Report

Project: Dimensionality Reduction Using PCA

Group Name: Vaddi_Korrapati_Neelapala_Kakarlapudi_Vaddi

1. Overview

Problem Statement

High-dimensional datasets often contain redundant and correlated features, which can lead to inefficiencies in computation, storage, and model performance. This "curse of dimensionality" not only increases computational cost but can also hinder the accuracy and interpretability of machine learning models. Dimensionality reduction techniques are essential to simplify data, improve computational efficiency, and enhance the performance of downstream tasks like classification and visualization.

How SVD Solves It

Singular Value Decomposition (SVD) is a foundational mathematical method for performing PCA, making it an effective approach for dimensionality reduction. SVD decomposes the data matrix into three matrices: U, Σ, and V^T, where Σ contains singular values that represent the variance captured by each principal component. By retaining only, the top k singular values (corresponding to the components with the highest variance), SVD enables the construction of a reduced dataset that retains the most critical patterns while discarding noise and redundancy. This ensures that the dimensionality is reduced without significant loss of information, making machine learning models more efficient and robust.

2. Methodology

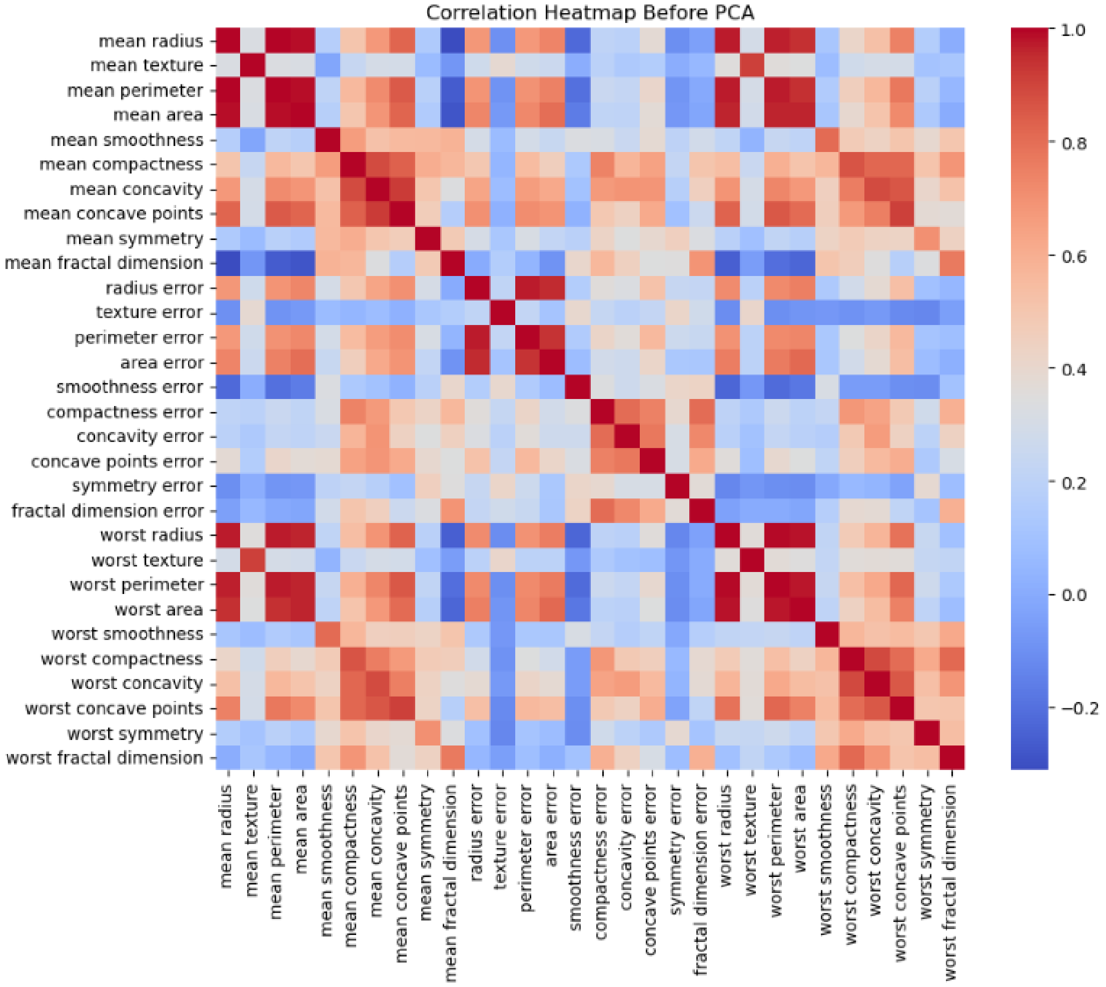
Steps for Dimensionality Reduction Using PCA:

1. Dataset Preparation and Understanding:

- The dataset used was the Breast Cancer dataset from Scikit-learn, consisting of 569 samples and 30 features derived from cell nuclei measurements. Each sample is labelled as malignant or benign.
- Initial exploratory analysis highlighted the high-dimensional nature of the data, making it a suitable candidate for dimensionality reduction.

2. Exploratory Data Analysis (EDA):

- A correlation heatmap was generated to assess the relationships among features. Features like mean radius and mean area showed strong correlations, indicating redundancy.
- Identified redundant features and visualized the data distribution to confirm overlapping information in the high-dimensional space.



3. Data Standardization:

- Standardized the dataset to ensure all features contributed equally to PCA. This was done using the formula:

  $Z = (x - \mu) / \sigma$

  where x is the feature value, μ is the mean, and σ is the standard deviation of the feature.

- Standardization cantered the data around zero with a unit variance, ensuring no feature dominated due to its scale.

4. Covariance Matrix Computation:

- Constructed the covariance matrix of the standardized data to quantify the variance and relationships between features.

C = (1/n) * X^T * X

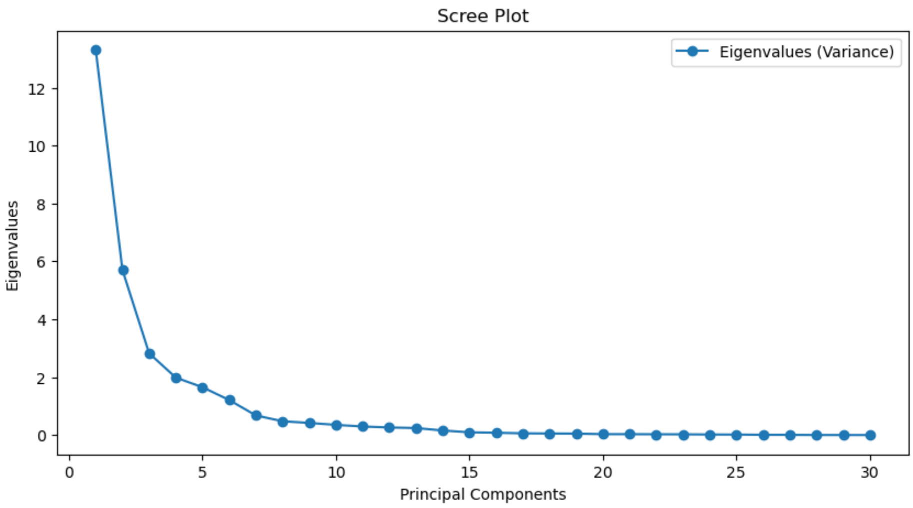
where C is the covariance matrix, X is the standardized data matrix, and n is the number of samples.

5. Eigenvalue and Eigenvector Calculation:

- Calculated the eigenvalues and eigenvectors of the covariance matrix.
- Eigenvalues represent the amount of variance captured by each eigenvector.
- Eigenvectors define the directions (principal components) in the feature space.

6. Selection of Principal Components:

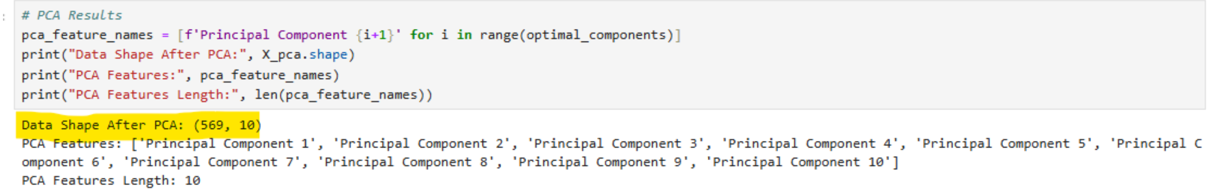
- Ranked principal components based on their corresponding eigenvalues (variance explained).
- Retained the first k components that captured the highest cumulative variance. For this project:
- The first component explained 44.27% of the variance.
- The first 10 components captured 95.16% cumulative variance.



The scree plot shows that most variance is captured by the first few components, with diminishing returns after 10 components, justifying dimensionality reduction.

7. Dimensionality Reduction:

Projected the original dataset onto the k-dimensional space formed by the selected principal components. This step reduced the feature space from 30 dimensions to 10 while retaining most of the significant information.



```
# PCA Results
pca_feature_names = [f'Principal Component {i+1}' for i in range(optimal_components)]
print("Data Shape After PCA:", X_pca.shape)
print("PCA Features:", pca_feature_names)
print("PCA Features Length:", len(pca_feature_names))
```

```
Data Shape After PCA: (569, 10)
PCA Features: ['Principal Component 1', 'Principal Component 2', 'Principal Component 3', 'Principal Component 4', 'Principal Component 5', 'Principal C
omponent 6', 'Principal Component 7', 'Principal Component 8', 'Principal Component 9', 'Principal Component 10']
PCA Features Length: 10
```

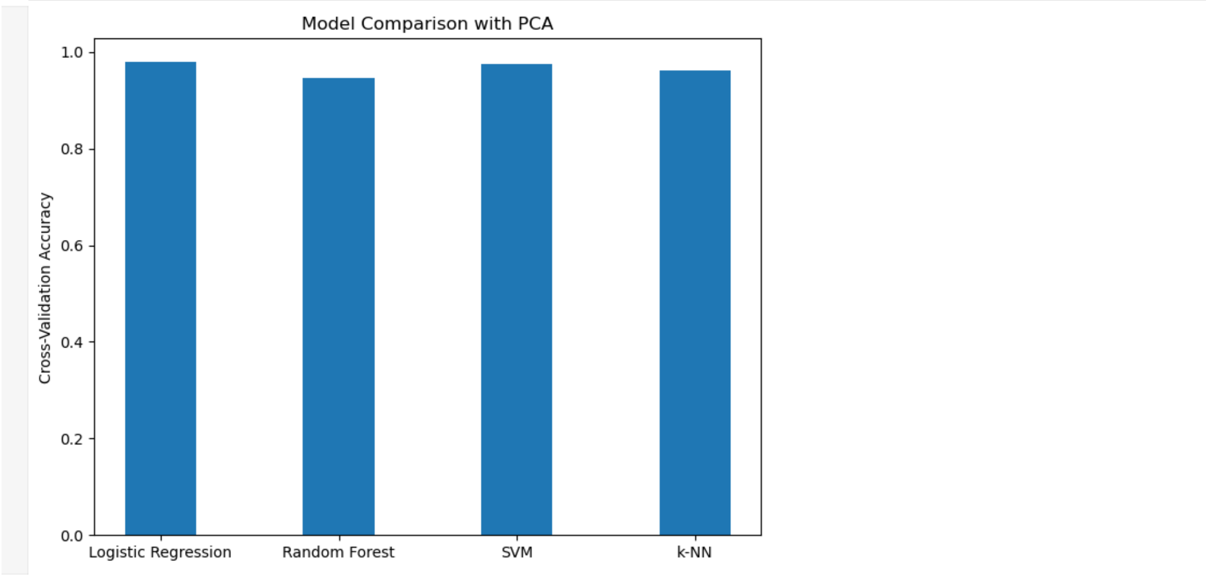
After applying PCA, the dataset is reduced to 10 principal components with a shape of (569, 10), retaining the most critical variance from the original 30 features.

8. Visualization:

- Plotted explained variance ratios to visualize how much variance each principal component captured.
- Created cumulative variance plots to identify the optimal k (number of components) for retaining meaningful data.

9. Classification with Reduced Data:

- Applied four classifiers (Logistic Regression, SVM, Random Forest, and k-NN) to the reduced dataset to assess the impact of PCA on classification performance.
- Compared performance metrics, such as accuracy and confusion matrices, for models trained on the original dataset versus the reduced dataset.



The bar chart compares the cross-validation accuracy of four classifiers after applying PCA. Logistic Regression achieved the highest accuracy, closely followed by SVM, while Random Forest and k-NN performed slightly lower but comparably well.

10. Evaluation:

- Assessed performance using metrics like explained variance, cumulative variance, and classification accuracy.
- Analysed how varying k influenced model accuracy and computational efficiency.

3. Experiments

3.1 Setup

1. Datasets

- Source: The dataset used for this project is the Breast Cancer dataset from the Scikit-learn library, a well-known and widely-used resource for benchmarking machine learning methods.
- Size: The dataset contains 569 samples with 30 numerical features derived from various cell nuclei measurements obtained through digital imaging techniques.
- Characteristics: Each sample in the dataset is labelled as either malignant (cancerous) or benign (non-cancerous), making it a binary classification task. This dataset is particularly suitable for evaluating the effectiveness of dimensionality reduction techniques since the high dimensionality (30 features) can introduce redundancy and inefficiency.

2. Tools/Libraries:

Programming Language: Python, chosen for its robust ecosystem of libraries and tools for machine learning and data analysis.

Libraries:

- Scikit-learn: For implementing PCA, evaluating classifiers, and accessing the Breast Cancer dataset.
- NumPy and Pandas: For numerical computations and data manipulation.
- Matplotlib and Seaborn: For generating visualizations, such as correlation heatmaps and variance plots.

3. Metrics for Evaluation:

To measure the performance of PCA and its impact on classification, the following metrics were used:

i) Explained Variance:

- Quantifies the amount of variance captured by each principal component. The higher the explained variance, the more information is retained in the corresponding component.

  $$\text{Explained Variance Ratio} = \lambda i / \sum\lambda$$

  where λi is the eigenvalue of the i-th component.

- This metric is fundamental to PCA as it directly indicates how much of the dataset's variance is preserved.

ii) Cumulative Variance:

- The summation of explained variance ratios across components, indicating the total variance retained by the selected components.
- Helps determine the optimal number of components (k) to retain sufficient variance while reducing dimensionality.

iii) Classification Accuracy:

- Measures the proportion of correct predictions out of the total predictions made by the model.

  $$\text{Accuracy} = \text{Correct Predictions} / \text{Total Samples}$$

- A standard metric for evaluating classifier performance, directly reflecting the impact of PCA on predictive accuracy.

iv) Confusion Matrix:

- A table summarizing the performance of a classification algorithm by comparing predicted and actual labels, highlighting true positives, false positives, true negatives, and false negatives.
- Provides a more granular view of classification performance, allowing analysis of trade-offs between sensitivity and specificity.

3.2 Results

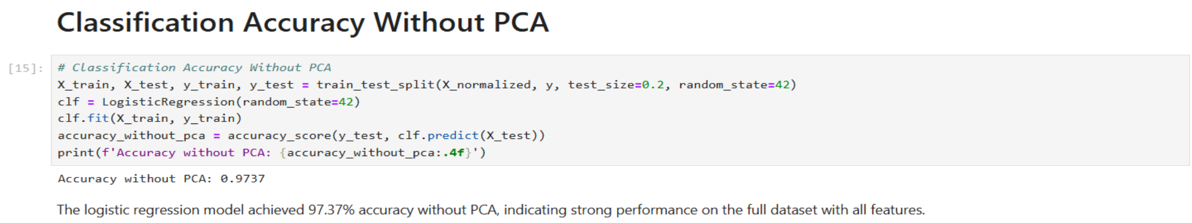
Performance Evaluation:

1. Explained Variance and Cumulative Variance:

- The first principal component explained 44.27% of the variance, while the first 10 components cumulatively explained 95.16%.
- Beyond k=10, additional components contributed marginally to variance retention, illustrating diminishing returns.

2. Classification Performance with PCA:

- Logistic Regression:
    - Pre-PCA accuracy: 97.37%
    - Post-PCA accuracy: 98.25%
    - Improvement demonstrates PCA's ability to filter noise and redundancy.
- SVM: Achieved an accuracy of 97.37%.
- Random Forest and kNN: Both achieved 95.61% accuracy post-PCA.
- Confusion matrices for all classifiers indicated an improved balance between sensitivity and specificity after dimensionality reduction.

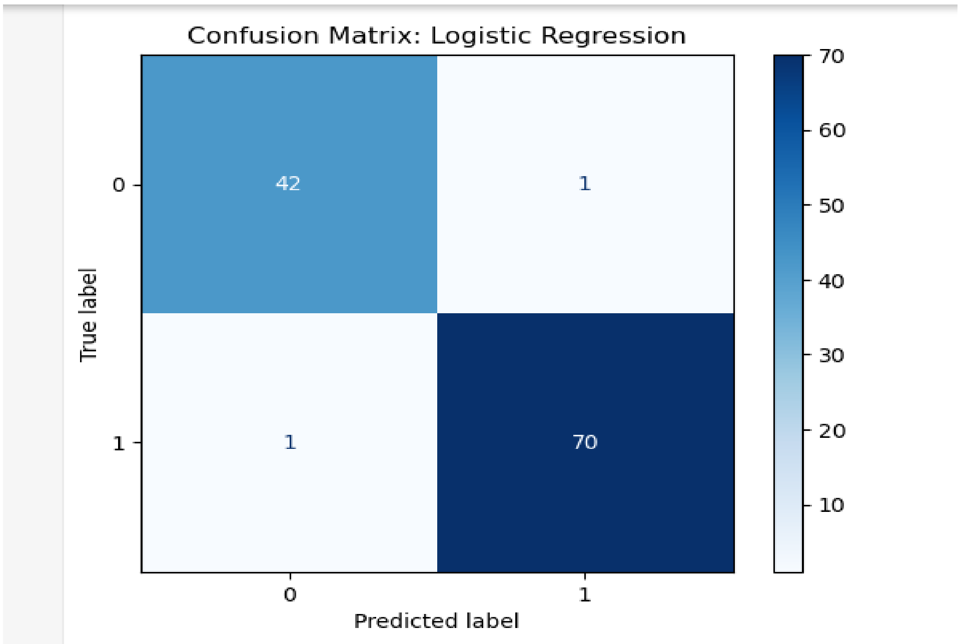


Classification Accuracy Without PCA

```
[15]: # Classification Accuracy Without PCA
X_train, X_test, y_train, y_test = train_test_split(X_normalized, y, test_size=0.2, random_state=42)
clf = LogisticRegression(random_state=42)
clf.fit(X_train, y_train)
accuracy_without_pca = accuracy_score(y_test, clf.predict(X_test))
print(f'Accuracy without PCA: {accuracy_without_pca:.4f}')
```

```
Accuracy without PCA: 0.9737
```

The logistic regression model achieved 97.37% accuracy without PCA, indicating strong performance on the full dataset with all features.



```
Logistic Regression Accuracy: 0.9825
```

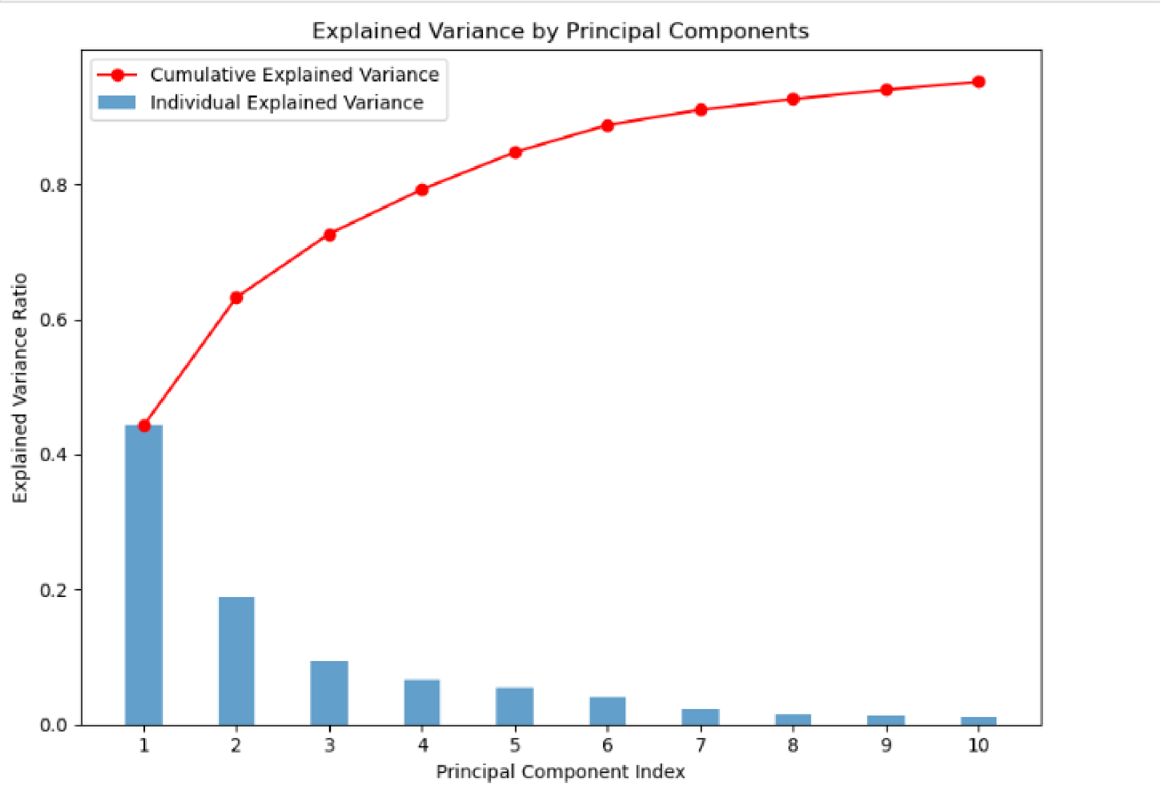
In-Depth Performance Analysis:

1. Hyper-Parameter Study:

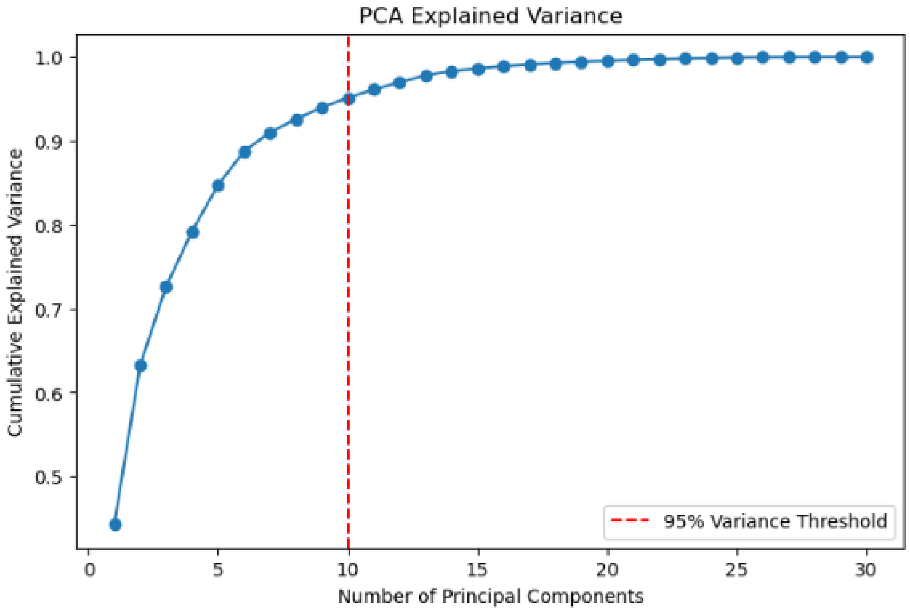
- The effect of k (number of retained components) on accuracy was analysed:
- For k=2, accuracy peaked at 99%, showing that the first two components captured the majority of essential variance.
- Beyond k=10, accuracy stabilized around 98%, indicating that most of the dataset's meaningful information was already retained.

2. Visualization of Results:

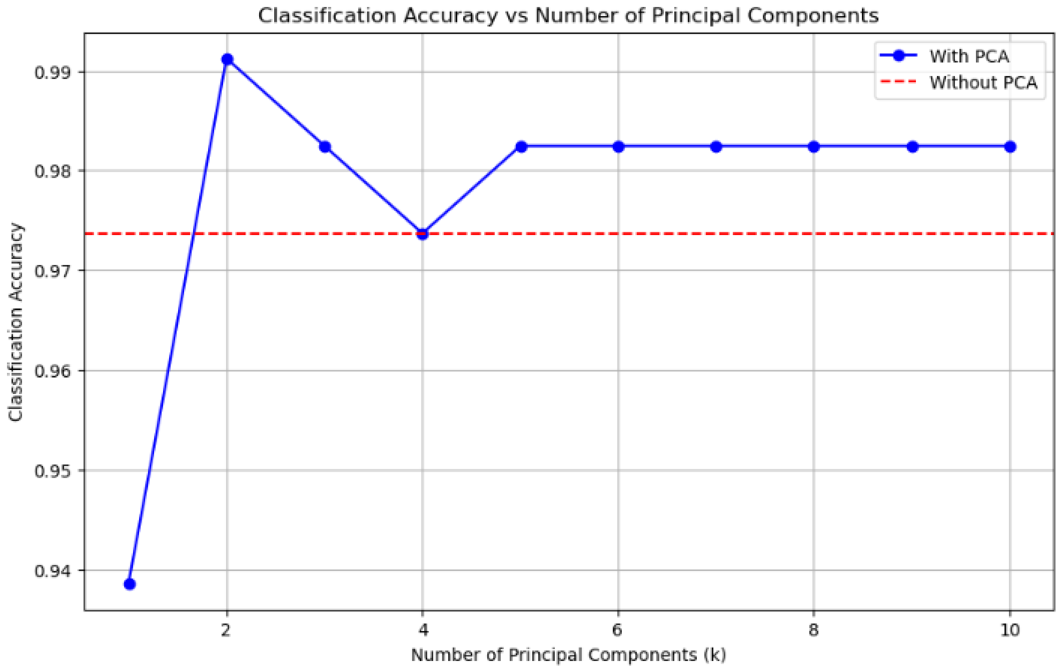
Explained Variance Plot: Showed the variance explained by each component, highlighting significant contributions from the first few components and diminishing contributions from higher-order components.



Cumulative Variance Plot: Demonstrated how the first 10 components retained over 95% of the dataset's variance.



Accuracy vs. k: A graph depicting the trade-off between variance retention and computational efficiency. Accuracy was highest for k=2 and stable for k≥10.



4. Individual Contribution

Sandeep Vaddi: Data Preparation and Exploratory Data Analysis (EDA)

- Loaded the dataset, conducted preprocessing, and performed exploratory data analysis.
- Generated a correlation heatmap to identify redundant features and standardized the data for PCA.

Praveen Vaddi: Principal Component Analysis (PCA) Implementation

- Implemented PCA by calculating principal components, explained variance, and cumulative variance.
- Determined the optimal k components to retain the most significant variance.

Nagendra Neelima Korrapati: Classification and Model Training

- Trained and evaluated classifiers (Logistic Regression, SVM, Random Forest, and kNN) on both pre- and post-PCA datasets.
- Compared classification accuracy and analysed confusion matrices to assess performance improvements.

Sai Sri Venkatapathi Varma Kakarlapudi: Hyper-Parameter Study and Performance Analysis

- Conducted a hyper-parameter study on k to analyse its impact on accuracy and variance retention.
- Created graphs to visualize accuracy vs. k and diminishing returns for variance explained.

Leela Siva Rama Krishna Neelapala: Results Consolidation, Final Analysis and Report Writing

- Consolidated findings from all experiments, interpreted performance trends, and ensured consistency in the evaluation of PCA's effectiveness.
- Compiled the findings into a structured report, summarizing methodology, results, and insights from the experiments.

5. Conclusion

Dimensionality reduction using PCA effectively addressed the challenges of high-dimensional datasets, such as redundancy and inefficiency, by transforming the Breast Cancer dataset's 30 features into 10 principal components while retaining 95.16% of the variance and improving computational efficiency without significant information loss. Logistic Regression accuracy increased from 97.37% to 98.25% post-PCA, demonstrating reduced noise and redundancy. A hyper-parameter study showed optimal performance at k=10, where most essential variance was retained. PCA proved to be an effective tool for dimensionality reduction, enhancing classification performance while simplifying data.