Alzheimer's disease classification using multi-feature fusion and ensemble learning

Sumedha Prathipati

May 4, 2020

Contents

1 Introduction

The Alzheimer's disease is a type of dementia that causes problems with an individual's thinking, memory and behavior. [1] This disease has become very common especially with the elderly people. According to the World Health Organization, there are nearly 10 million new cases every year worldwide. [2]

The patients experiencing such changes in their thinking and behavior can be categorized based on the intensity of their problem into the following:

1. Normal controls (CTL) : This condition is associated with an individual who does not display any changes in their memory or thinking abilities.

2. Mild cognitive impairment (MCI) : MCI is a condition in which an individual has mild but noticeable changes in their thinking abilities.

3. Alhzheimer's diease (AD) : This is the extreme condition where the symptoms becomes severe enough to intervene with daily tasks.

It is crucial to be able to distinguish the AD and MCI patients from the CTL patients in the early stages as there are no proper medications to cure the Alzheimer's once the symptoms have become severe. This remains a challenge because the conventional evaluation of the MRI scans requires manual intervention to a huge extent. Therefore, this project focuses on using pattern recognition and machine learning techniques to automate the process of classifying the patients into the CTL, MCI and AD categories using a multi-feature fusion and ensemble learning algorithm.

2 Problem statement

This project aims to perform a supervised learning based two-class classification of the Alzheimer's patients (AD vs MCI, AD vs CTL and MCI vs CTL) using a multi-feature fusion algorithm for feature selection and an ensemble of classifiers for classification. The leave-ten-out cross validation technique is used to create multiple splits of the dataset. The accuracy results of this proposed model are then evaluated with respect to other state-of-the models on this problem domain.

2.1 Goals

- Review the current state-of-the-art approach in Alzheimer's disease classification

- Implement feature selection on the Alzheimer's dataset using the multi-feature fusion algorithm from the base paper

- Experiment with various linear classifiers to implement an ensemble of models

- Compare this proposed model with other previous models and summarize the results

3 Related work

3.1 Dataset

The dataset used in this project is from [3]. It consisted of a set of 66 subjects selected from the Alzheimer's Disease Research Center categorized into the CTL, MCI and AD classes. Each subject was associated with around 55 million features and the top discriminative features were computed using the Augmented variance ratio metric.

3.2 Approach

There have been recent developments in classifying the alzheimer's disease subjects using structural features of the brain images. [4] uses the multi-feature fusion approach for feature selection to precisely identify the individuals with AD or MCI from CTL. The SVM - RFE (Reverse feature elimination) technique combined with each feature's covariance matrix is used to get a robust set of optimal features in this paper. These features are passed to an SVM classifier to achieve accuracy, sensitivity and specificity rates greater than their respective baselines.

[5] uses the structural features from the brain extracted according to the size of the hippocampus and amount of Gray matter, White matter and Cerebrospinal fluid to distinguish the individuals with AD from CTL. An ensemble of the SVM, Random forest and Neural network classifiers based on the majority voting are implemented to achieve specificity rates greater than the existing approaches, especially when the size of the hippocampus feature is considered in particular.

[6] is a very recent paper which uses the SVM - RFE with correlation coefficients method for feature selection and an ensemble of parametric and non-parametric classifiers based on majority voting to classify the subjects into AD affected or normal control. The morphometric and texture based features are initially extracted before the feature selection to achieve greater accuracy when compared with the CAD systems.

3.3 State of the art

The state of the art in this area achieves the following sensitivity and specificity values :

Model	**Accuracy**	**Sensitivity mean**	**Specificity mean**
State of the art ([3])		91 %	99 %
Multi-feature fusion + SVM ([4])	83.75 %	82.13 %	86.01 %
Latest work : Multi-feature fusion + Ensemble of classifiers ([6])	93 %		

Table 1: State of the art results for various models. The empty fields imply that the respective metrics were not evaluated in the work.

Sensitivity and specificity values of 91 % and 99 % are achieved when the same dataset which is used in this project is used. I am also comparing the accuracy values of the multi-feature fusion models on the ADNI dataset ([4], [6]) because of the structural similarity of this data with respect to the dataset used in this project. The ADNI dataset contains a total of 170 subjects, out of which 54, 58 and 58 subjects belong to the AD, MCI and CTL classes respectively. We can observe the difference in the reported values because the potential features extracted from the images in each of these cases is different. The fields left empty in Table 1 imply that the respective metrics were not evaluated in the work.

3.4 Comparison with existing work

This project differs from what has already been done, in the sense that it implements a SVM-RFE method along with each feature's covariance matrix to select an optimal subset of features. This is followed by a stacked ensemble of only linear classifiers to perform a two-class classification. The ensemble learning technique used in this project differs from [6]. In [6], they have have used both parametric and non-parametric models with majority voting for classification while we are using a simple stacking ensemble of models in this project. Secondly, the features extracted in each of these works are different. [6] uses features extracted from the ADNI dataset.

4 Dataset

The dataset includes 76 subjects categorized as follows:

1. 20 Alzheimer's disease (AD)
2. 26 Mild cognitive impairment (MCI)
3. 20 Normal controls (CTL)

This dataset contains the prominent features already extracted from the structural MRI images [3]. Therefore, each subject is associated with 1000 highest discriminative individual features, ranked according to the Augmented variance ratio score using the leave-ten-out cross validation. The labeled dataset hence, contains 3 classes, 76 subjects and 1000 features.

The data is divided such that the training data contains 60 subjects and the test data contains 16 subjects. The last column contains the class labels and is appended to the 1000 features. The leave-ten-out cross validation is used to divide the data into multiple splits. Each cross validation split contains the following:

- trainingSpread : 60 x 1001 matrix
- testSpread : 16 x 1001 matrix
- mergedFeatureIDs : 1 x 1000 vector containing feature type for each feature
  There are a total of 18 feature types as follows :

- 1, 2, 3 : x, y and z components of a deformation field
- 4 : length of the deformation field vector (R)
- 5 : Jacobian (J)
- 6 : log of Jacobian (LJ)
- 7 to 12 : absolute value of symmetry score for x, y, z, R, J and LJ respectively
- 13 to 18 : signed symmetry score for x, y, z, R, J and LJ respectively

- mergedMomentIDs : 1 x 1000 vector containing statistical moment for each feature

- mergedScaleIDs : 1 x 1000 vector containing image scales for each feature
  - 0 : image subsampled by the factor of 8
  - 1 : image subsampled by the factor of 4
  - 2 : image subsampled by the factor of 2
  - 3 : original image scale

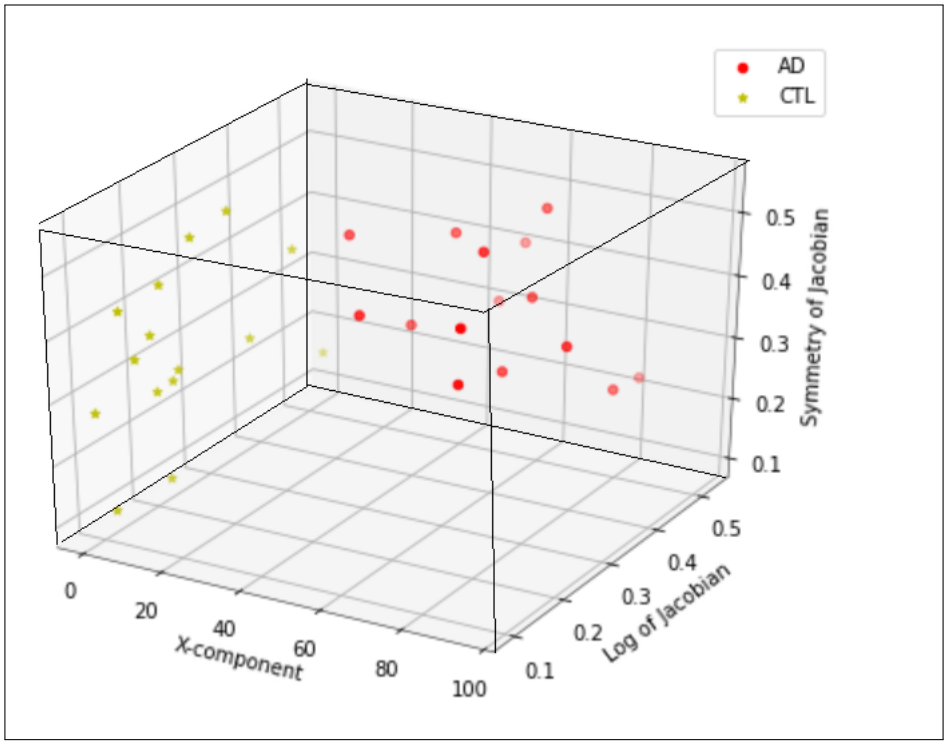
- mergedBestAVRScores : 1 x 1000 vector containing Augmented variance ratio score [3] of each feature

4.1 Dataset size

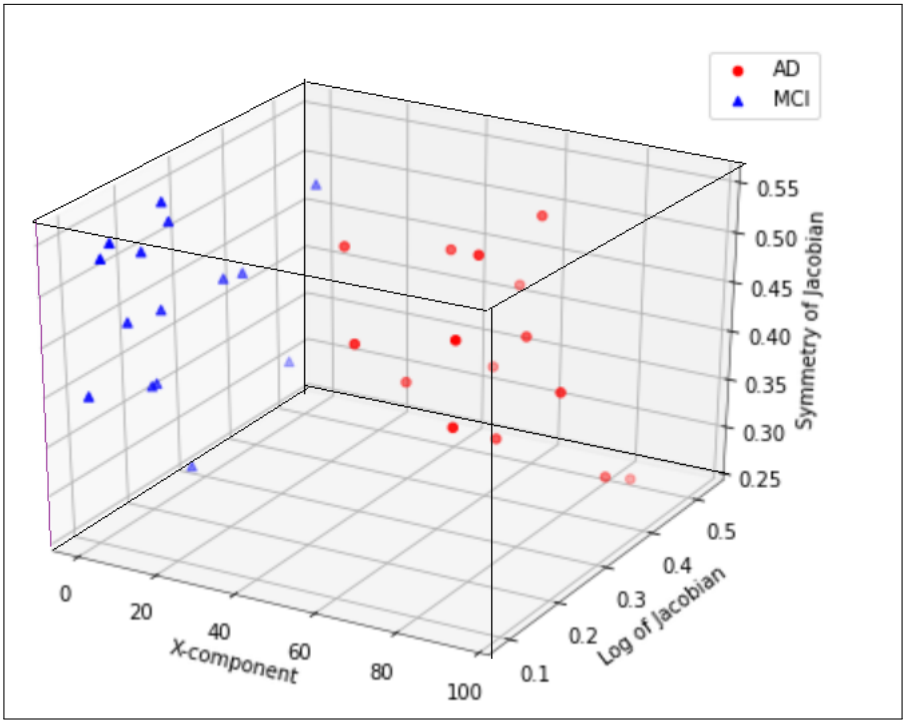
The rule of 10 states that the amount of training data you need for a well performing model is around 10 times the number of parameters in the model. [7] This project does not implement any deep learning models and only uses simple linear classifiers such as the SVM, KNN, Decision tree and LDA classifiers. This implies that the models are not very complex and deal with a limited number of parameters. Hence, the training data available for this project will be sufficient as it is nearly 10 times the number of model parameters.

Additionally, from [3], we infer that the features are non independent implying that it is possible to find a subspace where the classes in the data are separable.

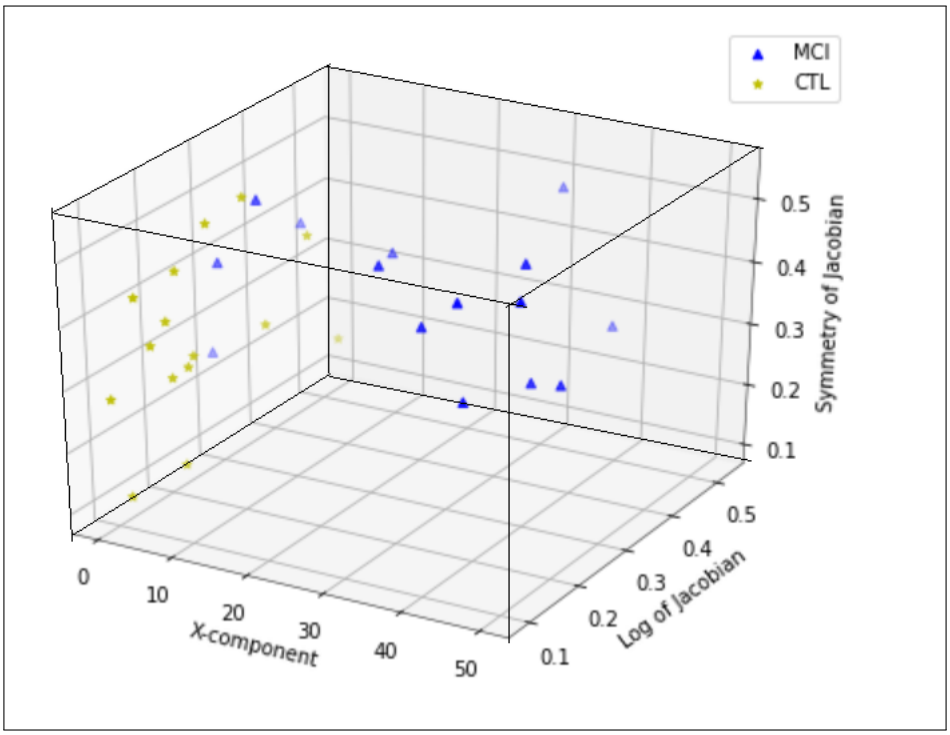
Visualizing the data corresponding to the AD-CTL, AD-MCI and MCI-CTL classes considering the X component, log of jacobian and the symmetry of jacobian features on the split 1 training data is as follows:



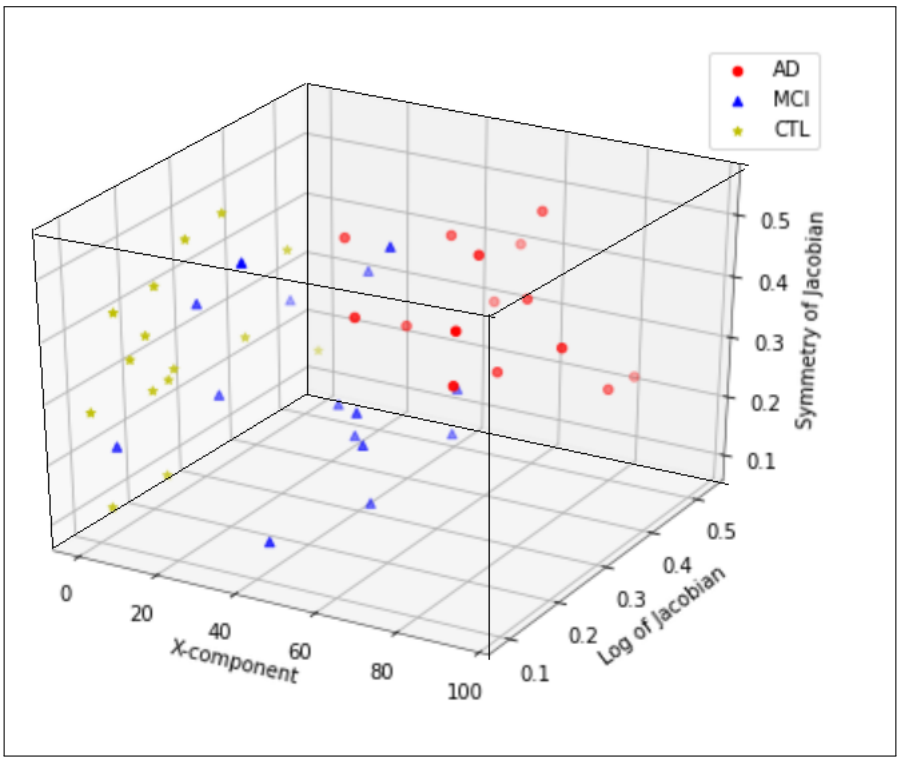
(a) The training set is separated completely by a set of discriminative triplet features for AD and normal control



(b) The training set is separated by a set of discriminative triplet features for AD and MCI



(a) The training set is separated by a set of discriminative triplet features for MCI and CTL



(b) When project the 26 MCI subjects into this AD-CTL discriminative feature subspace they are spread out between and around AD and control subjects

Therefore, from the above figures, we observe that the AD and CTL data is clearly separable and the MCI data is spread out uniformly between these data points.

Similarly, we can observe that the data corresponding to each of the classes is separable across all the data splits. It is also observed that the data corresponding to the dimensions and classes is distributed uniformly. The 1000 dimensionality of the data is another aspect to be considered as although the number of subjects is comparatively low, the features are prominent enough to get accurate results using a machine learning algorithm. In conclusion, the data dimensionality, less complexity of the classification models and the separability between the features and classes emphasize that the dataset is statistically significant to be used for the problem statement.

5 Method

5.1 Proposed method

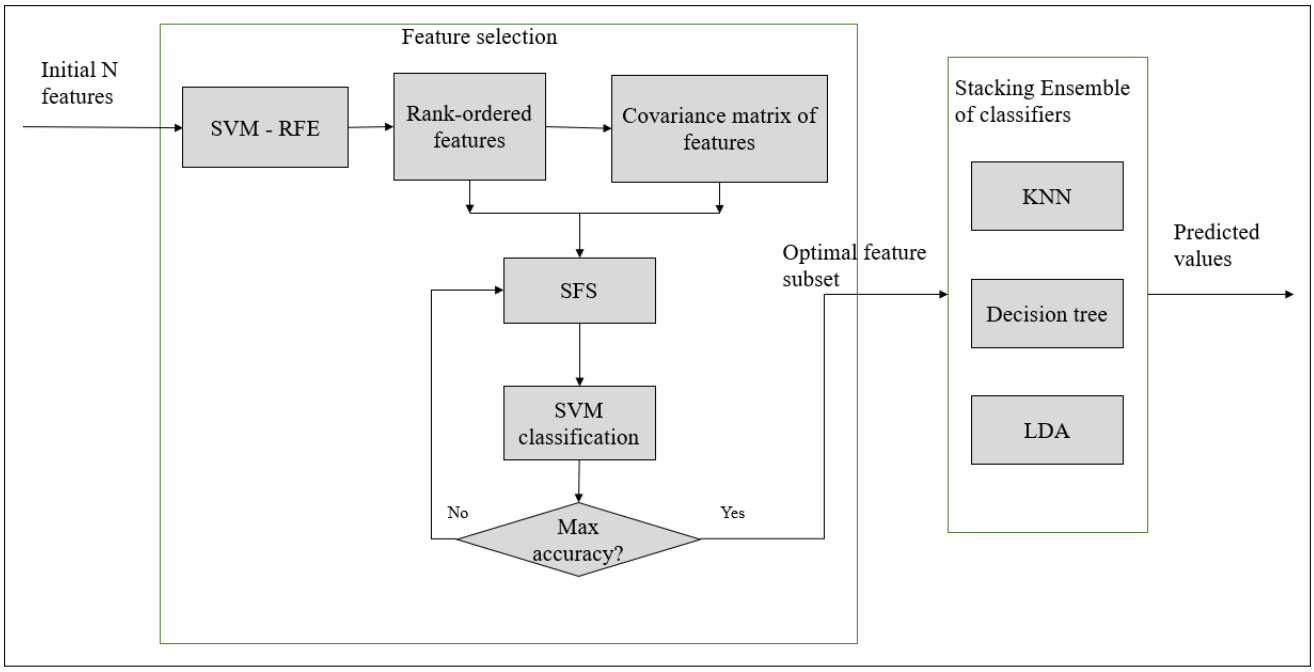


Figure 1: Proposed model architecture

The model proposed for achieving the Alzheimer's disease classification consists of passing the dataset containing the already extracted features to a feature selection module to obtain an optimal subset of features. This feature subset is then sent as input to a stacking ensemble of classifiers to get the predicted class values.

I am using the SVM - RFE (Reverse feature elimination) algorithm [4] to get the initial subset of features. The weights corresponding to the features are calculated according to parameters of the hyperplane obtained using the SVM classifier and then RFE removes

the features with less scores from the subset. These are combined with the feature's respective covariance matrix which is in turn passed to the Sequential Forward Selection algorithm to get the final optimal set of features. The accuracy from the SVM classifier is used as a metric to distinguish the feature subsets with an appropriate stopping criteria.

The KNN, Decision tree and LDA classifiers are used to create an ensemble using the stacking principle to get the appropriate predicted class values. These predicted values are compared with the originial class labels to get a measure of the performance of the models. The parameters of the constituent classifiers will be tuned appropriately to increase the model's accuracy.

5.2 Alternative method

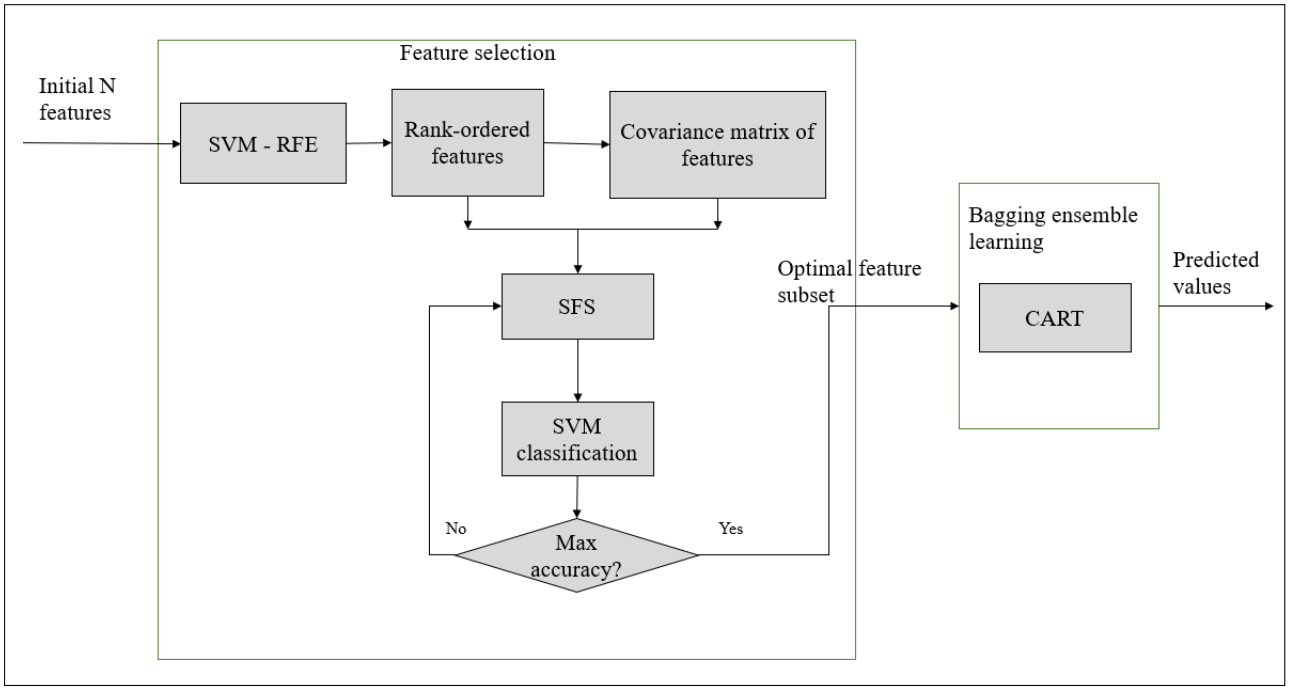


Figure 2: Alternative model architecture

The alternative model uses the same feature selection algorithm [4] to get the optimal subset of features. However, instead of using the stacking ensemble of classifiers as proposed above, I will use the CART based bagging ensemble classifier to perform the classification task. The CART classifier is used as an alternative if the proposed stacking ensemble of classifiers do not give satisfactory results. Additionally, I will experiment with various combinations of linear classifiers and will tune their parameters appropriately to observe the model's performance.

6 Steps to achieve the goals

Step 1 : Preprocess the extracted data features according to the respective cross validation splits.

Estimate time : 2 days

Step 2 : Implement the SVM - RFE algorithm for feature selection.
Estimate time : 2 days

Step 3: Implement Sequential forward selection on obtained initial set of features and integrate them with each feature's covariance matrix.
Estimate time : 4 days

Step 4: Tune parameters of SVM classifier in feature selection module to obtain maximum accuracy resulting in high discriminative features.
Estimate time : 2 days

Step 5: Explore and implement stacking ensemble of classifiers.
Estimate time : 8 days

Step 6: Analyze results and tune the classifier parameters to obtain maximum sensitivity and specificity scores.
Estimate time : 4 days

7 Final results quantitative validation method

The accuracy, sensitivity and specificity metrics are used to quantify the performance of the proposed model.

1. Accuracy = $\frac{TP+TN}{TP+TN+FP+FN}$
2. Sensitivity = $\frac{TP}{TP+FN}$
3. Specificity = $\frac{TN}{TN+FP}$

where TP, TN, FP and FN denote the true positives, true negatives, false positives and false negatives respectively as in [8] :

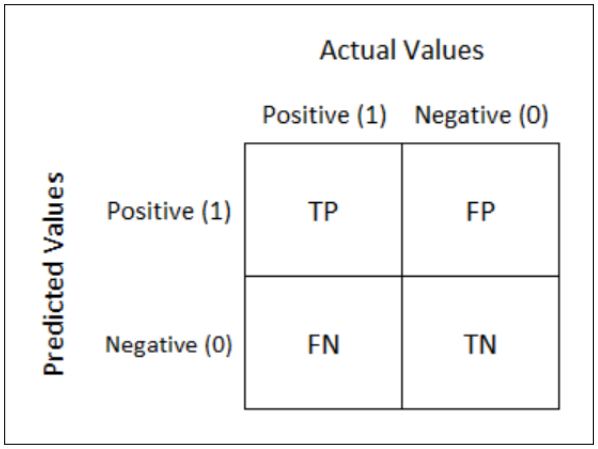


Figure 3: Confusion matrix structure

Sensitivity and specificity have been observed to be widely used statistics in the clinical domain and especially when a binary classification problem is considered. Also, most of the papers dealing with the Alzheimer's have been using these metrics to distinguish the base models. It is for these reasons these metrics are considered in addition to the accuracy metric.

8 Approach

8.1 Dataset preprocessing and analysis

The dataset consisted of 40 splits from leave-ten-out cross validation. This data was further cleaned to prune the data containing the label 4 from the splits. Additional preprocessing was performed to prune the subjects not associated with the corresponding binary classification problem.

For example, assume we are dealing with the AD-CTL classification problem, then it is required to remove the data associated with the label MCI from all the corresponding splits. A similar approach is carried out for the remaining AD-MCI and AD-CTL classification problems.

8.2 Feature selection

The multi-feature fusion approach as described in figure 1 is implemented to select the top 800 discriminative features from the total feature set. The accuracy obtained using the SVM classifier with a linear kernel is used as a metric to determine whether a feature can be added to the optimal feature subset or not. The recursive feature elimination is first implemented to get a preliminary subset of features which are later appended to the co-variance matrix of the features. This subset is further passed to the sequential forward selection approach to determine the final highly discriminative set of features.

8.3 Classification models

The classifiers used on the extracted subset of features from the multi-feature fusion approach are:

1. Linear Discriminant Analysis

2. Logistic Regression
   - Number of epochs : 40
   - Loss function : log loss
   - Learning rate : 1e - 4

3. Support Vector Machine
   - Kernel : linear

4. K Nearest Neighbors
   - K value : 28
   - The highest value of K which can be chosen is 28. This is because the maximum number of samples in each split for each class is 30. Choosing K values greater than 28 is overfitting the model and a steep decrease in the test accuracies is observed as shown in Figure 4 :

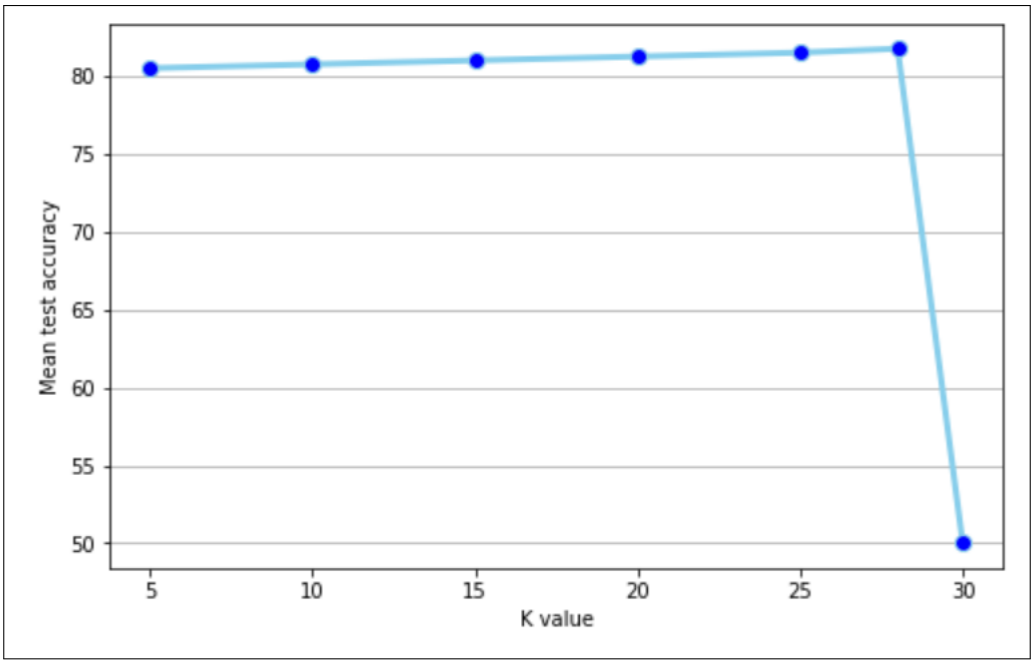


Figure 4: Plot of mean test accuracy of AD-CTL data observed for different K values of the KNN classifier. Highest accuracy is observed for K = 28.

5. Decision tree
   - Depth of tree : 5

6. Stacking ensemble of classifiers
   - Classifiers included in the ensemble : LDA, Decision tree, KNN

7. Random forest
8. Bagging ensemble of LDA classifiers
9. Naive Bayes

9 Results

9.1 Top features extracted after feature selection

The top 800 discriminative features are obtained after the multi-feature fusion feature selection approach for each of the cross validation splits. The top 50 features out of these discriminative features ranked according to their feature importance on the split 2 data is as follows:

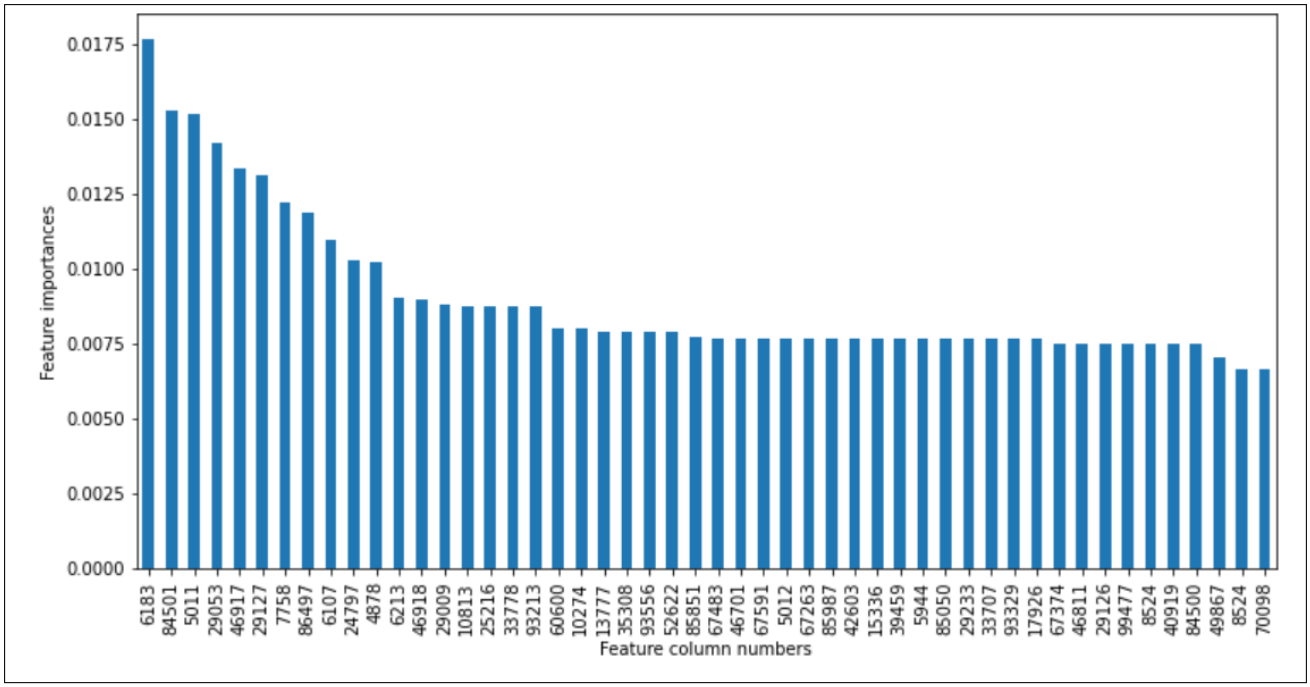
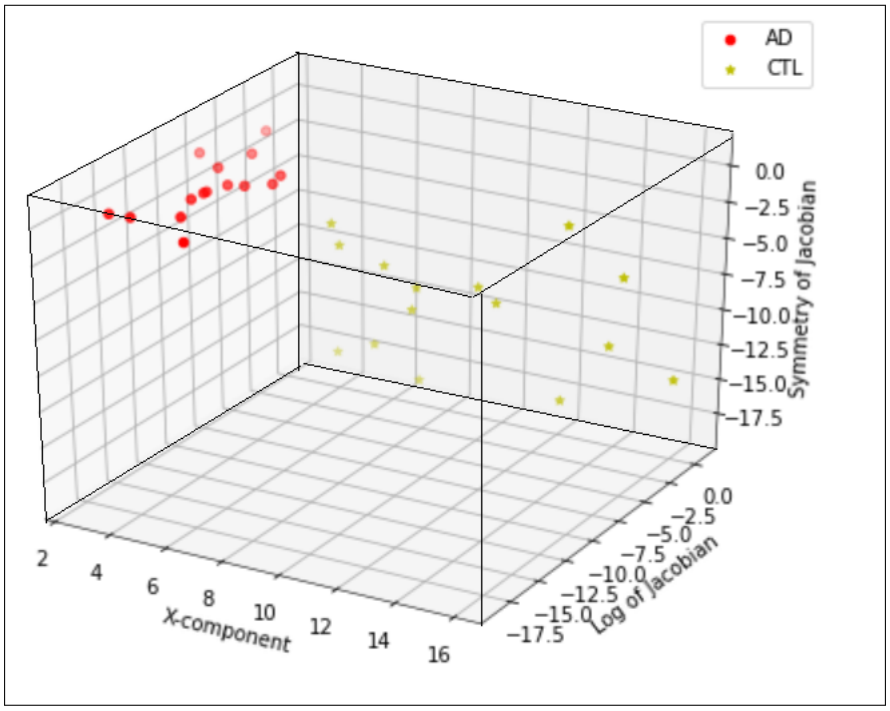


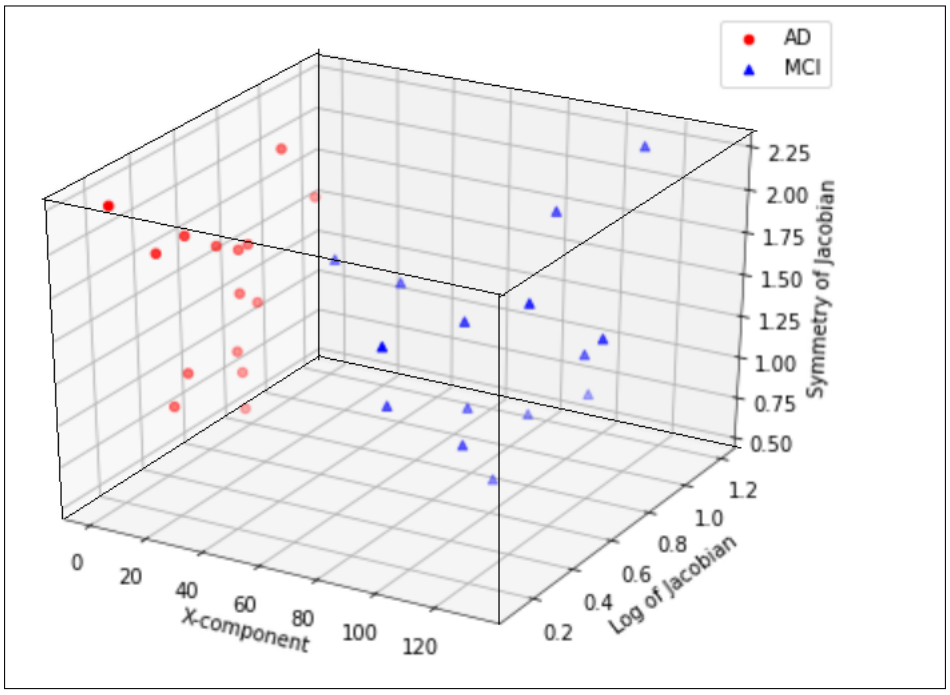
Figure 5: Top 50 discriminative features obtained after feature selection ranked according to their feature importance

From Figure 5, the features corresponding to the log of jacobian and symmetry of jacobian feature types are highly ranked towards the left of the graph. These features are having importance in the range of 0.0175. On the other hand, the features corresponding to the X component and the Y component are ranked relatively less than the jacobian feature types. The feature importance for the X, Y and Z components are around the range 0.0050 from the graph.

Considering three of these highly discriminative features, having the X component, log of jacobian and the symmetry of jacobian feature types as the axes, the visualization of the AD-CTL, AD-MCI and MCI-CTL data is:



(a) The training set is separated completely by the top set of features for AD and normal control



(b) The training set is separated by the top set of features for AD and MCI

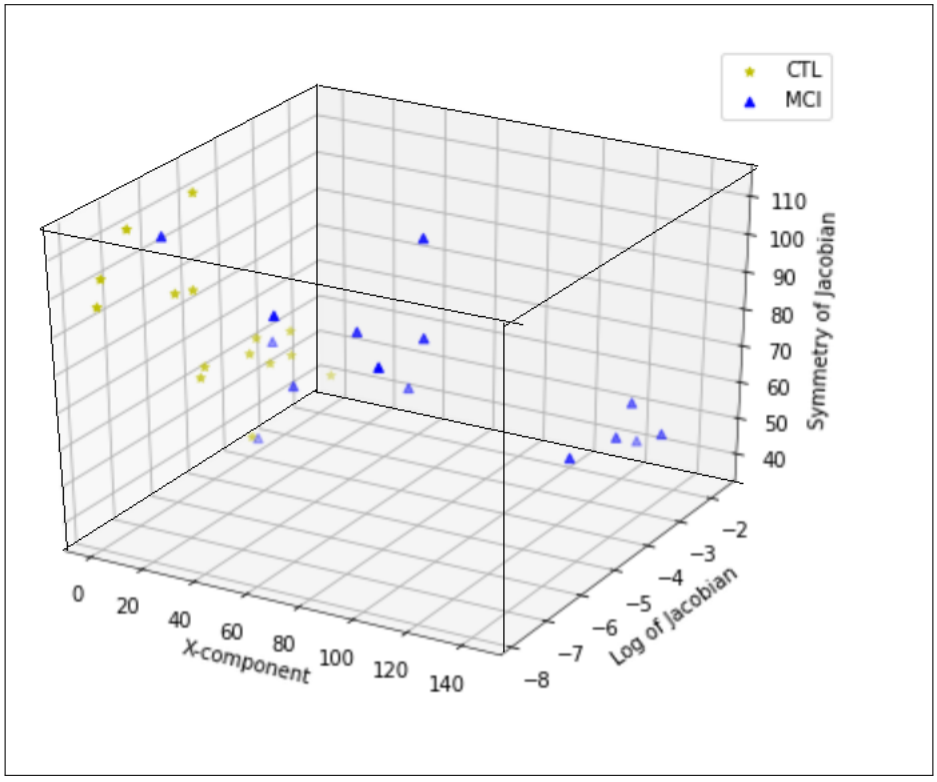


Figure 6: The training set is separated by the top set of features for MCI and CTL

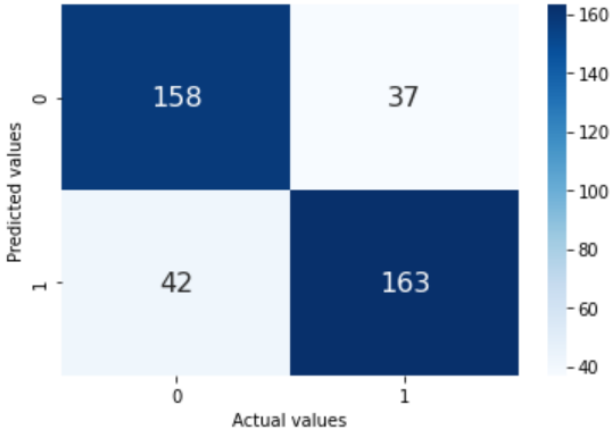
The AD-CTL data is completely separable when plotted onto the 3D space using the top features obtained after feature selection as the axes. The MCI data is however not that separable compared to these data points in this case as well.

9.2 Performance results of each classifier

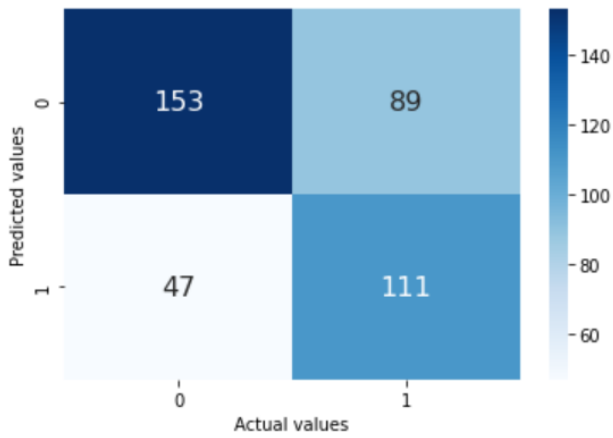
To evaluate the performance of each classifier, the confusion matrices across all the splits and the visualization of the classification of the training and test data across a particular split are reported. The confusion matrices denote the true positive, true negative, false positive and false negative values for the data belonging to all the 40 splits. A high number of true positives and true negatives is desired when dealing with clinical data as one would not want to tell a patient that he has a disease even though he is perfectly alright. A false positive denotes telling a patient that he is healthy even though he has a disease. A false negative denotes telling a patient that he has a disease even though he is healthy. The data split 2 is used to visualize the training and test samples of the AD-CTL data, split 5 is used to visualize the samples of the AD-MCI data and the split 3 is used to visualize the samples of the MCI-CTL data for each of the listed classifiers.

9.2.1 Linear Discriminant Analysis

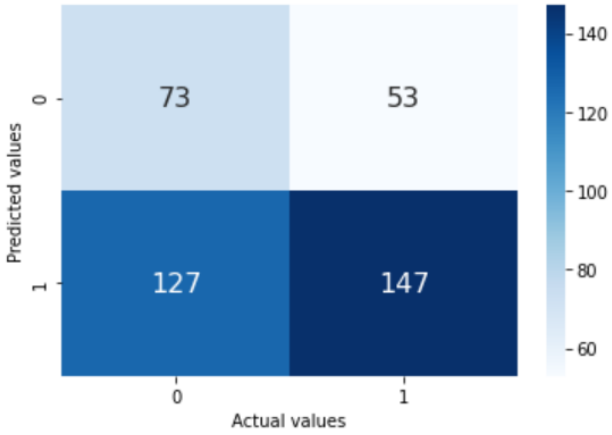
The confusion matrices obtained by training the LDA classifier on the AD-CTL, AD-MCI and MCI-CTL data are:



(a) AD-CTL



(b) AD-MCI

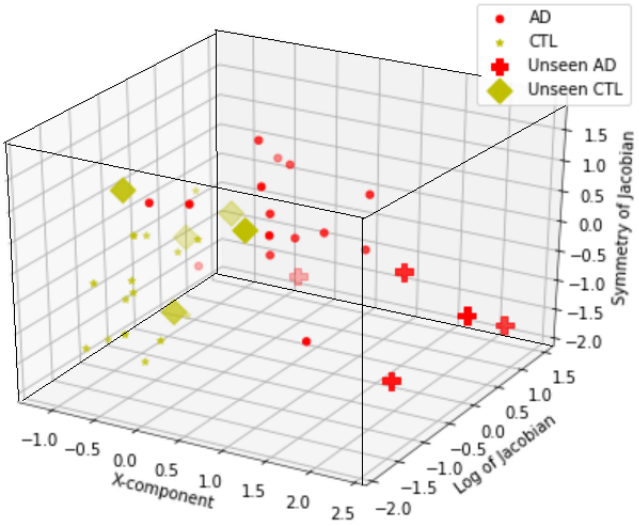


(c) MCI-CTL

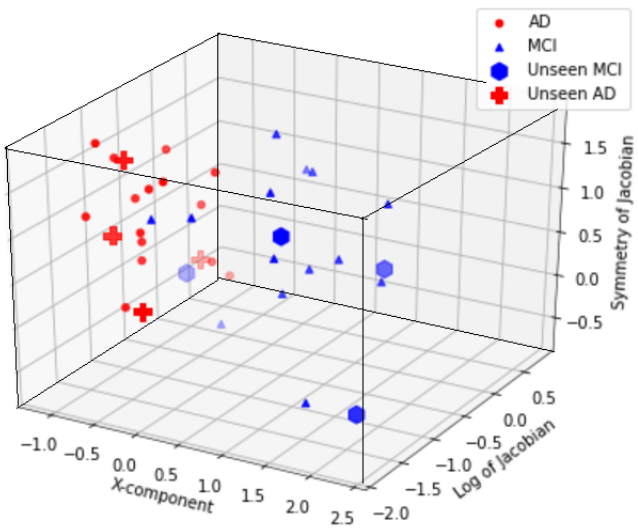
Figure 7: Confusion matrices of LDA across all splits of the AD-CTL, AD-MCI and MCI-CTL data

We can infer that the LDA classifier performs the best on all splits of AD-CTL data and it performs relatively poorly on the MCI-CTL data splits. However, considering the classifier on the whole, the number of true positives and true negatives is high which is desired when dealing with sensitive clinical information.

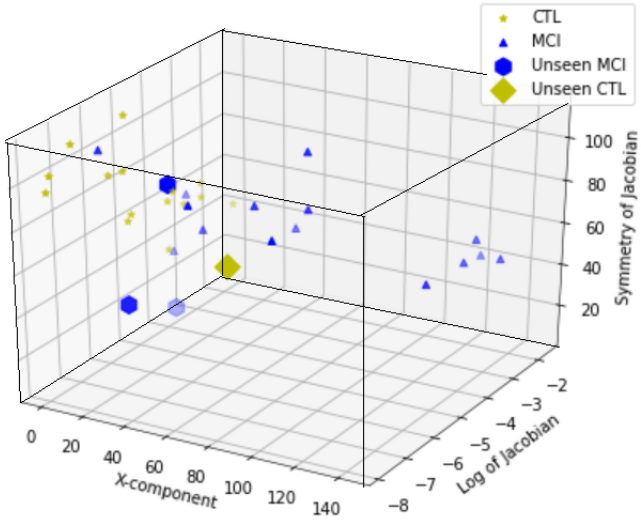
The test data points are classified into the following classes when the LDA classifier is used as follows:



(a) AD-CTL



(b) AD-MCI



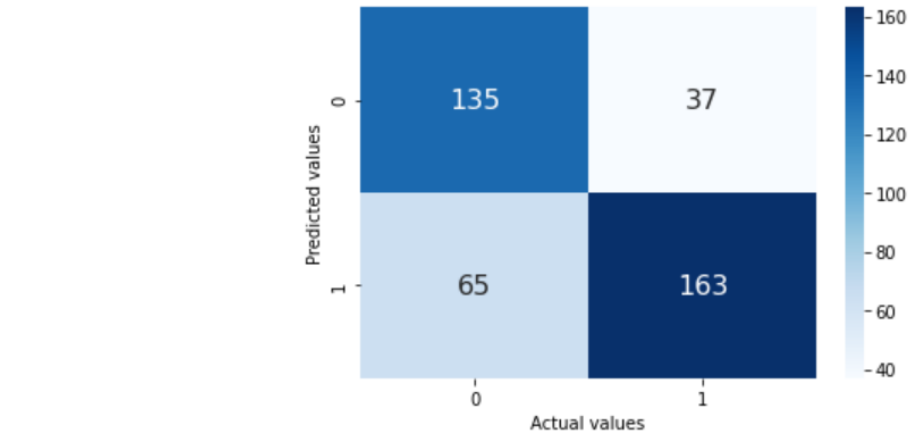
(c) MCI-CTL

Figure 8: Separation of the data points when the predicted test samples using the LDA classifier are plotted with respect to the training samples

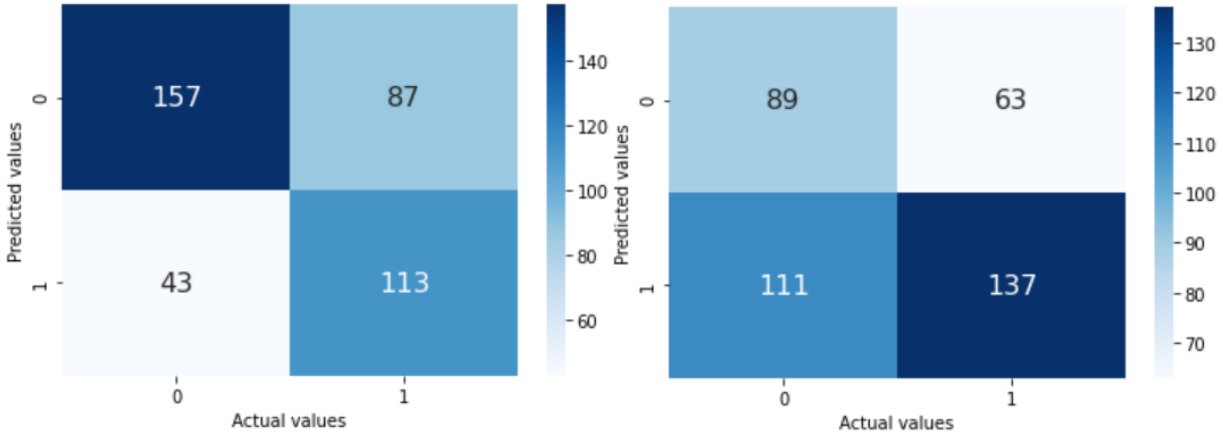
From Figure 8, two of the unseen CTL samples are incorrectly classified as the AD data points and three of the unseen MCI samples are incorrectly classified as the AD samples. The samples are comparatively more incorrectly classified for the MCI and CTL data and this explains the reduced performance of the LDA classifier on this specific dataset.

9.2.2 Logistic Regression

The confusion matrices obtained by training the logistic regression model on the AD-CTL, AD-MCI and MCI-CTL data are:



(a) AD-CTL

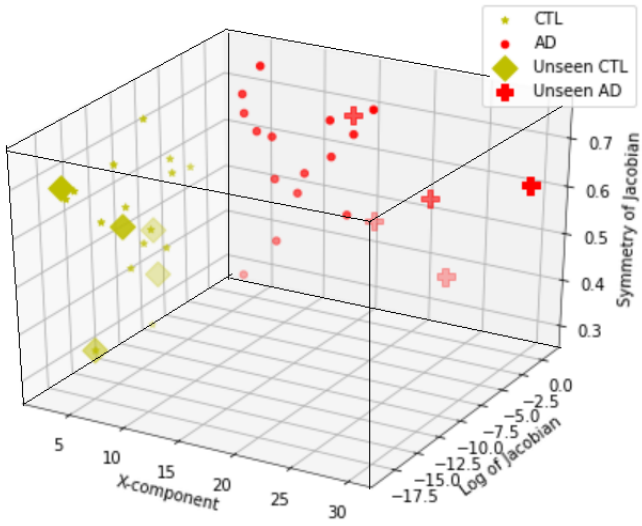


(b) AD-MCI (c) MCI-CTL

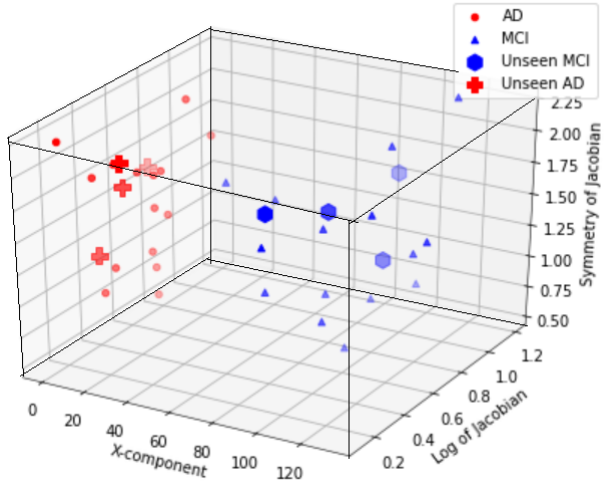
Figure 9: Confusion matrices of Logistic regression across all splits of the AD-CTL, AD-MCI and MCI-CTL data

The number of true positives and true negatives for each of these binary classification problems is high which is desired. The same trend of the performance decreasing when dealing with the MCI-CTL data is observed in this case as well.

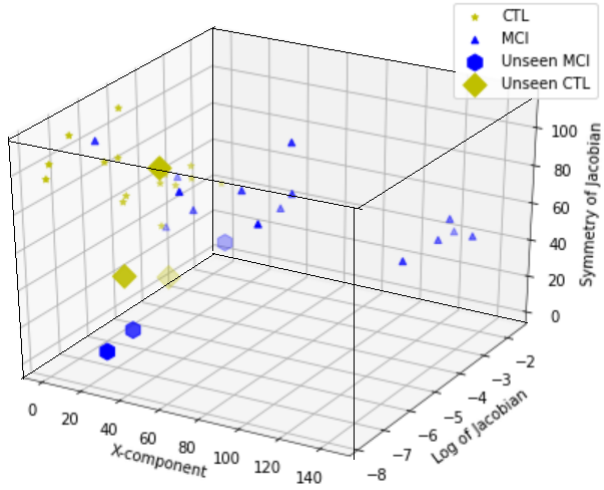
The test data points are classified into the following classes when the Logistic regression classifier is used as follows:



(a) AD-CTL



(b) AD-MCI



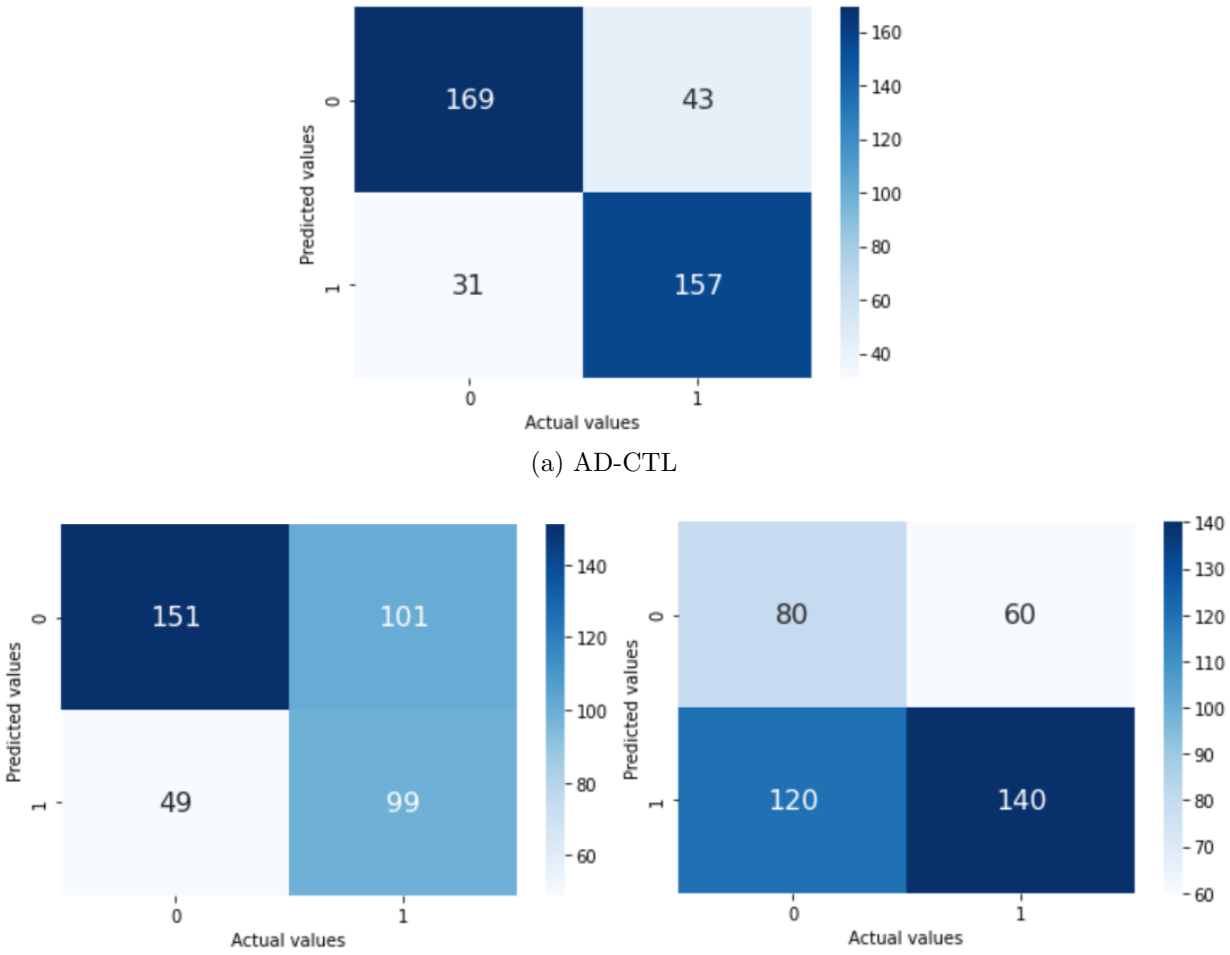
(c) MCI-CTL

Figure 10: Separation of the data points when the predicted test samples using the Logistic regression classifier are plotted with respect to the training samples

From Figure 10, all the unseen points in the AD-CTL and AD-MCI classes are correctly classified. Three of the unseen points belonging to the MCI class are incorrectly classified as part of the CTL class.

9.2.3 Support Vector Machine

The confusion matrices obtained by training the SVM classifier on the AD-CTL, AD-MCI and MCI-CTL data are:



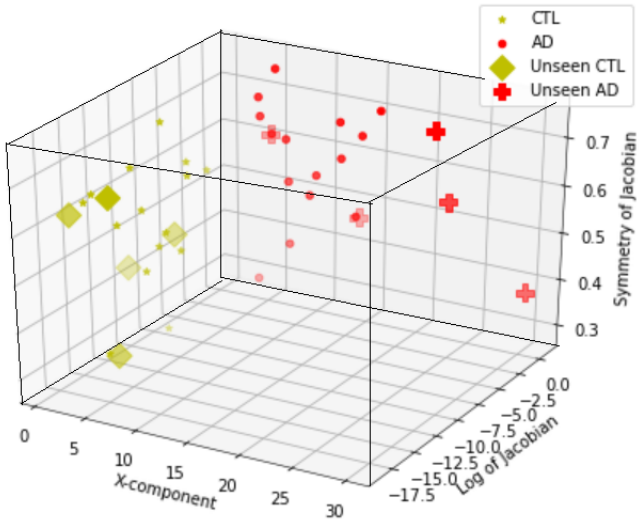
(a) AD-CTL

(b) AD-MCI

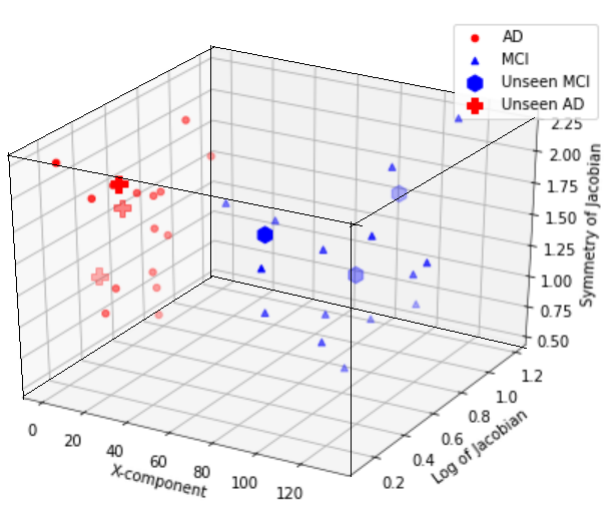
(c) MCI-CTL

Figure 11: Confusion matrices of SVM across all splits of the AD-CTL, AD-MCI and MCI-CTL data

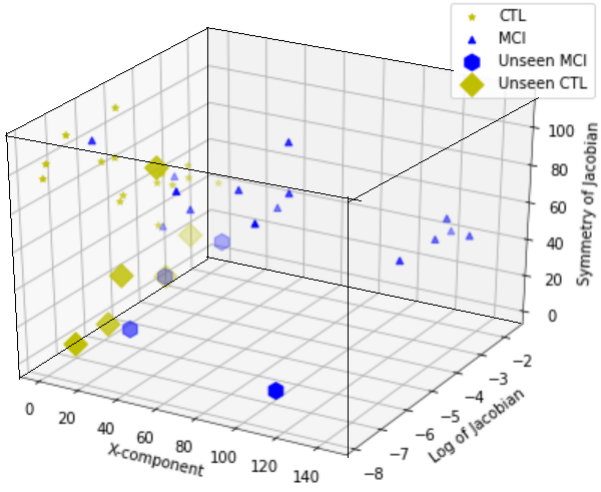
The test data points are classified into the following classes when the SVM classifier is used as follows:



(a) AD-CTL



(b) AD-MCI



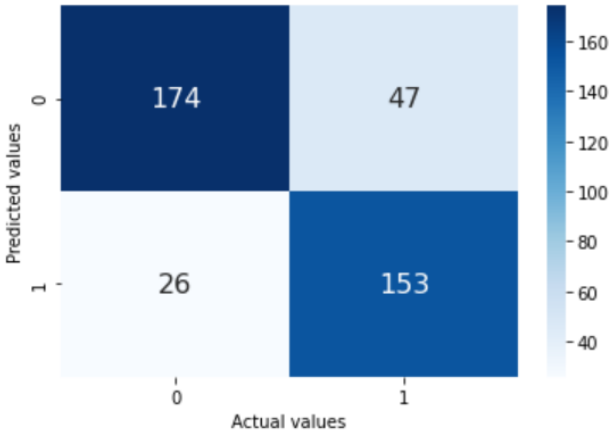
(c) MCI-CTL

Figure 12: Separation of the data points when the predicted test samples using the SVM classifier are plotted with respect to the training samples

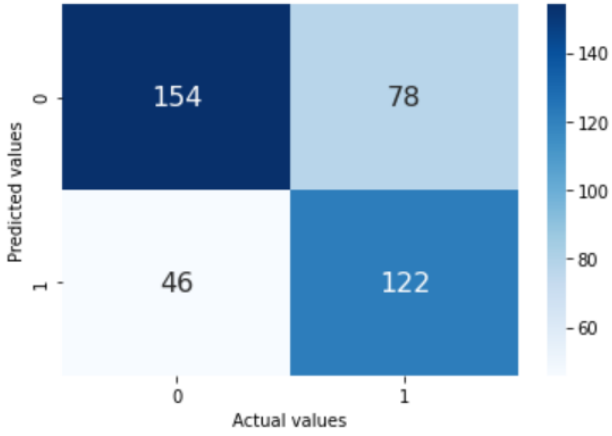
From Figure 12, the decision boundary precisely separates the AD and CTL data points. The boundary, denoted by the planar surface, however leaves a few points incorrectly divided into the respective AD and MCI classes. This separation is comparatively further decreased with the MCI and CTL data samples. This explains the confusion matrices above, where the number of true positives and true negatives is much higher for the AD-CTL data, followed by the AD-MCI and MCI-CTL data respectively.

9.2.4 K Nearest Neighbors

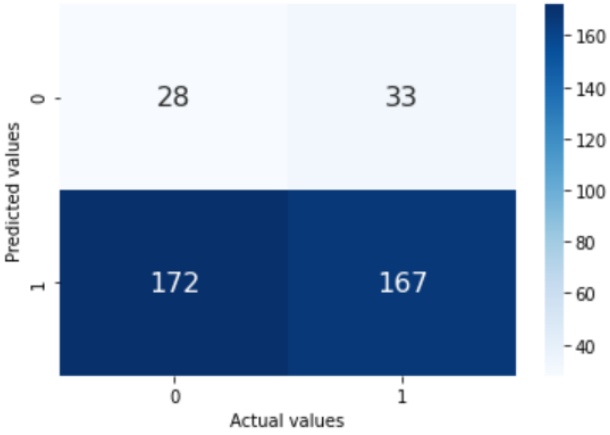
The confusion matrices obtained by training the KNN classifier on the AD-CTL, AD-MCI and MCI-CTL data are:



(a) AD-CTL



(b) AD-MCI

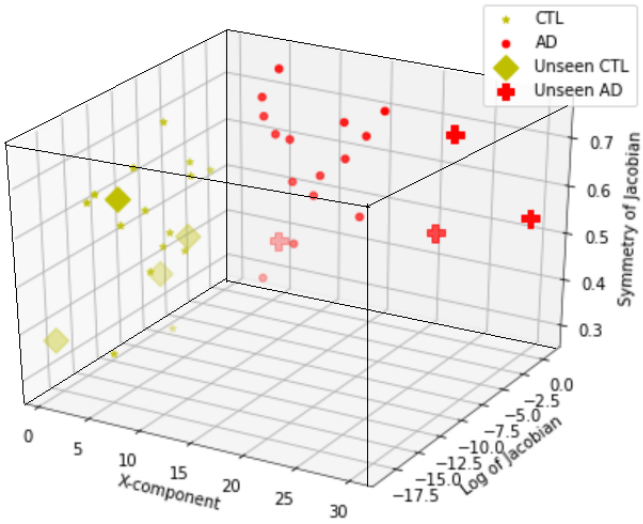


(c) MCI-CTL

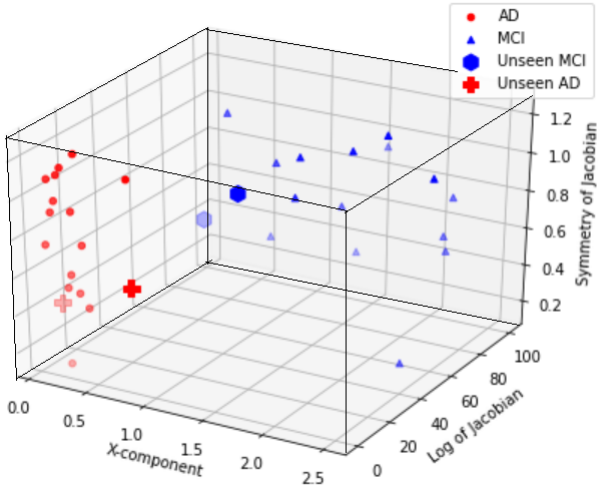
Figure 13: Confusion matrices of KNN across all splits of the AD-CTL, AD-MCI and MCI-CTL data

We can infer that the KNN classifier performs well on the data belonging to the AD-CTL classification problem and the least on the MCI-CTL classification problem. The value of K chosen here is 28. Since the number of samples in the AD-CTL data splits is relatively more than the number of data samples in the MCI-CTL data splits, the high value of K seems to be overfitting the MCI-CTL data.

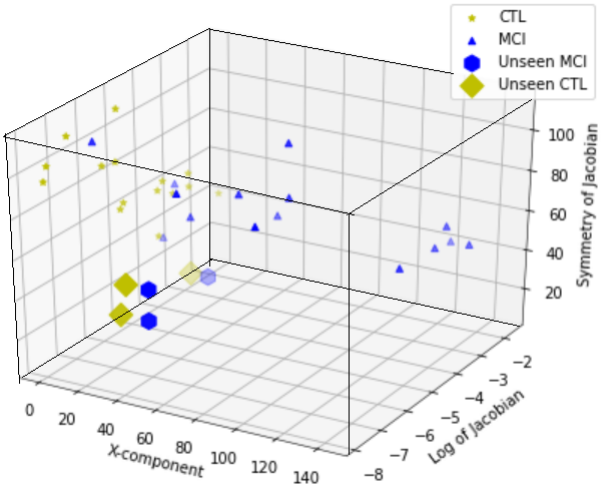
The test data points are classified into the following classes when the KNN classifier is used as follows:



(a) AD-CTL



(b) AD-MCI



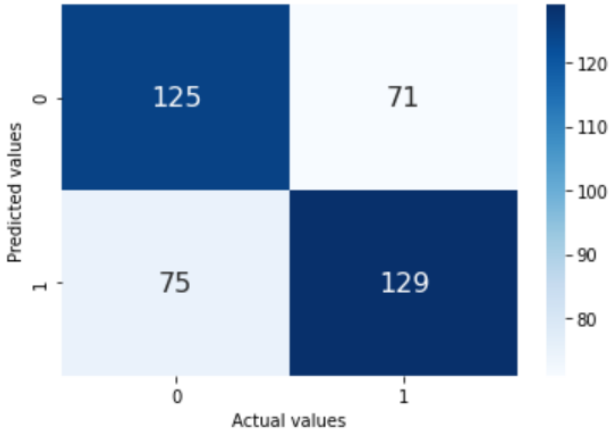
(c) MCI-CTL

Figure 14: Separation of the data points when the predicted test samples using the KNN classifier are plotted with respect to the training samples

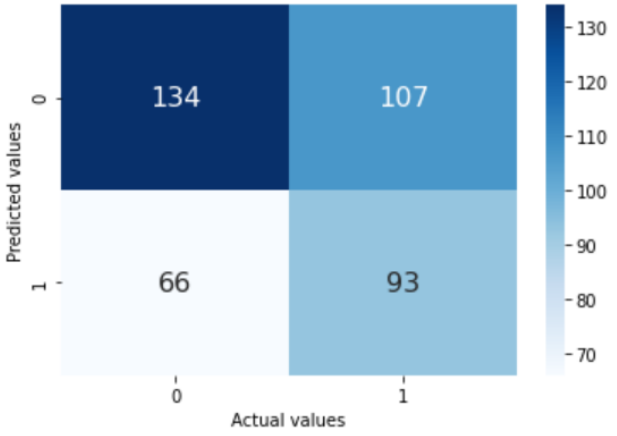
From Figure 14, all the unseen points in the AD-CTL classes are correctly classified. One of the MCI unseen samples is incorrectly classified as the AD sample. Five of the unseen points belonging to the MCI class are incorrectly classified as part of the CTL class.

9.2.5 Decision tree

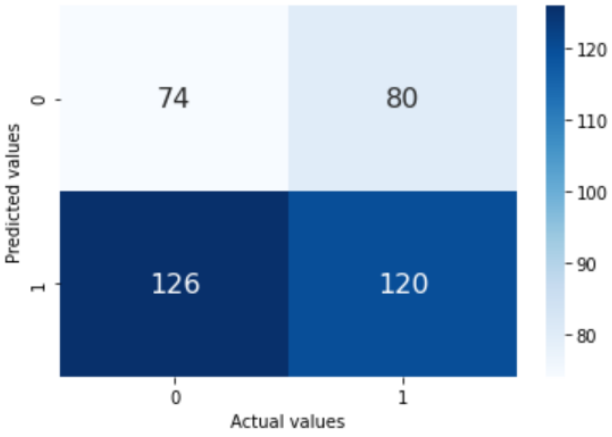
The confusion matrices obtained by training the Decision tree classifier on the AD-CTL, AD-MCI and MCI-CTL data are:



(a) AD-CTL



(b) AD-MCI

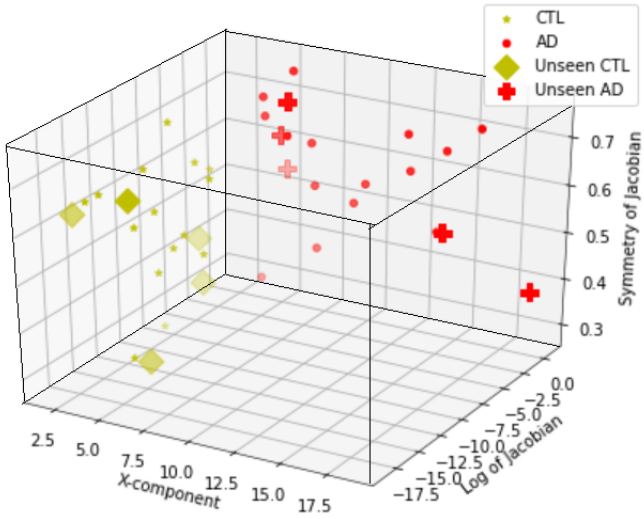


(c) MCI-CTL

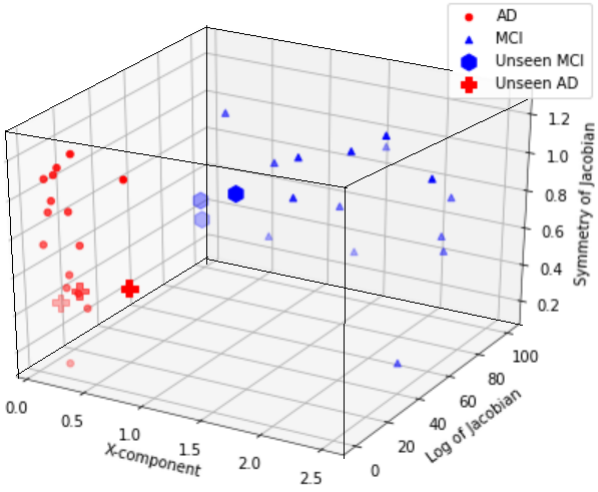
Figure 15: Confusion matrices of Decision tree across all splits of the AD-CTL, AD-MCI and MCI-CTL data

From the above matrices, we infer that the decision tree does not perform satisfactorily well compared to the other models on all the data belonging to the splits of the AD-CTL, AD-MCI and MCI-CTL classification problems.

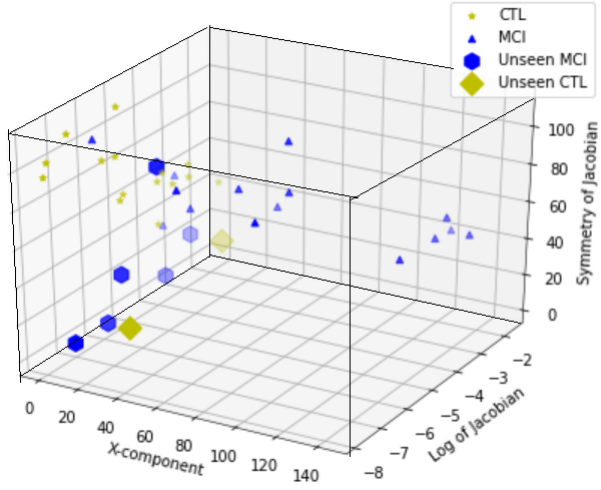
The test data points are classified into the following classes when the Decision tree is used as follows:



(a) AD-CTL



(b) AD-MCI



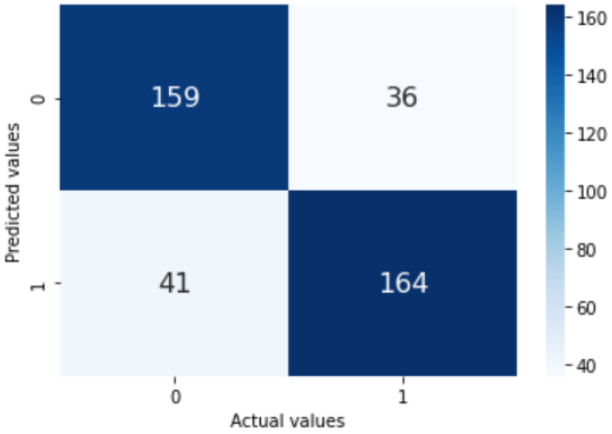
(c) MCI-CTL

Figure 16: Separation of the data points when the predicted test samples using the Decision tree classifier are plotted with respect to the training samples

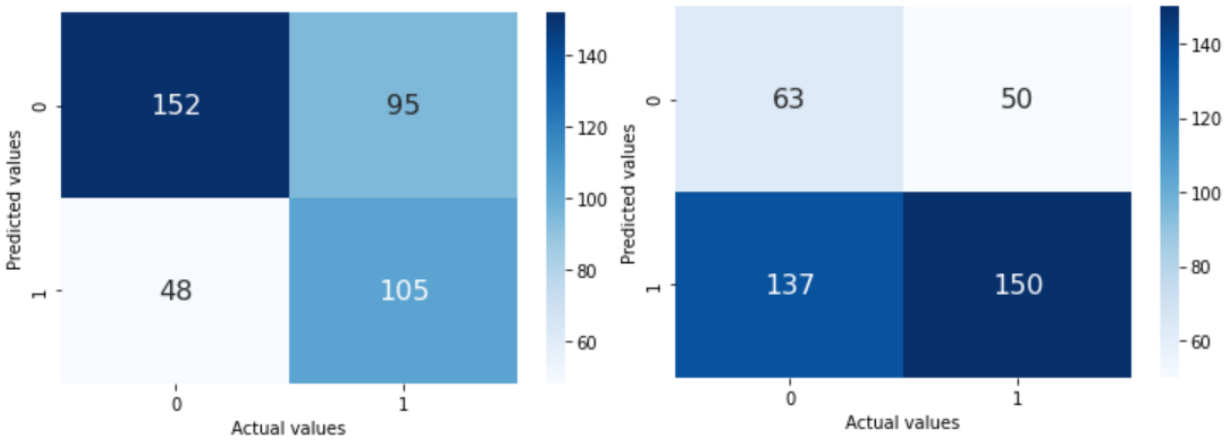
From Figure 16, all the unseen points in the AD-CTL classes are correctly classified. Two of the MCI unseen samples is incorrectly classified as the AD sample. Several unseen points belonging to the MCI class are incorrectly classified as part of the CTL class. This explains the reduced number of false positives and false negatives in the confusion matrices above.

9.2.6 Stacking ensemble of models

The confusion matrices obtained by training the stacking ensemble of classifiers on the AD-CTL, AD-MCI and MCI-CTL data are:



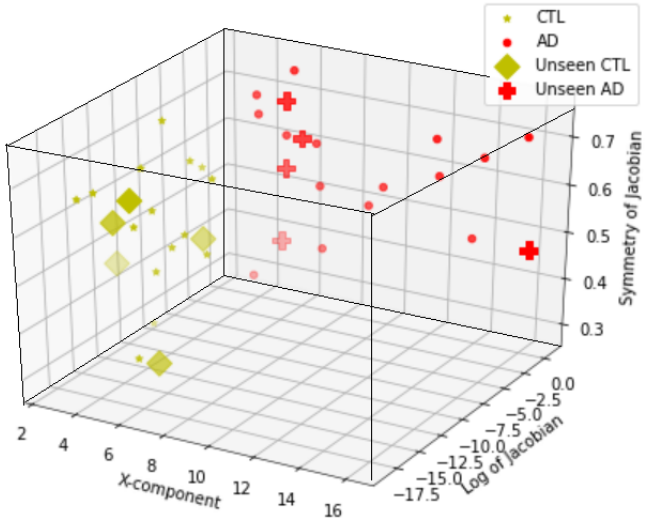
(a) AD-CTL



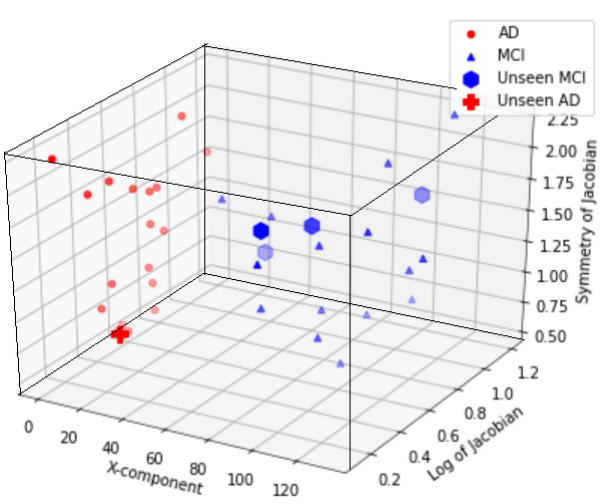
(b) AD-MCI (c) MCI-CTL

Figure 17: Confusion matrices of Stacking ensemble of classifiers across all splits of the AD-CTL, AD-MCI and MCI-CTL data

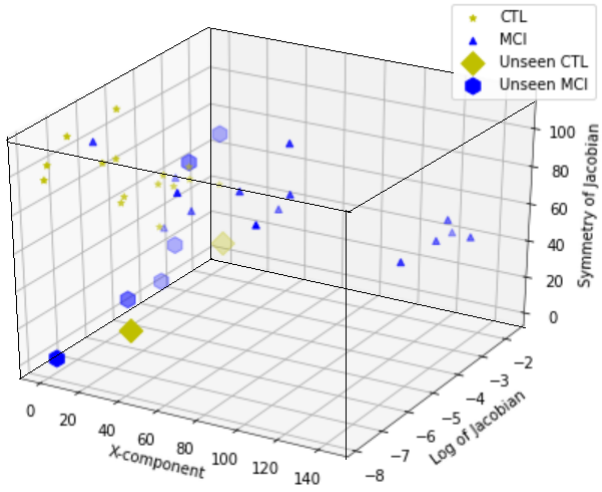
The test data points are classified into the following classes when the Stacking ensemble of classifiers is used as follows:



(a) AD-CTL



(b) AD-MCI



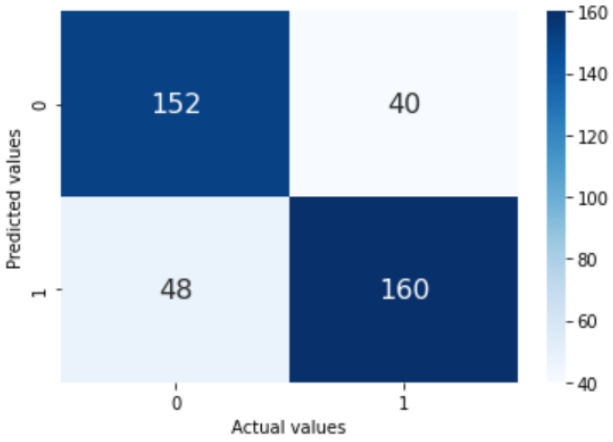
(c) MCI-CTL

Figure 18: Separation of the data points when the predicted test samples using the stacking ensemble of classifiers are plotted with respect to the training samples

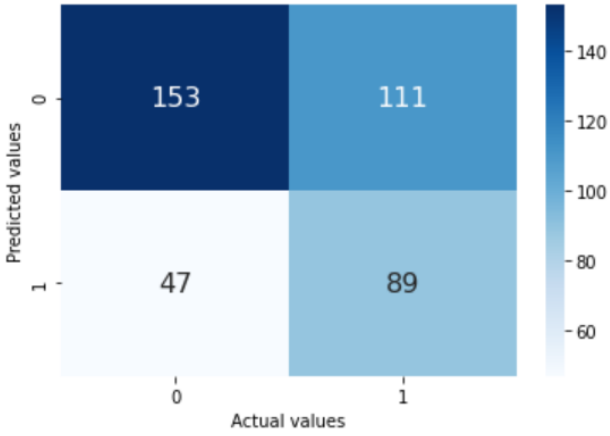
From Figure 18, all the unseen points in the AD-CTL and AD-MCI classes are correctly classified. Several unseen points belonging to the MCI class are incorrectly classified as part of the CTL class. This explains the reduced number of false positives and false negatives in the confusion matrices above.

9.2.7 Random forest

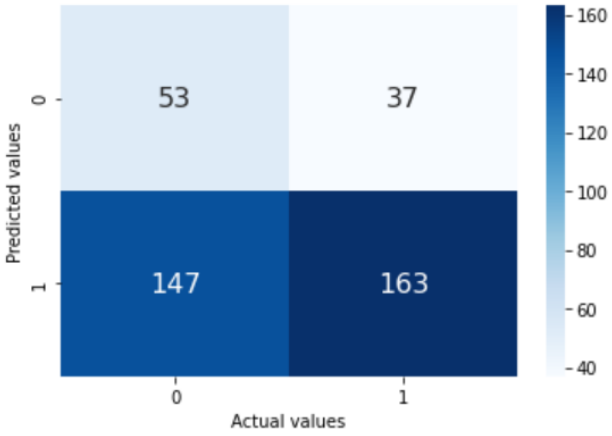
The confusion matrices obtained by training the random forest on the AD-CTL, AD-MCI and MCI-CTL data are:



(a) AD-CTL



(b) AD-MCI

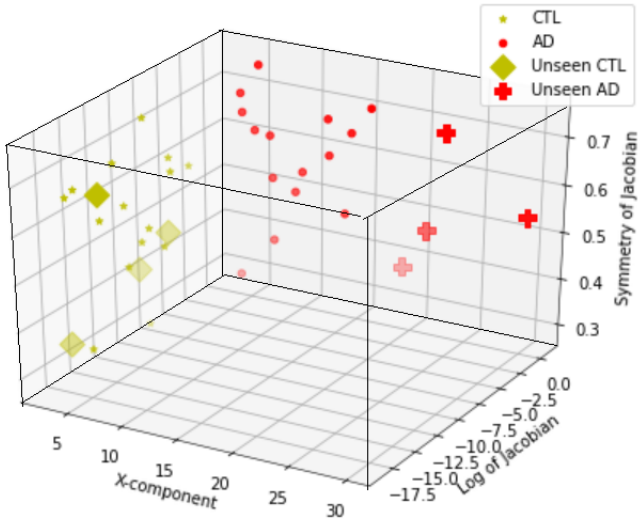


(c) MCI-CTL

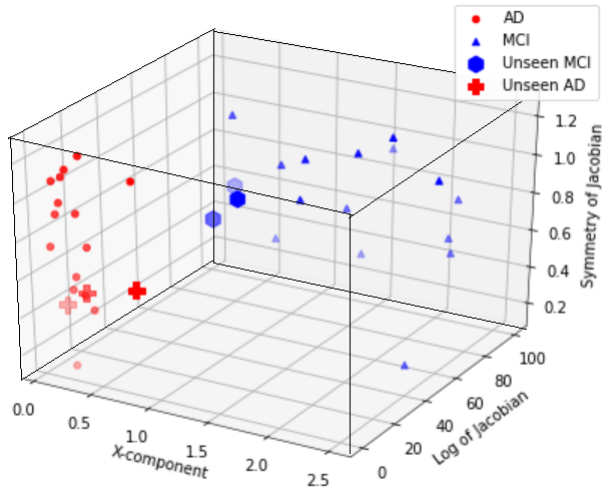
Figure 19: Confusion matrices of random forest across all splits of the AD-CTL, AD-MCI and MCI-CTL data

The number of true positives and true negatives is high in the AD-CTL data while these values gradually decrease as the random forest classifier is applied on the MCI-CTL test data points. The high number of false negatives in the MCI-CTL classification problem suggest that the patients are assumed to be mild cognitive impaired even though they are normal control patients.

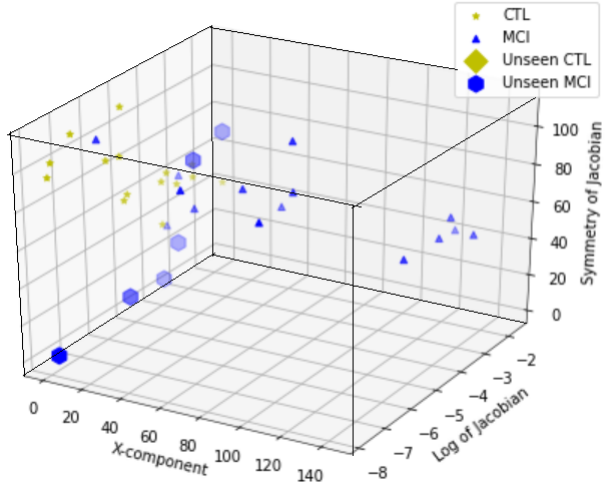
The test data points are classified into the following classes when the random forest classifier is used as follows:



(a) AD-CTL



(b) AD-MCI



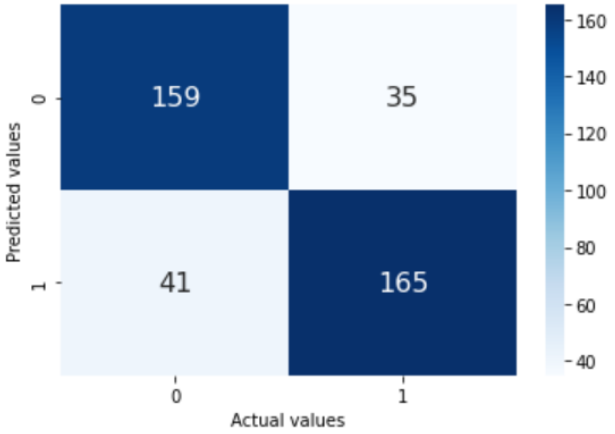
(c) MCI-CTL

Figure 20: Separation of the data points when the predicted test samples using the Random forest classifier are plotted with respect to the training samples

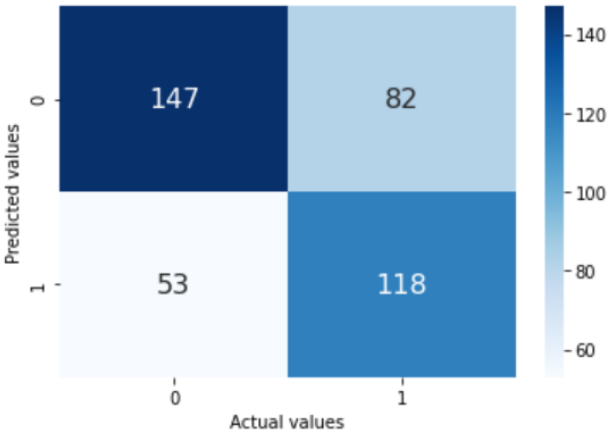
From Figure 20, all the unseen points in the AD-CTL are correctly classified. Several unseen points belonging to the MCI class are incorrectly classified as part of the CTL class. This explains the reduced number of false positives and false negatives in the confusion matrices above.

9.2.8 Bagging ensemble of classifiers

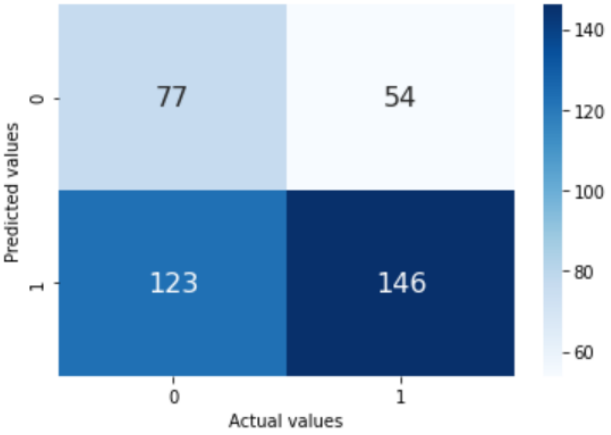
The confusion matrices obtained by training the bagging ensemble of classifiers on the AD-CTL, AD-MCI and MCI-CTL data are:



(a) AD-CTL



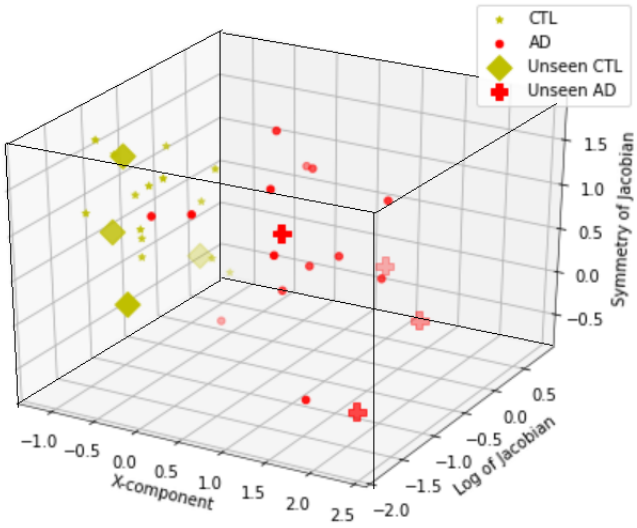
(b) AD-MCI



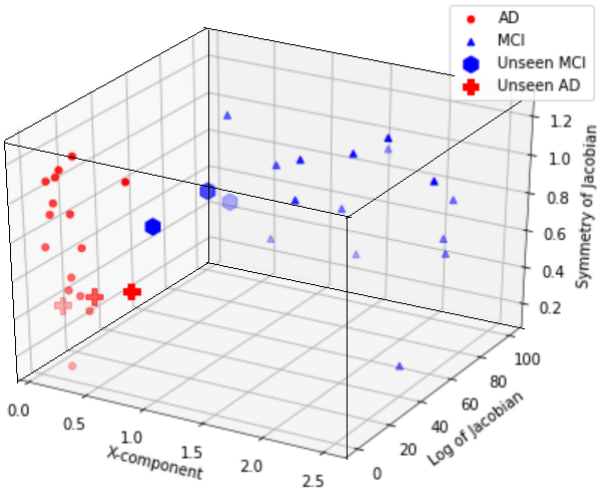
(c) MCI-CTL

Figure 21: Confusion matrices of bagging ensemble of classifiers across all splits of the AD-CTL, AD-MCI and MCI-CTL data

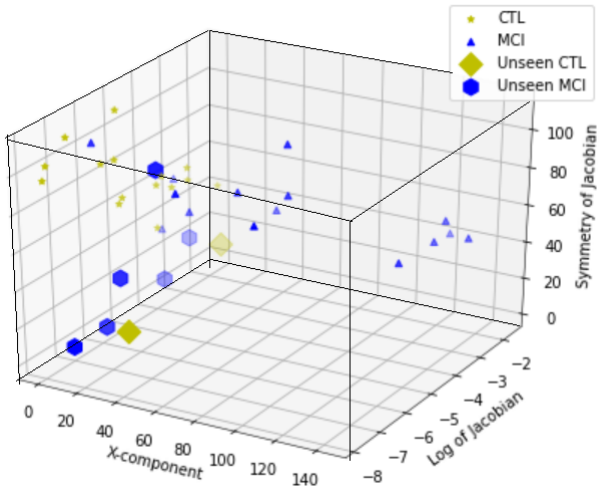
The test data points are classified into the following classes when the bagging ensemble of classifiers is used as follows:



(a) AD-CTL



(b) AD-MCI



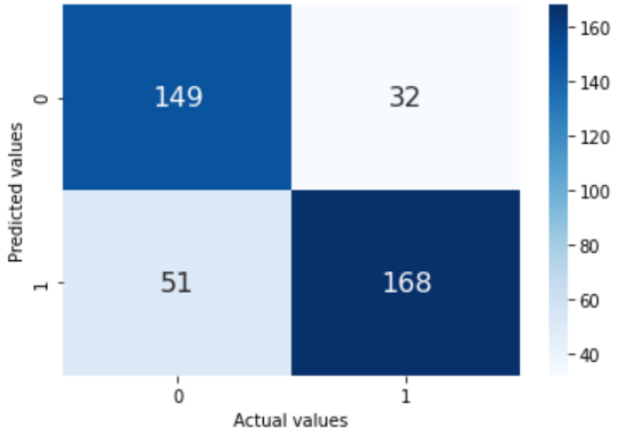
(c) MCI-CTL

Figure 22: Separation of the data points when the predicted test samples using the bagging ensemble of classifiers are plotted with respect to the training samples

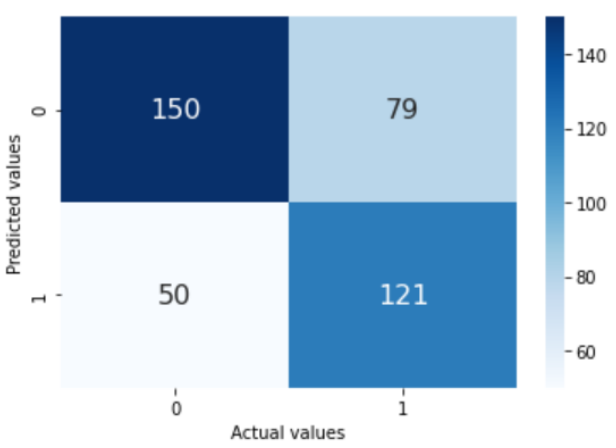
From Figure 22, a few of the AD unseen points are incorrectly classified as the CTL samples. The same is the case with the AD-MCI data where around two MCI samples are incorrectly classified. Several unseen points belonging to the MCI class are incorrectly classified as part of the CTL class. This explains the reduced number of false positives and false negatives in the confusion matrices above.

9.2.9 Naive Bayes

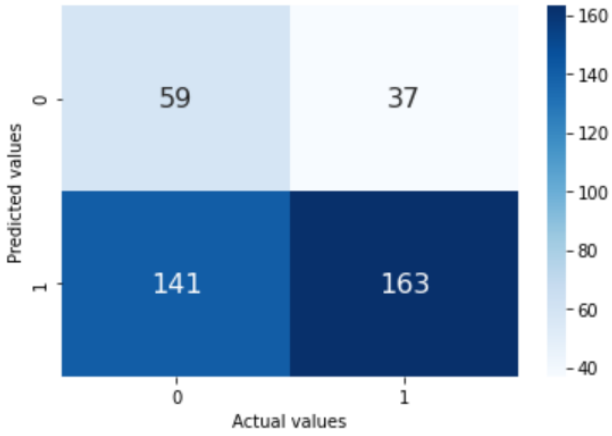
The confusion matrices obtained by training the Naive bayes classifier on the AD-CTL, AD-MCI and MCI-CTL data are:



(a) AD-CTL



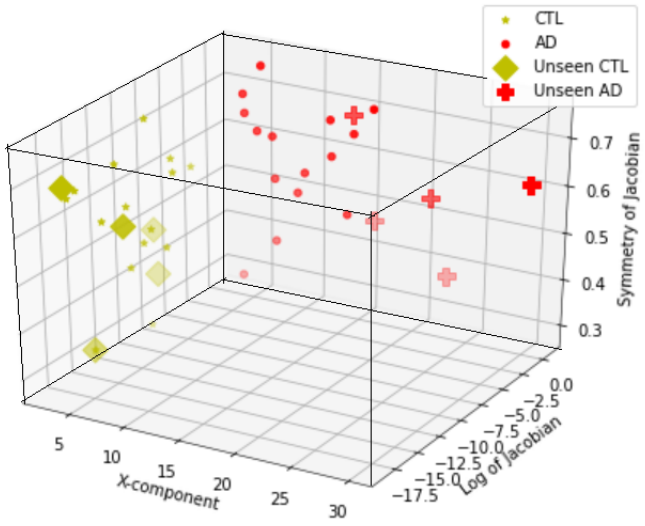
(b) AD-MCI



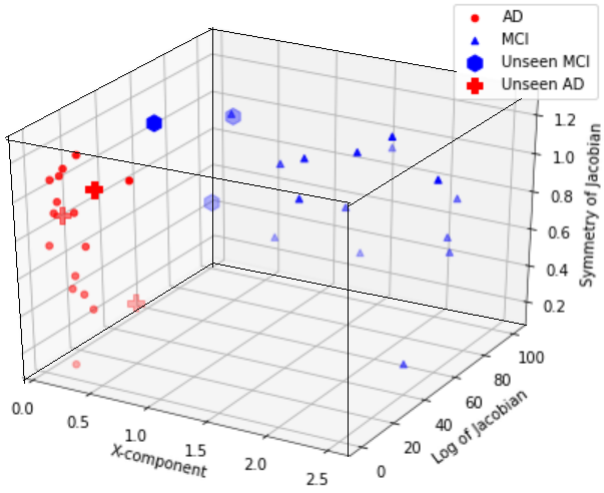
(c) MCI-CTL

Figure 23: Confusion matrices of naive bayes classifier across all splits of the AD-CTL, AD-MCI and MCI-CTL data

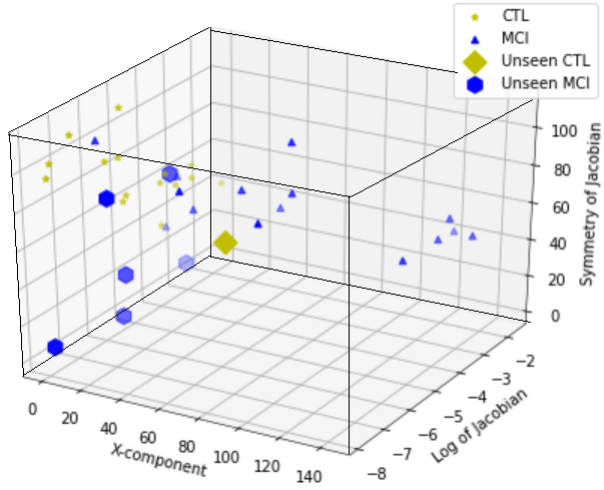
The test data points are classified into the following classes when the naive bayes classifier is used as follows:



(a) AD-CTL



(b) AD-MCI



(c) MCI-CTL

Figure 24: Separation of the data points when the predicted test samples using the Naive bayes classifier are plotted with respect to the training samples

From Figure 24, a few of the AD-MCI data are incorrectly classified. Several unseen points belonging to the MCI class are incorrectly classified as part of the CTL class. This explains the reduced number of false positives and false negatives in the confusion matrices above.

9.3 Overall performance results of classifiers on the dataset

The classifiers mentioned in Section 8.3 are trained on the highly discriminative features obtained after feature selection corresponding to the AD-CTL, AD-MCI and MCI-CTL data respectively. The metrics used to evaluate the performance of each of these models are the mean value of the test accuracy, the mean value of the test sensitivity, the mean

value of the test specificity, the mean value of the F1 scores and the mean absolute error across all the data splits. The standard deviation values for each of the accuracy, sensitivity and specificity metrics are reported as well.

For each binary classification problem, an analysis of the comparison of the performance of the above listed classifiers is reported along with the ROC curves for each classifiers. This will provide a deeper understanding of the performance of each classifer with respect to the others and will give a greater understanding of which classifier is more suitable for the problem statement.

ROC curves are an important measure for understanding the discriminative ability and for determining the sensitivity and specificity of a test. A highly sensitive test is useful for ruling out a disease if a patient has a negative result. On the other hand, a highly specific test rules out patients having positive results. A cutoff is then determined for each classifier based on the area under the ROC curve and depending on this threshold, the results can be divided into either of the classes.

9.3.1 AD - CTL classification

The plot of the test accuracies obtained by training and testing the respective classifiers across all the splits of the AD-CTL data is:

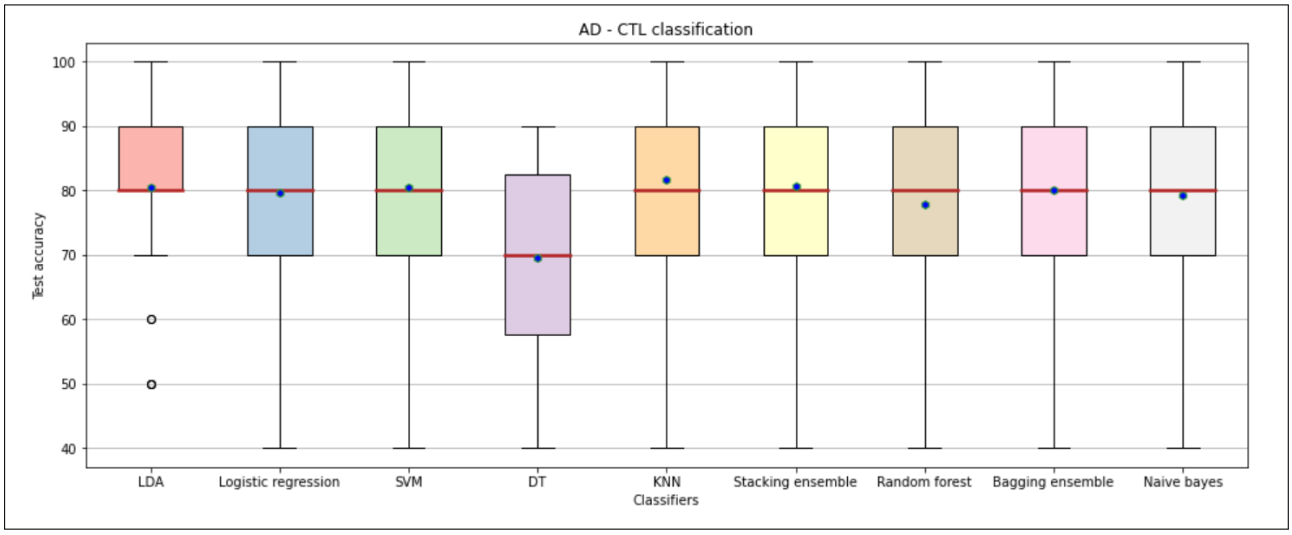


Figure 25: Test accuracies across all splits for each classifier on the AD-CTL data (• : mean accuracy, -- : median accuracy)

From the above figure, we infer that the KNN classifier performs the best across all the splits as it has the highest mean and median accuracy when compared with the other classifiers. The maximum test accuracy obtained across the splits is 100 % and the minimum test accuracy is 40 %. The decision tree classifier is not observed to perform accurately as it has not achieved maximum accuracy values of 100 % for any of the data splits, unlike the rest of the classifiers. The mean test accuracy is also comparatively much lesser than

the other classifiers.

The comparison of the test sensitivity and test specificity values obtained using the respective classifiers across all the splits of the AD-CTL data is:

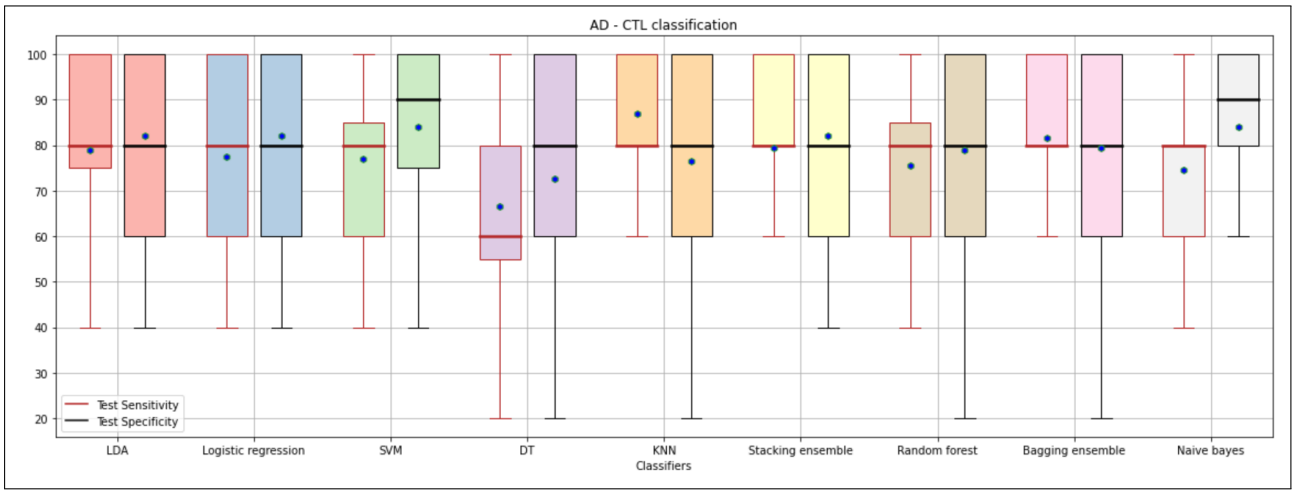


Figure 26: Test sensitivity and test specificity across all splits for each classifier on the AD-CTL data (• : mean sensitivity, specificity, -- : median sensitivity, -- : median specificity)

From the above figure, the KNN classifier has the highest mean sensitivity and specificty values of around 80 %. The maximum and minimum sensitivity and specificity values achieved across all the splits is 100 % and 20 % respectively. The Decision tree classifier is observed to have the least mean sensitivity and specificity across the splits.

The ROC curves for each of these classifiers across the data in the cross validation split 14 is:

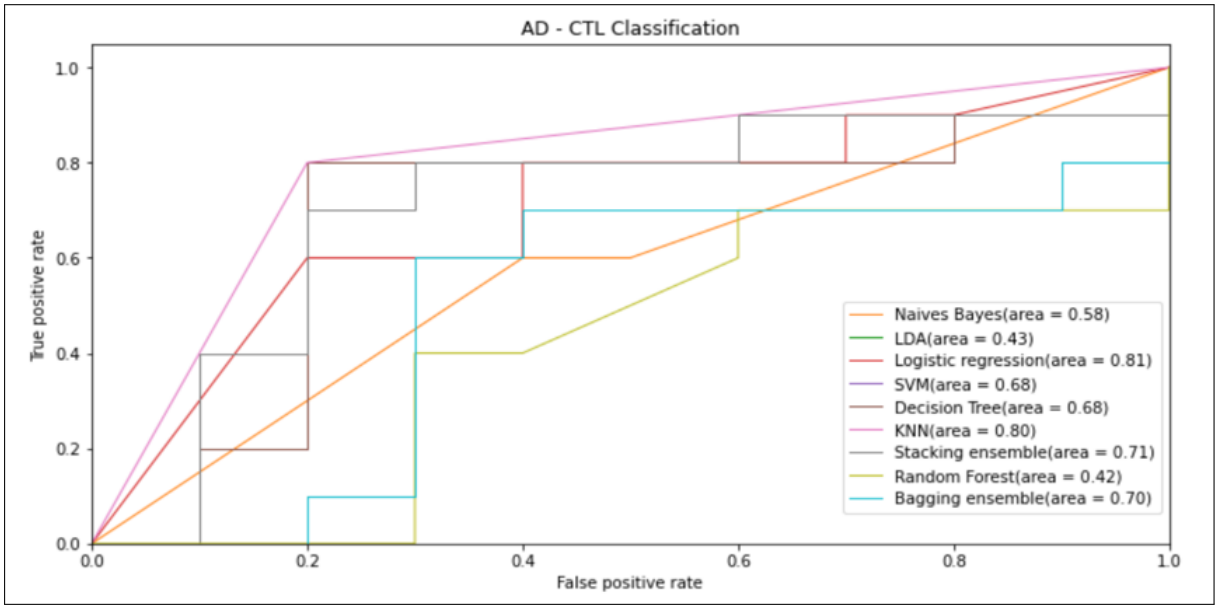


Figure 27: ROC curves for each classifier on the AD-CTL split 14 data. Area under the curve is mentioned in the legend next to the classifier.

From Figure 27, the logistic regression model followed by the KNN classifier has the highest area under the curve on the split 14 data. This translates into these models being highly discriminative tests when dealing with the AD-CTL split 14 test data.

Summarizing, the results obtained by training and testing the classifiers on the AD-CTL data are:

Metric / Model	Test accuracy mean (std)	Test sensitivity mean (std)	Test specificity mean (std)	F1 score mean	Mean absolute error
LDA	80.5% (13.6%)	79.0% (16.5%)	82.0% (16%)	0.799	0.390
Logistic Regression	79.75% (13.8%)	77.5% (17.2%)	82.0% (15.6%)	0.788	0.405
SVM	80.5% (14.1%)	77.0% (17.5%)	84.0% (18%)	0.793	0.391
KNN (K=28)	**81.75% (14.1%)**	**79.0% (14.7%)**	**76.5% (16.2%)**	**0.830**	**0.365**
Decision Tree	69.5% (17.7%)	66.5% (18.2%)	72.5% (16.8%)	0.672	0.636
Stacking Ensemble	80.75% (14.2%)	79.5% (16.4%)	82.0% (17.1%)	0.801	0.385
Random Forest	78.75% (14.4%)	76.5% (17%)	81.0% (18.5%)	0.778	0.425
Bagging Ensemble	81.0% (13.1%)	80.0% (15%)	79.0% (16.2%)	0.813	0.380
Naive Bayes	79.25% (14.2%)	74.5% (17.6%)	84.0% (18.8%)	0.776	0.415

Table 2: Leave-Ten-Out cross validation results of the respective classifiers on the AD-CTL data

From table 2, we infer that the KNN classifier performs the best on the AD-CTL data splits. The test accuracy is 81.75 % with a standard deviation of 14.1%. The test sensitivity and specificity scores of around 75-80 % with a standard deviation of around 15 % are achieved on an average for all the classifiers, which is desired. We can also observe that the SVM and LDA classifiers perform competitively well with the KNN classifier and achieve high F1 scores. The decision tree classifier is not well suited for this binary classification problem.

9.3.2 AD - MCI classification

The plot of the test accuracies obtained by training and testing the respective classifiers across all the splits of the AD-MCI data is:

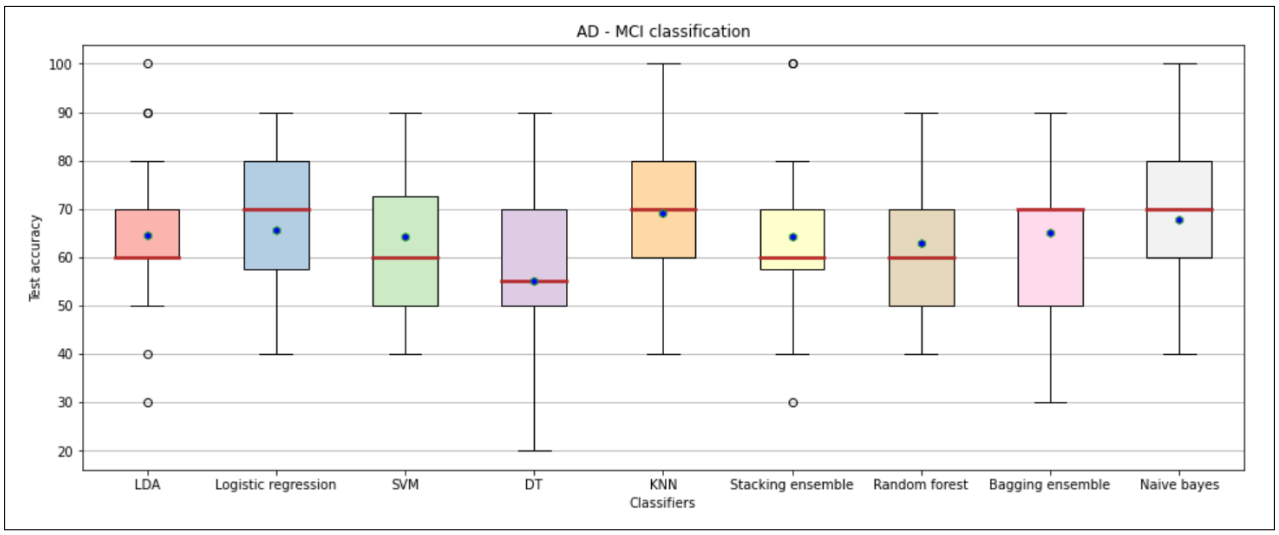


Figure 28: Test accuracies across all splits for each classifier on the AD-MCI data (• : mean accuracy, -- : median accuracy)

From the figure above, the KNN classifier has the greatest mean and median test accuracies. The maximum accuracy acheived across all the splits of the AD-MCI data is 100 % and the minimum test accuracy is 20 %. This minimum accuracy is much lesser than what was achieved on the AD-CTL data splits. The reasoning behind this could be that the AD-CTL data is clearly separable while the AD-MCI is not that clearly separable. The LDA classifier is not much suitable for these data splits.

The comparison of the test sensitivity and test specificity values obtained using the respective classifiers across all the splits of the AD-MCI data is:

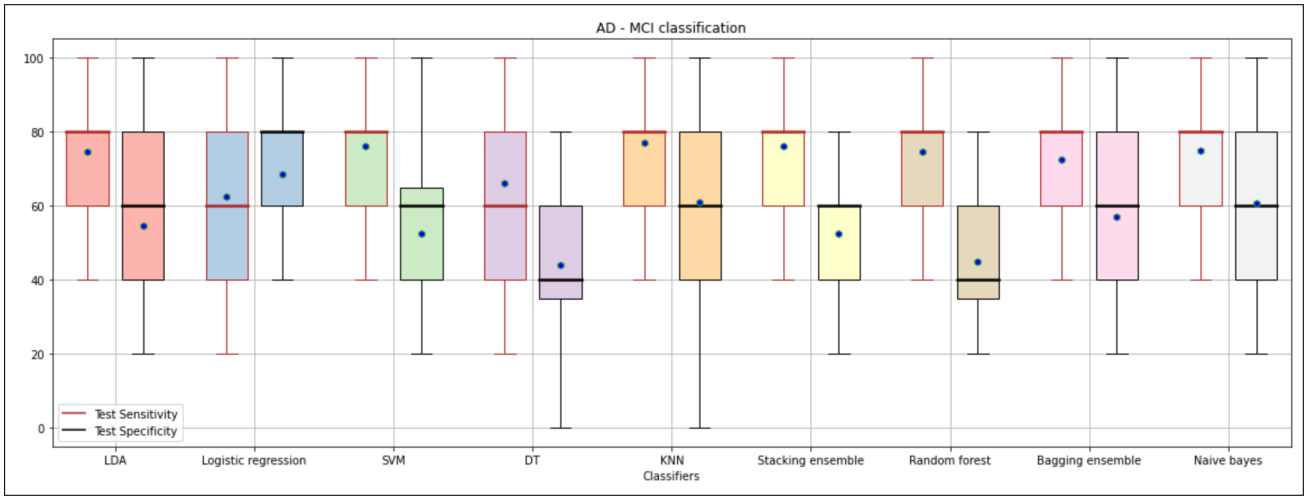


Figure 29: Test sensitivity and test specificity across all splits for each classifer on the AD-MCI data (• : mean sensitivity, specificity, -- : median sensitivity, -- : median specificity)

From the figure above, the KNN classifier has the highest mean and median sensitivity and specificity values. The maximum and minimum values achieved through each of the splits is 100 % and 0 % respectively, as in the case of the Decision tree classifier. This is much lower than in the case of AD-CTL data, and accounts to the data separability between the datasets.

The ROC curves for each of these classifiers across the data in the cross validation split 14 is:

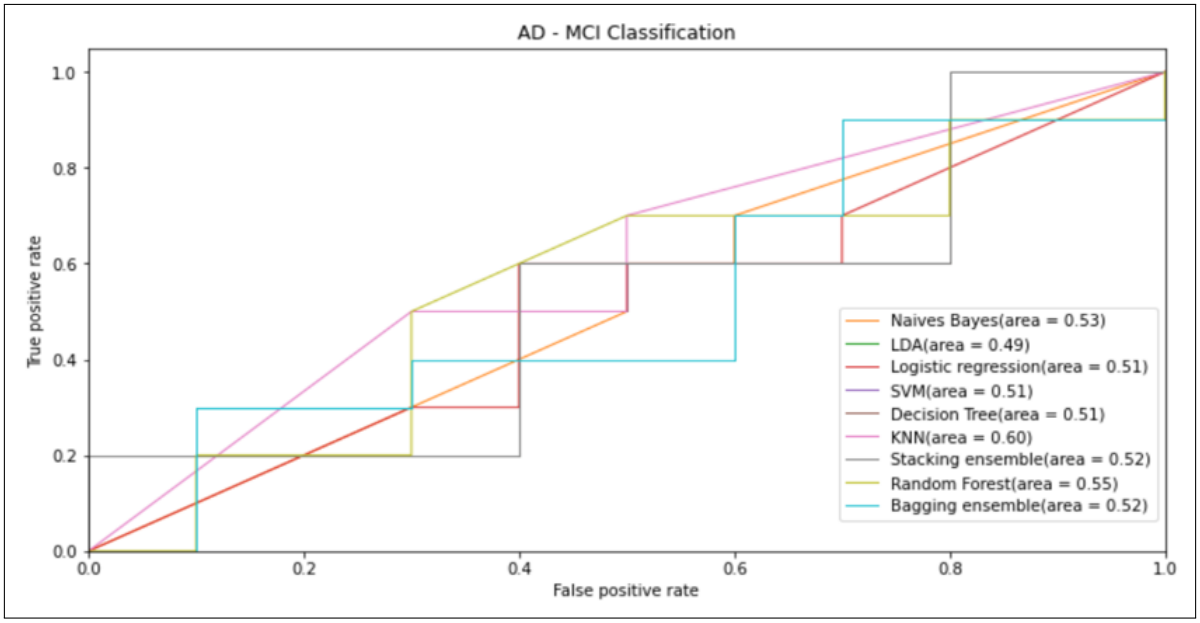


Figure 30: ROC curves for each classifier on the AD-MCI split 14 data. Area under the curve is mentioned in the legend next to the classifier.

From Figure 30, the KNN classifier has the greatest area under ROC curve of 0.60. The greater value of the area under the curve translates into a better discriminative test, which is useful in differentiating which of the patients have the Alzheimers disease and which of the patients are in the mild cognitive impairment stage.

The results obtained by training and testing the classifiers on the AD-MCI data are:

Metric / Model	Test accuracy mean (std)	Test sensitivity mean (std)	Test specificity mean (std)	F1 score mean	Mean absolute error
LDA	64.5% (13.3%)	74.5% (16.5%)	54.5% (17.6%)	0.674	0.355
Logistic Regression	65.5% (13.5%)	62.5% (15.5%)	68.5% (17.5%)	0.636	0.345
SVM	64.25% (13.7%)	76.0% (14.4%)	52.5% (17.8%)	0.681	0.357
KNN (K=28)	**69.0% (13.4%)**	**70.0% (14.2%)**	**61.0% (18.2%)**	**0.710**	**0.310**
Decision Tree	55.0% (14%)	66.0% (17.5%)	44.0% (18%)	0.581	0.472
Stacking Ensemble	64.25% (13.3%)	76.0% (17.6%)	52.5% (17.8%)	0.679	0.357
Random Forest	61.5% (13.5%)	79.5% (13.8%)	43.5% (18.2%)	0.675	0.385
Bagging Ensemble	65.25% (13.8%)	73.0% (17.4%)	57.5% (16.8%)	0.676	0.347
Naive Bayes	67.75% (13%)	75.0% (13.2%)	60.5% (16%)	0.697	0.322

Table 3: Leave-Ten-Out cross validation results of the respective classifiers on the AD-MCI data

From table 3, we conclude that the KNN classifier performs the best on the AD-MCI data. The test accuracy achieved is 69 % and the standard deviation is 13.4 %. Test sensitivity and specificity scores in the range of 60 - 70 % with a standard deviation of around 13 - 16 % are achieved on an average for all the classifiers on this data. This is however comparatively lower than the scores obtained using these classifiers on the AD-CTL data. The decision tree does not perform well in this case either.

9.3.3 MCI - CTL classification

The plot of the test accuracies obtained by training and testing the respective classifiers on the MCI-CTL data is:

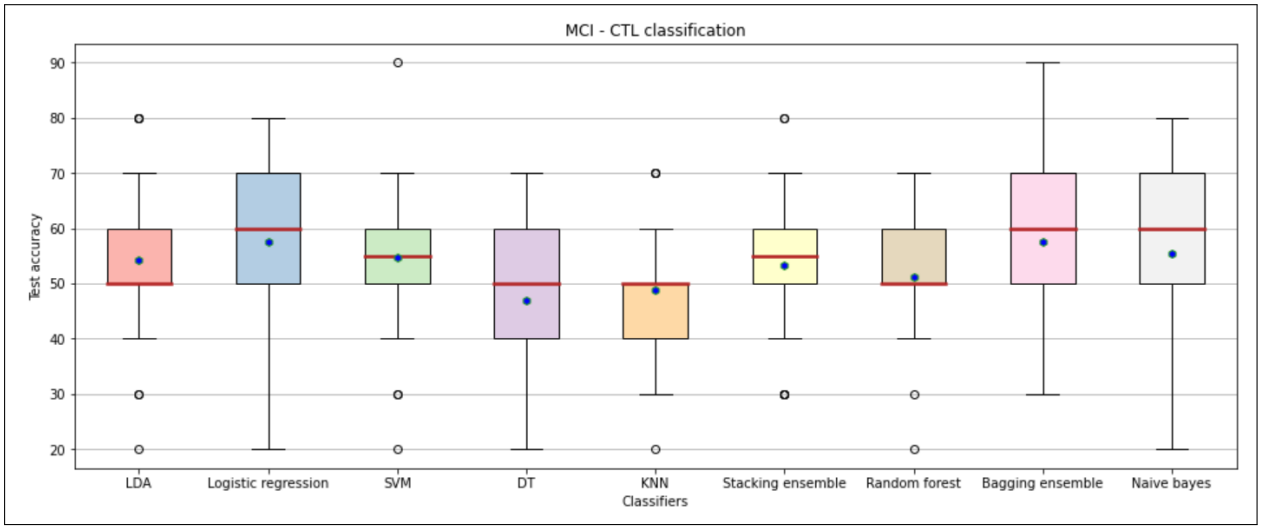


Figure 31: Test accuracies across all splits for each classifier on the MCI-CTL data (• : mean accuracy, -- : median accuracy)

From the figure above, the logistic regression classifier has the greatest mean and median test accuracy value across all the splits. The maximum accuracy achieved is 90 % and the minimum accuracy achieved is 20 %. This is much lower than both the other binary classification problems because of the partial data separability. The decision tree classifier performs the least effectively.

The comparison of the test sensitivity and test specificity values obtained using the respective classifiers across all the splits of the MCI-CTL data is:

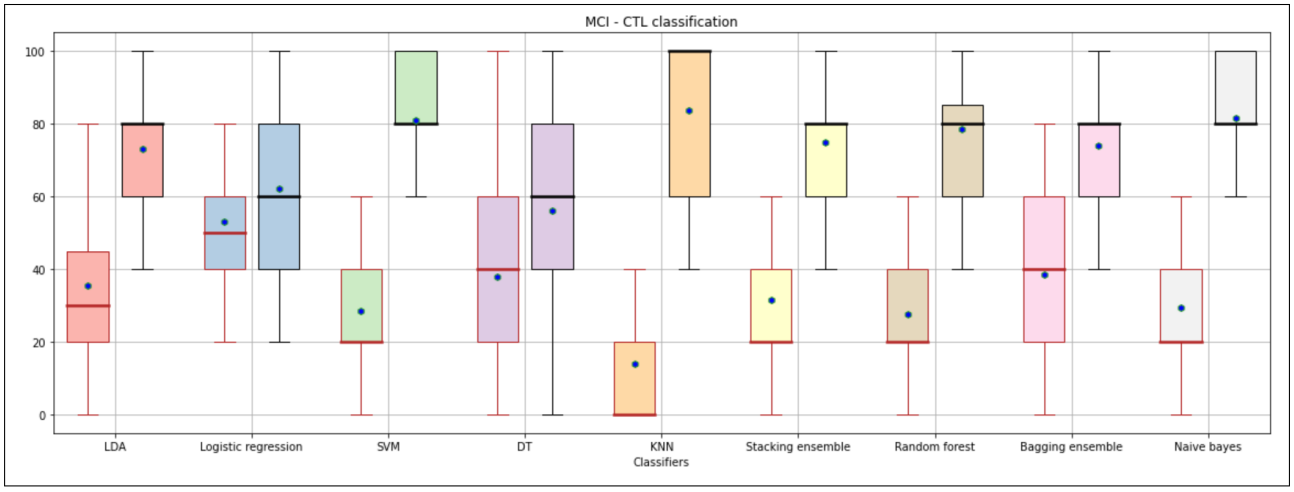


Figure 32: Test sensitivity and test specificity across all splits for each classifier on the MCI-CTL data (• : mean sensitivity, specificity, -- : median sensitivity, -- : median specificity)

From the figure above, the logistic regression classifier displays the uniformity in the mean sensitivity and specificity values. The maximum and minimum values achieved across all the splits of this dataset are 100 % and 0 % respectively. It is observed that a

few classifiers display a much higher sensitivity and a relatively very low specificity, such as the Random forest and the Naives Bayes classifiers. The reasoning behind this is the separability between the data points.

The ROC curves for each of these classifiers across the data in the cross validation split 14 is:

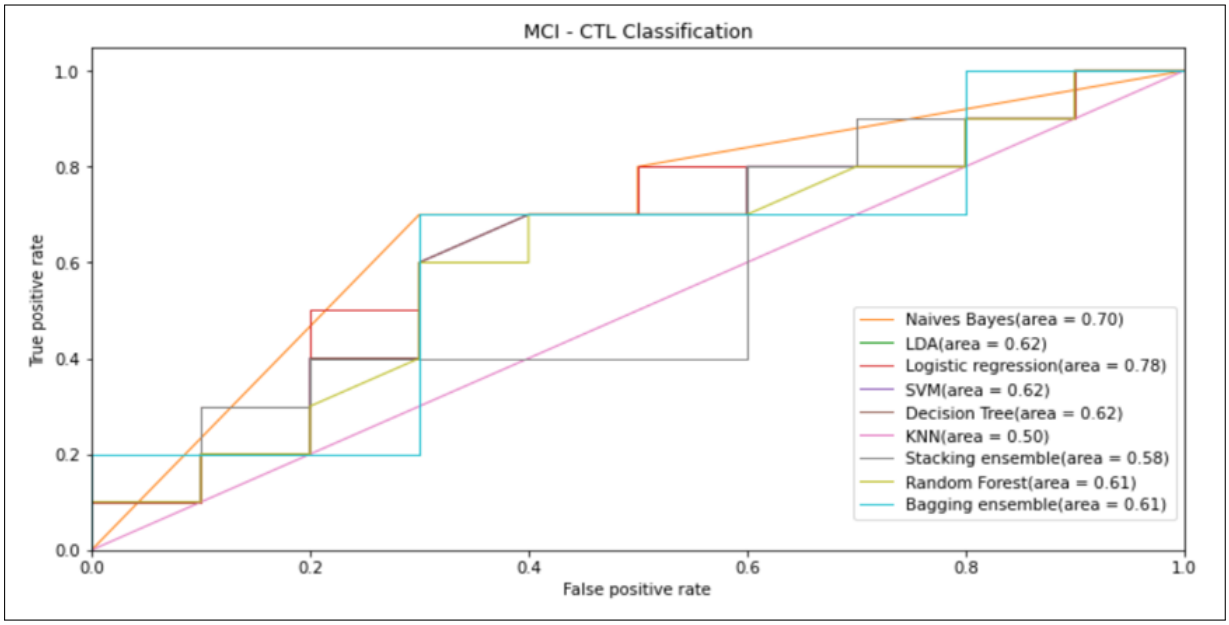


Figure 33: ROC curves for each classifier on the MCI-CTL split 14 data. Area under the curve is mentioned in the legend next to the classifier.

From Figure 33, the logistic regression classifier has the greatest area under the curve value of 0.78. This translates into a better sensitive and specific test.

Therefore, summarizing, the results obtained by training and testing the classifiers on the MCI-CTL data are:

Model \ Metric	Test accuracy mean (std)	Test sensitivity mean (std)	Test specificity mean (std)	F1 score mean	Mean absolute error
LDA	54.25% (14.8%)	35.5% (18.1%)	73.0% (19%)	0.560	0.457
Logistic Regression	**57.5% (12.2%)**	**53.0% (16%)**	**62.0% (17.2%)**	**0.610**	**0.425**
SVM	54.75% (13.5%)	28.5% (18.2%)	81.0% (17.9%)	0.575	0.452
KNN (K=28)	48.75% (10.6%)	14.0% (18.7%)	83.5% (20.1%)	0.565	0.512
Decision Tree	47.0% (14.4%)	38.0% (18.6%)	56.0% (19%)	0.510	0.554
Stacking Ensemble	53.25% (12.8%)	31.5% (16.6%)	75.0% (17.8%)	0.565	0.467
Random Forest	52.75% (13.6%)	26.5% (16.4%)	79.0% (19.1%)	0.545	0.472
Bagging Ensemble	55.25% (13.5%)	37.5% (16.8%)	73.0% (18.6%)	0.595	0.447
Naive Bayes	55.50% (13%)	29.5% (17.8%)	81.5% (18.2%)	0.605	0.445

Table 4: Leave-Ten-Out cross validation results of the respective classifier models on the MCI-CTL data

From table 4, we infer that the scores across all the classifiers are very much lower compared to the other two binary classification problems. This is because of the inherent complexity of the MCI-CTL data. The logistic regression classifier is the most well suited for this dataset and it is observed that test sensitivity and specificity scores of around 50 - 60 % with a standard deviation of around 16 - 17 % are achieved across all the classifiers on an average. The test accuracy is 57.5 % with a standard deviation of 12.2 %. From the above three studies, we can conclude that the decision tree classifier is not able to effectively classify the data into the respective classes.

Therefore, we conclude that the KNN classifier performs the most effectively on the AD-CTL and AD-MCI data. The Logistic regression classifier perform well on the MCI-CTL binary classification problem.

10 Observations

We are able to obtain precise classification results on the highly dimensional clinical data using effective feature selection methods and leveraging the linear classifiers. The above experiments prove that these linear classifiers such as the KNN and the logistic regression model perform more effectively than the complex models such as the stacking and bagging ensemble of classifiers. Selecting an appropriate set of highly discriminative features plays an important role in determining the effectiveness of the classifier on the given data as observed in these experiments.

The results of each binary classification problem depends on the separability of the data in the respective discriminative feature subspace. This is emphasized by the observation

that the AD-CTL classification problem has achieved the highest test accuracy of 81.75 % because of the complete separability of the data. On the other hand, the AD-MCI and MCI-CTL data is not that separable and hence we only observe the highest test accuracies of 69.0 % and 57.5 % respectively. I also observe that the probabilistic classifier such as the Naive bayes classifier performs equally well on each binary classification problem as the linear classifiers.

11 Timetable

I have adhered to the timetable I have initially proposed and was able to achieve all the goals. There were no changes made to the timetable given below:

Week	Work distribution
3/22 - 3/28	Data preprocessing, implementing SVM + RFE algorithm
3/29 - 4/4	Implement SFS using covariance matrix of features and outputs from SVM + RFE algorithm
4/5 - 4/11	Train SVM classifier for feature selection module, Experiment with various classifiers on subset of features
4/12 - 4/18	Analyze results of classifiers and report metrics, Project mid progress review
4/19 - 4/25	Explore stacking ensemble classification, Implement ensemble classification using proposed classifiers
4/26 - 4/29	Visualization of results, Final project presentation and report

Table 5: Timetable for work distribution

12 Future work

For the future work, we can experiment with other linear classifiers and data sampling techniques in order to reduce the standard deviation of the test sensitivity and specificity values across the leave-ten-out cross validation splits.

We can also experiment with effective feature selection techniques in order to understand which features are more related to the MCI-CTL data. These discriminative features can be used to increase the mean test accuracies across the splits of this data.

13 Conclusion

In conclusion, I have experimented with several linear classifiers and ensemble of classifiers using a multi-feature fusion approach as a feature selection method to perform binary classification on the AD-CTL, AD-MCI and MCI-CTL datasets. Leave-ten-out cross validation was used to divide the Alzheimer's disease dataset into multiple splits. The classification results achieved are attributed to the highly discriminative set of features obtained after feature selection.

Summarizing, the classifier and the respective maximum test accuracy, sensitivity and specificity scores achieved on each of the binary classification problem is as follows:

Metric / Data	Classifier	Test accuracy mean (std)	Test sensitivity mean (std)	Test specificity mean (std)	F1 score mean	Mean absolute error
AD - CTL	KNN (K=28)	81.75% (14.1%)	79.0% (14.7%)	76.5% (16.2%)	0.830	0.365
AD - MCI	KNN (K=28)	69.0% (13.4%)	70.0% (14.2%)	61.0% (18.2%)	0.710	0.310
MCI - CTL	Logistic regression	57.5% (12.2%)	53.0% (16%)	62.0% (17.2%)	0.610	0.425

Table 6: Maximum classification results achieved by the classifiers for the respective binary classification problem

References

[1] "https://www.alz.org/alzheimers-dementia/what-is-alzheimers." 3

[2] "https://www.who.int/news-room/fact-sheets/detail/dementia." 3

[3] Y. Liu, L. A. Teverovskiy, O. L. Lopez, H. Aizenstein, C. C. Meltzer, and J. T. Becker, "Dicovery of "biomarkers" for alzheimer's disease prediction from structural mr images," in *2007 4th IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, April 2007, pp. 1344–1347. 4, 5, 6

[4] Z. Xiao, Y. Ding, T. Lan, C. Zhang, C. Luo, and Z. Qin, "Brain mr image classification for alzheimer's disease diagnosis based on multifeature fusion," *Computational and Mathematical Methods in Medicine*, vol. 2017, pp. 1–13, 05 2017. 4, 5, 9, 10

[5] S. Farhan, M. Fahiem, and H. Tauseef, "An ensemble-of-classifiers based approach for early diagnosis of alzheimer's disease: Classification using structural features of brain images," *Computational and mathematical methods in medicine*, vol. 2014, p. 862307, 09 2014. 4

[6] G. Janakasudha and P. Jayashree, "Early detection of alzheimer's disease using multi-feature fusion and an ensemble of classifiers," *Advances in Intelligent Systems and Computing, Springer*, vol. 1082, 02 2020. 4, 5

[7] "Pattern recognition and machine learning leture notes on dataset size." 6

[8] "https://towardsdatascience.com/understanding-confusion-matrix-a9ad42dcfd62." 11