Phishing Website and Gene Expression Prediction Using Supervised Learning
HW1: Supervised Learning
Kunal Agarwal

Objective:

The objective here was to understand the working of different supervised learning algorithms by analyzing two different datasets. In particular five algorithms namely, Decision Trees, Neural Networks, Boosting, Support Vector Machines and k-Nearest Neighbors have been applied. Scikit learn library has been used for implementing all the above algorithms.

Datasets:

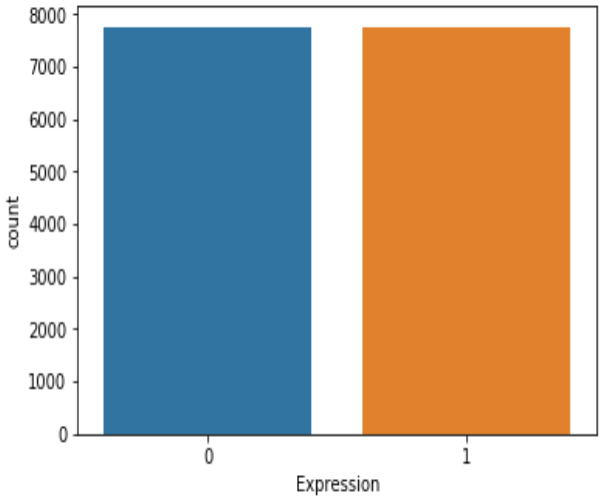
Both the datasets have been chosen from Kaggle. The first problem set deals with predicting the phishing website. Phishing websites are threat to online security. These websites have the layout of legitimate website and it is important for internet browser and firewall to detect it. Therefore, building machine learning models to predict phishing websites is of practical importance for cyber security. The phishing website dataset has 11055 instances and 30 attributes. The response variable has 2 classes indicating whether the website is phishing or legitimate.

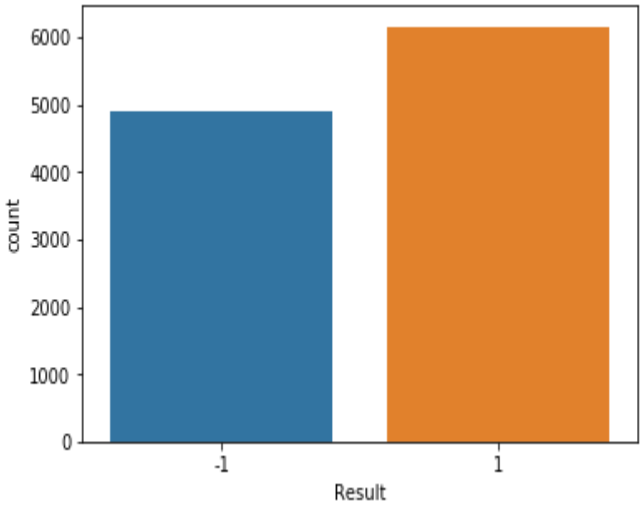
Other dataset deals with prediction of gene expression level based on histone modification signals. Gene expression is the process by which information from a gene is used in the synthesis of a functional gene product which can be a protein regulating cell function. Abnormal expression of a gene can be catastrophic for cells and therefore expression is regulated by different factors. Histone modifications are playing an important role in affecting gene regulation. Therefore, predicting gene expression from histone modification signals is a widely studied topic in scientific community. For collecting raw data, the 10,000 base-pair DNA region (+/-5000bp) around the transcription start site (TSS) of each gene was divided into bins of length 100 bp and then reads of 100bp were counted in each bin. Therefore, signal for each gene has shape of 100 x 5 and there are total 15485 genes. The raw gene expression data from Kaggle had the dimensions 1548500 x 7 with columns representing five histone modification marks, geneID and response/expression variable. The response variable has 2 classes: high expression (1) and low expression (0).

These datasets were chosen to compare the effect of high feature space on models. Gene expression dataset has highly dimensional data (515) features whereas phishing dataset has just 30 features.

<u>Distribution of response variable in both datasets:</u>

Left plot is for expression data while right plot is for phishing website data. In gene expression dataset, 49.90% instances have been classified as low expression (0) whereas 50.10% as high expression (1).
In phishing dataset, 44.30% instances have been classified as phishing (-1) whereas 55.70% (1) as legitimate websites.





Methodology:

There was no requirement of preprocessing for phishing dataset but gene expression dataset had to be reshaped so that each GeneID becomes one wide row with 500 columns/features. Each bin was made as individual feature for every histone modification mark corresponding to individual GeneID. 15 other derived features like sum, mean and median for 100 bins for each Histone Modification Mark for each GeneID were added as the additional columns. The final dimensionality after preprocessing was 15485 rows x 517 columns including GeneId and response variable.

Data before transformation (1548500 X 6)

GeneId	H3K4me3	H3K4me1	H3K36me3	H3K9me3	H3K27me3
1	2	1	4	1	0
1	0	2	1	1	1
1	0	0	4	1	1
1	0	2	2	0	1
1	2	0	0	0	0
1	1	2	0	0	1
1	2	2	2	0	1
1	1	1	4	2	2
1	2	3	4	3	1
1	0	2	3	2	2
1	2	0	3	4	2
1	0	0	4	1	2
1	2	0	5	3	2
1	1	0	3	9	2

Data after transformation and feature eng. (15485 X 517)

GeneId	Prediction	H3K4me3_1	H3K4me1_1	H3K36me3_1	H3K9me3_1	H3K27me3_1	H3K4me3_2	H3K4me1_2
1	0	2	1	4	1	0	0	2
2	0	1	0	1	0	0	0	0
3	1	1	6	3	1	1	1	6
4	1	0	4	3	2	1	0	2
5	1	0	1	2	0	0	0	0

After preprocessing each dataset was split into 70% training and 30% testing set. The hyperparameters were tuned using grid search on the training data and stratified 10-fold cross validation accuracy was used as the metric to choose best classifier (for the range of hyperparameters tested). We finally evaluate the testing accuracy of the best tuned classifiers for each algorithm on the testing set and plot learning curves for each algorithms. Learning curves have been plotted using the best parameters values obtained for the classifier using grid search.

Decision Tree

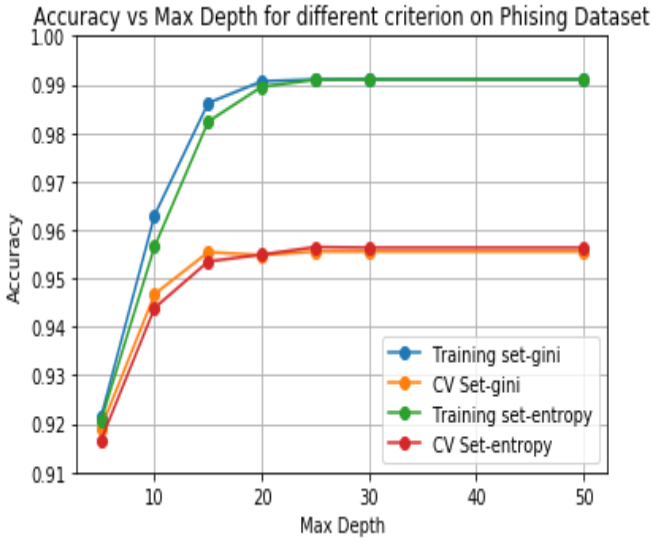
DecisionTreeClassifier module from scikit-learn library was used to analyze decision trees.
The following parameters were tuned using grid search module :

- Max_depth: Decision Trees tend to overfit if they are not pruned. This parameter helps us to pre-prune the tree by controlling the size of the decision tree created and it is known that as the depth of the tree increases there is more chance of overfitting as we will fit the model more according to training data.
- Criterion: This parameter controls how the tree is splitted. I used 'gini' and 'entropy' values.

Phishing Dataset

The Accuracy vs max_depth curve below shows the variation in training and cross validation accuracy because of max_depth and criterion parameters. We can see that the training accuracy as well as cross validation accuracy increases till max_depth = 25 and then remains constant for both criterion. Also, it can be seen that criterion parameter doesn't have significant effect on both accuracies. Therefore, max_depth =25 and criterion = 'entropy' (CV accuracy at max_depth= 25 was very slightly higher for entropy) were selected as best classifier parameters based on highest CV accuracy. Learning curve for the decision tree with best parameters is shown below. As expected as the training examples increase, training accuracy decreases whereas CV accuracy decreases and converges at the maximum training size. Since the curves are converging at maximum training size, more training examples will not help significantly in increasing CV accuracy. The training accuracy and testing accuracy is very high which indicates our algorithm fits the training data well and thus low bias.

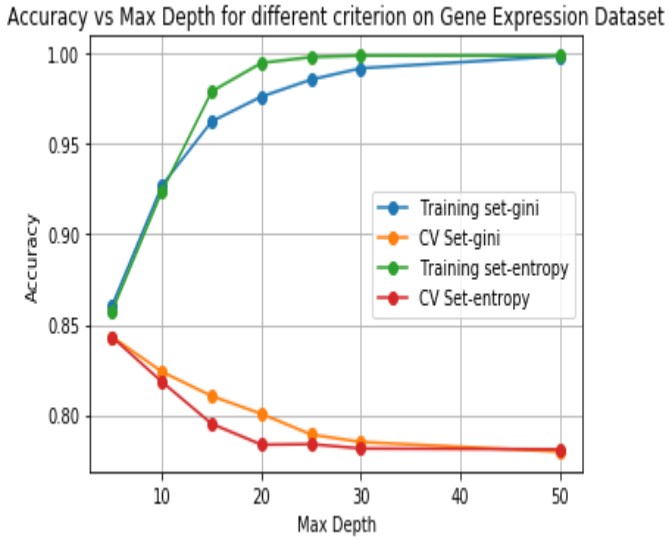
Testing Accuracy: The testing accuracy score for selected classifier on testing data was found to be 0.9611.



Gene Expression

For gene expression data, we can see from the Accuracy vs max_depth curve below that as the max depth increases, the training accuracy increases whereas CV accuracy decreases indicating overfitting of the data for high values of depth. Since, the highest CV accuracy is achieved at max_depth = 5 and for criterion = 'gini', these parameters are selected for the final decision tree. This may indicate that there are only few features in the dataset that are significant predictors and thus simple tree. The high

decreasing training accuracy and converging CV accuracy after 3000 examples in learning curve indicates that this decision tree classifier doesn't require large instances to fit the model accurately.



Testing Accuracy: The testing accuracy score for selected classifier on testing data was found to be 0.8275.

Gradient Boosting

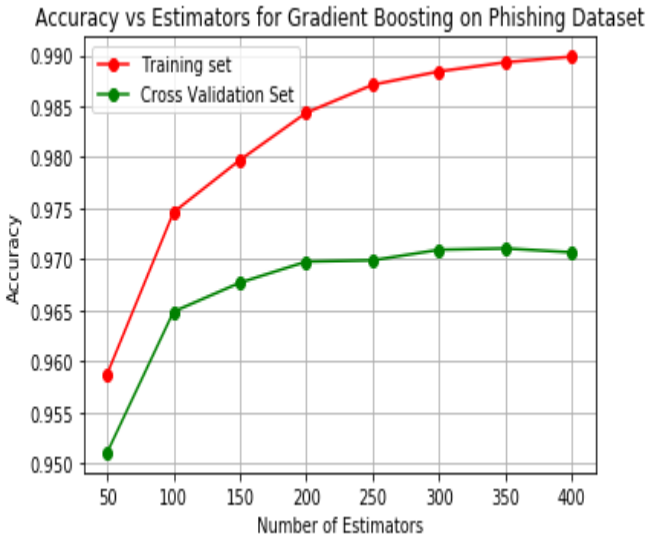
GradientBoostingClassifier from the sklearn library was used to analyze boosting algorithm. Typically decision tree is a weak learner and gradient boosting can significantly boost the decision tree performance. Boosting iteration number (n_estimators) have been tuned using grid search. Here also pre-pruning technique which controls the size of tree to combat overfitting has been used by setting the max_depth parameter = 5 on both datasets.

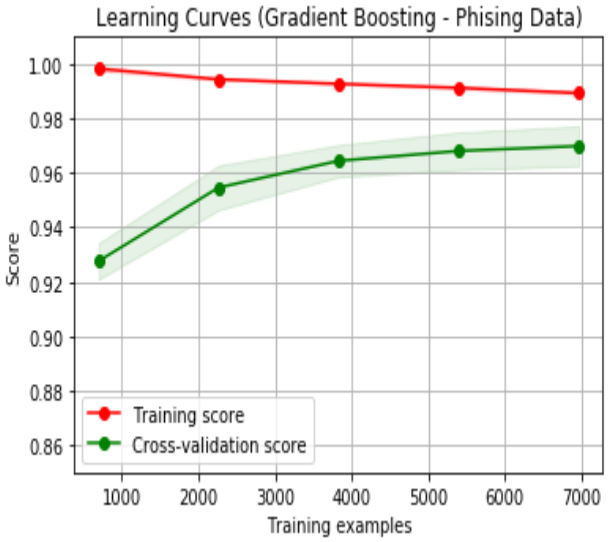
Phishing Dataset

As can be seen in the graph of Accuracy vs number of estimators below, both CV accuracy and training accuracy increased till 350 iterations. At 400 iteration, training accuracy is still increasing but CV accuracy is decreasing indicating overfitting. Therefore, n_estimators = 350 has been selected as optimal value. Overfitting will happen above 350 iterations.

As the CV accuracy curve in the learning curve below hasn't converged at maximum training example, it shows that more training examples could have helped the Gradient Boosting classifier. The smaller gap between training accuracy and CV accuracy curve at max training example indicates low variance.

Testing Accuracy: The testing accuracy score for selected GB classifier on testing data was found to be 0.9710. As expected, GB classifier has boosted the performance compared to single decision tree.

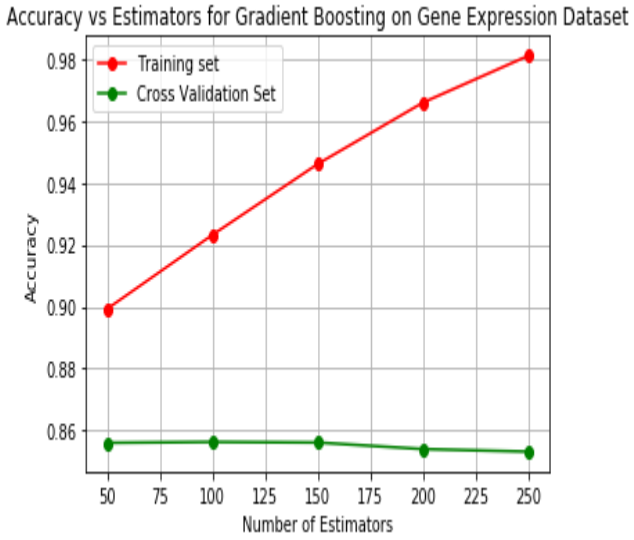


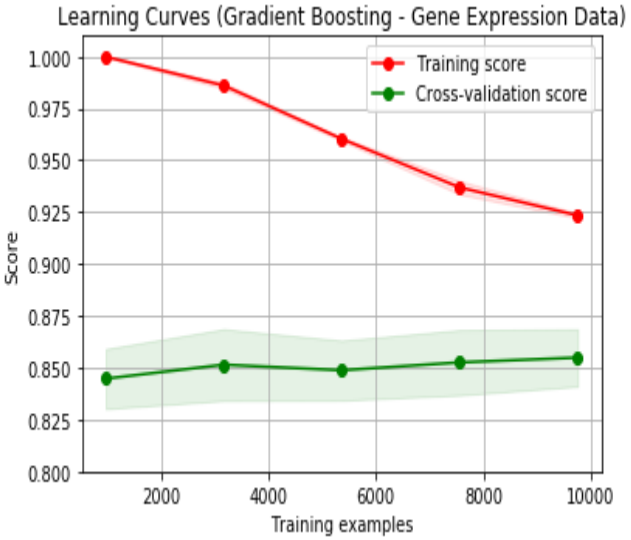


Gene Expression

As can be seen in the graph of Accuracy vs number of estimators, CV accuracy increases slightly at 100 iteration but decreases continuously as the iterations are further increases. Training accuracy on the other hand is increasing steeply indicating the data is being overfitted at high n_estimators (iterations) value. Therefore, n_estimators = 100 has been selected as optimal value based on highest CV accuracy.

Learning curve with the optimal value of n_estimators=100, shows there is not significant change in CV accuracy as the training examples increase indicating same performance can be obtained using less data as well. Indicates high variance issue as the gap between both curves is large. Reducing the complexity by decreasing number of features may help.





Testing Accuracy: The testing accuracy score for selected GB classifier on testing data was found to be 0.8495. As expected, GB classifier has boosted the performance compared to single decision tree.

Support Vector Machine (SVM)

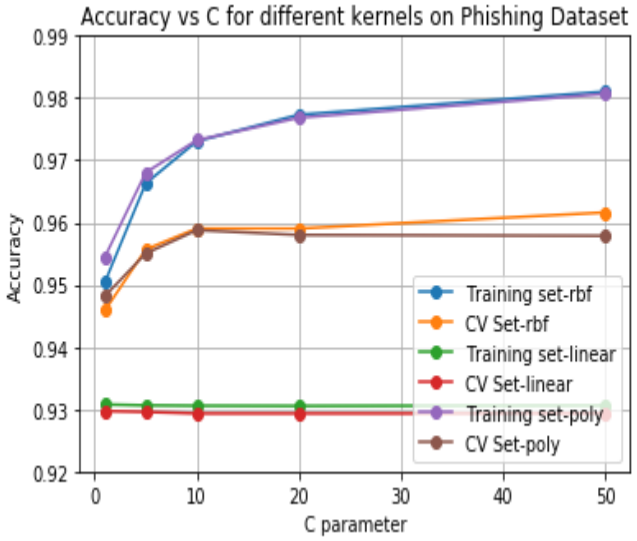
SVMs are based on the idea of finding a hyperplane that best divides a dataset into two classes.It tries to optimize the distance between two classes. SVC classifier from the sklearn library has been used to test the Support Vector Machine algorithm.The following parameters were tuned using grid search module:

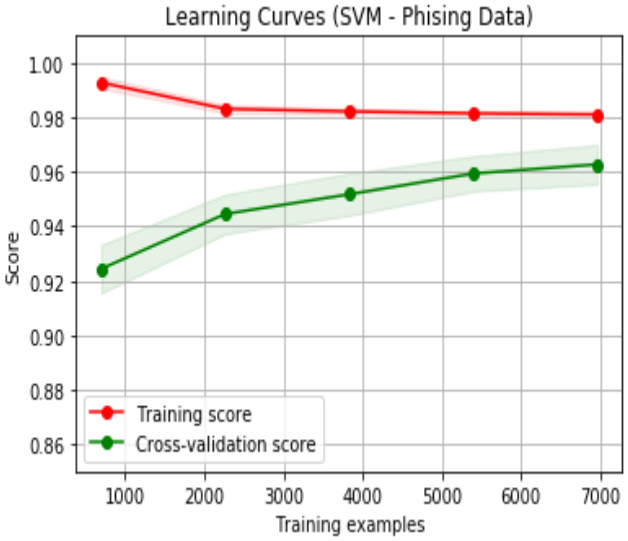
- Kernel: The shape of hyperplane is defined by kernel parameter. In this experiment, different kernels have been tried: linear, rbf and polynomial with degree 3 for both datasets.
- C: The C parameter or penalty parameter tells the SVM optimization how much we want to avoid misclassifying each training example. It has been tested on Phishing dataset only. Grid search on gene expression was taking a lot of time on expression data. C=1 was selected for expression data.

Scaling data : SVM in general was running very slowly for Gene Expression dataset because of high feature space. So gene expression data has been scaled (using StandardScaler in preprocessing module) in order to speed up the run time.

Phishing Dataset

As can been seen from the Accuracy vs C curve, RBF kernel at C= 50 gives the best cross-validation accuracy and has been chosen as the parameters for final SVM classifier. Since, RBF kernel projects data into multiple higher dimensions and is performing better than linear kernel, it can be inferred that data is not linearly separable. Learning curve suggests that additional training data can help in boosting the performance as the CV accuracy is continuously increasing and has not converged at maximum training example.

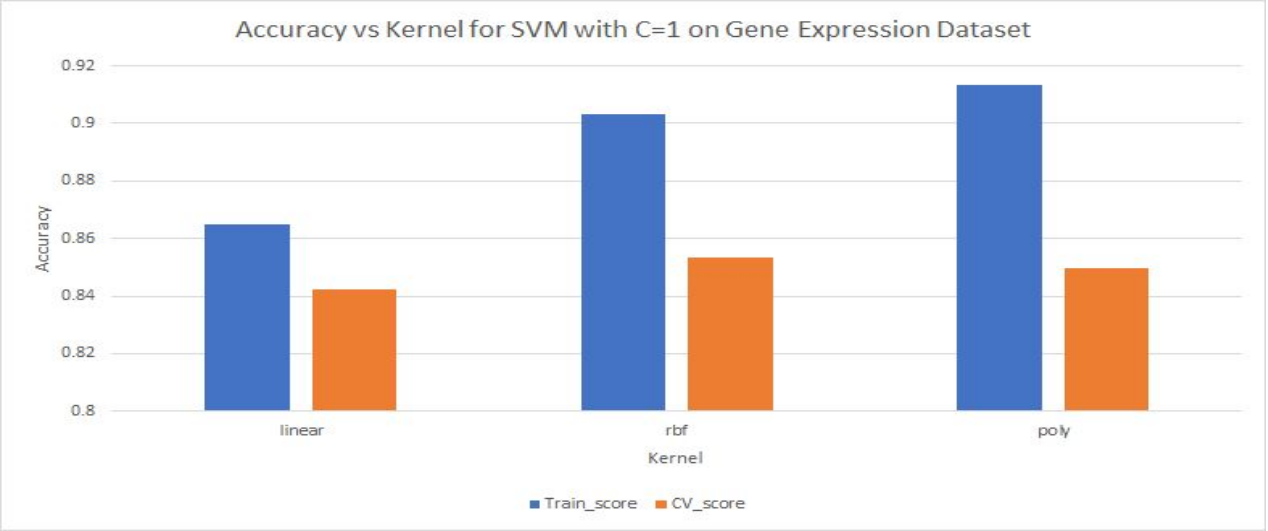




Testing Accuracy: The testing accuracy score for SVM classifier with best parameters on testing data was found to be 0.9620.

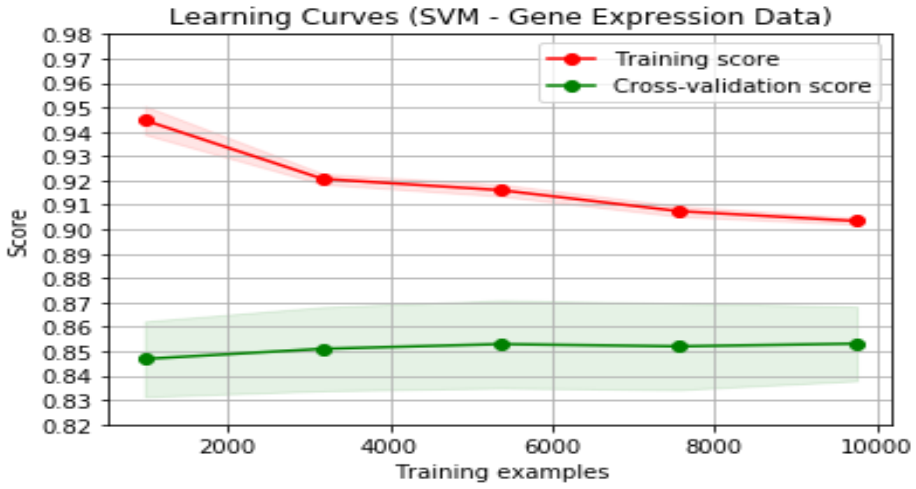
Gene Expression Dataset

As can been seen from the Accuracy vs Kernel curve below, RBF kernel gives the best cross-validation accuracy and therefore has been chosen as the parameters for final SVM classifier on gene expression dataset. Again since RBF performing better than linear kernel, it can be inferred that data is not linearly separable.



Learning curve below suggests high variance issue as the gap between both curves is large. Also, additional data isn't likely to solve the issue. Reducing the complexity by decreasing number of features may help in lowering variance.

Testing Accuracy: The testing accuracy score for SVM classifier with best parameters on testing data was found to be 0.8476.



K-Nearest Neighbors (KNN)

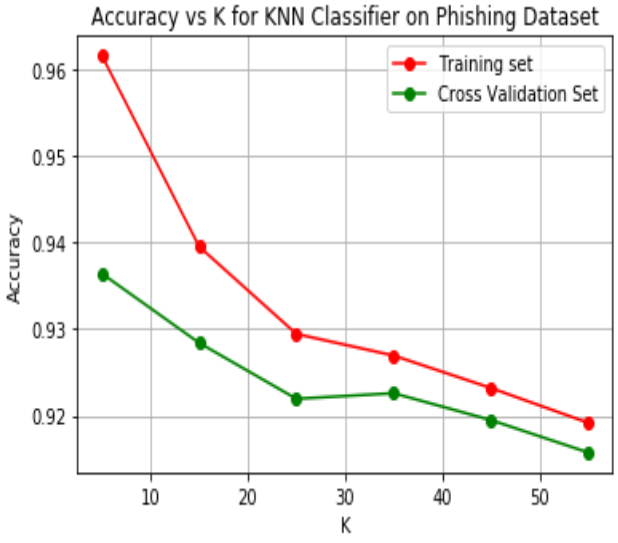
KNN is a non-parametric, "simple" algorithms..The KNeighborsClassifier from the sklearn library has been used to test the kNN algorithm.

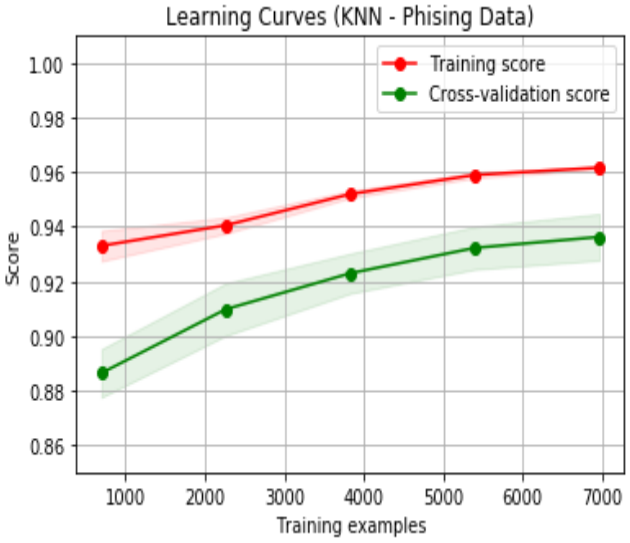
- k-value: The parameter we is used to modify the number of neighbours while training the model (k value). For phishing dataset, k-values ranging from 5 to 65 at a 10 step interval have been used. For gene expression dataset, k-values ranging from 5 to 45 at a 10 step interval and then ranging from 50 to 250 at a 50 step interval have been used.

Phishing Dataset

It can be observed in Accuracy vs K curve that the training as well as CV accuracy score decreases on increasing k which can be possibly due to increasing the neighbours, the impurity of selection can increase and we are selecting neighbours which don't actually help in predicting thereby reducing the accuracy.

In the learning curve below, unexpectedly both the training accuracy and CV accuracy is increasing continuously without convergence as number of training examples increase. Usually the training accuracy decrease with increasing training examples. It may be because of some potential noise in the data.
Testing Accuracy: The testing accuracy score for KNN classifier with best parameters on testing data was found to be 0.9372.

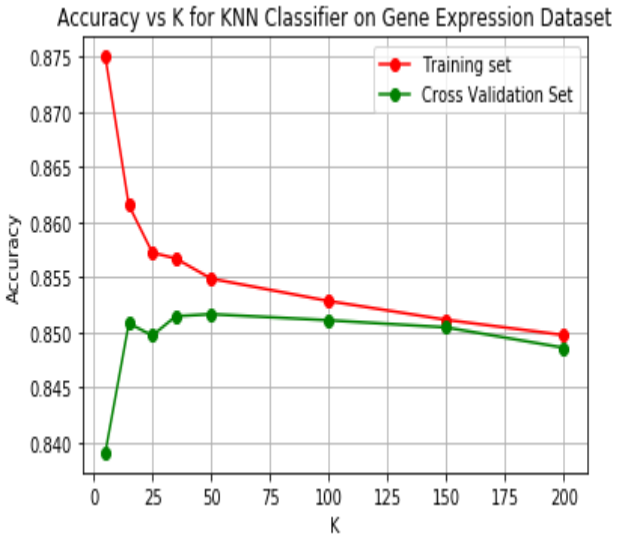


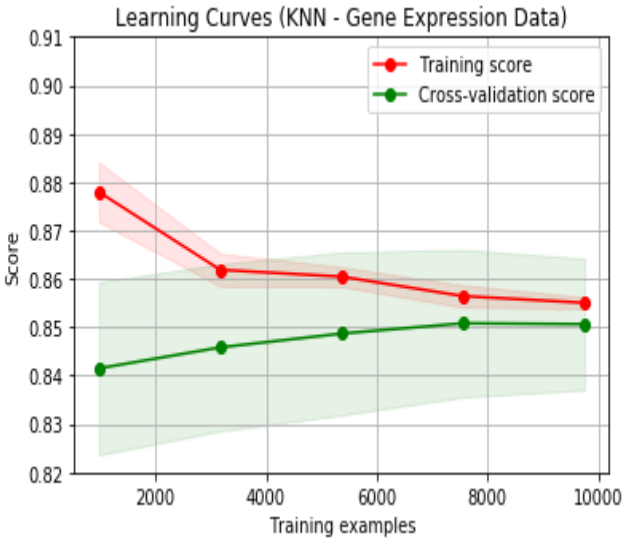


Gene Expression Dataset

It can be observed in Accuracy vs K curve for expression dataset that, CV accuracy increases significantly on increasing k-value from 5 to 15, slightly decreases at K=25 and increases again achieving maximum at K=50 and then decreases for high values of K. It implies that there are around

35-50 significant features in the dataset that are helping in prediction. Therefore, based on highest CV accuracy K=50 has been chosen as the best parameter value.





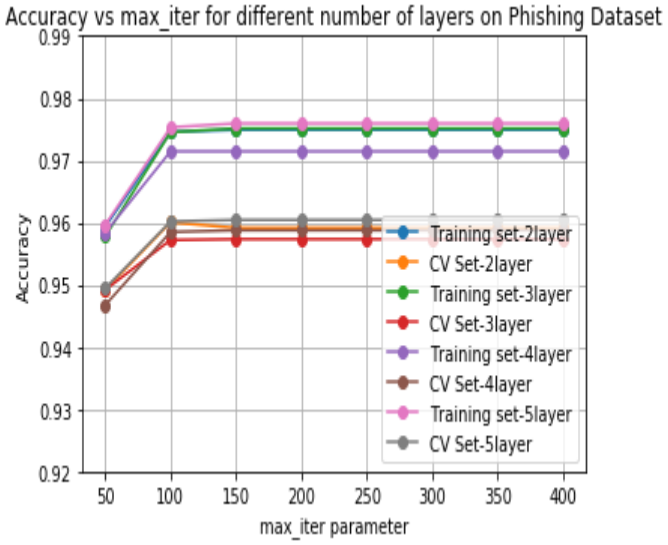
Learning curve for KNN with 50 neighbors is tending towards convergence and some additional training examples can surely help. Also, the gap between, training and CV curve is low indicating low variance. It is in contrast compared to other algorithms seen above. It is because we have reduced the dimensionality by selecting 50 neighbours whereas other algorithms have been built on whole 515 feature space. This confirms the problem of dimensionality in other algorithms.
Testing Accuracy: The testing accuracy score for KNN classifier with best parameters on testing data was found to be 0.8484.

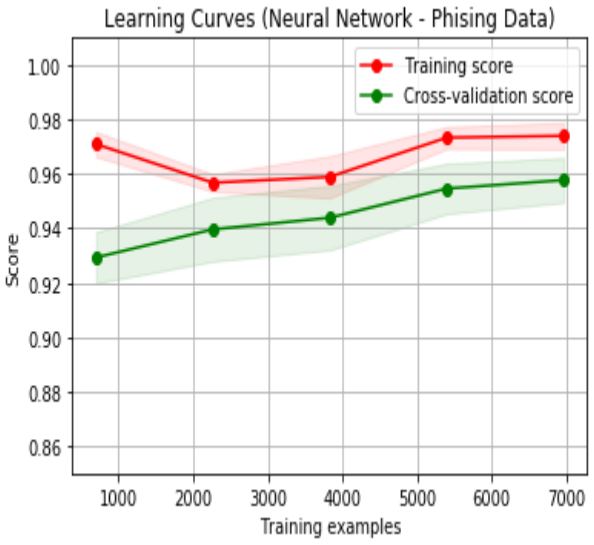
Neural Network

Multi-layer Perceptron classifier was used from the scikit-learn library to implement neural network. Normally neural networks need huge data to learn and provide a high performance. Both datasets have more than 10,000 instances which should provide good performance. I experimented with different parameters in both datasets which will be highlighted below.

Phishing Dataset

Neural network with activation function as ‘tanh’ and neurons=100, and default values for other parameters have been taken as base neural network before hyperparameter tuning the layer size and max_iter parameter. As can be seen in the Accuracy vs max_iter curve, training as well as CV accuracy increases from 50 to 150 iterations for all layer sizes and remains constant after that. This indicates neural network has reached its minima point at 150 iterations and therefore increasing the iterations has no effect on accuracy. Also, we can see that training as well as CV accuracy is highest for layer size of 5. Layer sizes above 5 were also (not shown here) tried but it yielded no or negligible performance enhancement. Therefore, finally max_iter=150 and layer_size=5 were chosen as best parameters.





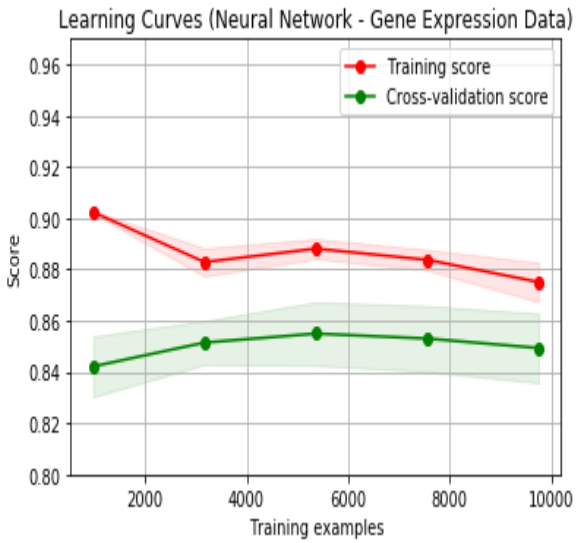
In the learning curve for Neural Network on Phishing dataset, we see some unexpected behaviour with training score curve as it decreases at first and then goes on increasing. Usually the training accuracy decrease with increasing training examples. It may be because of some potential noise in the data. Cross validation accuracy is continuously increasing therefore no overitting.

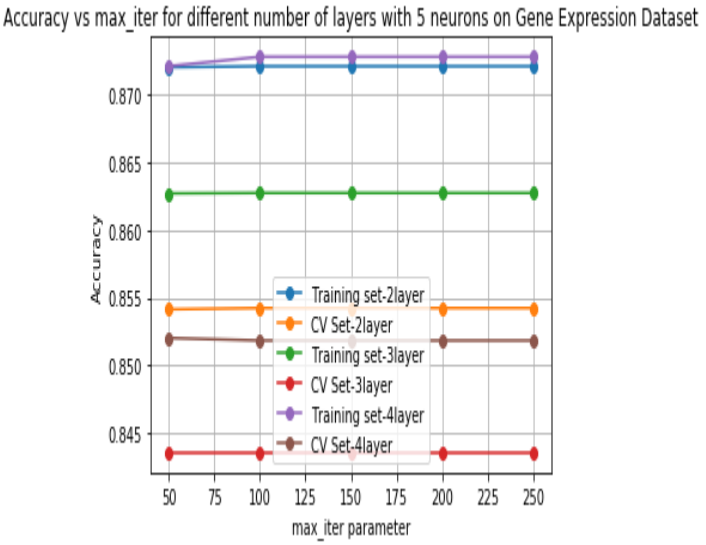
Testing Accuracy: The testing accuracy score for NN classifier (max_iter=150 and layer_size=5, neurons =100) with best parameters on testing data was found to be 0.9656.

Gene Expression Dataset

Neural network with activation function as 'tanh' was used as base neural network before tuning other parameters like number of neurons, max_iter, learning_rate_init and layer size values.

- Number of neurons: Different values like 5,20,50 and 100 were used but 5 provided the best CV accuracy.
- Learning_rate: Tried to increase the learning rate from default value of 0.001 but it lead to decrease in CV accuracy. So, default value was selected as optimal.
- Max_iter: The values were varied from 50 to 250 with step of 50. As can be seen in the Accuracy vs max_iter graph below, there is not significant change in accuracy on increasing iterations indicating neural network is converging very quickly.
- Layer Size: From the graph below can be seen that, 2 layer size model with 5 neurons gives the best CV accuracy indicating increasing the complexity isn't increasing the performance.





Learning curve for Neural network on expression data is showing some unexpected behaviour which maybe because of the data being very noisy (high dimensional feature space) in data set. Again reducing the complexity by reducing the number of features will be a better prospect.

Testing Accuracy: The testing accuracy score for NN classifier (max_iter=150 and layer_size=5, neurons =100) with best parameters on testing data was found to be 0.8465.

Conclusion

In this section, the results of each algorithm have been consolidated for both the datasets and we try to analyze which one performs the best. The results have been tabulated below:

Model	Hyper Parameters	Train Accuracy	10-fold CV Accuracy	Test Accuracy	Train Time
Phishing Dataset					
DT	Depth=25, Criterion=entropy	0.9909	0.9564	0.9611	0.0174
Boosting	max_depth: 5, n_estimators:350	0.9893	0.9710	0.9710	6.0410
SVM	C: 50, kernel: rbf	0.9810	0.9616	0.9620	1.2847
KNN	k=5	0.9615	0.9364	0.9372	0.0764
NN	hidden_layer_sizes: (100, 5), max_iter: 150	0.9759	0.9604	0.9656	3.8286

Model	Hyper Parameters	Train Accuracy	10-fold CV Accuracy	Test Accuracy	Train Time
Gene Expression Dataset					
DT	Depth=5, Criterion=gini	0.8610	0.8431	0.8275	0.7098
Boosting	max_depth: 5, n_estimators:100	0.9232	0.8562	0.8495	43.33
SVM	C: 1, kernel: rbf	0.9032	0.8535	0.8476	59.98
KNN	k=50	0.8548	0.8516	0.8484	1.1640
NN	hidden_layer_sizes: (5, 2), max_iter: 100	0.8721	0.8542	0.8465	19.3374

Observations

- It can be observed that Gradient Boosting is the best classifier for both the datasets based on 10-fold accuracy and Testing Accuracy. Though, for gene expression dataset KNN gives almost same accuracy with less fit time and simpler model. Therefore, for gene expression dataset it should be chosen as best classifier. KNN is supposed to performs well with noisy data which is the case with expression data due to high dimensional feature space.
- Phishing website dataset has achieved good performance on all the classifiers and there is no significant difference in accuracy except with KNN.
- Gene Expression data has achieved similar testing accuracy for all the classifiers except Decision Tree (a bit lower than others).
- Gene Expression dataset tends to seek for simple classifier with less features and therefore dimensionality reduction or feature selection can enhance the performance of the classifiers.
- SVM has very high learning time for high dimensional data and expression data should be scaled at the pre-processing step.

References

1. Gene Expression Dataset: https://www.kaggle.com/c/gene-expression-prediction
2. Phishing Dataset: https://www.kaggle.com/akashkr/phishing-website-dataset
3. Scikit-Learn Library: Scikit-learn: Machine Learning in Python, Pedregosa et al., JMLR 12, pp. 2825-2830, 2011.