Classification of p53 mutant Activity

Abstract

This project aims at conducting research and analysis on p53 mutant dataset. p53, also known as TP53 or tumor protein is a gene that codes for a protein that regulates the cell cycle and hence functions as a tumor suppression. It is very important for cells in multicellular organisms to suppress cancer. Our Data Set contains Biophysical models of mutant p53 proteins yield features which can be used to predict p53 transcriptional activity. The goal is to model mutant p53 transcriptional activity (active vs inactive) based on data extracted from biophysical simulations.

1 Introduction

1.1 Dataset Description

There is a total of 21799 instances with 5409 Features per instance.
Features 1-4826 represent 2D electrostatic and surface-based, it includes the steric and depth information, and the sphere beingmapped to a plane.
Features 4827-5408 represent 3D distance-based features, it includes the 3D distance difference map between mutant and wild-type p53.
Features 5409 is the class attribute, which is either active or inactive.The class labels are to be interpreted as follows:
'ACTIVE' labels represents transcriptionally competent, active p53
'INACTIVE' label represents cancerous, inactive p53.

1.2 Problem Description

The primary goal of our project is to provide classification whether the predicted mutant value is 'Active' or 'Inactive'. We achieve this by dividing our main goal into set of goals as:

1. Pre-processing of the Dataset by converting it into appropriate format.
2. Reduce the Data in chunks and Apply Stochastic Gradient algorithm classifier and check the accuracy.
3. Comparison of Performance of classifiers like PA, SGD, Logistic, ASGD and Perceptron using Learning Curve.
4. Reduce the Dimensionality of the data applying principal component analysis.
5. Apply the non linear methods on the reduced dimension (Using PCA) to boost the accuracy of the model.

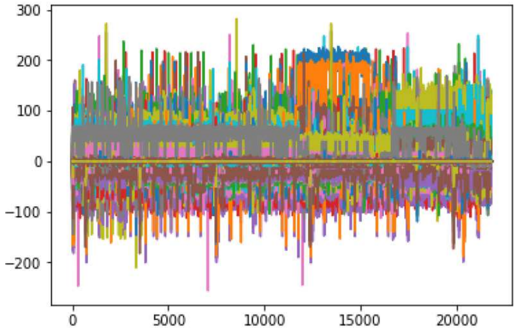
2 Initial exploration and visualization of the dataset

2.1 Data Pre-processing

There are 5408 predictor variables in this Dataset. A lot of missing values in the data was found. Initially, we replaced those values with NaN. Then fill those NaN values with the mean of the respective column. Assigning values to the y columns, 0 as "inactive" and 1 as "active". We reduced the Dataset into 1 set of observations. There were total of 40K observations. We took the 1st set of observations. i.e. 21799 rows * 5409 columns.

2.2 Visualization

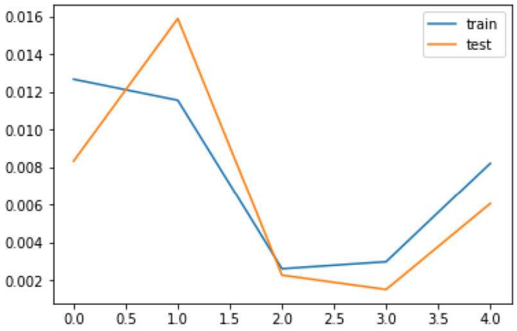
We visualize the Data using a normal plot, we can see the distribution of the X and y in the Data. Being a classification problem, we actually need to find the find the probablity curve for the scatterplot for the y(dependent) variable. The data is divided into a matrix of [21799*5409].



3 Classification

3.1 Stochastic gradient classifier (SGD)

SGD is a simple yet very efficient approach to discriminative learning of linear classifiers under convex loss functions. Using the loss function as ”hinge”, the mean squared error was found to be 0.0048 and Accuracy was 99.6 for the entire Dataset. After that, we applied the SGD-Partial-fit for chunks of the Data and observed the Train-Test loss in chunks. The variation in the test loss can be observed through the graph.
First iteration - trainloss:0.0127,testloss:0.0083
Last iteration - trainloss:0.0082,testloss:0.0061
Maximum Accuracy for the chunk = 99.48

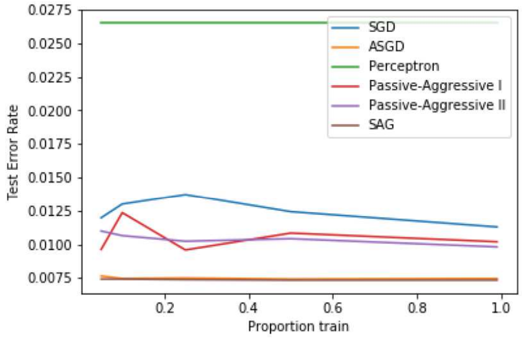


3.2 Passive Aggresive Algorithm (PA)

The passive-aggressive algorithms are a family of algorithms for large-scale learning.They do not require a learning rate but they do require a regularization parameter. The regularization was assumed to be 1.0. Using the loss function as ’hinge’, the accuracy was found to be 98.2575. The accuracy of PA algorithm is less compared to the SGD Classifier.

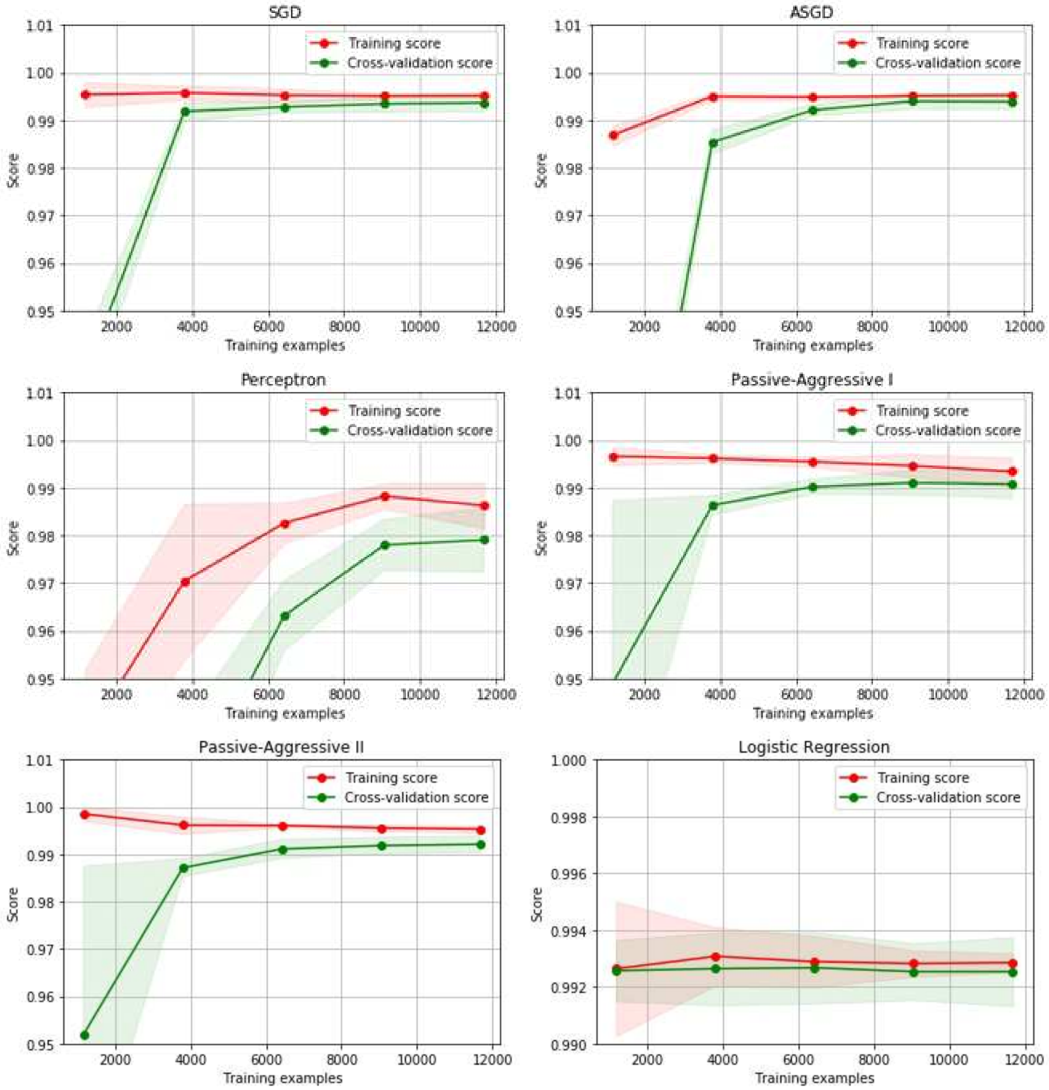
3.3 Comparison of Classifiers

We compare the Performance for various solvers.The comparison includes the performance of following solvers:
Stochastic Gradient Descent
Logistic Regression(SAG solver for l2 penalty)
Stochastic Gradient Descent (average=true)
Passive Aggresive Algorithm - I (hinge)
Passive Aggresive Algorithm - II (squared-hinge)
Perceptron
We compare the Test error of all these models using Classification testing error as a function of training examples.



The Test error rate remains constant for ASGD, SAG, Perceptron Classfiers whereas it gradually decreases for SGD and Passive algorithm (PA-I and PA-II) Classifiers.

3.4 Learning Curve Comparison

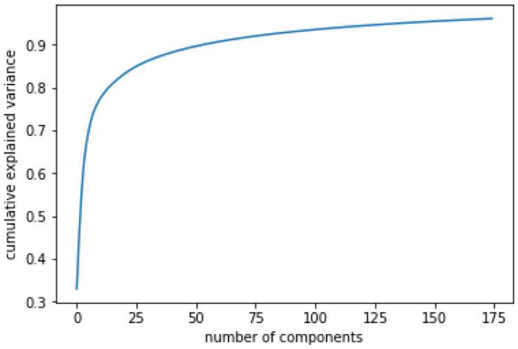


1. SGD: Curve for the Stochastic gradient descent Classifier looks good with highest accuracy amongst all.
2. ASGD: ASGD is SGD (average = true), the curve gets better as the number of samples increases.
3. Perceptron: It doesn't require learning rate, regularization is not included. This model is faster to

train than the SGD with hinge loss, hence the results are sparser.
4. Passive Aggresive Algorithm (PA-I and PA-II): These models are similar to Perceptron with no learning rate but they include a regularization parameter 'C'. The curve looks much better compared to Perceptron.
5. Logistic Regression: The sag solver uses a Stochastic Average Gradient descent. It is faster than other solvers for large datasets, when both the number of samples and the number of features are large. Curve looks better with good accuracy.

4 Principal Component Analysis (PCA) as Dimensionality Reduction

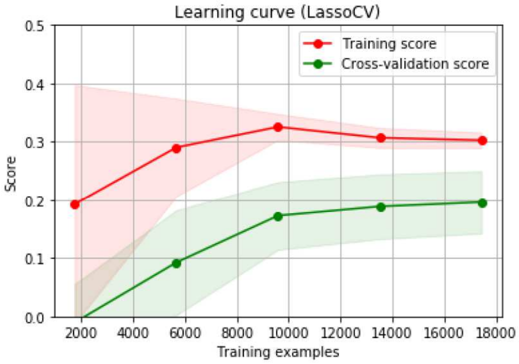
When the data is high dimensional, it is highly possible that the features are collinear, so there is redundancy in the data. So, we reduce the high dimensional data into low dimensions and perform the classifiers on the reduced dimension. Dimension Reduction process ensures that the reduced feature set conveys similar information. PCA reduces the redundancy in the data.



To validate the result, we used the cross-validation for the reduced feature using the Decision tree classifier. The Accuracy turned out to be 95.8 and also more than 90 percent of the variance is retained, so the reduced feature 175-D is correct.

5 Feature Selection using LassoCV

When features are colinear, it not only contains redundant information, but also make solutions unstable. Lasso (minimizing L1) is a feature selection method that select a small set of features that best completes the task on hand. In this sense, it is filtering extra information and treating colinearity. It is a feature selection method, in which we drop some features. We used LassoCV and the reduced number of features were 270.
However the accuracy was very less for LassoCV feature selection.

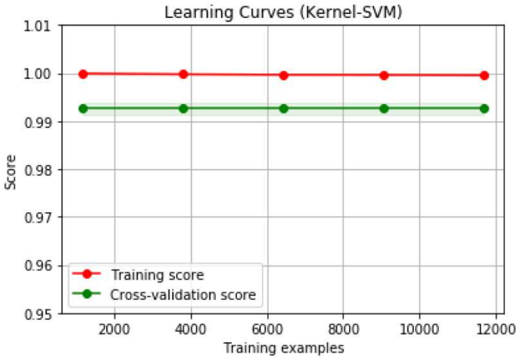


Comparing the Accuracy of LassoCV and PCA, we chose PCA as it is more accurate and lower number of dimensions.

6 Non-linear methods

6.1 Support Vector Machine (RBF)

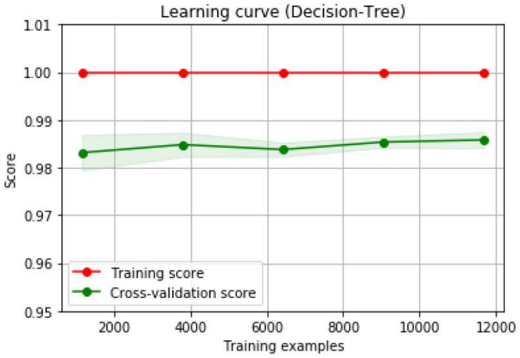
After reducing the components to 175 using PCA, we implemented the SVM Classifier. Support Vector Machine uses kernel tricks and transforms the data, and based on the transformation, it calculates an optimal boundary between class labels. SVM creates hyperplane to separate the classes by creating largest margin between closest points and decision boundary. We are using SVC with Radial Basis Function from svm class of sklearn package. Accuracy was found to be 99.43 percent.



With the best accuracy of 99.43 percent compared to all the other classifiers we applied, SVM is our best fitted model.

6.2 Decision Tree Classifier

Decision tree is a type of supervised learning algorithm (having a pre-defined target variable) that is mostly used in classification problems. By Applying the Decision Tree Algorithm, we got an accuracy of 98.3 percent.



7 Result

1. Initially, reading the data in chunks, the best fitted model was for SGD classifier with accuracy of 99.4 percent.
2. Between LassoCV and PCA, PCA turned out to be the best method for Dimension reduction with accuracy of 98.3 percent.
3. The Best fitted model for this Dataset after doing PCA is the model using Support Vector Machine (SVM) with the best accuracy of 99.43 percent comparing to the other Classifiers applied.

8 Conclusions

While implementing different machine learning algorithms, we figured out following conclusions:
1. While reading the data in chunks, we found that SGD classifier performs better compared to Logistic regression, Passive Aggresive Algorithm - I and II.
2. We considered all the features from the dataset in order to capture the complex relationships

between them. This, along with PCA, helped us in getting higher accuracy of 98.3%. We were able to boost the accuracy to 99.3% by performing the SVM on the reduced Dataset.
3. It is computationally expensive for SVM to process 5408 features. Hence we had to apply PCA to reduce the number to 200 features that helped us processing SVM smoothly.
4. When we compared the accuracies of LASSOCV and PCA, we figured out PCA has the best accuracy using the cross validation score and fitting the model using Decision tree classifier. This concluded that few change patterns are relevant in out Data. So, PCA performs better compared to Lasso, Lasso is good when few variables are relevant.
5. PCA performs the reduction by combining the variables and making new ones whereas Lasso simply drops the irrelevant variables.
SVM ended up having the highest accuracy among all the classification models that we worked with. SVM is able to capture the complex relationship between the features as it considers each feature as an independent and identically distributed variable.
6. Decision Tree Classifier ended up with the accuracy of about 98 percent.

9 Future work

9.1 Most informative Positve (MIP) Active learning

The dataset from UCI is more about biology and research. Using Machine learning in the field of biology, MIP is a integrated computational/biological approach for the biological recovery of the mutants. With such a highly accurate Data, it is infact easier to predict the activity of the produced mutant whether it'll be active or inactive. MIP can be used to select regions in the p53 tumor suppressor protein and visualize such a multi-Dimensional data by using the Chimera package in Python.

References

[1]: Danziger, S.A., Baronio, R., Ho, L., Hall, L., Salmon, K., Hatfield, G.W., Kaiser, P., and Lathrop, R.H. (2009) Predicting Positive p53 Cancer Rescue Regions Using Most Informative Positive (MIP) Active Learning, PLOS Computational Biology, 5(9), e1000498
[2]: Danziger, S.A., Zeng, J., Wang, Y., Brachmann, R.K. and Lathrop, R.H. (2007) Choosing where to look next in a mutation sequence space: Active Learning of informative p53 cancer rescue mutants, Bioinformatics, 23(13), 104-114.
[3]: Blog: Analytics Vidhya
[4]: Blog: Towards Data Science