Data Mining Project Report

URL Classification

Gaurav Tiwari and Snigdha Dubey

MS in Applied Data Analytics

Supervisor: Dr Jae Lee Young

Metropolitan College
Boston University

Declaration

This is to declare that the following report titled “URL Classification”, is uniquely prepared for the Data Mining project. The report is prepared for academic requirementsand not for any other purposes.

Task performed by Gaurav:

- Data Preprocessing
- Handling missing values
- Performance metrics Calculation
- Fit and run classification Algorithms
- Conclusion

Tasks performed by Snigdha:

- Applying Feature Selection Methods
- Confusion Matrix
- Fit and run classification Algorithms
- Performance metrics Calculation
- Conclusion

Submitted by,

Gaurav Tiwari
gtiwari@bu.edu

Snigdha Dubey
sndubey@bu.edu

Contents

Goal

The goal of this project is to classify if a URL is Legitimate or phishing, based on the features of a URL that are rule based.

Dataset

The dataset used was acquired from **https://datacite.org/** for this project. There are a total of 49 attributes including 1 Labelling attribute "CLASS_LABEL" and 10000 tuples.

The description of the attributes is described below in detail:

Attributes	**Description**
NumDots	Number of Dots present in URL
SubdomainLevel	Number of levels in a Subdomain
PathLevel	Number of levels in path
UrlLength	Length of URL
NumDash	Number of dashes in URL
NumDashInHostname	Number of dashes in URL Hostname
AtSymbol	Presence of @ symbol
TildeSymbol	Presence of ~ symbol
NumUnderscore	Number of _ present in an URL
NumPercent	Number of % present in an URL
NumQueryComponents	Total number of components in a query
NumAmpersand	Number of & present in an URL
NumHash	Number of # present in an URL
NumNumericChars	Number of numeric characters in Url
NoHttps	HTTPS used?
RandomString	Random Strings used?
IpAddress	IP Address used in Url?
DomainInSubdomains	Whether Domain is used in subdomains
DomainInPaths	Whether Domain is used in paths
HttpsInHostname	HTTPS used in Hostname
HostnameLength	Length of Hostname
PathLength	Length of Path
QueryLength	Length of Query
DoubleSlashInPath	DOUBLE Slash used in Path
NumSensitiveWords	NUM of sensitive Words
EmbeddedBrandName	brand Name embedded in URL
PctExtHyperlinks	URL encoded Hyperlinks
PctExtResourceUrls	URL encoded resource URLs
ExtFavicon	Favicon symbol present ?
InsecureForms	Insecure Login Form
RelativeFormAction	Relative URL used
ExtFormAction	Extended action on form allowed
AbnormalFormAction	Abnormal action on form visible

PctNullSelfRedirectHyperlinks	Percentage encoded Redirect Hyperlinks
FrequentDomainNameMismatch	FREQUENT Domain used name Mismatch
FakeLinkInStatusBar	Presence of fake link in Status bar
RightClickDisabled	Right click Disabled
PopUpWindow	POP up Window opened?
SubmitInfoToEmail	SUBMIT Info used to Email
IframeOrFrame	IFRAME used or Frame
MissingTitle	Title Missing
ImagesOnlyInForm	IMAGES Only used in Form
SubdomainLevelRT	Subdomain Level Routing
UrlLengthRT	URL Length changed to service routing
PctExtResourceUrlsRT	Ext Resource URLs routing
AbnormalExtFormActionR	Abnormal action on form routing action
ExtMetaScriptLinkRT	Ext Meta Script routing link
PctExtNullSelfRedirectHyperlinksRT	Percentage encoded null self-Redirect Hyperlinks routing
CLASS_LABEL	Class label

***Class_LABEL is the class attribute**

Tools and Libraries

This project has been implemented using Python3 on Spyder IDE of Anaconda The project notebook contains all standard procedures used to perform classification, such as data preprocessing, exploratory data analysis. Missing Values are handled followed by the train test split, K fold cross validation label encoding and fitting models. Finally, after using different feature selection methods, performance evaluation is done by calculating metrics on all five classification models and the preprocessed dataset.

All the Libraries used in the project are mentioned below:

Library	**Description**
Pandas	Loading Dataset, creating Data frames
Numpy	Performing mathematical operations
Matplotlib	Visual representation of attributes
Seaborn	Visual representation of attributes
sklearn.metrics	Calculating performance metrics, such as roc_auc_score, matthews_corrcoef, roc_curve
sklearn.model_selection	Performing train-test split on the dataset

sklearn.preprocessing	Used to perform Standard Scaling
sklearn.feature_selection /sklearn.ensemble	For using feature selection methods, such as Select K Best, Select From Model, Sequential Feature Selector, RFE, Variance Threshold and ensemble for Adaboost Classifier and RandomForestClassifier
Sklearn.linear_model	Logistic Regression and Linear Regression classification model
tabulate	Performance metrics as table format
Sklearn. naive_bayes / sklearn. neighbors	For Naïve Bayesian and KNN classifier

Classification Algorithms

Classification algorithms categorize the target attribute into a class or category. They are of three types: binary, multiclass and multilabel. In this project, binary classification is performed to find the status of a URL.

The algorithms used for classification are briefly described below:

Algorithm	**Brief Description**
Logistic Regression	The Classifier finds a best-fitting relationship between the dependent variable and a set of independent variables. The best-fitting line in thisalgorithm looks like S-shape.
Random Forest	It fits a number of decision tree classifiers on various samples of the datasetand uses averaging to improve the accuracy
K Nearest Neighbour	This algorithm relies on distance for classification. It calculates the distance of the sample with its nearest k units. The function is only approximated locally and all computation is deferred until function evaluation.
Naive Bayesian	Naive Bayes classifiers are probabilistic classifier based on applying Bayes' theorem with strong (naive) independence assumptions between the features
Adaboost	AdaBoost is an ensemble learning method increases the efficiency of binary classifiers. AdaBoost uses an iterative approach to learn from the mistakes of weak classifiers, and turn them into strong ones.

Attribute Selection Methods

Attribute Selection Method is a technique used for data reduction in the data mining process. Many times, the data may have a large number of attributes and not all of those attributes are relevant. The goal is to reduce the irrelevant attributes such that it does not affect the cost of data analysis.

The selection methods used in the project are briefly described below:

Selection Method	**Brief Description**
Select K Best	This method selects the features according to the k highest score. For classification, the default 'f_classif' method is used as a scoring function.
Variance Threshold	This method eliminates features with very low variance, features withlow variance do not provide much useful information.
Recursive feature selection	select features by recursively considering smaller and smaller sets of features by assigning them weights. Estimator used with this is Logistic regression.
Selectfrommodel	Meta-transformer for selecting features based on importance weights. This is used with estimator as Logistic Regression to assign weights to features.
Sequential feature selection	Sequential Feature Selector adds (forward selection) or removes (backward selection) features to form a feature subset in a greedy fashion. It uses Linear Regression as Estimator

Selected Attributes

1. Select K best

   10 attributes were selected:

   - NumDots
   - NumDash
   - NumSensitiveWords
   - PctExtHyperlinks
   - InsecureForms
   - PctNullSelfRedirectHyperlinks
   - FrequentDomainNameMismatch
   - SubmitInfoToEmail
   - IframeOrFrame
   - PctExtNullSelfRedirectHyperlinksRT

2. Variance Threshold

   11 attributes were selected

   - NumDots
   - PathLevel
   - UrlLength
   - NumDash
   - NumUnderscore
   - NumQueryComponents
   - NumAmpersand
   - NumNumericChars
   - HostnameLength
   - PathLength
   - QueryLength

3. Recursive feature selection

10 attributes were selected

- NoHttps
- NumSensitiveWords
- PctExtHyperlinks
- PctExtResourceUrls
- ExtFavicon
- InsecureForms
- PctNullSelfRedirectHyperlinks
- FrequentDomainNameMismatch
- SubmitInfoToEmail
- PctExtNullSelfRedirectHyperlinksRT

4. Selectfrommodel

14 attributes were selected:

- SubdomainLevel
- NumDashInHostname
- NumPercent
- NumQueryComponents
- NoHttps
- NumSensitiveWords
- PctExtHyperlinks
- ExtFavicon
- InsecureForms
- FrequentDomainNameMismatch
- SubmitInfoToEmail
- IframeOrFrame
- MissingTitle
- PctExtNullSelfRedirectHyperlinksRT

5. Sequential Feature Selector

10 attributes were selected:

- NumDots

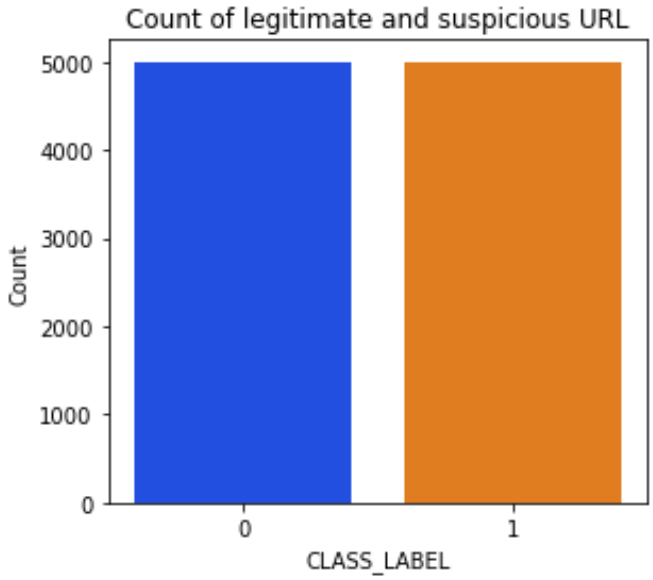
- SubdomainLevel
- HostnameLength
- PctExtHyperlinks
- PctExtResourceUrls
- InsecureForms
- FrequentDomainNameMismatch
- SubmitInfoToEmail
- UrlLengthRT
- PctExtNullSelfRedirectHyperlinksRT

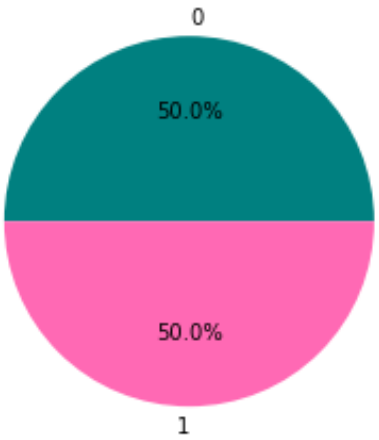
Course of Action

This section discusses the procedure and steps performed for this Classification project. The dataset is trained and tested on 5 different classification models, using 5 feature selection methods, giving us 25 different results for various performance metrics. These metrics help us determine which model performs best under different feature selection methods.

The dataset contains the 49 attributes. The target variable, 'CLASS_LABEL, shows that there is no class imbalance between the Phishing URL and Legitimate URL. The dataset contains label values as 0 and 1 and it means below:

1 means → legitimate
0 means → suspicious

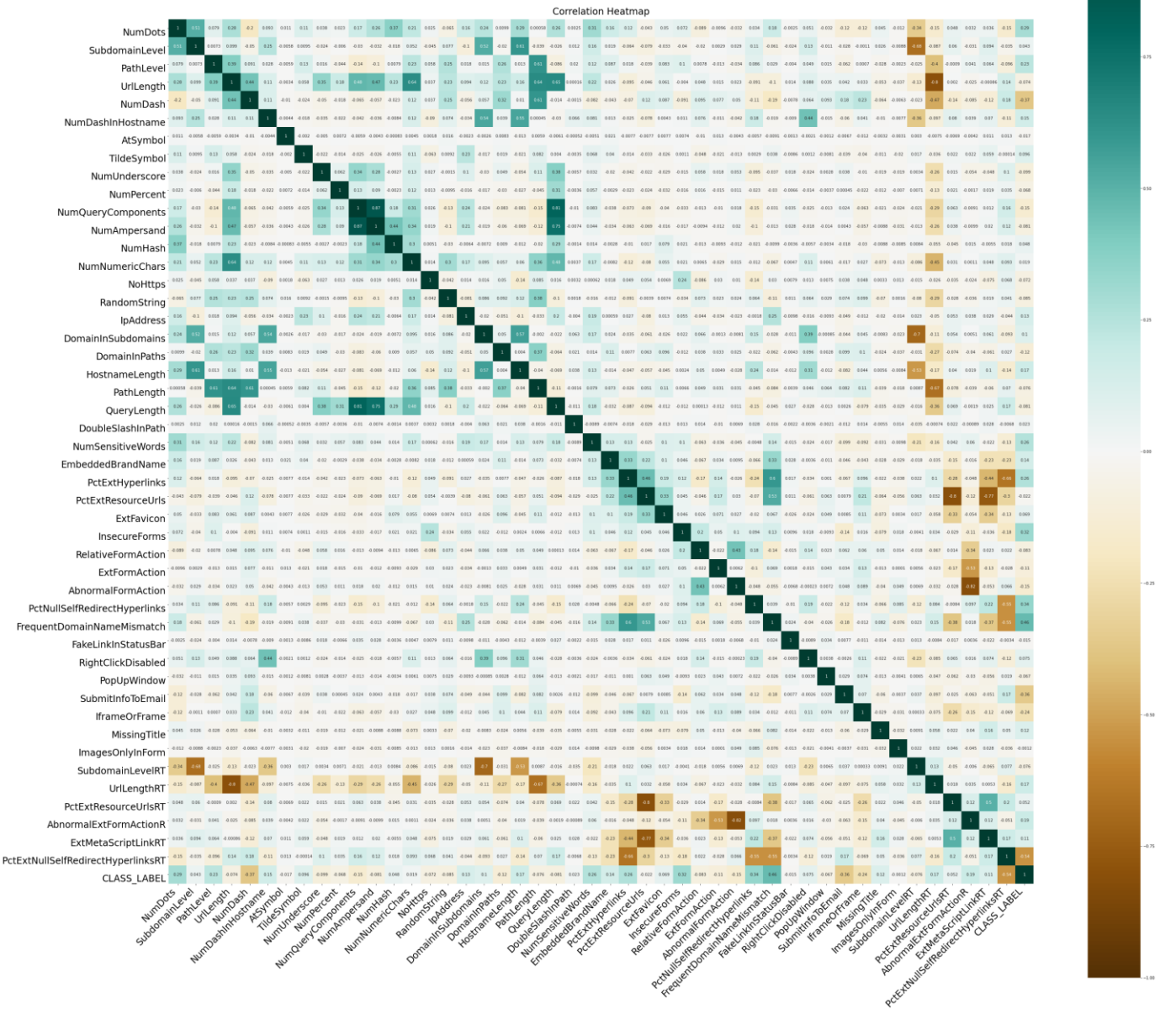




While performing data preprocessing, it was observed that the dataset does not contain similar

data spread across different columns and so there was no need to club any columns together or do preprocessing for the same. Also, it was found that the value in one of the columns “HttpsInHostname” was constant and was 0 in all the 10000 tuples. So, this column was dropped as part of preprocessing.
The Dataset was checked for Null values but no Null values were found in the dataset. The Preprocessed Dataset was also later saved and is attached with the report.

In the last step of pre-processing, correlation of the attributes with the target values is calculated. Since none of the attributes share a high correlation with the target value, we can not deduce any important information. After finding correlation between attributes, there are 2 pairs of attributes that share correlation greater than 80%.



Five popular and effective classification algorithms were chosen: Logistic Regression, Random Forests, KNN, Naïve Bayesian, Adaboost. All these algorithms were implemented and tested using 10-fold cross validation. Finally, the best performing algorithm (based on highest mean cross validation score) was selected as the best model so far.

Five popular and effective attribution selection methods were chosen, to reduce the dimensionality of the dataset and the complexity of the model.

The 5 datasets that were obtained after attribute selection are trained on the five chosen classification models. The function 'splitfn' fits and trains the training data and post that produces the predicted values by calling the respective functions for each classification. With the predicted values and y_test data, the performance metrics are calculated and are appended to a list.

With the list obtained, the measures help in choosing the best model, continued in the next section.

Steps Involved

- Preprocessing of the datasets (Checked for null values and whether the dataset is balanced)
- Selected five classification algorithms (Logistic Regression, Random Forest, KNN, Naives Bayesian, Adaboost)
- Stratified Sampled the dataset and into Test dataset (34%) and Train dataset (66%).
- Using the selected classification algorithms, trained on the training dataset and tested on the test dataset using K-fold Cross Validation with K as 10, saved the predicted results to be used later for comparison and computation of various metrics.
- Choose five attribution selection methods (Select K-best, Variance Threshold, Recursive Feature Selection, Select from Model with Logistic regression, Sequential Feature Selection) and prepared five reduced training datasets and test datasets selecting the attributes from these methods and saved them.
- Using each selected classification algorithms, trained on the reduced training dataset and tested on the reduced test dataset using K-fold Cross-Validation with K as 10, saved the predicted results to be used later for comparison and computation of various metrics.
- Computed various metrics (FPR, TPR, Precision, Recall, F1_Score, MCC, ROCAUCScore, Accuracy) for each of the predicted results from the 25 models on reduced datasets and 5 models on full datasets
- Compared them and selected one model as the best model out of the 25 models on reduced datasets.
- Compared the performance of the best model with the performance of the model that was built using the same classification algorithm from the dataset with all attributes (the dataset after preprocessing) and drew the conclusion.

Steps to replicate result

- Place the dataset in the working directory of python.
- Install Spyder or Anaconda or Jupyter Notebook.
- You must read the csv file in the Spyder or any IDE and store the result in any variable in this case as df.
- To replicate the result, you must then just call the splitfn function with the df as a parameter.

***Results tend to change in case of Random Forest as at the back-end trees classification criteria differs(though the changes are very small)**

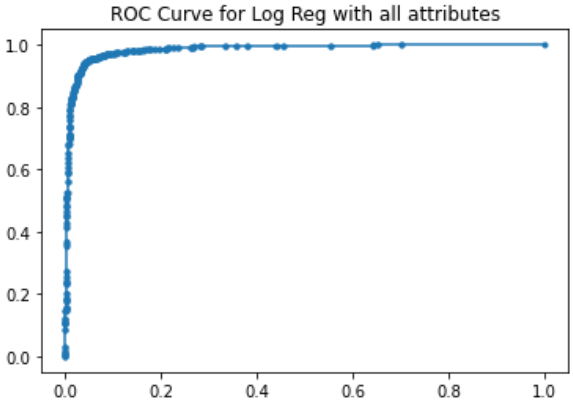
Results and Evaluation

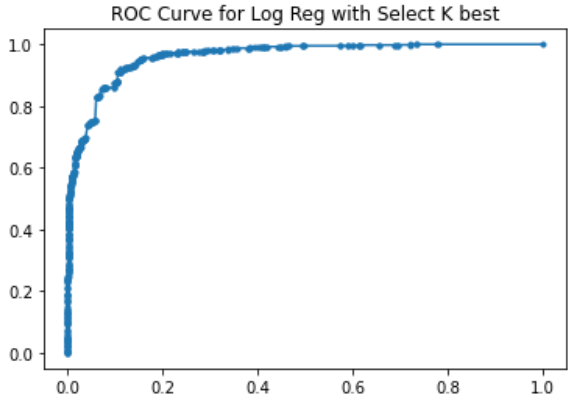
The performance metrics data was collected and showed as form of a table. The performance metrics include.

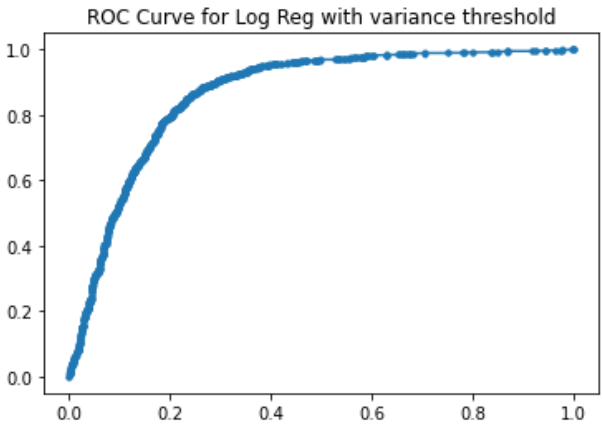
1. TP Rate & FP Rate
2. Precision & Recall
3. F1 score
4. MCC
5. ROC Area
6. Accuracy

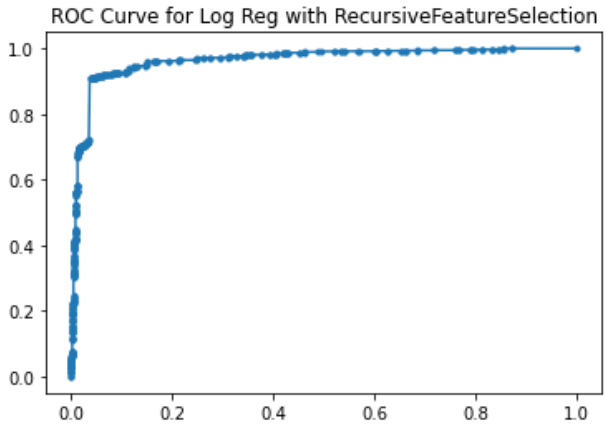
We also computed ROC curves for all 30 models to have a visualisation of performance of the models at different classification thresholds. The computed ROC curves are as below:

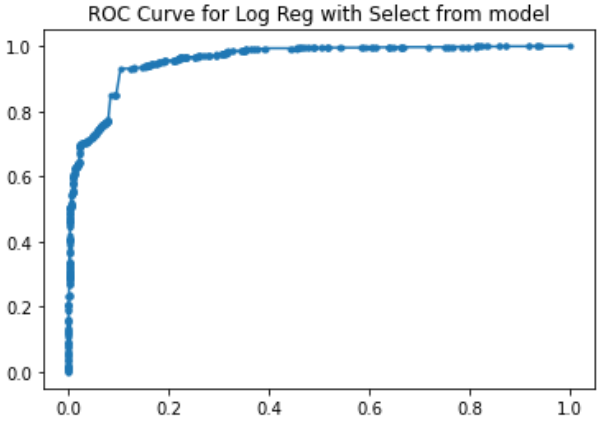
For Logistic Regression:

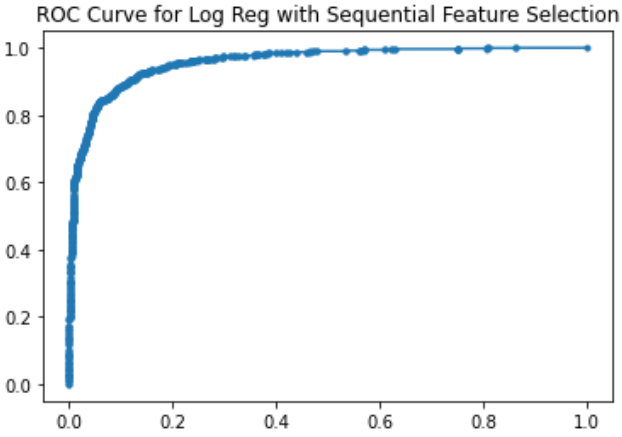




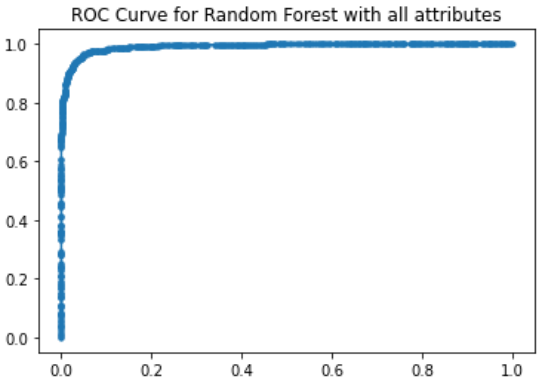


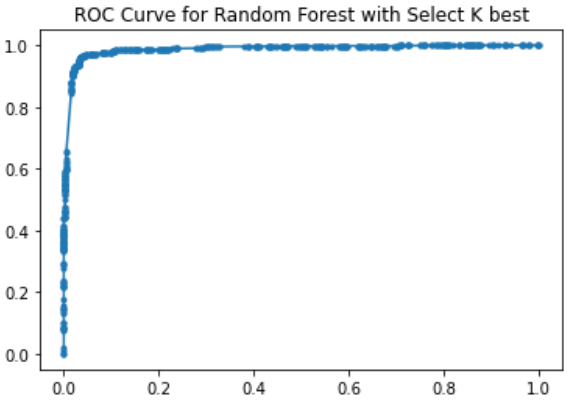


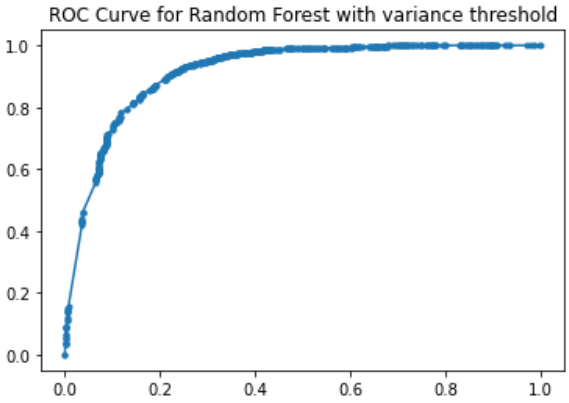


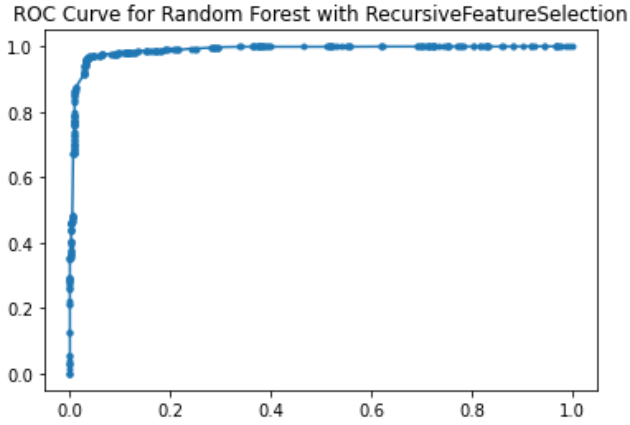


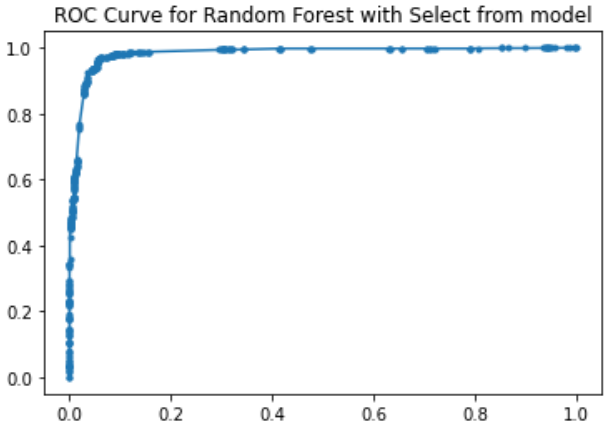
For Random Forest:

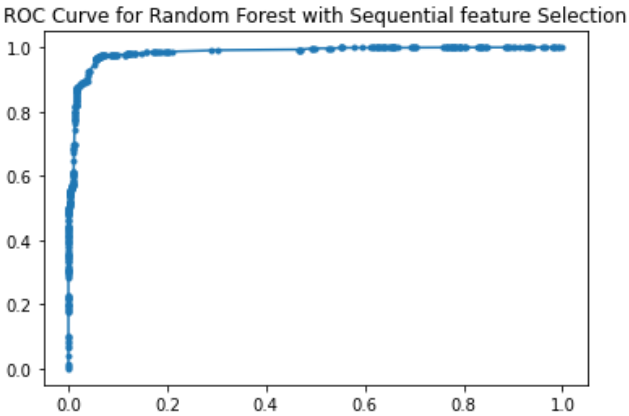




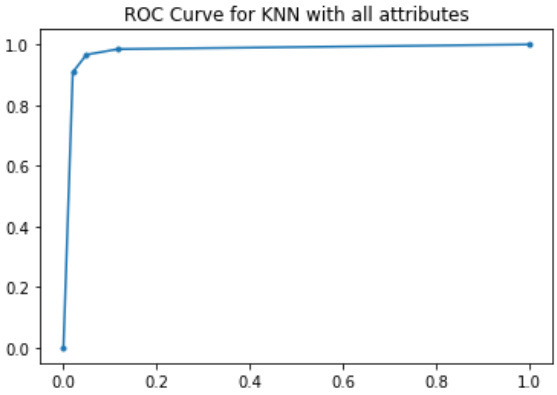

ROC Curve for Random Forest with variance threshold
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0

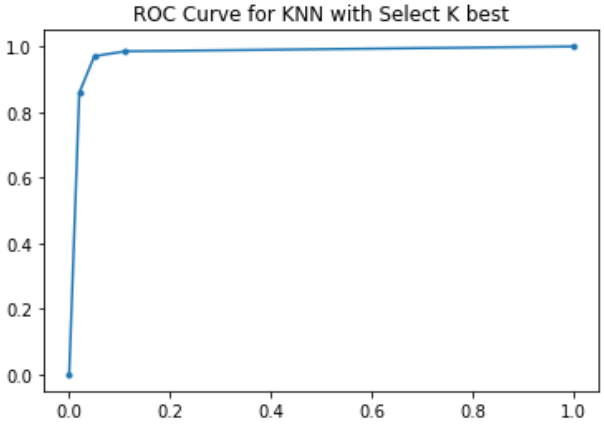

ROC Curve for Random Forest with RecursiveFeatureSelection
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0

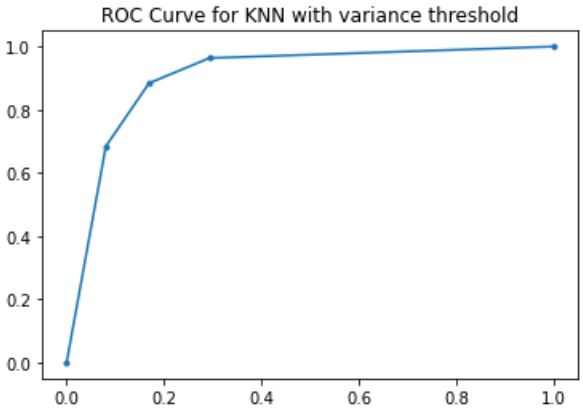

ROC Curve for Random Forest with Select from model
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0

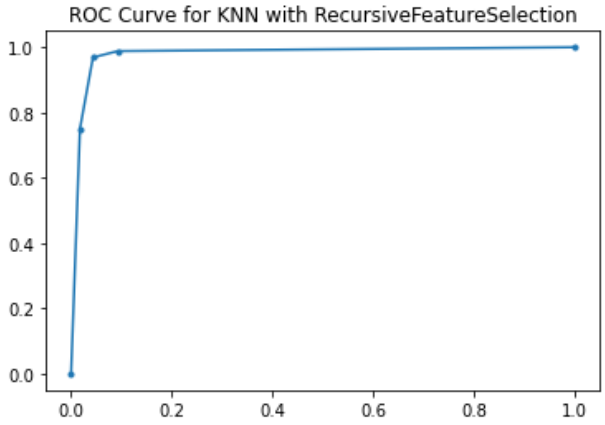

ROC Curve for Random Forest with Sequential feature Selection
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0

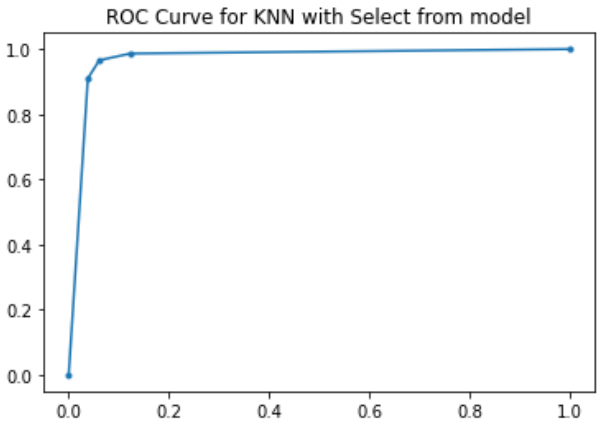
For KNN:

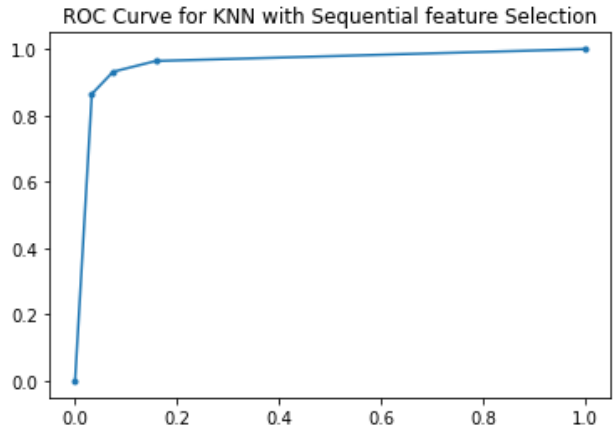

ROC Curve for KNN with all attributes
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0


ROC Curve for KNN with Select K best
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0

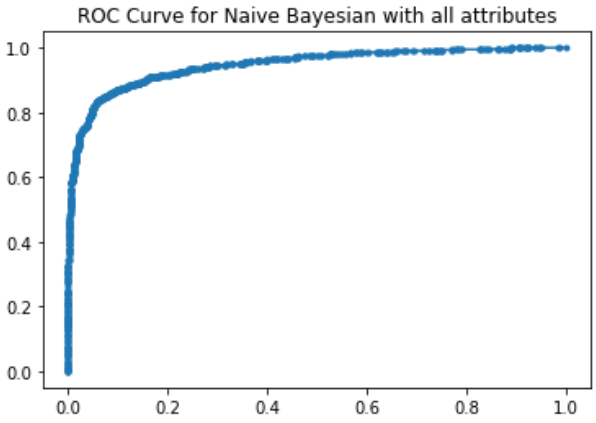


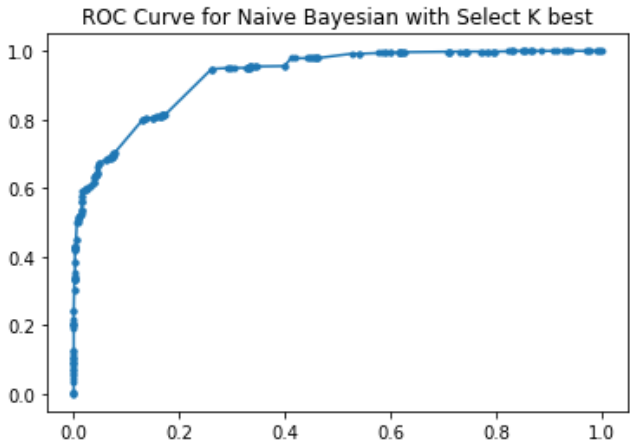


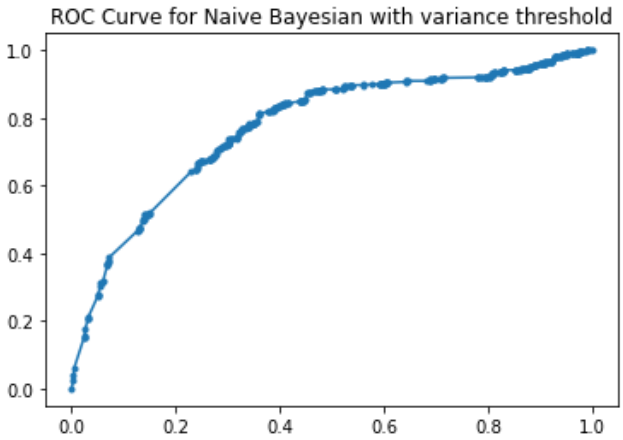


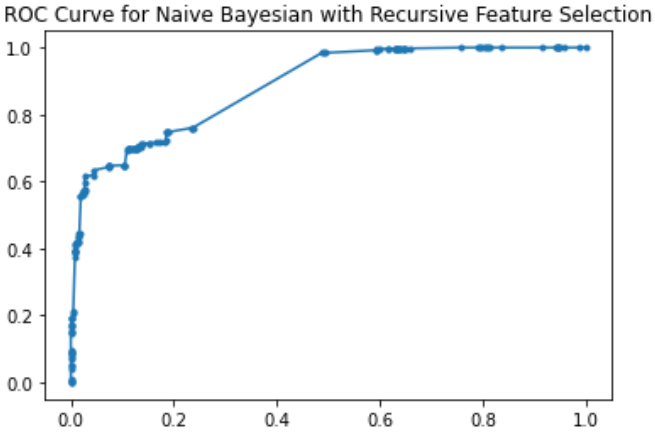


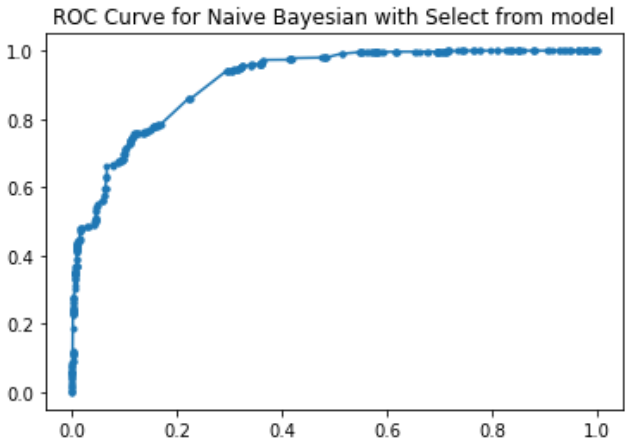
For Naïve Bayesian:

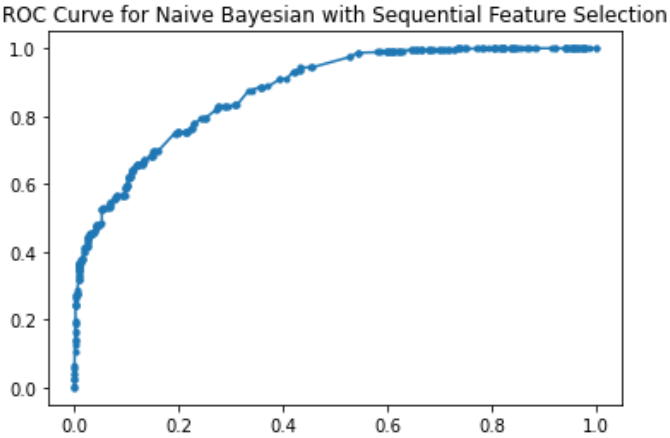




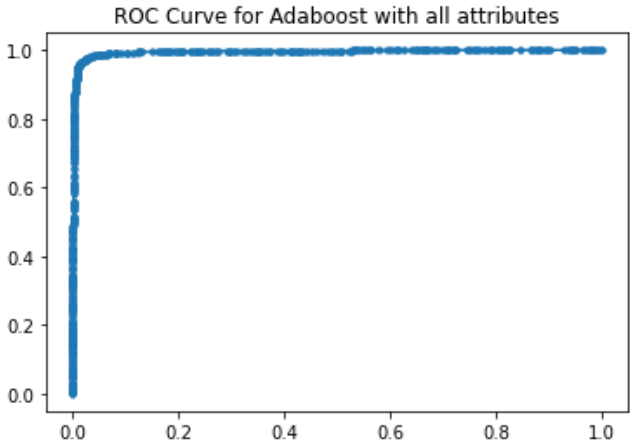


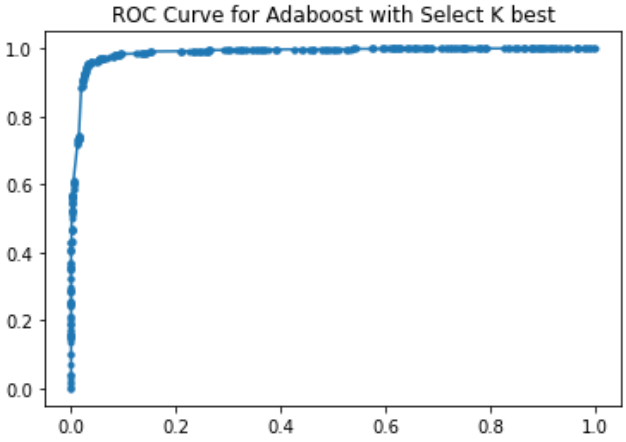


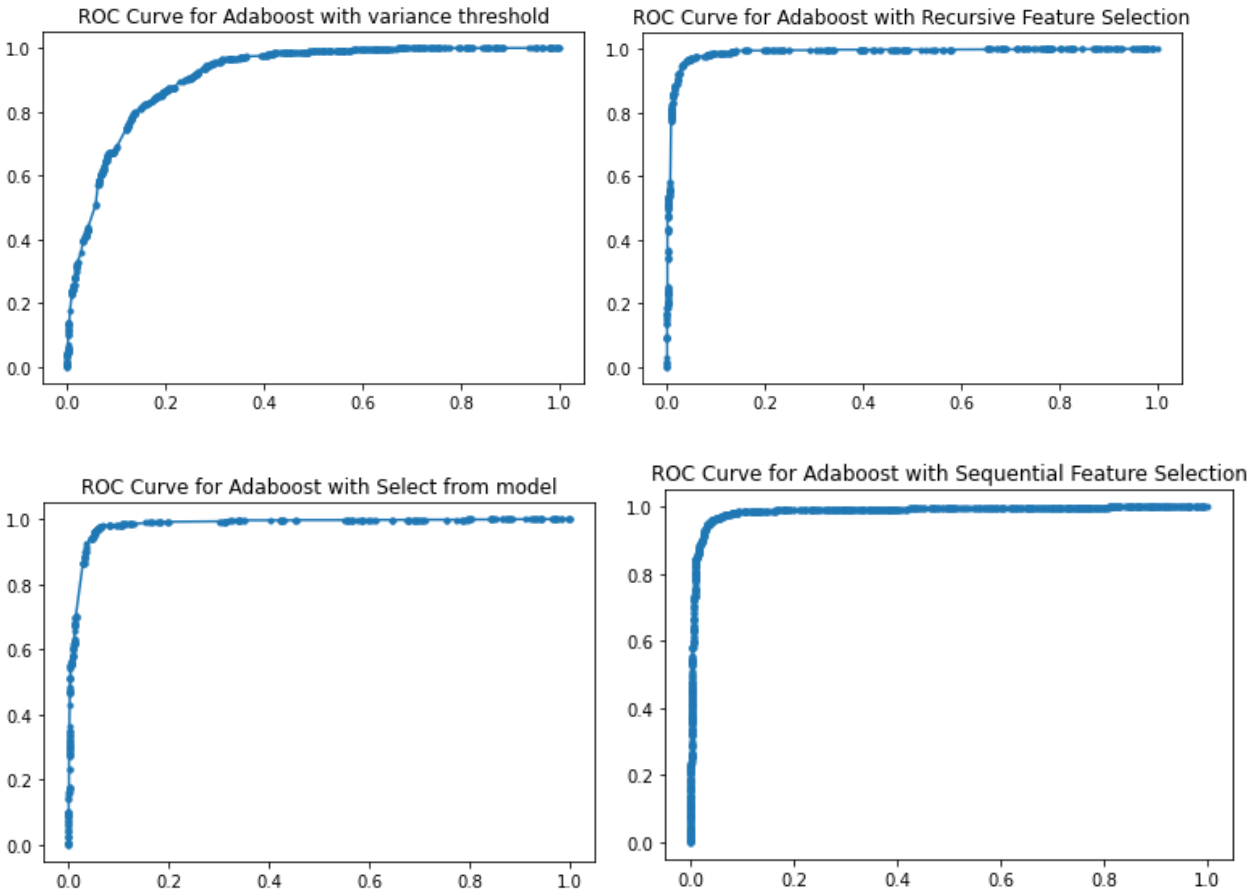




For Adaboost:







On the basis of Accuracy. The best performing classification algorithms was observed as KNN.

The feature selection techniques that give us the best results is RFE
Considering these observations, we choose KNN with RFE model selection as it performs best on the basis of overall accuracy of the model. The accuracy of model KNN with RFE is more than its base model KNN with all attributes.
Below are the results in table format for all the models with their performance metrices.

Classifier	FP Rate	TP Rate	Precision	Recall	F_Measure	MCC	ROC_Area	Accuracy
Log regression with all attributes	0.0558464	0.954221	0.943563	0.954221	0.948862	0.898322	0.986301	0.949133
Log regression with Select K best	0.122426	0.923317	0.870664	0.923317	0.896218	0.7996	0.962533	0.899147
Log regression with variance threshold	0.21826	0.836774	0.762493	0.836774	0.797908	0.616087	0.864928	0.806822
Log regression with Recursive Feature Selection	0.0911188	0.925615	0.907113	0.925615	0.916271	0.834334	0.966387	0.917083
Log regression with Select from model	0.107163	0.928133	0.888301	0.928133	0.90778	0.820223	0.959366	0.909732
Log regression with Sequential Feature Selection	0.10492	0.889148	0.895944	0.889148	0.892533	0.784204	0.959301	0.892091
RandomForest with all attributes	0.0256571	0.920688	0.975897	0.920688	0.947489	0.893403	0.991408	0.945898
RandomForest with Select K best	0.0495963	0.968806	0.949442	0.968806	0.959026	0.919032	0.988138	0.959424
RandomForest with variance threshold	0.208797	0.90513	0.757202	0.90513	0.824584	0.686999	0.915459	0.838871
RandomForest with Recursive Feature Selection	0.0580866	0.972036	0.940035	0.972036	0.955768	0.913463	0.988922	0.956483
RandomForest with Select from model	0.0533175	0.940455	0.94709	0.940455	0.943761	0.887114	0.983379	0.943546
RandomForest with Sequential Feature Selection	0.058319	0.967918	0.940035	0.967918	0.953773	0.909229	0.984551	0.954425
KNN with all attributes	0.0486393	0.96595	0.950617	0.96595	0.958222	0.917199	0.978835	0.958542
KNN with Select K best	0.0489914	0.969988	0.950029	0.969988	0.959905	0.920807	0.978153	0.960306
KNN with variance threshold	0.160804	0.87764	0.830688	0.87764	0.853519	0.71582	0.90551	0.857395
KNN with Recursive Feature Selection	0.0429983	0.968452	0.956496	0.968452	0.962437	0.925387	0.978957	0.962658
KNN with Select from model	0.0595647	0.964955	0.938859	0.964955	0.951728	0.905066	0.970337	0.952367
KNN with Sequential Feature Selection	0.0742256	0.931361	0.925338	0.931361	0.92834	0.857119	0.957615	0.92855
NaiveBayesian with all attributes	0.130987	0.883693	0.866549	0.883693	0.875037	0.752569	0.94798	0.876213
NaiveBayesian with Select K best	0.144201	0.814507	0.864785	0.814507	0.838894	0.669015	0.928036	0.833872
NaiveBayesian with variance threshold	0.29866	0.744792	0.672546	0.744792	0.706827	0.44404	0.77408	0.720964
NaiveBayesian with Recursive Feature Selection	0.15461	0.744852	0.87184	0.744852	0.803359	0.581565	0.886354	0.786533
NaiveBayesian with Select from model	0.14944	0.783209	0.866549	0.783209	0.822774	0.630149	0.91406	0.81329
NaiveBayesian with Sequential Feature Selection	0.16543	0.719922	0.868901	0.719922	0.787427	0.542448	0.875072	0.765363
Adaboost with all attributes	0.0258824	0.974133	0.974133	0.974133	0.974133	0.948251	0.995563	0.974125
Adaboost with Select K best	0.0371462	0.960704	0.962963	0.960704	0.961832	0.923554	0.988308	0.961776
Adaboost with variance threshold	0.182682	0.852886	0.80776	0.852886	0.82971	0.669276	0.912377	0.834166
Adaboost with Recursive Feature Selection	0.0398827	0.962854	0.960024	0.962854	0.961437	0.922968	0.989797	0.961482
Adaboost with Select from model	0.0510737	0.961263	0.948266	0.961263	0.95472	0.91011	0.984302	0.955013
Adaboost with Sequential Feature Selection	0.0460641	0.96204	0.953557	0.96204	0.95778	0.915943	0.9884	0.957954

***Results tend to change in case of Random Forest as at the back-end trees classification criteria differs (though the changes are very small)**

Observation

- Overall, the best model was the Adaboost with all attributes having accuracy of 0.9741
- From the reduced datasets and on comparison of the results of the 25 models, KNN with Recursive Features Selection gave the highest accuracy i.e., 0.9626 and when the same is compared with base model with all features (having 0.9585) is slightly better.
- On comparison of the best model on the reduced dataset, with the same classification algorithm i.e., KNN with all attributes having the accuracy of 0.9584
- TPR is highest for the Adaboost with all features of 0.974 whereas in the 25 models on the reduced dataset TPR was found to be the highest for Random Forest with Recursive Feature Selection 0.9720 which when compared to the base model with all attributes (having 0.9206) is slightly higher.
- ROC AUC Score is also higher for the Adaboost with all features having score of 0.9955 whereas in the 25 models on the reduced dataset ROC AUC score was found to be the highest for Adaboost with recursive feature selection 0.989 which when compared to the base model with all attributes (having 0.9955) is slightly lower.

- Precision is highest for the Random Forest with all features having score of 0.9758 whereas in the 25 models on the reduced dataset Precision was found to be the highest for Adaboost with Select K-Best selection 0.9629 which when compared to the base model with all attributes (having 0.9741) is slightly lower.
- Recall is highest for the Adaboost with all features having score of 0.9741 whereas in the 25 models on the reduced dataset Recall was found to be the highest for Random Forest with Recursive Feature selection 0.9720 which when compared to the base model with all attributes (having 0.9206) is slightly higher.
- FPR is lowest for the Random Forest with all features of 0.0256 whereas in the 25 models on the reduced dataset TPR was found to be the highest for Adaboost with Select K-Best 0.0371 which when compared to the base model with all attributes (having 0.0258) is slightly higher.

Conclusion

The several attributes of URL features led us to the revelation of what all features have impact in determining if a URL is legitimate or phishing. The project also exemplifies how having a large number of attributes affects the 'Precision', 'Recall' and other performance metrics. By selecting a few, relevant attributes to train a model, the performance of the model improves drastically.
The accuracy from the final model obtained is 0.96 (KNN, RFE). This is clearly a better model than the initial best one obtained, using 10-fold cross validation, since the accuracy of the model was 0.95 This implies that the feature selection leads to an increase in accuracy, but this increase is not very significant in our case. The model seems to be doing well (>90% accuracy) even with high dimensionality and complexity

***These observation and conclusion is based on the result as depicted in the table above**

References:

- https://pandas.pydata.org/docs/reference/
- https://scikit-learn.org/stable/
- https://www.kaggle.com/learn
- https://towardsdatascience.com/
- https://machinelearningmastery.com/
- https://datacite.org/