Epileptic Seizure Recognition

Abstract-

To learn the purpose and use of complex feature selection algorithms in machine learning and data mining field. Implementation of various classification algorithms on given data set along with a PCA analysis to reduce the computation time by selecting important features from a dataset. As part of this task, the Epileptic dataset is chosen, and the algorithms are applied using WEKA to perform the analysis. The dataset is processed, and the algorithms are applied to compare for its performance. The Performance is evaluated with the help of standard measurements, significance test and Area under curve to depict it. In this report, we will be discussing the details of Naïve Bayes and Random Forest algorithms along with the results that we have obtained by implementing these functions in the dataset using WEKA.

Introduction -

Machine learning and data science are the two emerging fields in the era of computer science. Many techniques are used to predict the state of a model. As we know that the greater the size of the model the more accurate will be the prediction. But in most cases, the size of data becomes large and too complex to handle. As such types of data set consist of noise and we need to properly normalize the dataset to get accurate results. In machine learning, we use different selection methods to reduce the size of the data set and eliminate anomalies. Several structures and unstructured algorithms are available for the construction of data. These algorithms play a vital role in data understanding and analysis. As for large datasets requires more time and storage capacity and sometimes the whole process becomes very costly and time taking so to minimize the computational cost and time, we use Feature selection Techniques in data science. For analysis, we will be using the Epileptic Seizure Recognition dataset (UCI, 2020). we will implement classification techniques (Jadhav, n.d.) to learn the behavior of the dataset.

PCA analysis(A. Matin, n.d.) to reduce the size of data set according to the best variance value. As well as we will perform a feature selection technique (A. Matin, n.d.) to reduce the size of the dataset and will critically analyze the behavior of the dataset before and after implementing the feature selection technique. Critically analyze the impact of feature selection technique on the dataset by comparing the accuracy and time taken by the dataset in the processing of classification algorithms. In the next sections we will be discussing the detail of Dataset, Classification Algorithms implemented on dataset along with PCA analysis. 6 System Description The dataset that we will be using in this project is the Epileptic Seizure Recognition dataset (UCI, 2020). It's a multivariate, preprocessed dataset which means that for performing classification we don't need to operate removing Null values, resampling, removing duplicates, normalizing, etc. as the attributes are in integer and float values so we can simply use this to perform classification techniques.

Some other characteristics of the dataset are as follows (UCI Machine Learning Repository: Epileptic Seizure Recognition Data Set, 2020):

- Attributes: 179
- Instances: 11500
- Data Type: Integer, Real
- Classes: 5
- Missing Values: Null
- Associated Task: Classification and clustering

The dataset is divided into five different folders each folder having 100 files and each file represents the data of one person. Column y has 179 from X1 to X178 total attributes but these attributes are classified against five major states.

- State 1 shows the recording of EEG seizure activity.
- State 2 shows the location of EEG where the tumor was located.
- State 3 defines the actual location of a brain tumor concerning the one located by EEG.
- State 4 shows that the eyes of the person where closed while the EEG recording.
- State 5 shows that the eyes of the person where open while the EEG recording.

All the persons who fall in category 2-5 do not have an epileptic seizure and those who fall in category 1 have an epileptic seizure.

Experimental Setup -

The UCI dataset for epileptic seizure recognition contains 5 different classes, and the need from the dataset is to predict the seizure or non-seizure based on the analysis of EEG signals. The Experimental setup comprises of the approach and tool to be used, algorithms to be compared, Methodology for dataset processing using attributes selection, Principal Component analysis for Dimensionality reduction and the details of the evaluation parameters.

Solution approach -

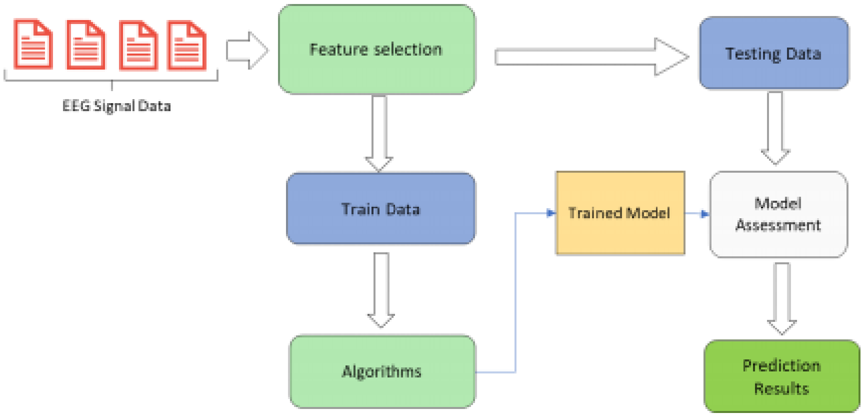


Figure 4 Solution Approach

The EEG signal data from the dataset provides information on the seizure or non-seizure data.

Feature Selection -

There are various feature selection and extraction techniques available for deriving the useful attributes/characteristics from the data. The use of a selection algorithm to reduce the dimensionality of the dataset to minimize the computation cost. Another important aspect of the feature selection is to improves the accuracy of the model as well and removes the problem of overfitting. the dataset used nowadays has many instances, which means such a dataset requires huge storage and computation time to predict the state of the model. So, to reduce the dimensionality of such large dataset PCA or principal Component analysis technique (A. Matin, n.d.) is used.

It's the most widely used data compression technique as at reduces the size of data effectively without any information loss. 7.1.2 Algorithm and Training The processed data is split into train and test data with the split ratio. Naïve Bayes and Random forest classification algorithms are chosen for studying this dataset. The algorithms are used to train the dataset and the trained/learned model is built

Model Evaluation and Prediction - Once the trained model is available, it is evaluated by the testing data, and once it is completed, the trained model is used for prediction of results. In our case, feeding of any data yields classification results as seizure or non-seizure.31 7.2 Data Processing The dataset is processed to perform the binary classification and the classes can be categorized as seizure and non-seizure on column y. The data is processed by replacing class 1 with seizure and class 2, class 3, class 4, class 5 with non-seizure. Our goal is to use classification algorithms to predict the validity of the dataset. for this, we need to divide the dataset into a certain test and train ratio. It's important to divide data set into a certain test and train ratio normally 60:40, and 80:20 ration sets are used. 60 or 80 percent of data for training and 40 or 20 percent for testing. It's true that for large training sets the system has more accurate results. So, the greater would be the training set the more accurate results will system predict.

The data pre-processing includes Data cleaning, Data integration and Data transformation. Data preprocessing is needed for the big data sets to get more accuracy and reliability. Also preprocessing is much required in business, education and research industry for further cost reduction, ease in storage and report making.

> o Test Set: the test set contains data from all folders and runs over the trained model. Model accuracy is predicted based upon the test set results that are predicted correctly. So, for this problem, the test set has 20% of values from the actual dataset.
>
> o Train Set: to attain maximum accuracy, we assign train set maximum values. So, in this case, our train set contains 80% of actual dataset values. The model will be trained on the given train set values and will be then tested for validating accuracy.

Algorithms -

The dataset is valid for both classification and clustering problems. In machine learning, we have many supervised and unsupervised classification and clustering learning algorithms. Some common classification algorithms are Naïve Bayes, Linear Regression, Logistic Regression, Random Forest, Neural Network, etc.

Naïve Bayes - For a structured dataset where it requires predicting the class of unknown dataset naïve Bayes classification algorithm is used. it's the simplest supervised machine learning classification algorithm that predicts the class of any feature based on the values of other unrelated features. It works based on the Bayes algorithm, easy to build, and ideal for large dataset problems. Its widely used in recommendation systems and real-time predictions system as well (Jadhav, n.d.). The reason that we choose this algorithm for our dataset is that our dataset has signal data and naïve Bayes works best for classification predictions.

Random Forest -Random forests are a combination of tree predictors such that each tree depends on the values of a random vector sampled independently and with the same distribution for all trees in the forest(Breiman, n.d.). It is an ensemble algorithm that works on the output from multiple base learners to improve the performance by using their weighted outcome to predict new data (D. K. Reddy, n.d.).

Evaluation of Model -

The trained model is evaluated using the test data and the details of the classification can be recorded. 7.5 Performance measurements for evaluation and performance measurement, the following parameters derived from the confusion matrix are used.

Accuracy = $(TP+TN)$ / $((TP+FN)$ +$(FP+TN))$ (Scikit, 2020)

Precision = TP / ((TP+FP)) (Scikit, 2020)

F1 Score = (2xTP) / ((2xTP+FP+FN)) (Scikit, 2020)

Recall = (TP) / ((TP+FN)) (Scikit, 2020)

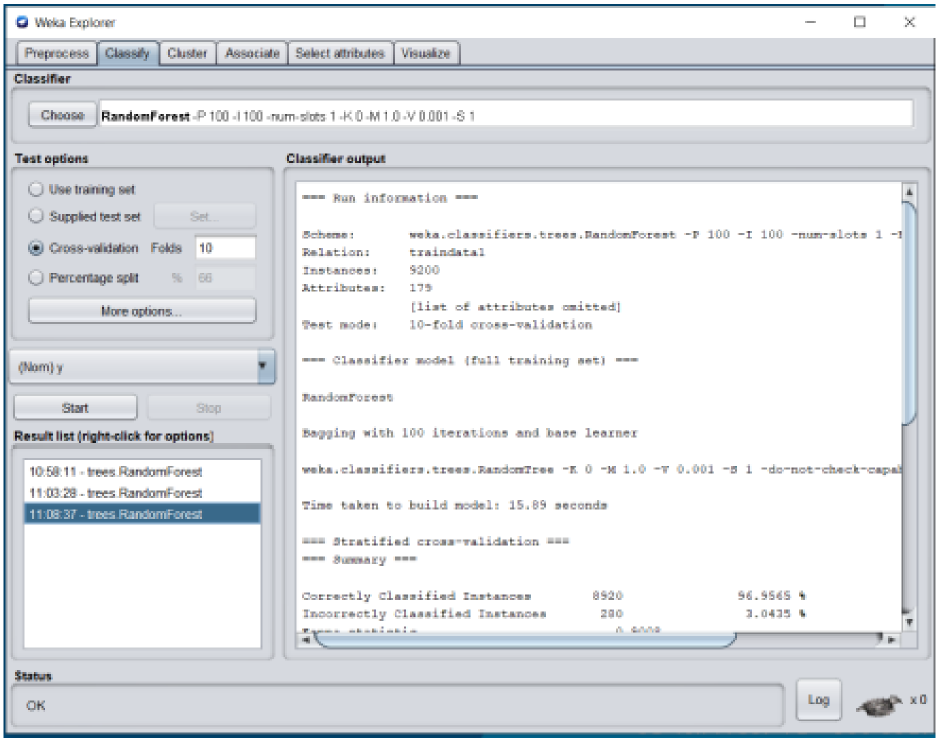
The following section details out the results carried out for this dataset using this algorithm in the WEKA tool.

Experimental Results -

Weka is the workbench for machine learning and can be used to perform or create models using algorithms for the dataset (WEKA, 2020). The tool is widely used for academic and learning purposes.

Training and Test Data - The training and test data can be created directly with the help of Resample mechanism in Weka. The Weka Explorer offers this filter under Preprocessing and can be leveraged for the creation of data. As part of Resampling, the sample size percentage to be given as 80 which indicates the split of data as 80-20 in to training and test dataset. After the resampling, the dataset is split into train and test datasets. It can be saved as CSV in WEKA. The Training dataset is loaded again on to the tool for further processing.

Apply of Classification Algorithms - The Algorithms Naïve-Bayes and Random Forest are selected for this classification problem. In the WEKA, there is a classify tab and the algorithms can be selected and checked for the training of this data. The classify tab have options for the user to select the options of choosing cross validation, training percentage, percentage split. For our study, cross validation with 10 folds is chosen and the algorithm is applied. A sample screenshot for this is given below:



Classifier

RandomForest -P 100 -I 100 -num-slots 1 -K 0 -M 1.0 -V 0.001 -S 1

Test options

○ Use training set
○ Supplied test set
● Cross-validation Folds 10
○ Percentage split % 66

(Nom) y

Result list (right-click for options)

10:58:11 - trees.RandomForest
11:03:28 - trees.RandomForest
11:08:37 - trees.RandomForest

Classifier output

```
=== Run information ===

Scheme:       weka.classifiers.trees.RandomForest -P 100 -I 100 -num-slots 1 -I
Relation:     traindata1
Instances:    9200
Attributes:   179
              [list of attributes omitted]
Test mode:    10-fold cross-validation

=== Classifier model (full training set) ===

RandomForest

Bagging with 100 iterations and base learner

weka.classifiers.trees.RandomTree -K 0 -M 1.0 -V 0.001 -S 1 -do-not-check-capal

Time taken to build model: 15.89 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances        8920               96.9565 %
Incorrectly Classified Instances       280                3.0435 %
```

Status

OK

Figure 5 Sample screenshot

The results of the algorithms for the classification is given below:

Algorithm Name	***Correctly classified Instances***	***Incorrectly classified Instances***	***Time taken to build model (seconds)***
Naive Bayes	*95.9348*	*4.0652*	*0.17*
Random Forest	*96.9565*	*3.0435*	*10.86*

Table 4 Comparison between algorithms

Naïve Bayes

The detailed analysis with the performance evaluation parameters are given below:

Naïve Bayes	**TPRate**	**FPRate**	**Precision**	**Recall**	**F-Measure**	**MCC**	**ROCArea**	**PRCArea**	**Class**
	0.975	0.104	0.975	0.975	0.975	0.871	0.954	0.979	nonseizure
	0.896	0.025	0.897	0.896	0.897	0.871	0.985	0.902	seizure
Weighted	0.959	0.088	0.959	0.959	0.959	0.871	0.96	0.964	

Table 5 Naive Bayes - Performance evaluation Parameters

The classifier output from Weka is given below:

```
Time taken to build model: 0.17 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances        8826               95.9348 %
Incorrectly Classified Instances       374                4.0652 %
Kappa statistic                          0.8713
Mean absolute error                      0.0405
Root mean squared error                  0.2012
Relative absolute error                 12.9022 %
Root relative squared error             50.6201 %
Total Number of Instances             9200

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.975    0.104    0.975      0.975    0.975      0.871    0.984     0.979     nonseizure
                 0.896    0.025    0.897      0.896    0.897      0.871    0.985     0.902     seizure
Weighted Avg.    0.959    0.088    0.959      0.959    0.959      0.871    0.960     0.984

=== Confusion Matrix ===

    a    b   <-- classified as
 7205  186 |    a = nonseizure
  188 1621 |    b = seizure
```

The results can be seen through the Area Under ROC curve as below:

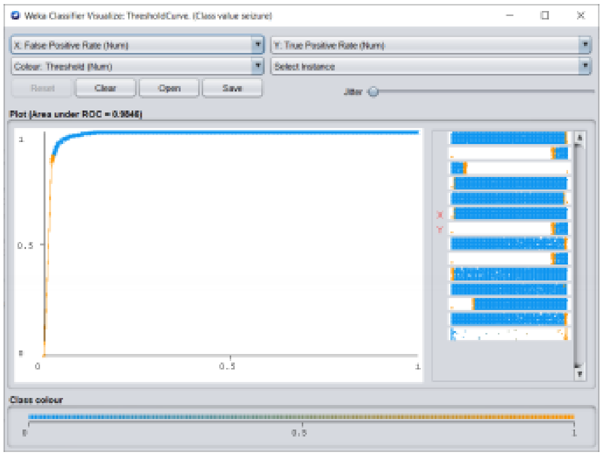

Weka Classifier Visualize: ThresholdCurve. (Class value seizure)
X: False Positive Rate (Num)
Y: True Positive Rate (Num)
Colour: Threshold (Num)
Select Instance
Reset
Clear
Open
Save
Jitter
Plot (Area under ROC = 0.9846)
1
0.5
0
0
0.5
1
X
Y
Class colour
0
0.5
1

Figure 6 Area under ROC curve - Naive Bayes

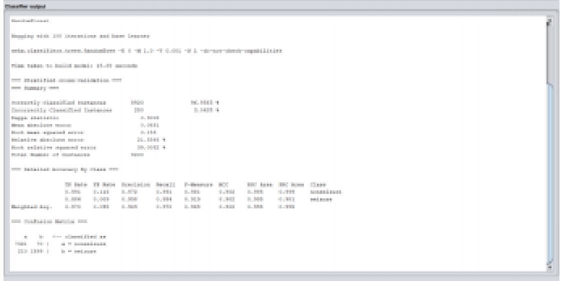
Random Forest

The detailed analysis with the performance evaluation parameters are given below:

Random Forest	TPRate	FPRate	Precision	Recall	F-Measure	MCC	ROCArea	PRCArea	Class
	0.991	0.116	0.972	0.991	0.981	0.902	0.995	0.999	nonseizure
	0.884	0.009	0.958	0.884	0.919	0.902	0.995	0.981	seizure
Weighted	0.97	0.095	0.969	0.97	0.969	0.902	0.995	0.995	

Table 6 Random Forest - Performance Evaluation Parameters

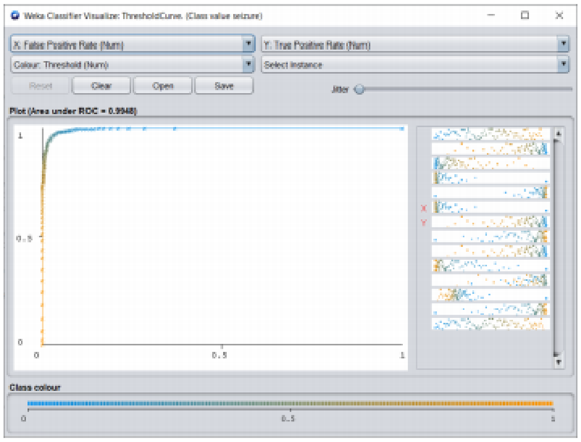
The classifier output from Weka is given below:



Classifier output

```
RandomForest

Bagging with 100 iterations and base learner

weka.classifiers.trees.RandomTree -K 0 -M 1.0 -V 0.001 -S 1 -do-not-check-capabilities

Time taken to build model: [illegible] seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances       [illegible]            [illegible] %
Incorrectly Classified Instances      [illegible]            [illegible] %
Kappa statistic                          [illegible]
Mean absolute error                      [illegible]
Root mean squared error                  [illegible]
Relative absolute error                 [illegible] %
Root relative squared error             [illegible] %
Total Number of Instances            [illegible]

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 [illegible]    [illegible]    [illegible]      [illegible]    [illegible]      [illegible]    [illegible]     [illegible]     nonseizure
                 [illegible]    [illegible]    [illegible]      [illegible]    [illegible]      [illegible]    [illegible]     [illegible]     seizure
Weighted Avg.    [illegible]    [illegible]    [illegible]      [illegible]    [illegible]      [illegible]    [illegible]     [illegible]

=== Confusion Matrix ===

    a    b   <-- classified as
 [illegible]   [illegible] |    a = nonseizure
  [illegible] [illegible] |    b = seizure
```

The results can also be seen in through are under graph curve.

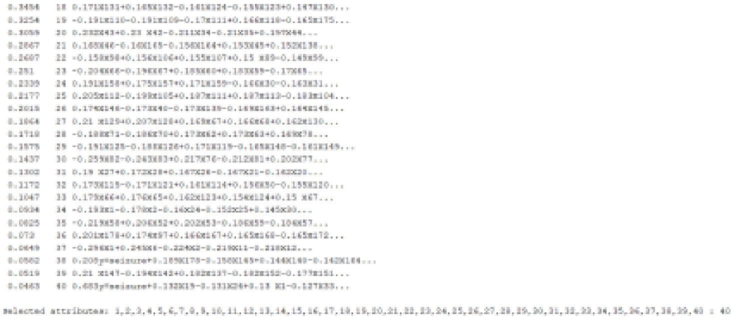


Weka Classifier Visualize: ThresholdCurve. (Class value seizure)
X: False Positive Rate (Num)
Y: True Positive Rate (Num)
Colour: Threshold (Num)
Select Instance
Reset
Clear
Open
Save
Jitter
Plot (Area under ROC = 0.9948)
1
0.5
0
0
0.5
1
X
Y
Class colour
0
0.5
1

Figure 7Area under ROC -Random forest

Based on the results from the Table 4, it is evident that the Random Forest classifier is having better accuracy than the Naïve Bayes classifier. Even though the time taken to build the model is higher, the prediction from random forest is higher which makes it an ideal candidate for the Epileptic Seizure Recognition.

PCA Analysis - In the WEKA tool, you can perform a PCA analysis of the dataset easily. After loading the dataset from Select Attributes tab choose PCA analysis and click on start to begin the PCA analysis of dataset. Following is the sample screenshot and the summary of PCA analysis with different variance covered values:



```
0.3454   18 0.171X131+0.165X132-0.161X124-0.155X123+0.147X130...
0.3254   19 -0.191x110-0.191x109-0.17x111+0.166x118-0.165x175...
0.3059   20 0.232X43+0.23 X42-0.211X34-0.21X35+0.197X44...
0.2867   21 0.165X46-0.16X165-0.156X164+0.155X45+0.152X138...
0.2607   22 -0.158x58+0.156x106+0.155x107+0.15 x89-0.145x59...
0.251    23 -0.204X66-0.196X67+0.185X60+0.183X59-0.17X65...
0.2339   24 0.191X158+0.175X157+0.171X159-0.166X30-0.163X31...
0.2177   25 0.205x112-0.198x105+0.187x111+0.187x113-0.183x104...
0.2015   26 0.174X146-0.173X40-0.173X139-0.169X163+0.164X145...
0.1864   27 0.21 x129+0.207x128+0.169x67+0.166x68+0.162x130...
0.1718   28 -0.188X71-0.186X70+0.173X62+0.173X63+0.169X78...
0.1575   29 -0.191X125-0.188X126+0.171X119-0.165X148-0.161X145...
0.1437   30 -0.255X82-0.243X83+0.217X76-0.212X81+0.202X77...
0.1302   31 0.19 X27+0.172X28+0.167X26-0.167X21-0.162X20...
0.1172   32 0.173X115-0.171X121+0.161X114+0.156X50-0.155X120...
0.1047   33 0.175X66+0.176X65+0.162x123+0.156x124+0.15 x67...
0.0934   34 -0.193x1-0.178x2-0.16x24-0.152x25+0.145x30...
0.0825   35 -0.215X58+0.206X52+0.202X53-0.196X55-0.194X57...
0.072    36 0.201x178+0.174x97+0.166x167+0.165x168-0.165x172...
0.0649   37 -0.256X1+0.245X6-0.224X2-0.219X11-0.218X12...
0.0582   38 0.205y=seizure+0.159X175-0.156X165+0.144X169-0.142X164...
0.0519   39 0.21 x147-0.194x142+0.182x137-0.182x152-0.177x151...
0.0463   40 0.683y=seizure+0.132X15-0.131X26+0.13 X1-0.127X33...

Selected attributes: 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40 : 40
```

Dataset name	*Instances*	*Attributes*	*Variance Covered*	*Ranked Attributes*
Epileptic Dataset	*11500*	*179*	*X*	*X*
			0.95	*40*
			0.85	*30*
			0.75	*24*

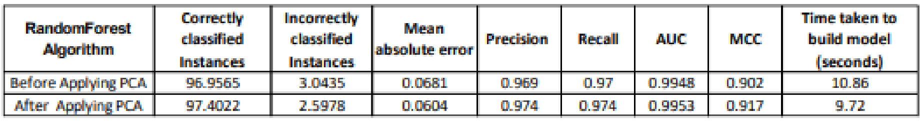
Table 7 Comparison table with different Variance Covered Values

In the dimensionality of dataset is reduced to a Total of the 40 attributes that have been selected based on the values of standard deviation with the variance covered as 0.95. The Table 7 shown above displays the attributes reduction with different variance covered values. As in CA analysis, only those fields are chosen whose stdDev value is greater than the tolerance value (Y, n.d.). PCA can be performed manually by setting the value of Tolerance and comparing it with stdDev value of each attribute. for large data set, it is not feasible to run PCA analysis manually, so we use automated tools for this purpose. Now based on our analysis we can keep all the selected 40 attributes and remove the rest of them. This will reduce the computation cost.

Feature Selection Algorithm -

The goal of a feature selection algorithm is to reduce the dimensionality of the dataset to minimize the computation cost. The selection algorithm that this study uses is the principal component analysis.

In the WEKA tool, the feature selection PCA is chosen from the Attribute selection tab. From choose, select attribute evaluator and from options select PCA with Variance covered value as 0.95. The dimensionality is reduced and the attributes contributing to the overall dataset reduced to 40 from 179. Now, with these ranked attributes, the algorithms are executed again, and the results are documented.



RandomForest Algorithm	Correctly classified Instances	Incorrectly classified Instances	Mean absolute error	Precision	Recall	AUC	MCC	Time taken to build model (seconds)
Before Applying PCA	96.9565	3.0435	0.0681	0.969	0.97	0.9948	0.902	10.86
After Applying PCA	97.4022	2.5978	0.0604	0.974	0.974	0.9953	0.917	9.72

Table 8 Performance Parameters before and after applying PCA

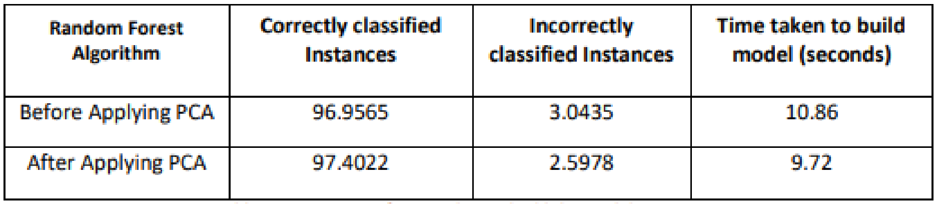
From the results, it is evident that the accuracy improved with the limited set of features with PCA than without PCA. The Error, precision, recall, AUC and MCC performance parameters shown a good improvement over without PCA.

Challenges and Implications - The challenges in building a model depends on various factors such as quality of data, accuracy, time taken to build the model and the selection of algorithms for the problem set. Based on the use cases, the algorithms to be chosen wisely and appropriate optimization techniques to be applied for better feature extraction of the input data. All the algorithms require time to build it based on how it is implemented. The time taken is a key factor as it will determine the speed the algorithm is trained for the data since the datasets are always huge. Time taken to build the model should not be characterized based on the speed alone but the accuracy, less error and other metrics such as Precision, Recall, Sensitivity etc. The complexity measurement is driven by the attributes, classes available in the dataset. It can be Multi class, Linear or Non-linear data which contributes to the time taken for an algorithm to build a model.

On specific fields, the time taken factor is over-run by accuracy of classification and prediction. Particularly, in health care, the accuracy of the classification is more important factor for an algorithm to be chosen. There lies a balance between time and accuracy in these domains.

The feature selection algorithms are introduced to augment the speed and maintaining the accuracy of the classification. The algorithms are statistical based and filter-based methods. These facilitate the extraction of informative features from the dataset.

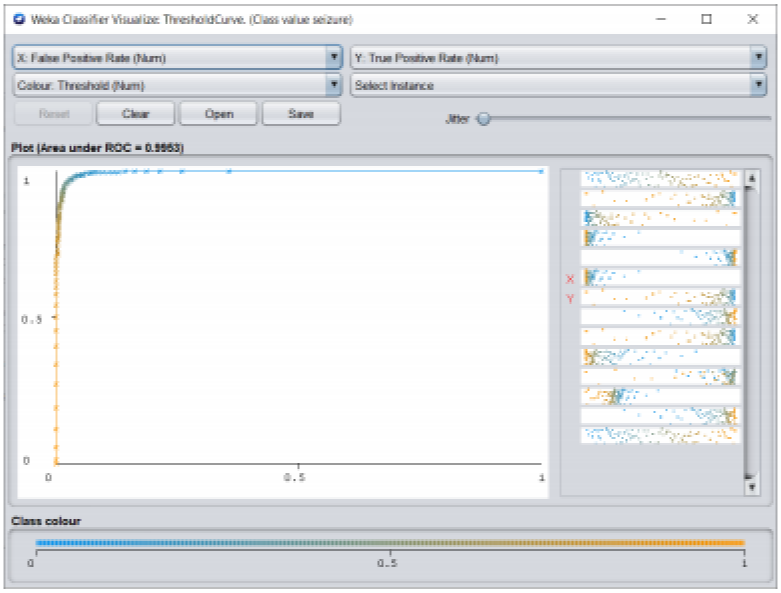
Once these techniques are employed, the ranked attributes are available which covers the complete dataset characteristics. In the current study, the PCA is applied and the machine learning algorithm is compared for its performance. With the feature selection, the time taken to build the model and the accuracy improvises for the classification.



Random Forest Algorithm	Correctly classified Instances	Incorrectly classified Instances	Time taken to build model (seconds)
Before Applying PCA	96.9565	3.0435	10.86
After Applying PCA	97.4022	2.5978	9.72

Table 9 Comparison of Time taken to build the model

From the table, the time taken to build the model reduces and accuracy of correctly classified instances increases. The Area under ROC curve is given below:


Weka Classifier Visualize: ThresholdCurve. (Class value seizure)

X: False Positive Rate (Num)

Y: True Positive Rate (Num)

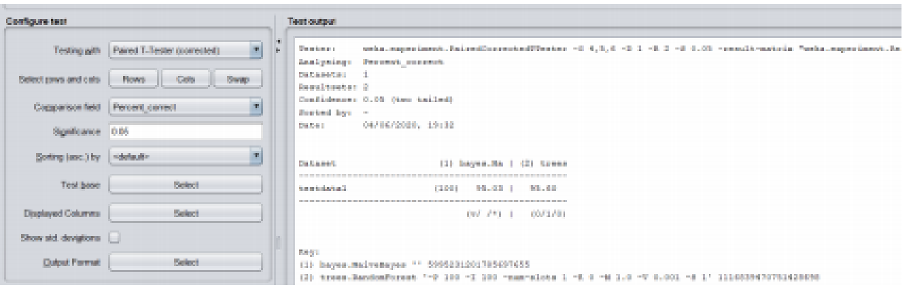
Colour: Threshold (Num)

Select Instance

Plot (Area under ROC = 0.9953)

Figure 8 Area Under ROC for Random Forest after PCA

Performance Comparison of algorithms - The performance comparison of algorithms can be done using the experimenter in WEKA. The process of Design Run and review of results to be followed. The chosen algorithms are tested with samples of data and the results are analyzed for its performance. The significance test is performed, and the results are found. Random forest scores better than Naïve Bayes.


Configure test

Testing with: Paired T-Tester (corrected)

Comparison field: Percent_correct

Significance: 0.05

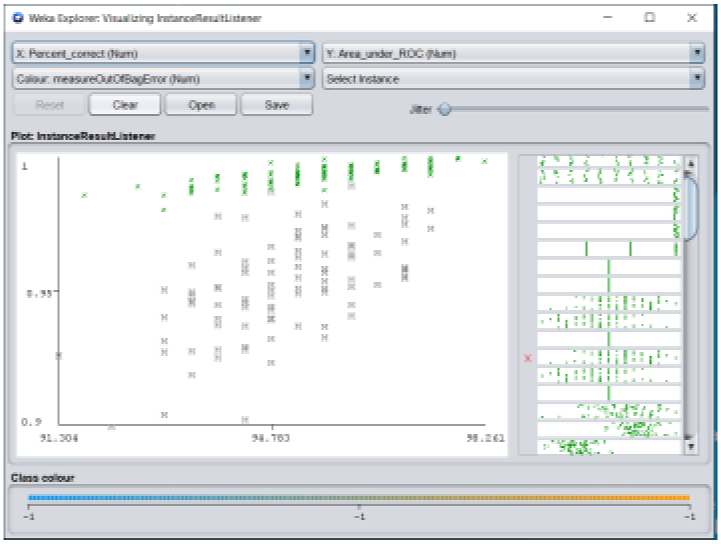
Sorting (asc.) by: <default>

Show std. deviations: □

```
Tester:     weka.experiment.PairedCorrectedTTester -G 4,5,6 -D 1 -R 2 -S 0.05 -result-matrix "weka.experiment.Re
Analysing:  Percent_correct
Datasets:   1
Resultsets: 2
Confidence: 0.05 (two tailed)
Sorted by:  -
Date:       04/06/2020, 19:32


Dataset                   (1) bayes.Na | (2) trees
--------------------------------------------------
testdata1             (100)   98.03 |   98.68
--------------------------------------------------
                               (v/ /*) |   (0/1/0)


Key:
(1) bayes.NaiveBayes '' 5995231201785697655
(2) trees.RandomForest '-P 100 -I 100 -num-slots 1 -K 0 -M 1.0 -V 0.001 -S 1' 1116839470751428698
```

Figure 9 Performance Comparison in Weka

The plot of the percent correct vs Area under ROC is given below. The graph indicates the data samples are classified correctly for Random forest (Green in color) than Naïve Bayes.



Weka Explorer: Visualizing InstanceResultListener
Reset
Clear
Open
Save
Jitter
Plot: InstanceResultListener
1
0.95
0.9
91.304
94.783
98.261
Class colour
-1
-1
-1

X: Percent_correct (Num)

Y: Area_under_ROC (Num)

Colour: measureOutOfBagError (Num)

Select Instance

Figure 10 Comparison between Models

Conclusion -

In this growing era of data complexity and volume, for predicting states of a large number of datasets its important to reduce the computational cost. Several machine learning algorithms are available to solve classification problems. In the current study, two classification algorithms namely naïve Bayes and Random Forest are implemented and the results are compared. Random forest works well compared to the Naïve Bayes classifier in terms of accuracy which is very much needed for the medical data analysis. The dimensionality reduction concepts are explained through the Principal component analysis and the results are documented, explained for different variance covered values. PCA analysis certainly reduces the dimensionality of the dataset and reduces the computational cost. The feature selection algorithm is chosen as PCA and the impact of the dimensionality reduction on the performance of algorithms are clearly explained.

Bibliography -

A. Matin, R. A. (n.d.). A Hybrid Scheme Using PCA and ICA Based Statistical Feature for Epileptic Seizure Recognition from EEG Signal. 2019 Joint 8th International Conference on Informatics, Electronics & Vision (ICIEV) and 2019 3rd International Conference on Imaging, Vision & Pattern Recognition (icIVPR). Breiman, L. (n.d.).

Random Forests Machine Learning. D. K. Reddy, A. M. (n.d.).

Feature extraction and classification of Electroencephalogram signals for vigilance level detection. 2013 International Conference on Control, Automation, Robotics and Embedded Systems (CARE). DB Modeling. (2020). Retrieved from https://www.vertabelo.com/blog/database-design-more-than-just-an-erd/ Jadhav, S. D. (n.d.).

Comparative Study of K-NN, Naive Bayes and Decision Tree Classification Techniques. International Journal of Science and Research (IJSR). MongoDB. (n.d.). MongoDB. Retrieved from https://docs.mongodb.com/ PC technicalPro. (2020). PC technicalPro.

Retrieved from https://pctechnicalpro.blogspot.com/2017/04/advantages-disadvantages-er-model dbms.html Scikit. (2020). Scikit Learn.

Retrievedfrom https://scikitlearn.org/stable/modules/generated/sklearn.metrics.precision_recall_fscore_suppo t.html UCI. (2020).

Datasets. Retrieved from https://archive.ics.uci.edu/ml/datasets/Epileptic+Seizure+Recognition UrbanBus. (n.d.).

Urban Bus. Retrieved from 2020: https://ppiaf.org/sites/ppiaf.org/files/documents/toolkits/UrbanBusToolkit/assets/43 /1d/1d4.html WEKA. (2020).

WEKA. Retrieved from https://www.cs.waikato.ac.nz/ml/weka/ Wikipedia. (2020). MongoDB. Retrieved from https://en.m.wikipedia.org/wiki/MongoDB Y, L. Z. (n.d.).

Eigen-analysis of nonlinear PCA with polynomial kernels. Statistical Analysis and Data Mining