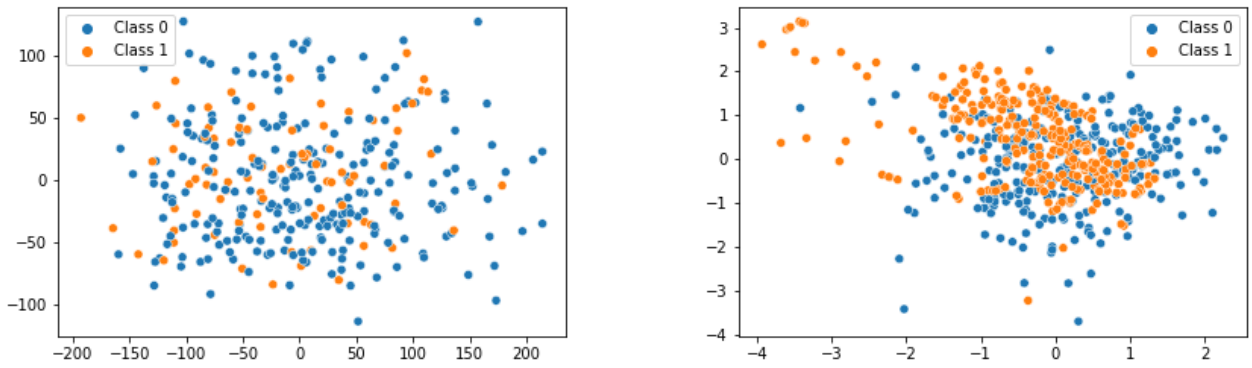

Figure 15: Descriptor C0:3

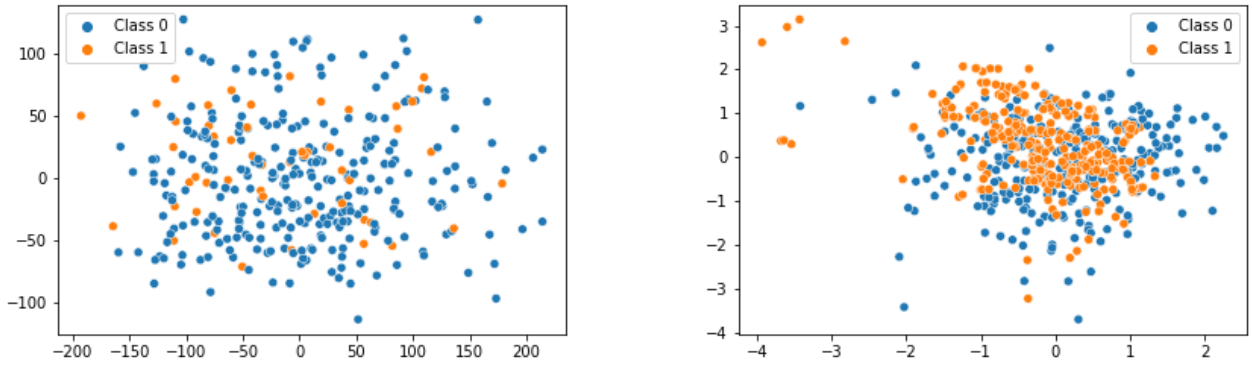

Figure 16: Descriptor C0:4

Here the left image is the image of the original data and the right side image we obtained turns out to be the image obtained after re-sampling the data.
The presence of very large number of feature vectors compared to the number of data points also tuned out to be of some nuisance, which we tried to handle using feature restriction techniques

such as PCA or SVD. It also did not improve the performance of the model what so ever, so we used it for just visualization of the data and not model development.
These were the major challenges faced during this Data Contest regarding the difficult to predict descriptors C0:3 and C0:4. But the descriptor C0:6 needs more understanding and exploration which constitute the future works of this procedures.