SML Assignment 3
Report

Aditya Prasad
2022036

Assumptions:

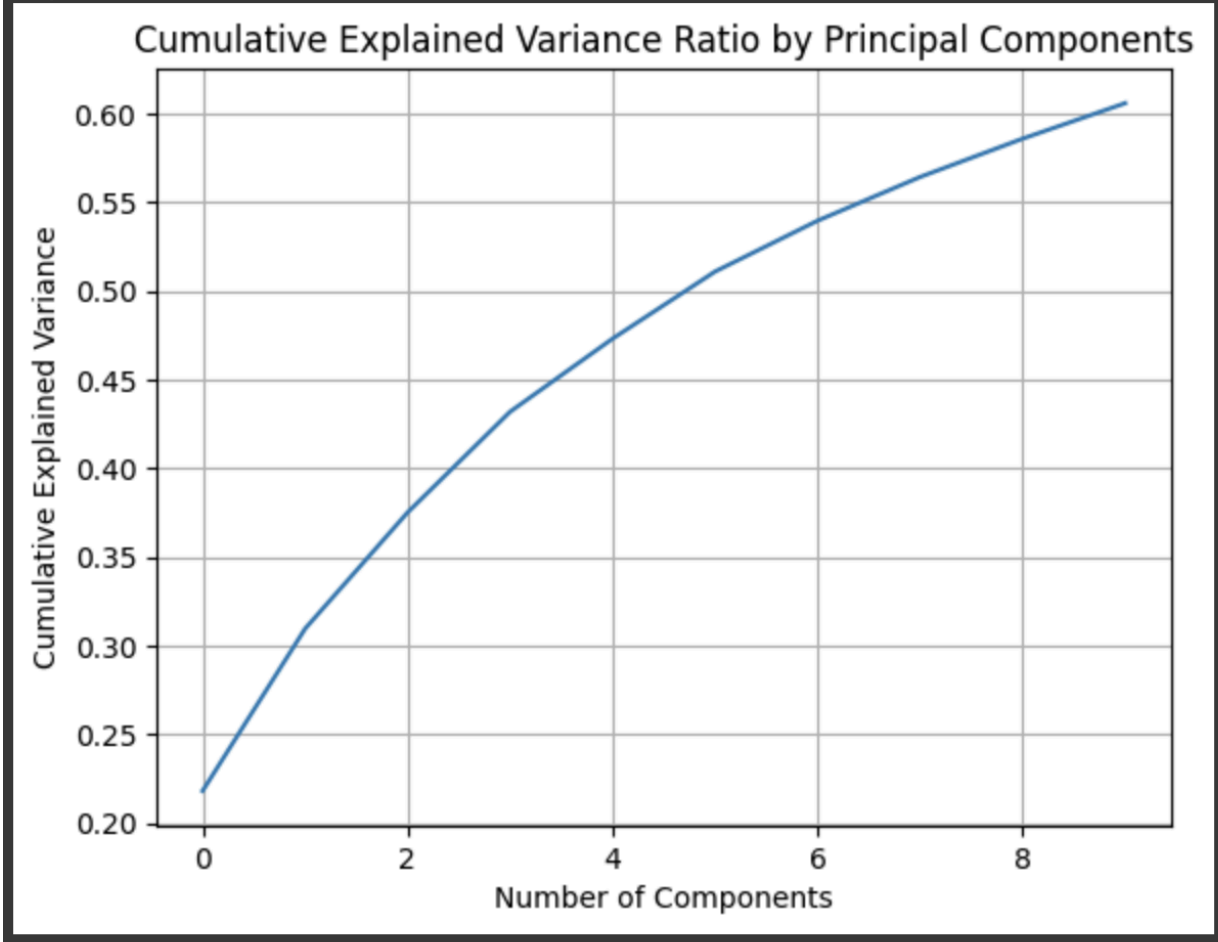
- The MNIST dataset is used, and only classes 0, 1, and 2 are selected.
- Principal Component Analysis (PCA) is applied to reduce the dimensionality of the data to 10 components.

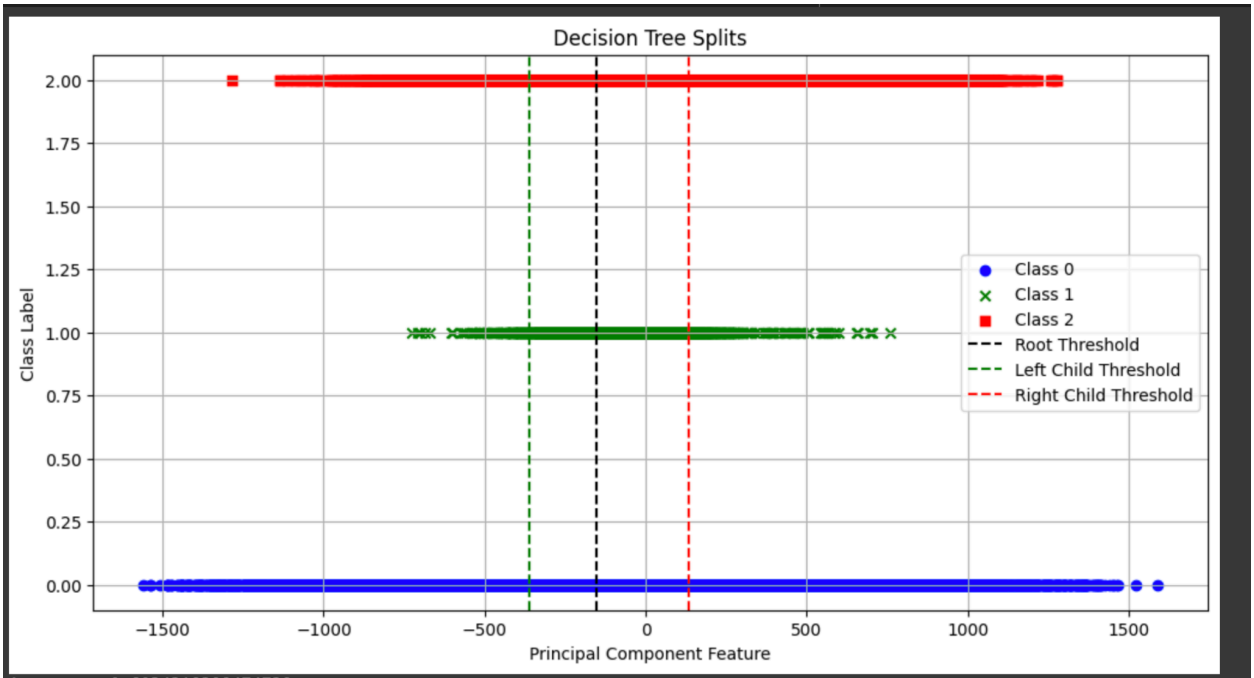
Approach:

1. Load the MNIST dataset and select classes 0, 1, and 2 for both training and testing sets.
2. Apply PCA to reduce the dimensionality of the training data to 10 components.
3. Implement functions to calculate the Gini index and find the best split for a node in the decision tree.
4. Grow a decision tree with 3 terminal nodes using the training data.
5. Make predictions on the test data using the decision tree and calculate the accuracy and class-wise accuracy.
6. Implement a function for bootstrapping the dataset.
7. Build 5 different decision trees using bagging (bootstrapping the dataset).
8. Make predictions on the test data using majority voting from the 5 decision trees.
9. Calculate the overall accuracy and class-wise accuracy for the bagging approach.

Results:

```
Explained variance ratio: 0.606064935212838
```

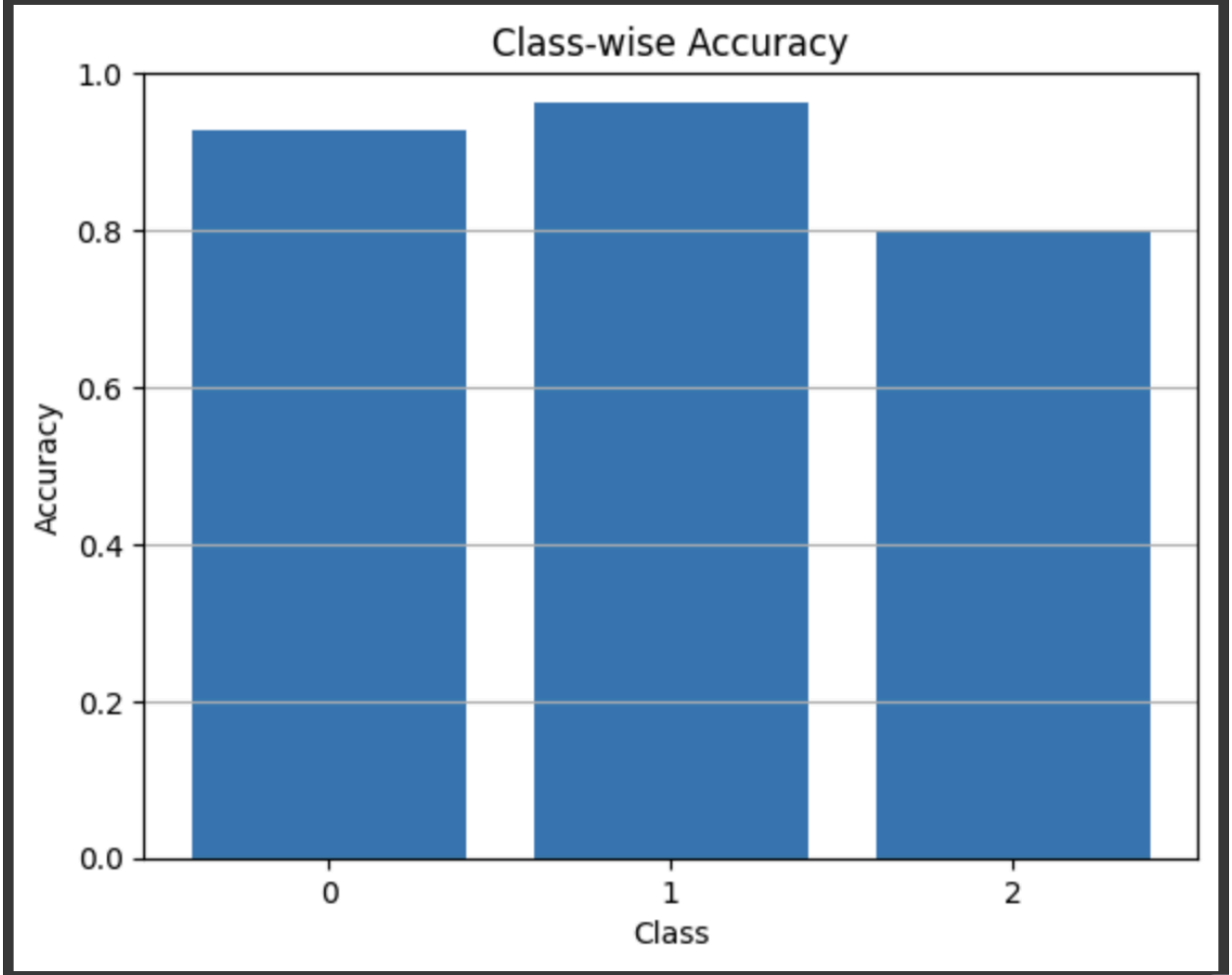





```
Accuracy: 0.6034318398474738
```

```
Overall Accuracy: 0.6034318398474738
Class-wise Accuracy:
Class 0: 0.3183673469387755
Class 1: 0.9251101321585903
Class 2: 0.5203488372093024
```

```
Overall Accuracy: 0.8976803304734668
Class-wise Accuracy:
Class 0: 0.926530612244898
Class 1: 0.9629955947136564
Class 2: 0.7984496124031008
```



Methodology:

Principal Component Analysis (PCA):

Principal Component Analysis (PCA) is a statistical technique used for dimensionality reduction. It transforms a high-dimensional dataset into a lower-dimensional subspace by finding a new set of orthogonal variables, called principal components, that capture the maximum variance in the data. PCA is particularly useful when dealing with datasets that have a large number of features, as it can help reduce computational complexity and mitigate the curse of dimensionality.

In this analysis, PCA was applied to the MNIST dataset to reduce the dimensionality of the data from 784 (28x28 pixels) to 10 components. The explained variance ratio, which measures the proportion of variance in the data that is retained by the principal components, was used to determine the number of components to keep. The choice of 10 components was based on the cumulative explained variance ratio, which showed that these 10 components accounted for a significant portion of the variance in the data.

Decision Tree Learning:

Decision tree algorithms are a type of supervised learning technique used for both classification and regression tasks. They work by recursively partitioning the feature space into smaller regions based on the values of the input features. The partitioning process is guided by a metric that measures the impurity or heterogeneity of the data in each region, such as the Gini index or information gain.

The Gini index is a measure of the probability of misclassifying a randomly chosen instance in a given dataset. It ranges from 0 (pure) to 1 (impure), with lower values indicating a better separation of classes. Information gain, on the other hand, quantifies the reduction in entropy or impurity achieved by splitting the data based on a particular feature.

In this analysis, a decision tree with 3 terminal nodes was grown using the training data. The process involved finding the best split at each node by evaluating the Gini index for all possible splits across all features. The feature and threshold that minimized the weighted sum of the Gini indices of the resulting partitions were chosen as the best split. This process was repeated recursively for each child node until the desired number of terminal nodes was reached.

Bagging:

Bagging, short for Bootstrap Aggregating, is an ensemble learning technique that combines multiple models to improve the overall accuracy and stability of predictions. It works by creating multiple bootstrap samples from the original dataset, training a separate model (e.g., decision tree) on each bootstrap sample, and then aggregating the predictions of all models.

The advantages of bagging include reducing the variance of individual models, improving the overall accuracy, and providing a measure of confidence in the predictions through the agreement of multiple models.

In this analysis, bagging was applied by creating 5 different datasets from the original training data using bootstrapping. For each bootstrap sample, a decision tree with 3 terminal nodes was trained. To make predictions on the test data, majority voting was used, where the class predicted by at least 3 out of the 5 decision trees was assigned to a given test instance. In case of a tie (two trees predicting one class and two predicting another), either class could be chosen.