Project 2: Gene Expression microarray data analysis and cancer classification

INTRODUCTION:

Given a gene expression microarray data with phenotype information, we are required to analyse which genes are responsible for given cancer types. We further want to build machine learning models to predict the cancer class given the genetic information of a particular patient.

The input dataset contains 802 samples for the corresponding 802 people who have been detected with different types of cancer. Each sample contains expression values of more than 20,532 genes. Samples have one of the types of tumours: BRCA, KIRC, COAD, LUAD, and PRAD.

The dataset is high dimensional, where number of features are greater than number of samples, so careful need is required before applying statistical techniques as these techniques suffer from the curse of dimensionality.

WEEK 1:

TRAIN – TEST SPLIT

To build a classification model, we must set aside certain proportion of our dataset for testing the performance of the model. However, there is a tricky issue with the current dataset. This gene expression dataset is very noisy. It has been found that there are genes which are only expressed in 20 percent or less samples. This means when we split the data n number of times, there might be certain splits where all the 0s fall in the training set, and all expression in the test set respectively. So, when we standardize the data using parameters of the training set, we might get a division by 0 on both the training and testing sets since the standard deviation of the genes in training set would be 0.

Now since the sample set is relatively small, we must only choose the most important of all the informative features, and those features must be found only using the training set. For genes whose expression is mostly 0, there are 2 possibilities. Either the gene is expressed equally in all classes, in which case it is noise, or it is differentially expressed, in which case its relative importance w.r.t other genes must be considered. But given the training size, we might end up completely ignoring these genes.

In order to keep them, we must find out the maximum proportion of 0s in significant genes and split the data so that it doesn't exclude these genes automatically, except if this proportion is larger than 0.85, which is our training set size. In order to give such genes a fair chance, we can do the following: First run ANOVA on the entire dataset to find out all differentially expressed genes, then find out the proportion of their zeros and split at the maximum proportion of zeros. That is, the splitting size

s = min(0.85, max(proportion of zeros of all informative genes))

This will ensure that training set contains at least one sample of such genes and so can be standardized and be directly fed to week 2 analysis of dimensionality reduction as well as wrapper-based feature selection algorithms. If there are genes which are zero on more than 85% of samples and still significant, then those genes can be studied separately and will not be considered for classification model as they would only create the dataset noisy in different iterations. It was found that maximum proportion of zeros in differentially expressed genes was much greater than 0.85, so we ended up removing them. This method can be compared to Filter based feature selection algorithms.

FEATURE SELECTION:

Feature selection techniques can roughly be considered of 3 types:

1. Filter Based: ANOVA, Low variance filter, Correlation
2. Wrapper Based: Stepwise selection, Recursive Feature Elimination
3. Embedded: L1 Penalty in Logistic Regression

For the current dataset we first used ANOVA to select 500 genes out of already filtered genes. The idea was this. Each gene has an expression value in 5 classes of cancer. If the gene has similar expression in all 5 classes, then the gene is not discriminative. There were around 19000 genes selected at first step. Then 500 genes were selected based on the highest F-statistic. Then taking these 500 genes we tried Forward stepwise selection, backward stepwise selection using Support Vector Classifier and Random Forests. However, it was seen that after 8 or 9 features the evaluation metric (f1 score) stopped changing indicating that multiple features were redundant. The search was manually stopped early and the method was discarded. n their landmark paper Guyon, I., Weston, J., Barnhill, S., & Vapnik, V., "Gene selection for cancer classification using support vector machines" propose a new method of gene selection utilizing Support Vector Machine methods based on Recursive Feature Elimination (RFE). This method allows to remove multiple features at the same time by utilizing the coef_ or feature_importance_ attribute of a given algorithm, by discarding the least important variables, recursively. This method was used again using both Support Vector Classifier and Random Forest with 5-fold cross - validation. Then an intersection of both the sets of features was taken which gave final 47 most informative features.

WEEK – 2

DIMENSIONALITY REDUCTION:

There are 3 most important dimensionality reduction methods:

1. Principal Components Analysis
2. Linear Discriminant Analysis
3. t-SNE

Principal components analysis (PCA) is frequently used in biological sciences for global analysis of omics datasets. It provides fully unsupervised information on the dominant directions of highest variability in the data and can therefore be used to investigate similarities between individual samples, or formation of clusters. In most of these studies, scientists focus on the first two to four principal components (PCs), assuming that higher-order components mainly contain irrelevant information or noise].

One study performed by Schneckener et al., states that gene expression differences that are represented in higher than the first four PCs are rarely reproducible in new experiments.

Another study by Lukk et al. also suggests a surprisingly low dimensionality of gene expression data. They performed PCA and hierarchical clustering on a larger and heterogeneous microarray dataset and found that the first three PCs have clear biological interpretations. The hierarchical clustering analysis in week 3 supported these results, detecting the same major clusters that were found in the PCA.

It was seen that no matter the split and choice of training and validation datasets, the different classes of cancer can be well segregated into their clusters in 3 dimensions, when we use the 3 leading principal components. This reveals an inherent dimensionality of the dataset, which is stable enough over the choice of training and validation samples.

Choosing number of PCs: According to Joliffe(1982),The most obvious way of using PCA in a discriminant analysis is to reduce the dimensionality of the analysis by replacing x by the first m (high variance) PCs in the derivation of a discriminant rule. It is inadvisable to look only at high variance PCs, as the low-variance PCs can also be highly correlated with the dependent variable. According to Hastie et al, we can assume that the directions in which X1, . . . , Xp show the most variation are the directions that are associated with Y. While this assumption is not guaranteed to be true, it often turns out to be a reasonable enough approximation to give good results. Although, the number of principal components, M, is typically chosen by cross-validation.

We performed cross – validation using Random Forest and Support Vector Classifiers and choose the number of principal components which maximize the f1 score. It was found that Random Forest and Support Vector Machine need different number of components for maximum validation scores. It is because whether a feature is informative or not depends on the particular algorithm we are using. In fact, while SVM gives an ideal f1 score between 71 to 110, the same number of components in Random Forest cause a decrease in performance. So, we cannot choose one out of the two for model building. We tested models in week 4 using both the datasets. Similar methodology was performed to choose number of components in LDA, and finally 4 components gave the best f1 score.

WEEK – 3

CLUSTERING

Gene expression matrix can be analysed in two ways. For gene-based clustering, genes are treated as data objects, while samples are considered as features. Conversely, for sample-based clustering, samples serve as data objects to be clustered, while genes play the role of features.

Gene-based clustering:

The purpose of gene-based clustering is to group together co-expressed genes which indicate co-function and co-regulation. We consider two approaches: Hierarchical clustering and K-Means Clustering. These clusters may also be intersected with each other, and so one might also be interested in finding the relationship between these clusters, for which Hierarchical clustering is more suitable as K-Means clusters do not overlap.

The problem of finding optimal number of clusters: Since the number of clusters is not known in advance, clustering the genes boils down to finding the optimum number of clusters. There are many indices available for this purpose. Generally, the clustering algorithm is run several times for several values of k (number of clusters) and the value of k at which a particular score maximises (or minimises) is considered the most optimal number of clusters. We have here considered 3 such indices: Davies-Boulding index, Silhouette index and WCSS.

1. Hierarchical clustering: There are 2 types of hierarchical clustering: Divisive and Agglomerative. In the current task, we employ Agglomerative HC. There are many ways AHC can be performed, the parameters being: **Type of linkage**, one of centroid, ward, UPGMA, WPGMA, complete. Type of Distance: Pearson, Euclidean. Different methods could give rise to different dendrograms on different datasets. To choose the right type of linkage an analytical method is used based on correlation of cophenetic distance with the distance matrix. It was found that UPGMA linkage with Euclidean distance gave the most coherent clusters, with optimal number of clusters being 3. Eisen et al. first applied the agglomerative algorithm called UPGMA in literature.

Eisen's method is much favoured by many biologists and has become the most widely-used tool in gene expression data analysis, so there's evidence for meaningful clustering using this method.

2. K-Means clustering: No meaningful results were found using K-means clustering on DB index, Silhouette index or WCSS.

Sample-based clustering:

The goal of sample-based clustering is to find the phenotype structures or substructures of the samples. Previous studies have demonstrated that phenotypes of samples can be discriminated through only a small subset of genes whose expression levels strongly correlate with the class distinction. These genes are called informative genes. The remaining genes in the gene expression matrix are irrelevant to the division of samples of interest and thus are regarded as noise in the data set.

The existing methods of selecting informative genes to cluster samples fall into two major categories: supervised analysis (clustering based on supervised informative gene selection) and unsupervised analysis (unsupervised clustering and informative gene selection).

Measuring quality of clustering: For sample-based analysis, the number of clusters is always pre-defined, equal to the number of phenotypes. The quality of clustering was measured using the entropy index.

1. Clustering based on supervised informative gene selection: Using this method, first informative genes are selected with the help of a classifier, and then traditional clustering methods such as K-means are applied. Here we used the selected features from week 1 to reduce the dimensionality of the dataset and applies K-means and DBSCAN algorithms. Both the algorithms performed really good with entropy scores.

2. Unsupervised clustering and informative gene selection: In this approach, first the gene (feature) dimension is reduced, then the conventional clustering algorithms are applied. Here we used the selected PCs and LDs from week 2 and applied K-means and DBSCAN algorithms. Here only K-means with LDA performed good.

WEEK – 4

BUILDING A CLASSIFIER

The final task of this project was to build a classifier. 3 machine learning algorithms were tested: Random Forest Classifier, Support Vector Classifier and Artificial Neural Networks, on 3 datasets: Complete Dataset, Feature Selection Dataset and Dimensionality Reduction Dataset. Primary method of evaluation was accuracy and the final models were compared on ROC-AUC scores.

RESULTS:

1. **Best Random Forest Model:** Random Forest achieves a ROC-AUC score of 1.0 on complete dataset, feature selection dataset and LDA dimensionality reduction dataset, but slightly lags behind in PCA dataset. As we discovered in week 2 and week 3, the intrinsic dimensionality of this gene expression microarray data is very low, meaning that most of the features don't add much value in the process of classification, and so are rightly termed as noise. Random Forest is good at dealing with high dimensional datasets because it only considers square root or logarithm of total number of features at a time (feature subsampling), which in our case is ~sqrt(19000) = roughly 138 features and log2(19000) = roughly 15. However, a very high number of trees must be built so as

to ensure that all features are sampled in the final forest. This makes the training computationally very expensive, given that the optimal hyperparameters are found via grid search and 5 fold cross validation. Moreover, the complete dataset is of the form p>m i.e the number of features is much higher than number of training examples. It is known that most statistical methods don't work properly in such cases of high dimensionality. Even though the results obtained are good, they can't be considered as reliable. It is considered better to reduce the dimensions of the data before building a classifier on such high dimensional dataset. Therefore, we discard this model.

This leaves us with two other models: The Feature Selection Dataset and the LDA Dimensionality reduction dataset. These methods have 47 and 4 number of features respectively, so they are not computationally expensive. However, while applying LDA we lose interpretability of the model obtained, and the new features may not have real life scientific value. So, it is best to go with Random Forest on Feature Selection Data.

2. **Best Support Vector Classifier:** Support Vector classifier achieved perfect ROC-AUC score on all the four algorithms. There are few interesting observations: 1. In all the 4 cases, the kernel was selected as linear, meaning that this data is linearly separable. This is not surprising as we saw in week 2 and week 3, the clusters were well separated, so a more complex kernel was not required at all. 2. The Penalty term C got selected as follows: Complete (n_features=19000): 5e-5, PCA (n_features=71): 1e-4, Feature Selection (n_features=47): 1e-2 and LDA (n_features=4): 1e-3. As the number of features increase, higher regularization gives good results indicating the presence of noise in the dataset, and indicating that model avoided overfitting. It is noteworthy that LDA required a higher penalty than Feature Selection dataset, which indicates noise or redundancy in the data. SVC is a fast algorithm, so there was no computational burden in any of the datasets. However, we would rule out the original dataset for the same reason as above. Since LDA and PCA reduce interpretability of the model, here again we would go with the feature selection dataset.

3. **Best Deep Neural Network:** All 3 Neural networks achieved different scores for ROC-AUC. Going by that score alone, the Neural network performed best on LDA dataset. However, LDA model was slow in training, and showed greater fluctuations before convergence compared to other 2 models. In terms of computation, Feature Selection model was the quickest. Building a competent model on the complete dataset required a lot of parameters and hence a greater memory and gpu support, therefore it is not feasable. There are 2 misclassifications in the feature selection model, while 0 on LDA model. So, the best Deep Neural Network is LDA Model.

4. **Best Classifier:** We can judge the best model in terms of computation requirements, ROC-AUC score, interpretability and Cross-Validation.

   Computational Requirements: On the current dataset, both SVC and RF were quick with selected hyperparameters, while it took longer to train the neural network.

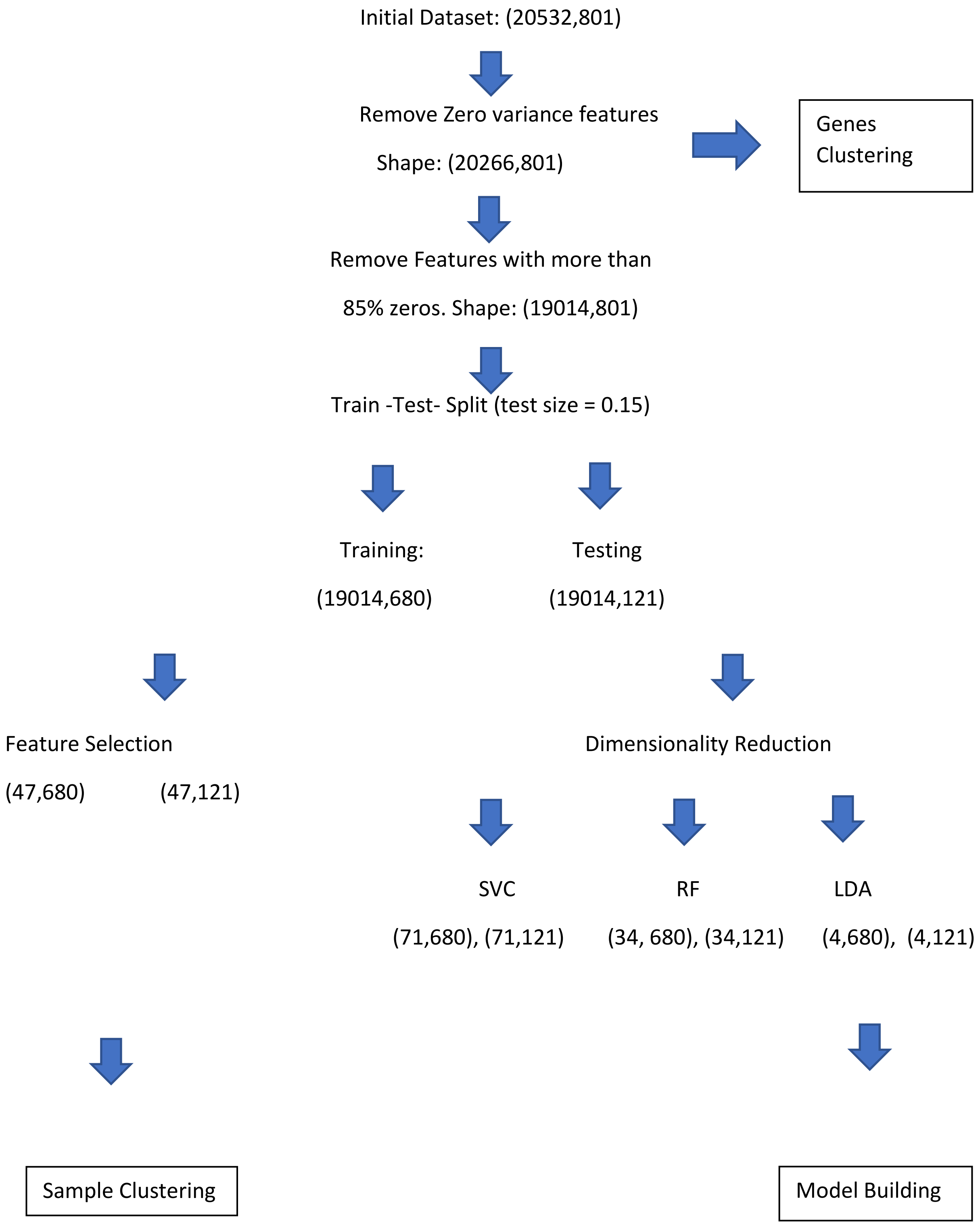
   ROC-AUC Score: All 3 models have achieved the perfect ROC-AUC score of 1.0

   Interpretability: Since the SVC chosen is with linear kernel, it provides coefficient of the original features, while random forest also provides feature_importances, whereas DNN doesn't. Moreover, DNN is built on LDA dataset which furthers worsens the interpretability.

Cross-Validation Error: Both SVC and RF were evaluated with 5-fold Cross Validation whereas DNN was evaluated using hold out set strategy, so the estimate is rather optimistic and not a true reflection of the error.

Based on the above criteria, it is clear that Neural Networks are not suitable for the given problem set. Both Support Vector Classifier and Random Forest perform equally good. However, the results are only valid for the current dataset, because of the No free lunch theorem.

Appendix: Datasets details


Initial Dataset: (20532,801)
Remove Zero variance features
Shape: (20266,801)
Genes Clustering
Remove Features with more than
85% zeros. Shape: (19014,801)
Train -Test- Split (test size = 0.15)
Training:
(19014,680)
Testing
(19014,121)
Feature Selection
(47,680)
(47,121)
Dimensionality Reduction
SVC
(71,680), (71,121)
RF
(34, 680), (34,121)
LDA
(4,680), (4,121)
Sample Clustering
Model Building