DS303 Course Project

Discovery of Genomic Biomarkers for Predicting Cancer State through High Resolution Genomic Data

SAKSHAM GAUTAM (180020088), CHIRAG RAJU (18B090003),
SHREY GUPTA (190100112), JAIDEEP CHAWLA (190110030)

GUIDE: PROF. BIPLAB BANERJEE

Motivation Behind the Problem

Discovery of Biomarkers from high resolution Genomic, proteomic, and metabolomic data is a very trending problem in the field of healthcare technology. The main motivation behind the discovery of biomarkers is, it is not possible to carry out numerous tests for each and every feature at the laboratory level to predict the state or early detection of a disease. Hence there is a need of defined number and combination of features which can predict the state of disease in an individual, and from the extracted markers, proteomic or metabolomic paths can be traced back to find the underlying cause of disease, which can lead to personalized medicine a reality in near future. In Metabolomics study, the target is to discover 4-5 biomarkers only to predict the disease, However, on the level of Genomics, the scale might be large. In this project, we are going to extract out Genomic biomarkers in the range of 50-100 Genes to predict the cancer with good accuracy.

Contents

Contents

Abstract

Using a high resolution genome data which contains RNA expression of ~20,000 genes as features, we build a best accuracy classification model to study cancer samples. We classify the samples into the right category of cancer for early detection, and explore discovery of biomarker features, which give optimum accuracy with ML models to predict the cancer state on a genomic level.

Benchmarking of various ML models along with detailed analysis of accuracies is done to get the optimum parameters along with feature selection.

Implementation of Machine Learning and Neural Network based Feature Selection Techniques are also carried out. Implementation and Analysis of approach used in paper - https://arxiv.org/pdf/1903.07045.pdf signifying Teacher Student network for feature selection with different kind of embeddings is done.

Dataset & Problem Description

https://archive.ics.uci.edu/ml/datasets/gene+expression+cancer+RNA-Seq

Number of samples: **801**
Number of features: **20,531**

Decision variables: **Cancer phenotypes**

Problem:
*Which genes**/combination of genes**** *contribute most to the occurrence of these phenotypes of cancer?*
*What machine learning algorithm is the most **efficient** way to classify these biomarkers?*
Can we salvage a classification/demarcation of cancer phenotypes from just unsupervised techniques?
*(Efficient ~ **Less features, Greater accuracy**)*
*(a combination of genes is also referred to as a **biomarker**)*

Methodology

Pre-processing & Data Preparation

- Data Labels of 5 types of cancers are encoded into integer labels
- By exploring 20,531 genomic features, got 267 such features whose values are 0 for all the features, hence these 267 features are useless
- Checked whether the standard deviation of any of the genomic features is 0, signifying same value for all the samples
- Divided the dataset into training and test, 70% training samples and 30% test samples
- Carried out dimensionality Reduction by elimination through Pearson's correlation coefficients, considered 4 thresholds {0.8, 0.6, 0.4, 0.2} of correlations. Got dataset with reduced dimensions to benchmark the models on.

Dimensionality Reduction

After Applying PCA, got 500 project features with explained variance around 99%, which we are going to use as a dataset for model benchmarking

Dimensionality Reduction

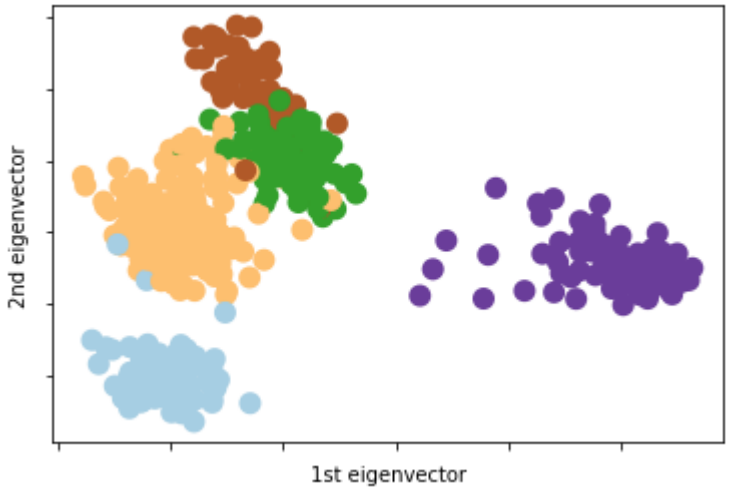
With 20264 features, we should be able to reduce the dimesionality of the dataset greatly while explaining the variance above a certain threshold. This is can be done through PCA. Or, we can perform feature selection by removing correlated variables.

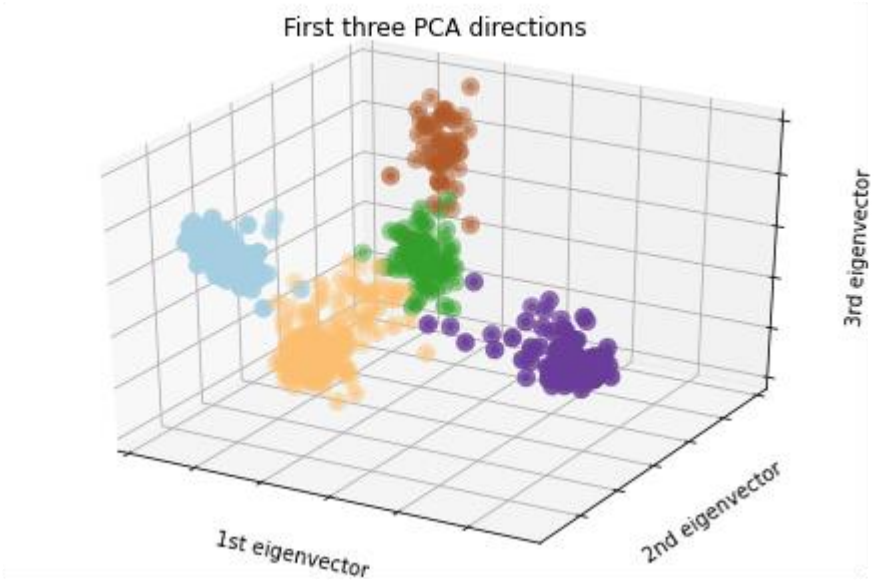
PCA

```
pca = PCA()
pca.fit_transform(X_train)
total = sum(pca.explained_variance_)
k = 0
current_variance = [0]
while sum(current_variance)/total < 0.99:
    current_variance.append(pca.explained_variance_[k])
    k = k + 1

print(k, "features explain around 99% of the variance.")

pca = PCA(n_components=k)
X_train_pca = pca.fit_transform(X_train)
X_test_pca = pca.transform(X_test)
```

```
500 features explain around 99% of the variance.
```

Principal Component Analysis





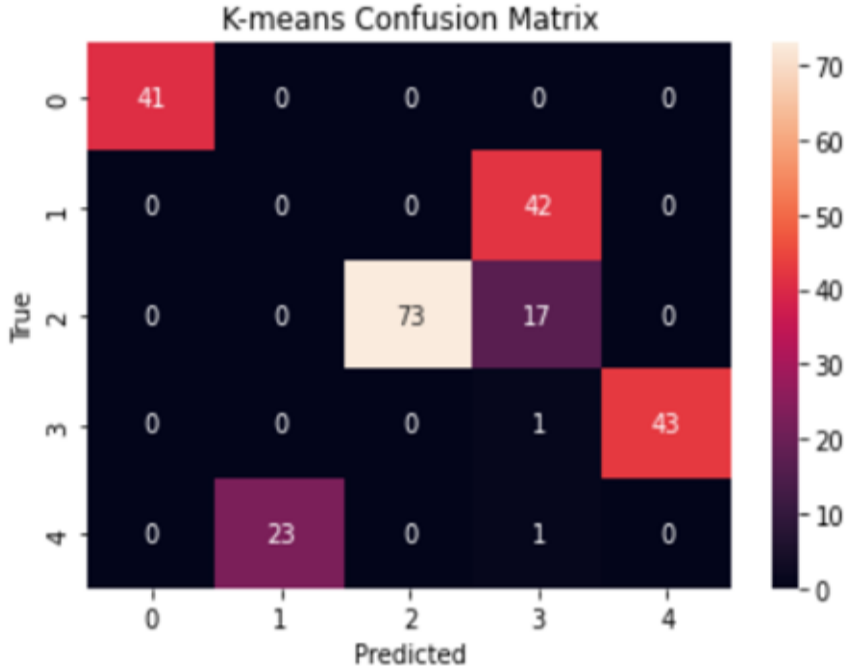
We can observe highly clustered data in the project of eigenvectors, signifying the cancer disease state is predictable from the pool of genomic features given.

Unsupervised Learning: Clustering

Implemented K-means clustering on all the 20,000 features to cluster the entire data into 5 classes.

Got quite less accuracy (47.7%), showing Unsupervised method is not reliable.

K-means accuracy: 0.477



Benchmarking Supervised Learning Models

We are going to analyze the performance of various ML models to get the best accuracy model for the current data distribution of genomic features, and same model will be used further as a reliable model to carry out performance evaluation on selected or ranked features by feature selection. Hyperparameter analysis will also be done to get the best set of parameters. Following are the models which will be analyzed along with the parameters:

- Logistic Regression
- Support Vector Classifier - Regularization Parameter, Kernels
- K Nearest Neighbors - Number of neighbors
- Random Forest Classifier - maximum depth, number of estimators, minimum sample split, minimum sample leaf
- Naive Bayes - Gaussian, Bernoulli, Complement, Multinomial
- Neural Networks - Number of Layers, Number of Neurons in a layer

Logistic Regression

	Dimensions	Description	Accuracy	F1 Score
0	18307	Correlation less than 0.8	1.000000	1.000000
1	8848	Correlation less than 0.6	1.000000	1.000000
2	1665	Correlation less than 0.4	1.000000	1.000000
3	500	PCA-explained variance of 99%	1.000000	1.000000
4	424	Correlation less than 0.3	0.946058	0.946211
5	51	Correlation less than 0.2	0.589212	0.588333

From the above table, we can see that we can find biomarkers through feature selection in the range of dimensions where correlation is less than 0.3

Only for datasets having dimensions 424 and 51, the accuracy is not 100% due to lack of features.

Support Vector Machine

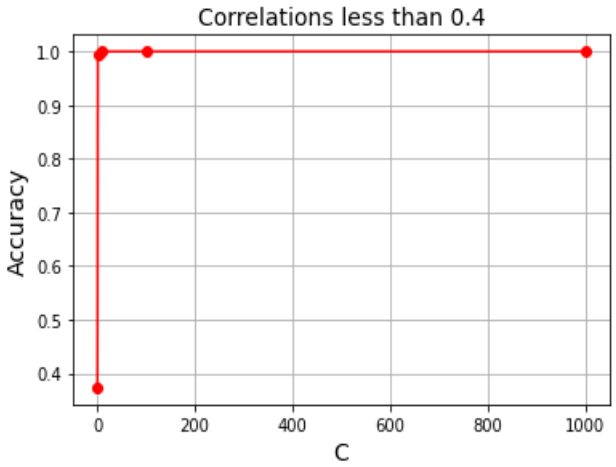
	Dimensions	Description	Accuracy	F1 Score
0	18307	Correlation less than 0.8	1.000000	1.000000
1	8848	Correlation less than 0.6	1.000000	1.000000
2	1665	Correlation less than 0.4	0.995851	0.995838
3	500	PCA-explained variance of 99%	1.000000	1.000000
4	424	Correlation less than 0.3	0.937759	0.937952
5	51	Correlation less than 0.2	0.373444	0.203081

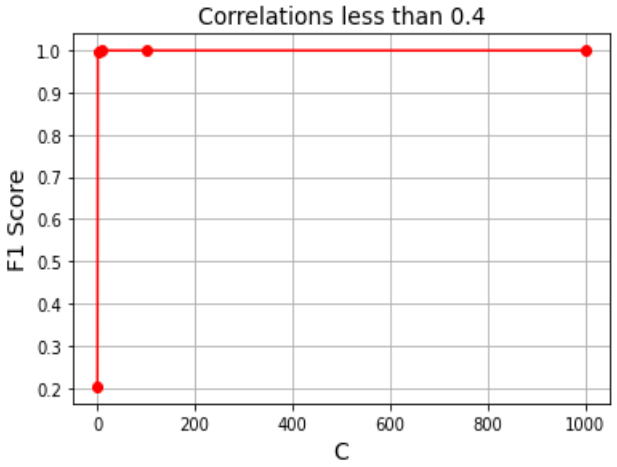
Here we can see for dimensions 1665, 424, and 51, the testing accuracy is not 100%. Hence we will carry out hyper-parameter tunning for these datasets.

Perform slightly worse than Logistic Regression, due to a certain margin of the linear classifier with the nearest sample. Performance can be improved by varying the Regularization parameter.

Hyperparameter Tuning of C with 1665 dims

	C	Accuracy	F1 Score
0	0.1	0.373444	0.203081
1	1.0	0.995851	0.995838
2	10.0	1.000000	1.000000
3	100.0	1.000000	1.000000
4	1000.0	1.000000	1.000000

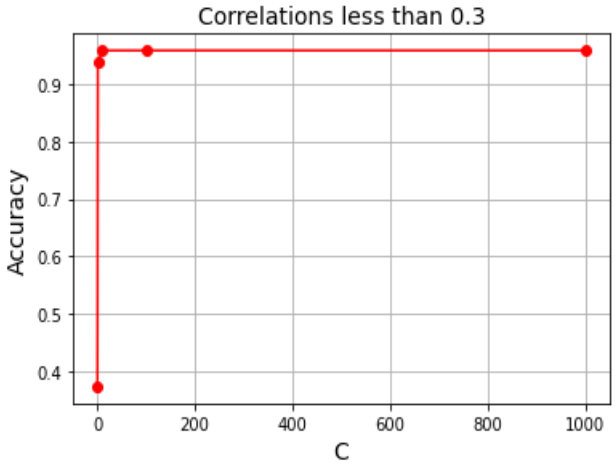


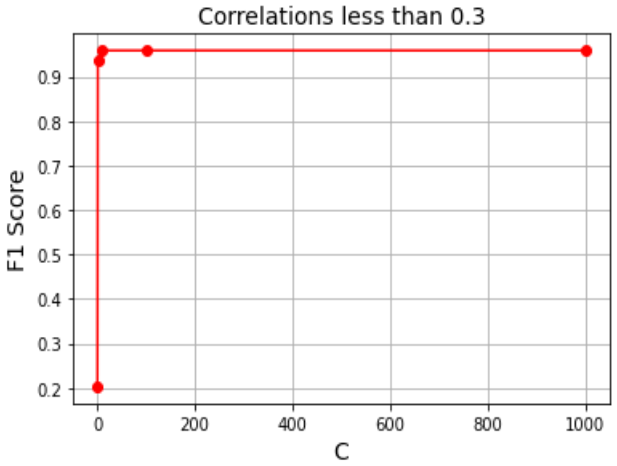


Increased performance as increasing the regularization parameter, due to good number of dimensions in the data.

Hyperparameter Tuning of C with 424 dims

	C	Accuracy	F1 Score
0	0.1	0.373444	0.203081
1	1.0	0.937759	0.937952
2	10.0	0.958506	0.958606
3	100.0	0.958506	0.958606
4	1000.0	0.958506	0.958606

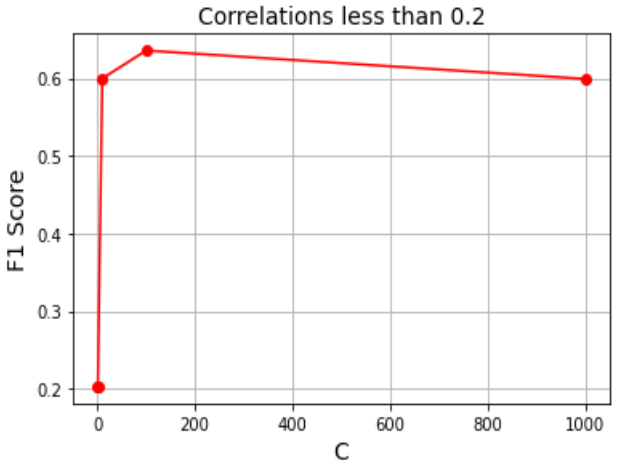


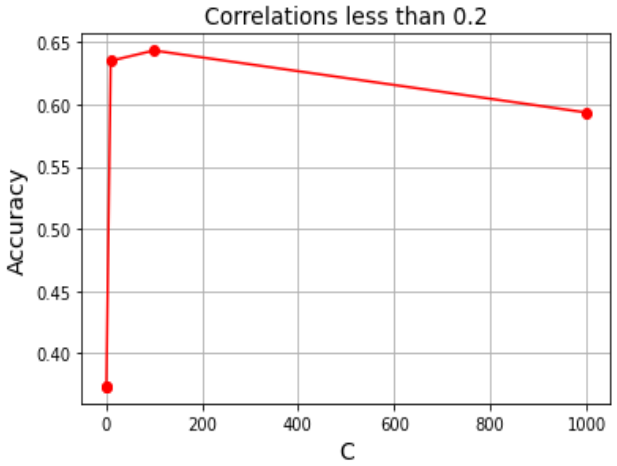


Performing better than logistic regression for high regularization parameter, due to significantly good number of dimensions.

Hyperparameter Tuning of C with 51 dims

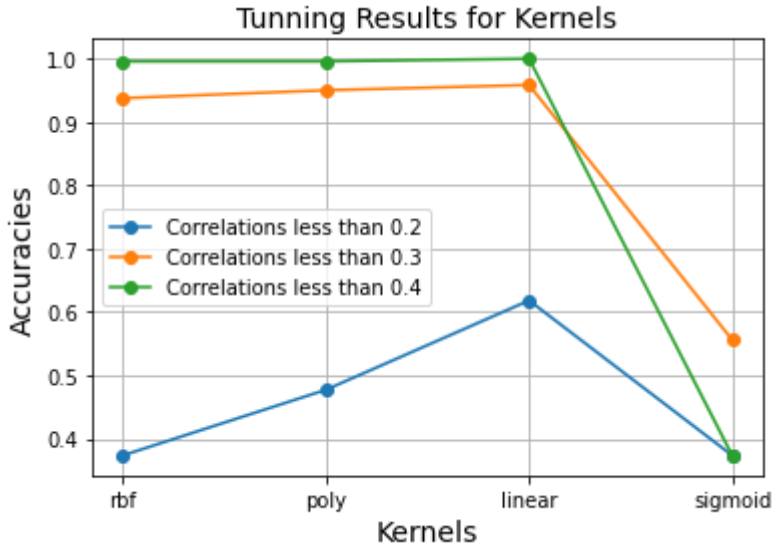
	C	Accuracy	F1 Score
0	0.1	0.373444	0.203081
1	1.0	0.373444	0.203081
2	10.0	0.634855	0.599769
3	100.0	0.643154	0.635839
4	1000.0	0.593361	0.599280

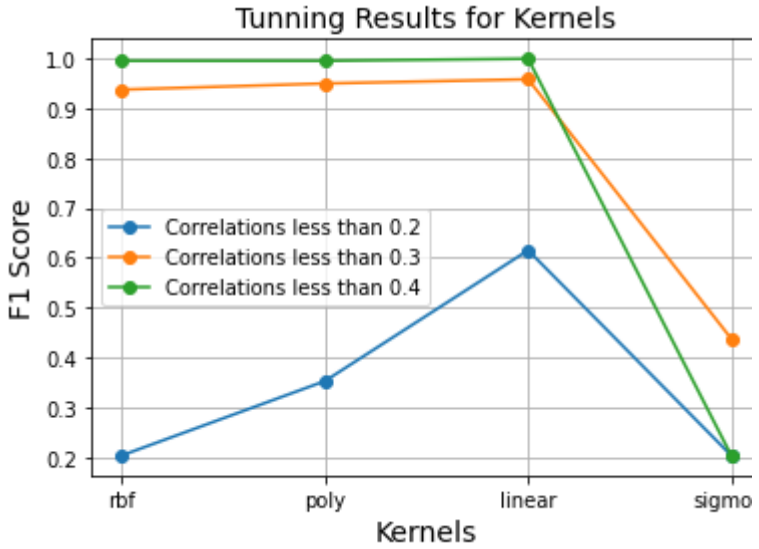




Performance is degrading for more regularization due to already less number of features, overfitting might happen.

Hyperparameter Tuning of Kernel





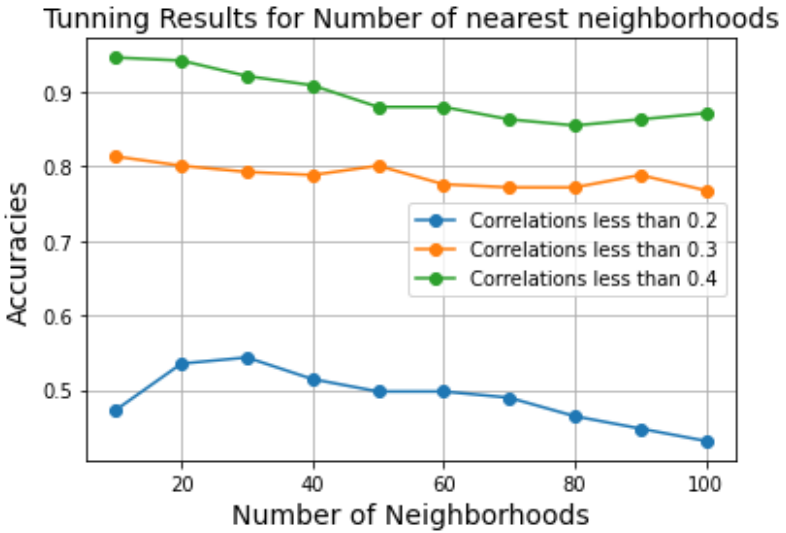
We can see from the above analysis, that sigmoid kernel is the worst, as sigmoid has a problem of vanishing gradients. And for dataset having correlations less than 0.2, linear kernel is best due to low number of dimensions in the data, hence linearly separable.

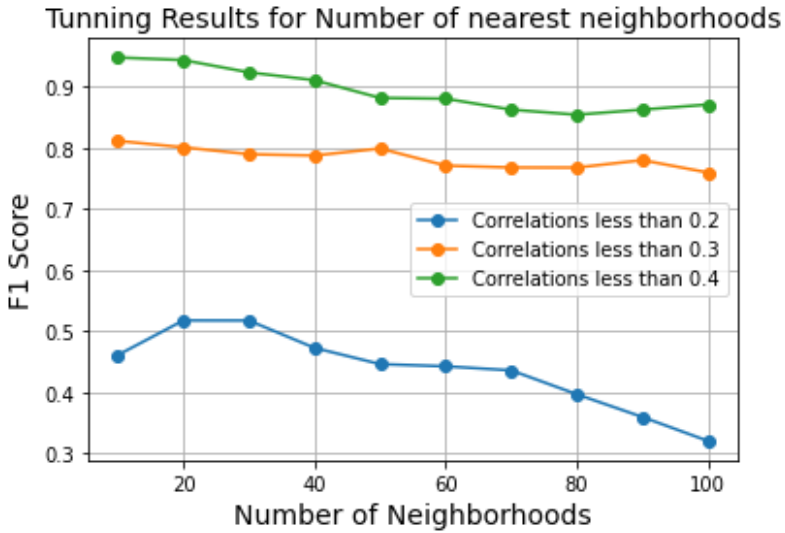
K-nearest Neighbours

	Dimensions	Description	Accuracy	F1 Score
0	18307	Correlation less than 0.8	0.991701	0.991679
1	8848	Correlation less than 0.6	0.995851	0.995831
2	1665	Correlation less than 0.4	0.871369	0.871235
3	500	PCA-explained variance of 99%	0.991701	0.991679
4	424	Correlation less than 0.3	0.767635	0.759863
5	51	Correlation less than 0.2	0.431535	0.320198

Here we can see for dimensions 1665, 424, and 51, the testing accuracy is not 100%. Hence we will carry out hyper-parameter tunning for these datasets

Hyperparameter Tuning of Number of Nearest Neighbourhoods





We can see from the above 2 graphs, n_neighbors = 30 is best for the data whose correlations are less than 0.2, whereas n_neighbors = 50 is best for correlations are less than 0.3. It can be seen that as the neighborhoods increase, accuracy decrease due to taking outlier data-points (or datapoints of other classes) also into account for prediction.

Random Forest Classifier

	Dimensions	Description	Accuracy	F1 Score
0	18307	Correlation less than 0.8	0.995851	0.995838
1	8848	Correlation less than 0.6	0.995851	0.995838
2	1665	Correlation less than 0.4	0.975104	0.975181
3	500	PCA-explained variance of 99%	0.987552	0.987587
4	424	Correlation less than 0.3	0.929461	0.929269
5	51	Correlation less than 0.2	0.659751	0.651427

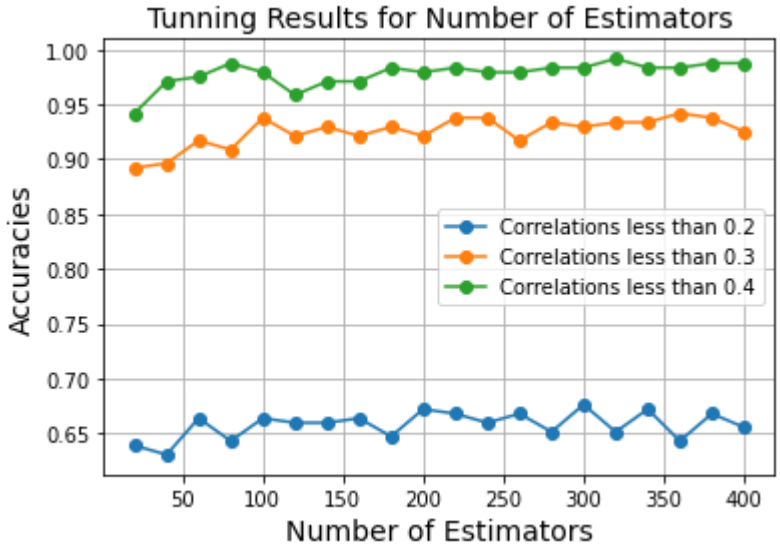
Here we can see for dimensions 1665, 424, and 51, the testing accuracy is not 100%. Hence we will carry out hyper-parameter tunning for these datasets

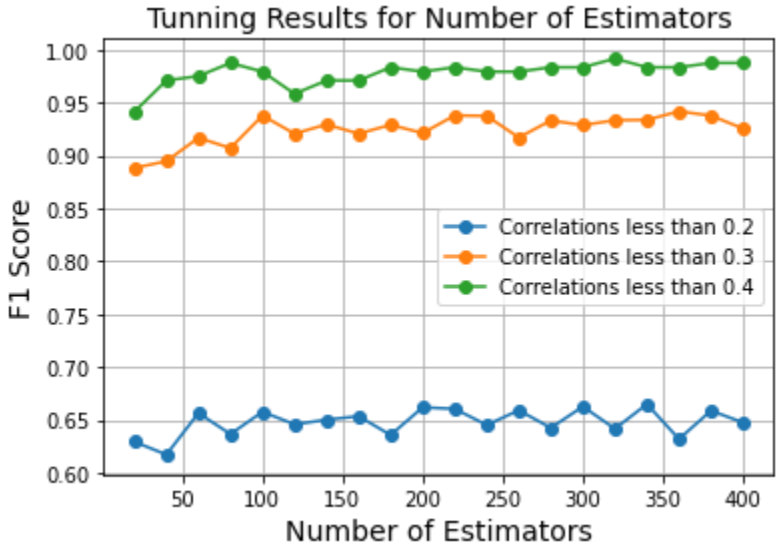
Giving almost best accuracy on dataset having 51 dimensions as compared to other classification models. Reduced accuracy for other datasets due to high variance.

Hyperparameter Tuning

- Number of Estimators
- Depth of Tree
- Minimum Samples Split
- Minimum Samples Leaf

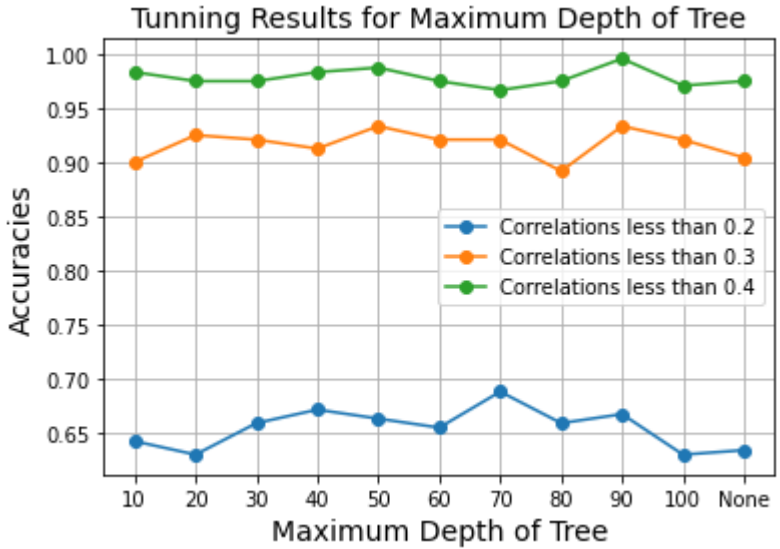
Hyperparameter Tuning of Number of Estimators

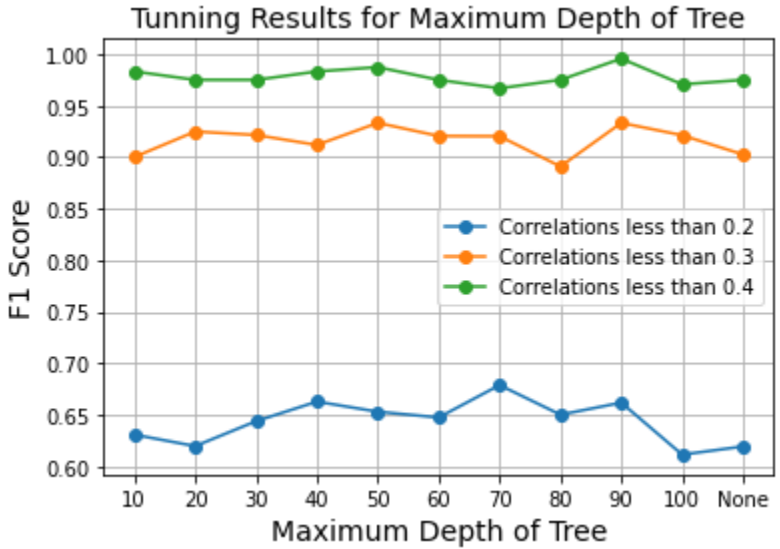




Fluctuating behaviour for number of estimators is observed for all the 3 datasets.

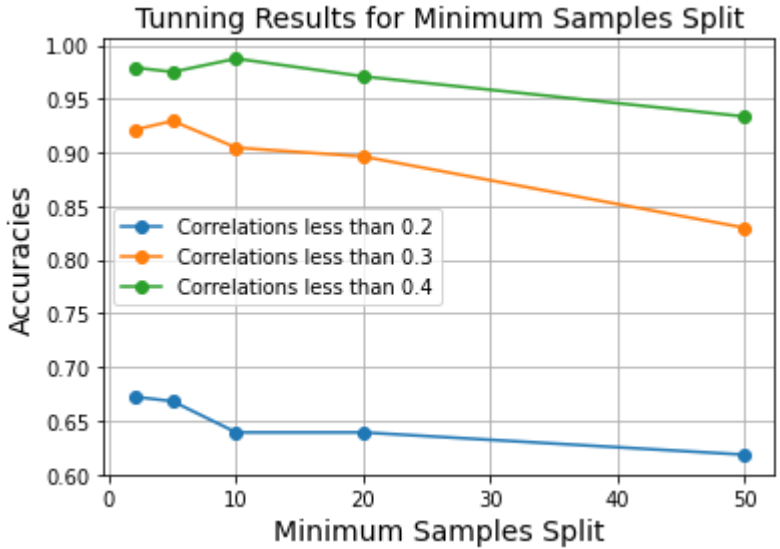
Hyperparameter Tuning of Maximum Tree Depth

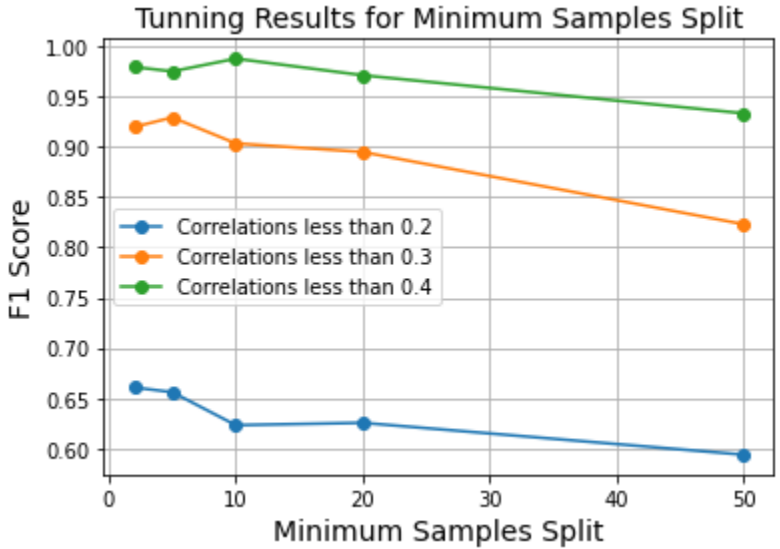




Fluctuating behaviour is observed for maximum depth of tree also, limited maximum tree depth is giving typically more accuracies as compared to infinite tree depth

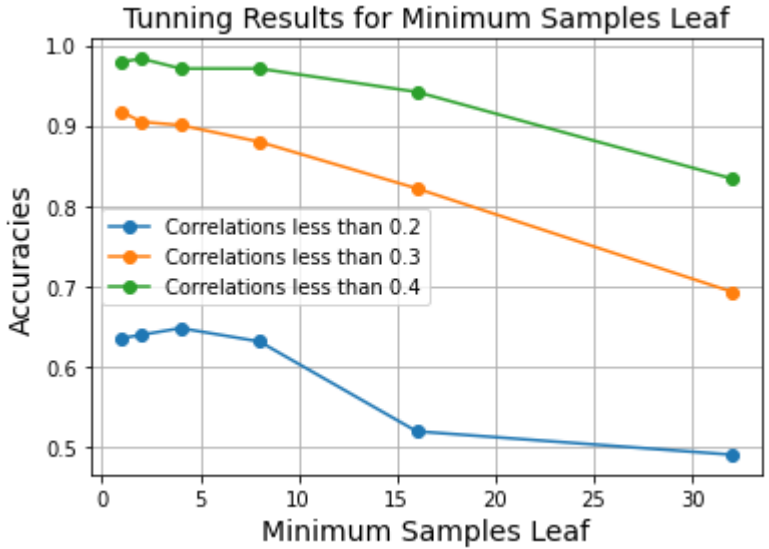
Hyperparameter Tuning of Minimum Samples Split

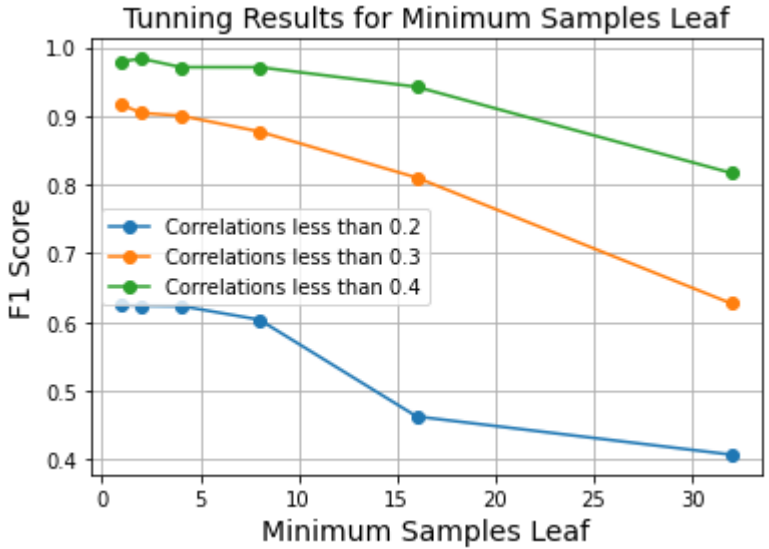




Accuracy is decreasing as the minimum sample split size increases. Accuracy is maximum in most of the cases when min sample split size is 2

Hyperparameter Tuning of Minimum Samples Leaf





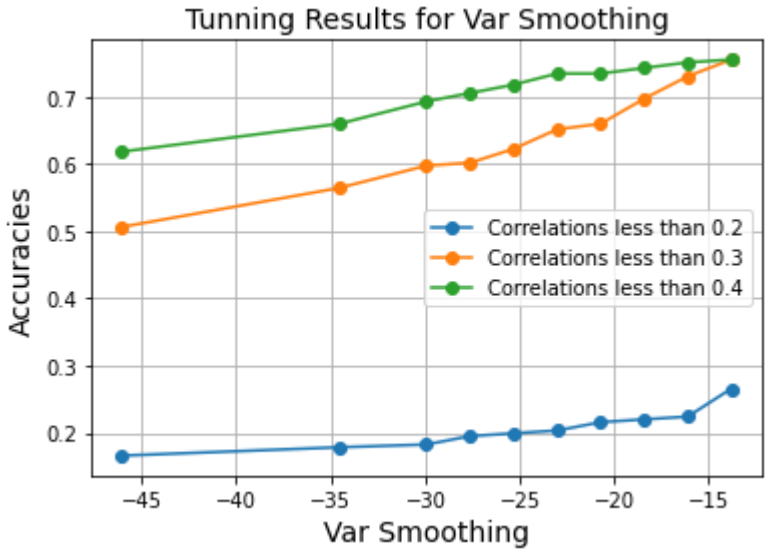
Accuracy is decreasing as the minimum sample leaf increases. Accuracy is maximum in most of the cases when min samples leaf size is 1

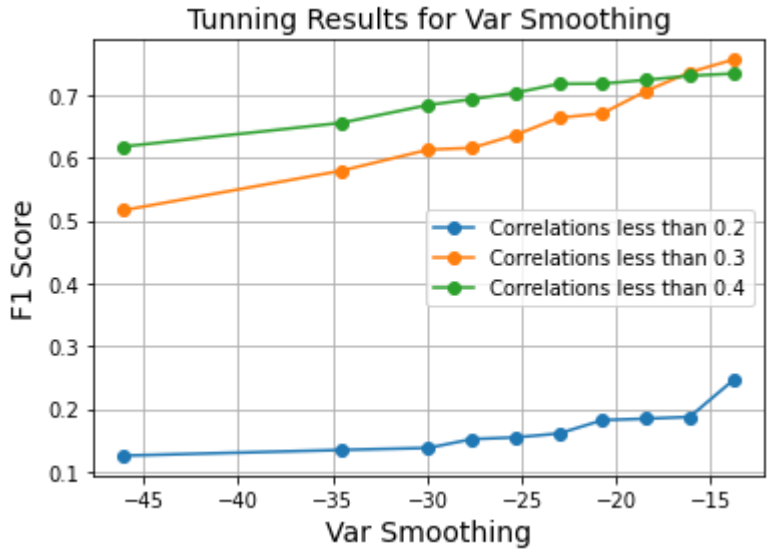
Naive Bayes Classifier

	Dimensions	Description	Accuracy	F1 Score
0	18307	Correlation less than 0.8	0.763485	0.741594
1	8848	Correlation less than 0.6	0.697095	0.666853
2	1665	Correlation less than 0.4	0.734440	0.717946
3	500	PCA-explained variance of 99%	0.979253	0.979617
4	424	Correlation less than 0.3	0.659751	0.670703
5	51	Correlation less than 0.2	0.215768	0.182395

Performing worst on all the datasets as compared to other models, due to the assumption of independent predictors coming from a distribution, which is not very likely, leading to worse performance.

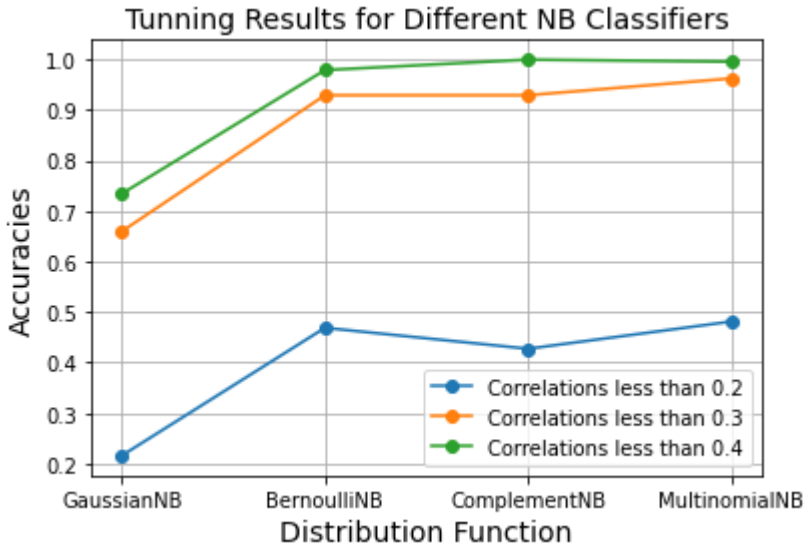
Hyperparameter Tuning of Var Smoothing

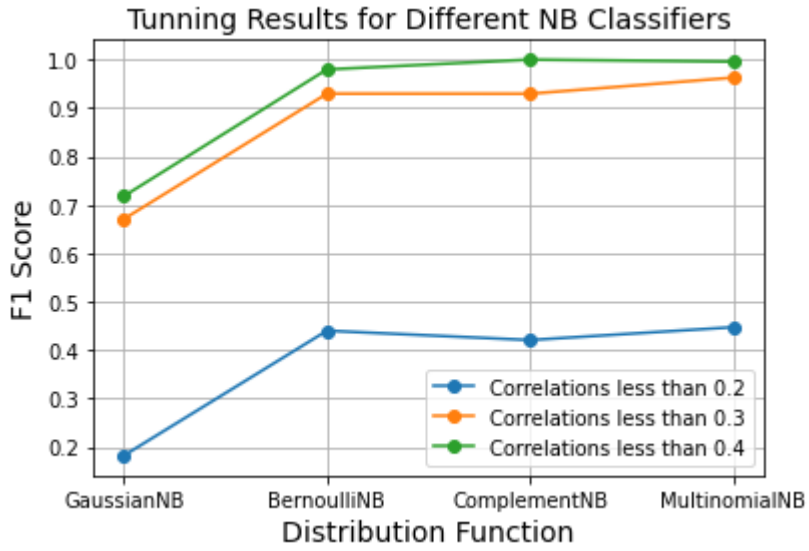




Increasing smoothing leads to increase in the performance as it helps tackle the problem of zero probability. Higher alpha values push the likelihood towards a value of 0.5, i.e., the probability of a word equal to 0.5 for both the positive and negative reviews

Hyperparameter Tuning of Classifier





NB on Gaussian distribution is performing the worst as compared to other distributions in all the datasets, MultinomialNB is performing the best among all

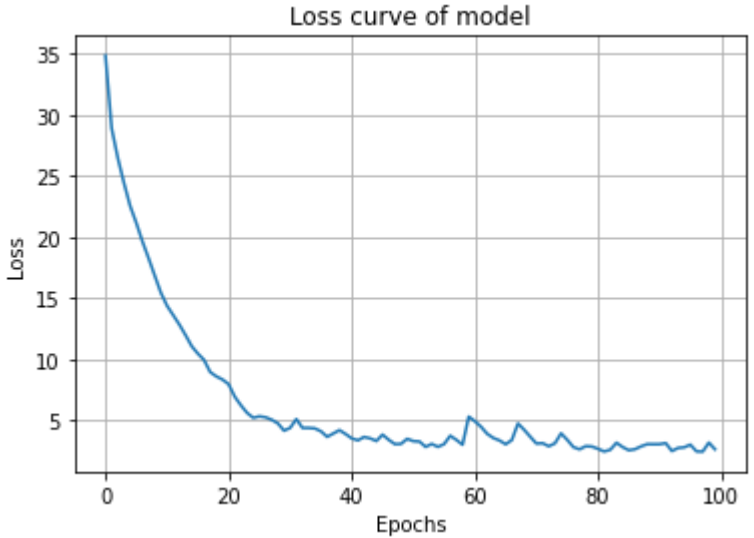
Neural Networks

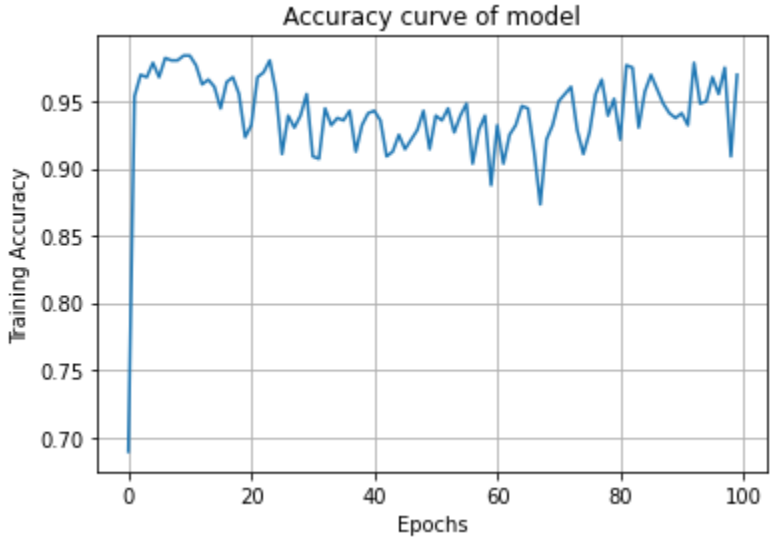
For evaluating the neural net on different parameters, we are taking certain parameters as common:
n_layers = 4
n_neurons = 256
l1 = 1e-3

	Dimensions	Description	Accuracy	F1 Score
0	1665	Correlation less than 0.4	0.510373	0.395715
1	500	PCA-explained variance of 99%	1.000000	1.000000
2	424	Correlation less than 0.3	0.842324	0.815901
3	51	Correlation less than 0.2	0.535270	0.516913

Neural Net is performing significantly good on data having dimension 51 as compared to other models due to introduced non-linearity, but performing worst on data having 1665 dimensions, due to very low bias and very high variance, leading to overfitting the model.

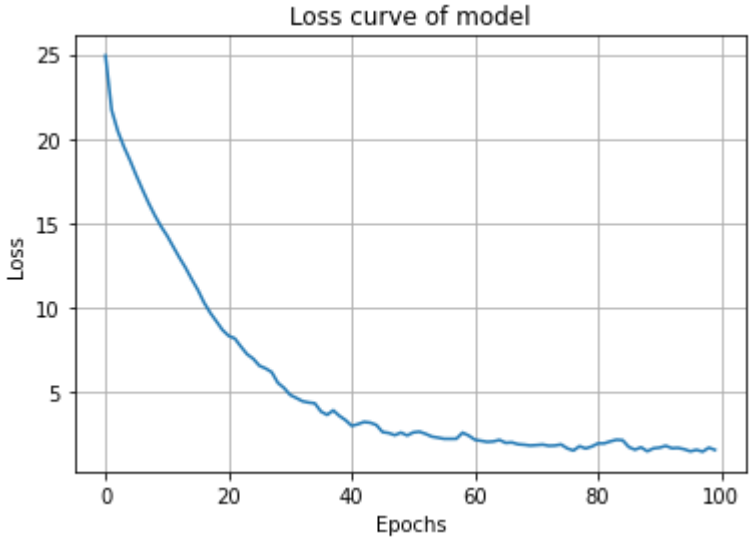
Model with 1665 dims

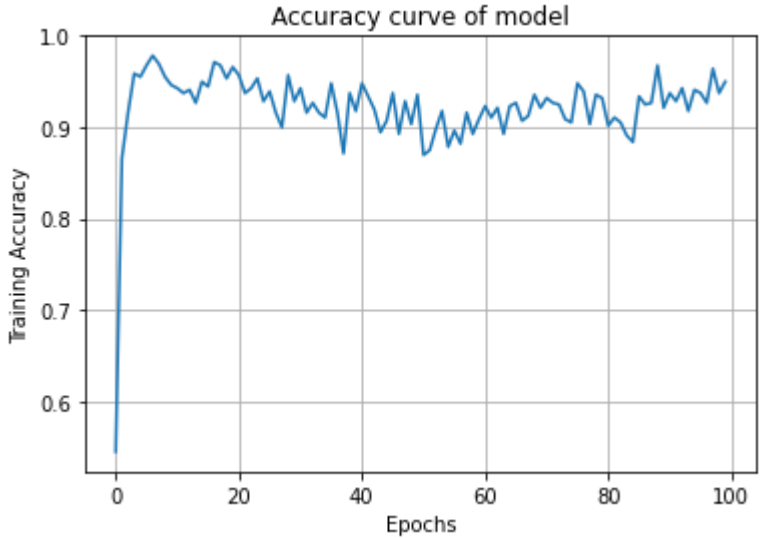




Loss decreases and training accuracy increases as epoch increases upto a certain point, then shows erratic behaviour

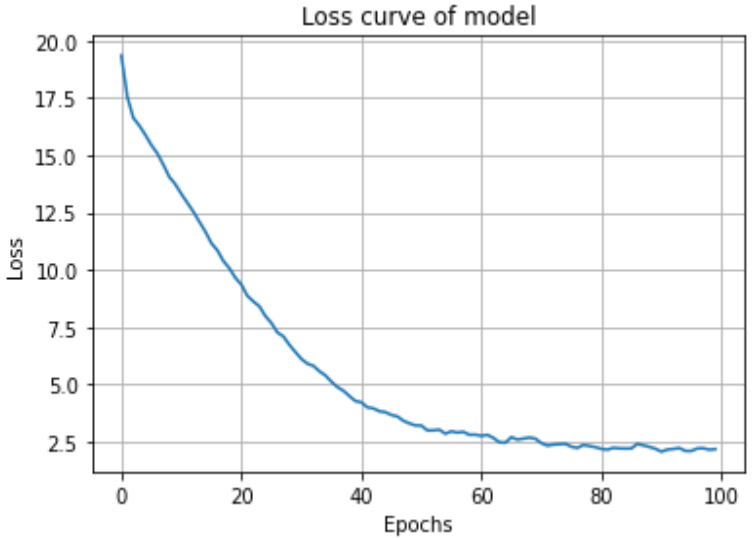
Model with 424 dims

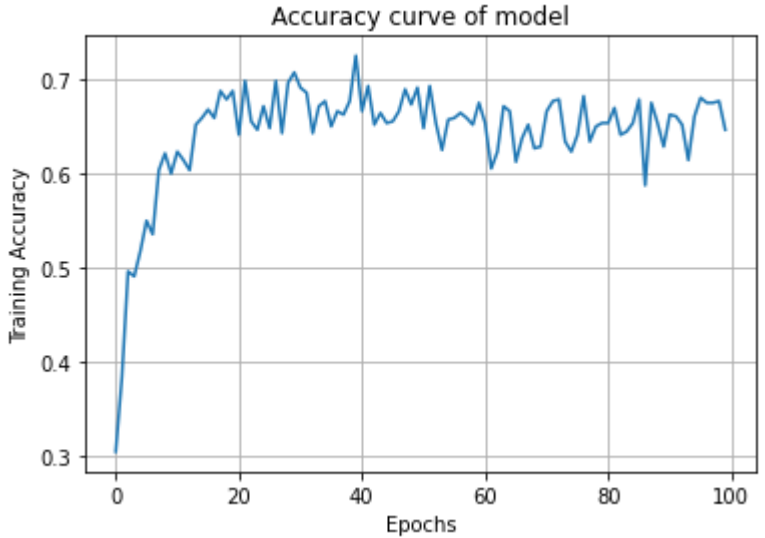




Loss decreases and training accuracy increases as epoch increases upto a certain point, then shows erratic behaviour. Loss decreases and accuracy increases at 0 epoch as number of dims decreases

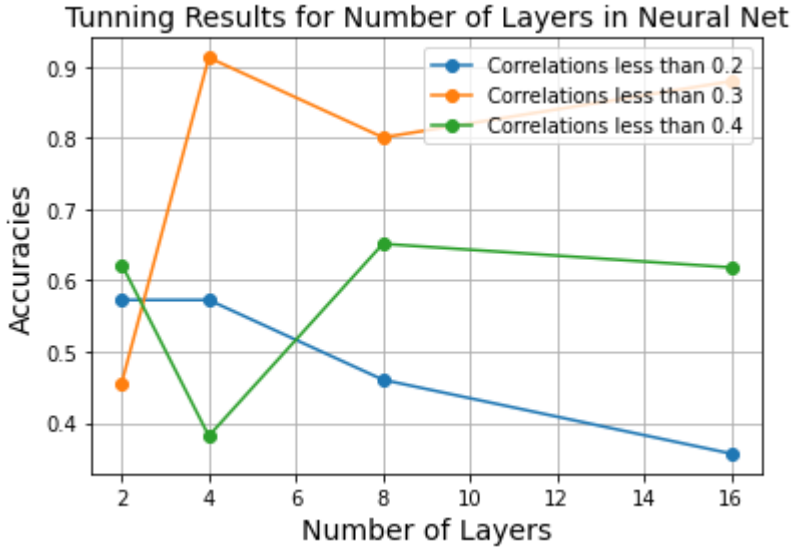
Model with 51 dims

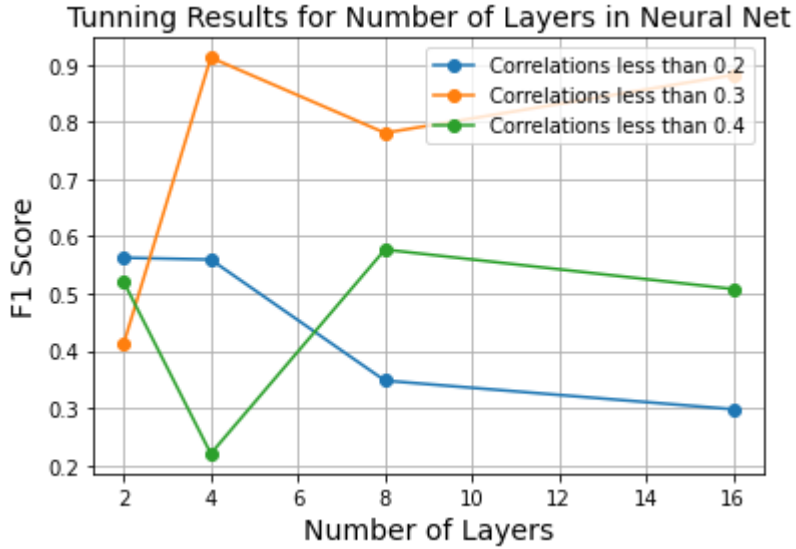




Loss decreases and training accuracy increases as epoch increases upto a certain point, then shows erratic behaviour. Loss decreases and accuracy increases at 0 epoch as number of dims decreases.

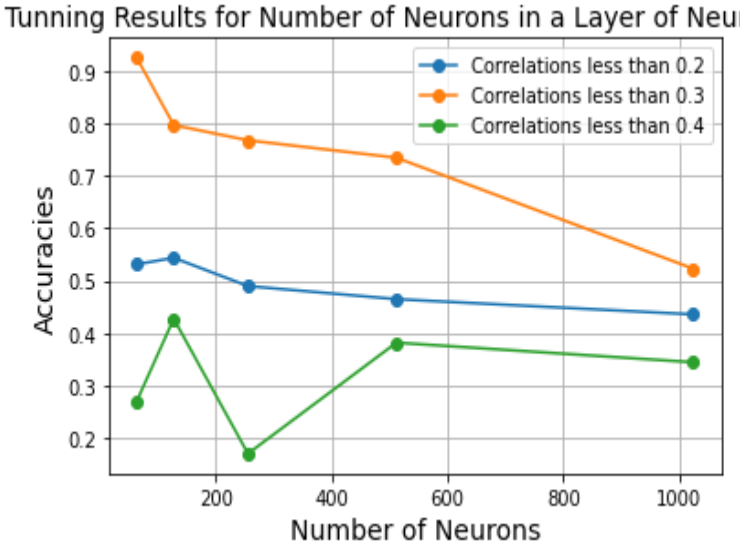
Hyperparameter Tuning of Number of Layers

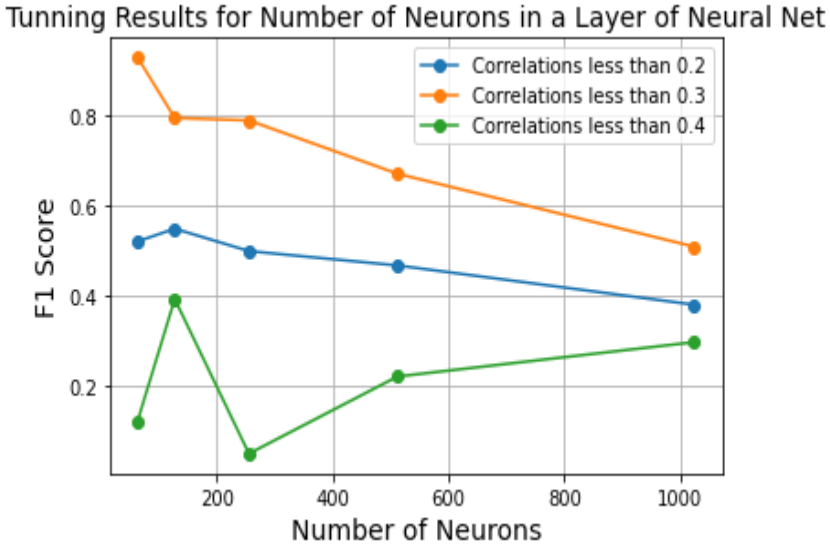




Increasing number of layers to 4 increases the performance, but after that performance decreases due to overfitting. Whereas data having correlations less than 0.4 has minimum performance for 4 layers, which is abnormal.

Hyperparameter Tuning of Number of Neurons





Deep networks are always better than wide networks, this can be seen in the above curve, as the network goes wide, the performance decreases. Incase of data having correlations less than 0.4, performance is increasing due to its high dimensionality, as wide networks will be required to extract the features from high dimensionality.

Results of Model Training

Comparison Between All The Models

	Datasets	Dimensions	Logistic Regression	SVM	KNN	RFC	Gaussian NB	Neural Net
0	Correlation less than 0.8	18307	1.000	1.000	0.992	0.996	0.763	NA
1	Correlation less than 0.6	8848	1.000	1.000	0.996	0.996	0.697	NA
2	Correlation less than 0.4	1665	1.000	0.996	0.871	0.976	0.734	0.51
3	PCA-explained variance of 99%	500	1.000	1.000	0.992	0.987	0.979	1
4	Correlation less than 0.3	424	0.946	0.938	0.768	0.929	0.659	0.842
5	Correlation less than 0.2	51	0.589	0.373	0.432	0.659	0.216	0.535

The dimensionally reduced projection of 500 dimensions using PCA is showing best accuracy on almost all the models, but since the feature identity is lost as it is projected to some other dimension, we can't rely on PCA, as it just reduces the model complexity, but doesn't carry out feature selection.

Summary

Based on the above analysis, we can conclude that we can move forward with the dataset having features whose correlations are less than 0.3, as that dataset is showing significant accuracy on 3 of the models.

We need to analyze the hyperparameters of these 3 models based on the hyper-parameter tuning done for this dataset on these 3 models.

These 3 models are:

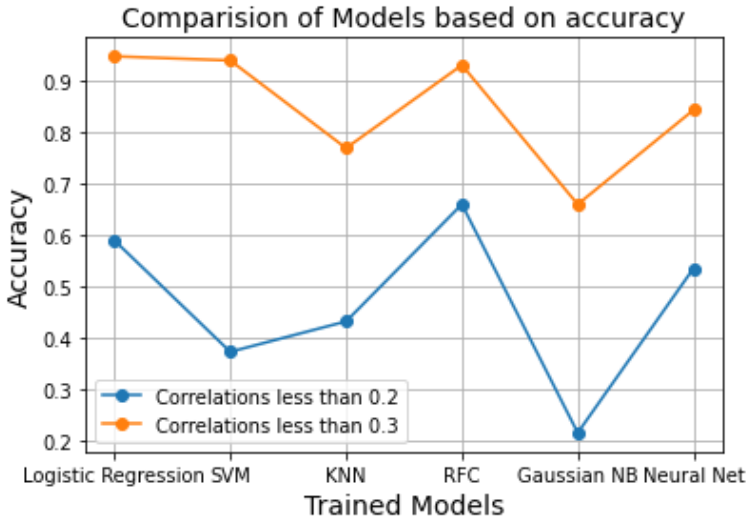
1. Logistic Regression
2. SVM
3. RFC

Logistic Regression has its accuracy of **0.946**

After analyzing the parameters, we got the best parameters using SVM is C=10 with 0.958 accuracy.

After analyzing the parameters, we got the best parameters using RFC are Number of Estimators=100, Maximum Depth of Tree = 90, min_samples_split= 5, and min_samples_leaf = 1. In the graph, we can see the accuracy of 0.913 from the best parameters.

So, from here we can conclude, that we can use Logistic Regression and SVC(C = 10) as best benchmarking models for the dataset for further analysis.



Feature Selection Methodology

Feature Selection Techniques

We have eliminated features whose correlations are greater than 0.3, hence we have got total 424 features which are giving good accuracies via ML models and Neural Networks.

On these 424 extracted genomic features, we are going to carry out feature selection through below techniques:

- Feature Selection through Teacher Student Network
- Feature Selection by selecting KBest through chi2 scores
- Backward elimination through P-values

Implementation of Official paper Deep Feature Selection using a Teacher-Student Network has been done to carry out feature rankings on the genomic features.

Feature Selection Through Teacher-Student Net

The teacher network can be trained in 2 ways to generate the low dimensional encodings - Supervised and Unsupervised.

Supervised Encoder is trained by training an end-to-end deep network with one layer having less number of neurons (the size of embeddings, typically 5 or 10), and the last layer contains the categorical scores. Hence, the softmax or sigmoid activation function is used in the last layer.

Unsupervised Encoder is trained by training an end-to-end deep network with the last layer containing the same features, hence the network first encodes the provided high dimensional features into low dimensions, and then those low dimensions are again retained back to the same features using a decoder.

After getting the low dimensional representations of the features, these representations are replicated by the student network containing one layer only with the number of neurons typically equal to 10 times the embedding size. Mean square error is used as a loss function to train the network. After training, the weights of the 1st layer of the student network are exploited to find the feature rankings of the high dimension features based on their weighted contribution according to the formula:

Feature score of i = SUM(j, Wij*Wij)

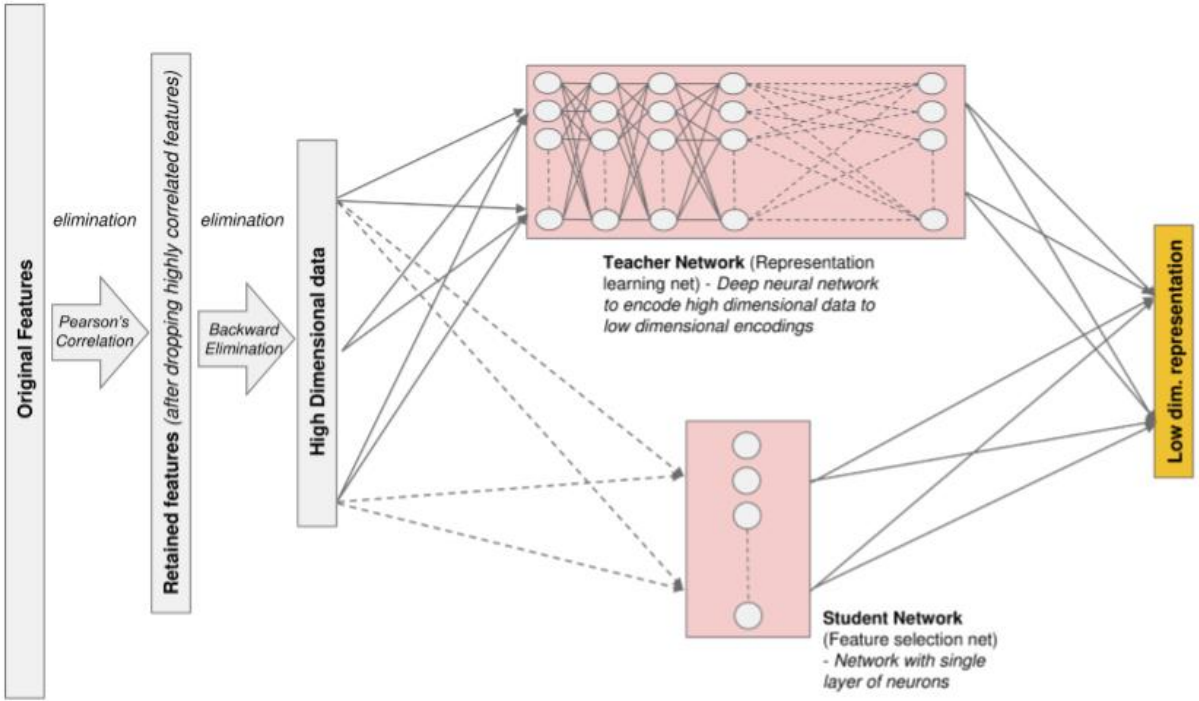
Feature Selection Through Teacher-Student Net

To generate the encodings, we are using total 7 encoders to produce the low dimensional representations of high dim(424) data, as per mentioned in the paper:

- **Isomap**
- **TSNE**
- **LocallyLinearEmbedding**
- **MDS**
- **SpectralEmbedding**
- **Supervised Neural Net Encoder**
- **Unsupervised Neural Net Encoder**

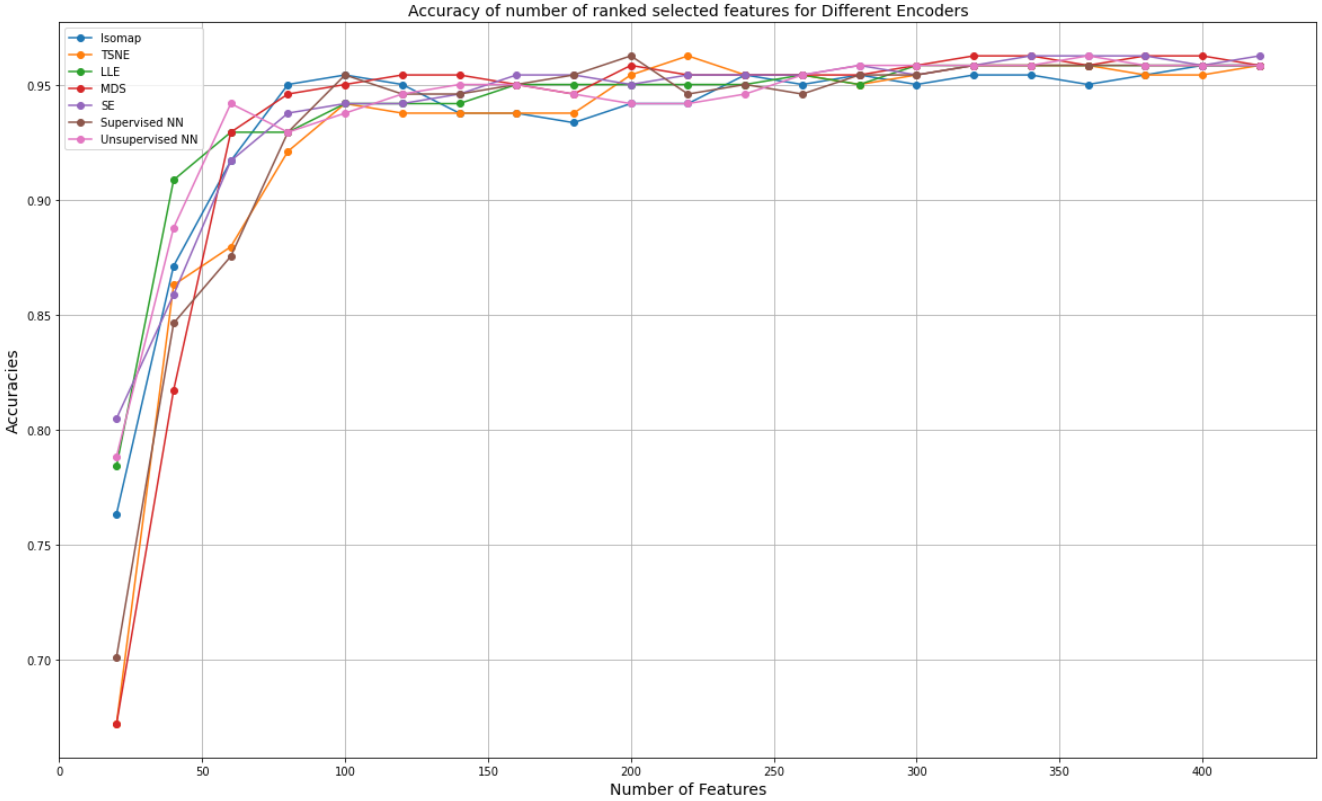
We are going to compare the results of feature selection by using these encodings on the test data by the best ML models (SVC and Logistic Regression) that we have got from the model benchmarking.

Teacher-Student Net: Flow Diagram

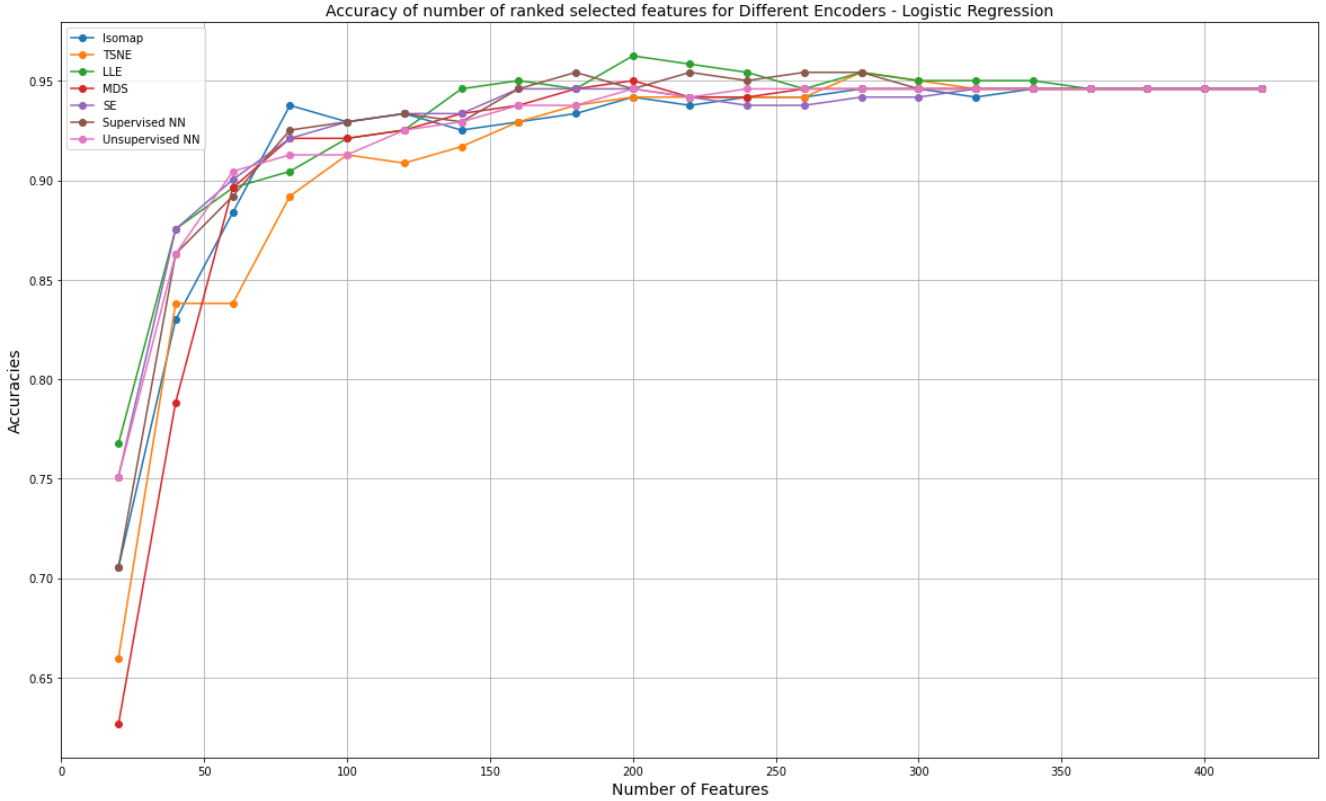


After implementing all the encodings to get the corresponding feature rankings from the student net, the top rankings of the features are trained over SVC(C=10) and Logistic regression.

Alongside is the curve for test accuracy through SVC.



Alongside is the curve for test accuracy through Logistic Regression.



Teacher-Student Net: Observations

From the last 2 curves trained over top n features and their corresponding accuracies, we can conclude:

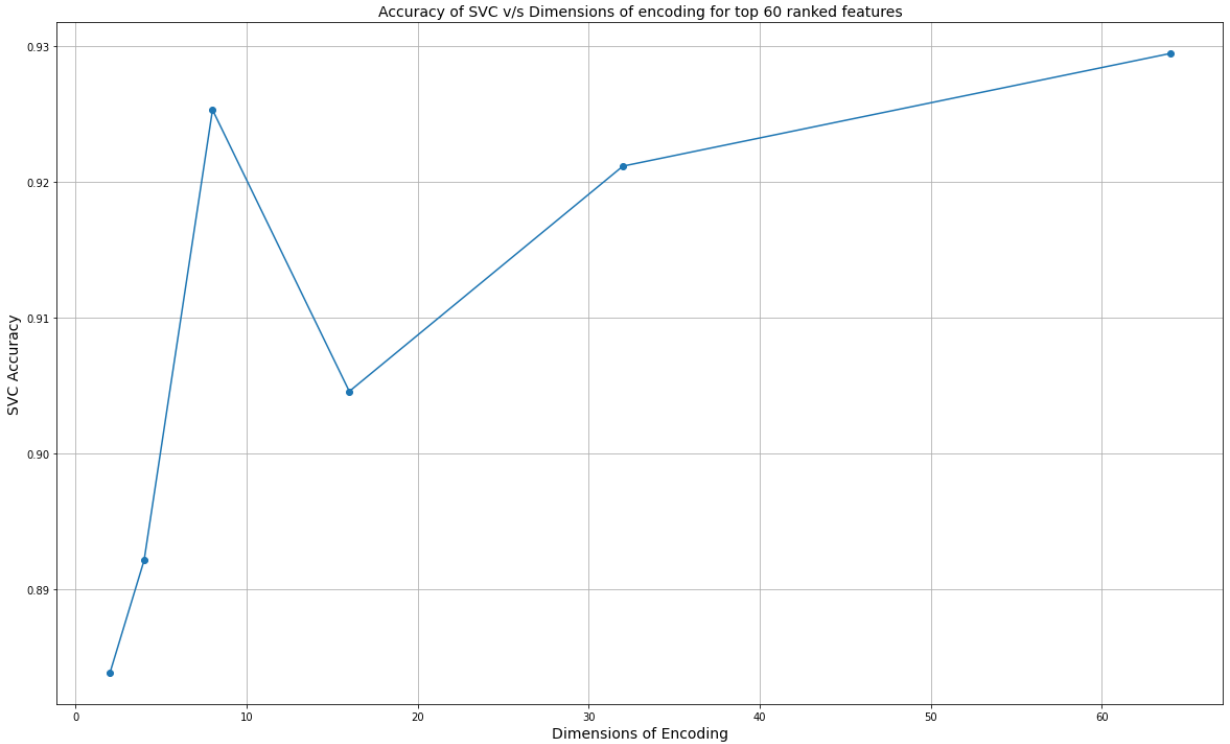
- From the SVC, we can see more accuracies than the Logistic Regression
- In SVC, for Number of Genes = 60, we got optimum accuracy using unsupervised encoder, which is matching the hypothesis mentioned in the paper (about state-of-art results of supervised and unsupervised encoder over other encoders) . Hence we are going to analyze the effect of dimensions of encodings on Unsupervised Encoder for FS by Student network using SVC.
- In SVC, for Number of Genes = 200, we got optimum accuracy for supervised classifier, which is the maximum among all the data-points on the curve.

Hence, we have got the ***top 60 genomic feature biomarkers*** which can bring optimum accuracy of **94.5%** over SVC model from 424 features

Teacher-Student Net: Parametric Analysis

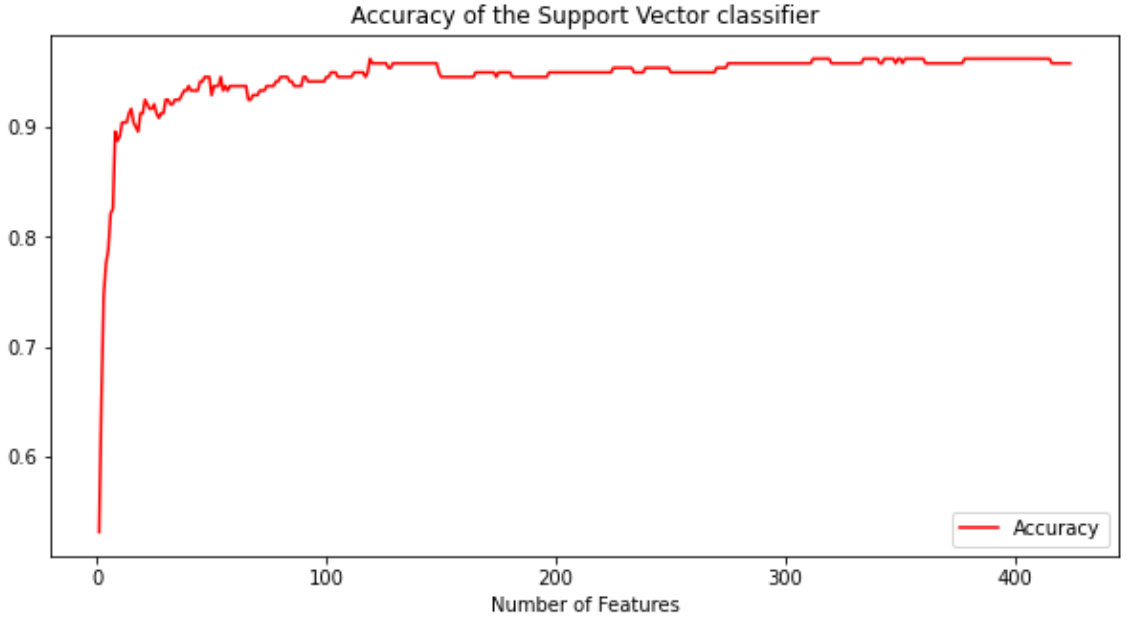
For Unsupervised encoder with top 60 features taken, dimensions of encoding is varied, and top 60 features came out is examined by SVC whose accuracy is reported in the graph alongside.

64 size of encodings is showing maximum accuracy, along with size 8 showing considerably good accuracy



Forward Elimination using chi2 Test

- Chi2 scores are calculated for all the 424 features
- Top k features based on the chi2 scores are selected and accuracy is measured over SVC(C=10)
- The point of plateauing off the curve is examined to get the optimal number of top features for good accuracy.



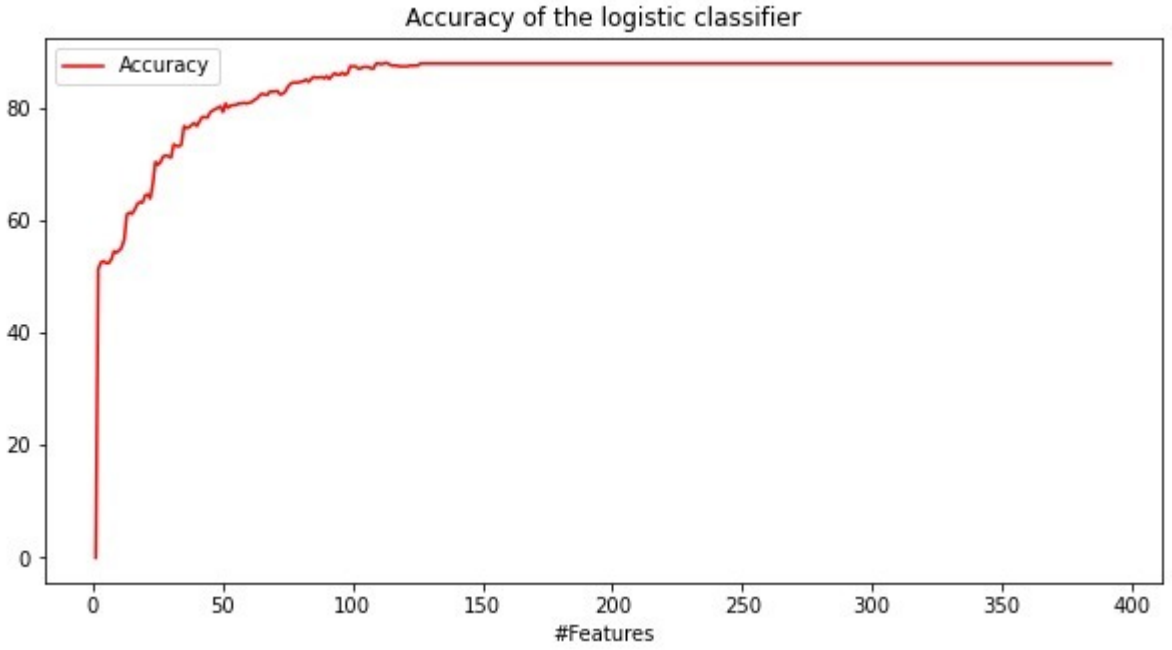
It can be seen that curve is plateauing off at 60 top features, and the corresponding ***accuracy on SVM is 93.77%***

Feature selection by backward elimination through correlation matrix

- Features can be eliminated on account of high correlation with other features in the dataset.
- Performing feature elimination,
  20,351 features had correlations <1
  20,093 features had correlations <0.9
  14,778 features had correlations <0.7
  5,121 features had correlations <0.4
  393 features had correlations <0.2

Feature selection by forward selection

A logistic classifier was trained on increasingly higher and higher number of features. We see that at around 120 features, the logistic classifier attains it's most accurate output.



Backward Elimination using p-values

We will be carrying out backward elimination on the basis of p-values. This approach is carried out in the following steps:

- Selecting a significance level, or P-value in other words. Usually, a 5% significance level (P-value of 0.05) is selected
- Fit the model with all possible features
- Remove the feature with the highest P-value
- Fit the model on the remaining features until the P-values of all the features will be less than the significance level that we selected

To fit the OLS regressor, we have chosen class 2 ('BRCA') as a binary variable, as it is the most occurring class in the dataset.

After backward elimination, ***101 features have p-values less than 0.05.***

These 101 features are trained over Logistic Regression and SVC to get the corresponding accuracies:

Logistic Regression: 90.87% **SVC: 93.36%**

Results from Feature Selection

Feature Selection Technique	Number of Genomic Features selected	Accuracy on SVC	Accuracy on Logistic Regression
Teacher-Student Net ***(Unsupervised Teacher Net)***	60	94.5%	91.5%
Chi2 Test	60	93.77%	-
Backward elimination through P-values	101	93.36%	90.87%

From the above table, we can conclude that Teacher-Student Net technique for feature selection seems most appropriate among all the 3 techniques, giving highest accuracy on SVC and Logistic Regression with lowest set of genomic features.

Unsupervised Teacher encoder with ***number of neurons*** = ***5*** at encoder layer will give the optimum 60 features for best accuracy (from parametric analysis)

Concluding Remarks for Project

The AIM of the project is to select the best combination of genomic biomarkers in a reduced number to predict the cancer state with optimum accuracy

- Benchmarking of all the ML models and Neural nets is done over the dataset to examine the model giving best accuracy for the given dataset. Logistic regression and SVC are the best one with appropriate parameters
- ML methods of Feature Selection (Backward and Forward elimination) are carried out on 424 uncorrelated dimensions to get the optimal number of features and accuracies
- Deep Feature selection approach using Teacher-student Network approach is implemented using 7 types of encoders, in which unsupervised and supervised teacher encoders showed significantly good performance(aligning with the thesis of paper). Overall best feature selection model came out to be the Teacher Student Net.
- Finally got ***60 Genomic Biomarkers***, from a pool of 20k+ features, which can be measured on a lab scale to successfully predict the cancer state in the healthcare industry.

Future Scope

- This research can be extended to other diseases which are affected by the gene distribution in the samples. Implementing same procedure roadmap to other healthcare datasets can help in the better understanding of feature selection.
- Choosing some dataset, which has reported compounds of genes will be more helpful while feature selection, as selected genomics biomarkers can be traced back through the genomic paths to find the underlying cause of the disease, which may lead to personalized medicine in the near future for the treatment
- This study can be extended to metabolomic and proteomic datasets also. Diabetes is a well known metabolomic disease (https://www.ebi.ac.uk/metabolights/MTBLS1/metabolites)
- Other feature selection techniques through ML and neural networks need to be explored to find the overlap among the selected top ranked features leading to accurate prediction, that is checking whether we are getting same features from other feature selection techniques or not.

Appendix

- Code (jupyter notebook): 180020088-18B090003-190100112-190110030.ipynb
- Link to the Paper: https://arxiv.org/pdf/1903.07045.pdf
- Github Repository Reference for Teacher-Student Net: https://github.com/alimirzaei/TSFS
- Dataset: UCI Machine Learning Repository: gene expression cancer RNA-Seq Data Set

Contribution of Each Member

- **Saksham:** Paper Implementation, Deep Feature Selection, Parametric Analysis of models
- **Chirag:** Paper Implementation, Feature Selection through ML models, Dataset analysis
- **Shrey:** Skimming through code, Presentation and Literature Review
- **Jaideep:** Exploratory Data Analysis, Visualization, Data Preparation, Model Training