Overview:

Dimensionality reduction is a essential technique in the data science, especially when working with the datasets which contains a large number of features. As the number of features in the dataset increases, it will become challenging to process, interpret, visualize the data and increases the computational costs. Principal Component Analysis(PCA) is a powerful method for addressing these issues by identifying the most important patterns in the data, PCA reduces the number of features while retaining the majority of the information. In this project, we have utilized the PCA to simplify the Wine dataset, which contains chemical properties of wines. Our objective is to improve visualization and the computational efficiency while maintaining the high classification performance. This project demonstrates the steps involved in applying PCA, evaluating its impact on machine learning models, and explores how it can enhance the data interpretation.

Problem Statement:

High-dimensional datasets often present several challenges. First, they are computationally expensive to process due to the large number of features. Second, visualizing patterns or relationships in such datasets is difficult because humans can only interpret data in two or three dimensions. Third, high-dimensional data may contain redundant or irrelevant features, which can negatively impact the performance of machine learning models. The Wine dataset, used in this project, exemplifies these challenges as it contains 13 features that describe the chemical composition of wines. These features can overlap in the information they provide, making dimensionality reduction necessary. This project aimed to apply PCA to reduce the dimensionality of this dataset while retaining the information required to perform effective classification and visualization.

How PCA and SVD Solve It:

PCA reduces the dimensionality of a dataset by identifying the new features, known as principal components, which are linear combinations of the original features. These principal components capture the maximum variance in the data while being uncorrelated with each other. PCA achieves this transformation using Singular Value Decomposition (SVD), a mathematical technique that decomposes the data matrix into three components: U, Sigma and $V^{T.}$ The singular values in Sigma quantify the importance of each principal component by measuring how much variance it explains. By truncating the smaller singular values, we can retain only the most significant components, which represent the key patterns in the data. This process effectively reduces the number of dimensions while preserving the dataset's structure and essential information.

Methodology:

Visualization of Original Data

To understand the dataset structure, we begin by visualizing the original data. Scatterplots of the first two features provided a basic view of how the classes were distributed. However, significant overlap between classes made it challenging to discern meaningful patterns. A pairplot of all features allowed us to examine relationships across multiple dimensions, but the complexity of high-dimensional data still hindered clear interpretation. Additionally, a heatmap of the correlation matrix revealed that many features were strongly correlated, indicating redundancy in the dataset. These observations highlighted the need for dimensionality reduction.

Data Normalization
The next step implemented is to normalize the dataset using StandardScaler. Normalization ensures that all features have the same scale, preventing features with larger ranges from dominating the PCA process. This step is crucial because PCA is sensitive to the scale of the data, and unnormalized features could lead to inaccurate principal components. By normalizing the data, we have ensured that each feature contributed equally to the variance calculation.

Applying SVD
We have applied the Singular Value Decomposition to the normalized dataset to compute the principal components. The singular values obtained from SVD allowed us to calculate the explained variance for each component. The explained variance ratio revealed how much of the total variance in the dataset was captured by each principal component. This step provided a clear understanding of the importance of each component and guided the selection of the optimal number of components to retain.

Choosing the Number of Components
The cumulative explained variance plot was used to determine the number of components (k) required to retain 95% of the variance. By choosing k=10, we reduced the dataset from 13 features to ten principal components while preserving most of the information. This step ensured that we achieved dimensionality reduction without losing significant patterns in the data.

Projecting Data into Reduced Space
After determining the optimal number of components, we projected the dataset into the reduced space defined by these components. The reduced dataset was visualized using the first two principal components, which showed clear separations between wine classes. This visualization demonstrated how PCA simplifies the dataset and uncovers underlying patterns that were difficult to observe in the original feature space.

Classification Performance
To evaluate the impact of PCA on classification, we trained Logistic Regression and Random Forest models on both the original and reduced datasets. These models were chosen for their complementary properties: Logistic Regression as a linear classifier and Random Forest as a non-linear, ensemble-based approach. Classification accuracy was measured for various values of k, the number of principal components retained. This experiment helped us understand the relationship between dimensionality reduction and model performance.

Hyperparameter Study
We conducted a hyperparameter study to explore how the number of components (k) affects classification accuracy. By varying k from 2 to 10, we analyzed the trade-off between retaining fewer dimensions and maintaining high accuracy. Scatterplots of the reduced data for each value of k illustrated how the structure of the dataset changes with the number of components.

Experiments:

Setup

The experiments used the Wine dataset from the sklearn.datasets library, which has 178 samples divided into three wine classes based on their chemical composition. Each sample includes 13 numerical features, such as alcohol, malic_acid, ash, flavonoids, and total_phenols. The goal of the experiments was to reduce the number of features using PCA and analyze its effect on both data visualization and classification performance.

```
In [7]: df=pd.DataFrame(data["data"],columns=data["feature_names"])
        df
```

Out[7]:

	alcohol	malic_acid	ash	alcalinity_of_ash	magnesium	total_phenols	flavanoids	nonflavanoid_phenols	proanthocyanins	color_intensity	hue	od280/od3
0	14.23	1.71	2.43	15.6	127.0	2.80	3.06	0.28	2.29	5.64	1.04	
1	13.20	1.78	2.14	11.2	100.0	2.65	2.76	0.26	1.28	4.38	1.05	
2	13.16	2.36	2.67	18.6	101.0	2.80	3.24	0.30	2.81	5.68	1.03	
3	14.37	1.95	2.50	16.8	113.0	3.85	3.49	0.24	2.18	7.80	0.86	
4	13.24	2.59	2.87	21.0	118.0	2.80	2.69	0.39	1.82	4.32	1.04	
...	...	...	...	...	...	...	...	...	...	...	...	
173	13.71	5.65	2.45	20.5	95.0	1.68	0.61	0.52	1.06	7.70	0.64	
174	13.40	3.91	2.48	23.0	102.0	1.80	0.75	0.43	1.41	7.30	0.70	
175	13.27	4.28	2.26	20.0	120.0	1.59	0.69	0.43	1.35	10.20	0.59	
176	13.17	2.59	2.37	20.0	120.0	1.65	0.68	0.53	1.46	9.30	0.60	
177	14.13	4.10	2.74	24.5	96.0	2.05	0.76	0.56	1.35	9.20	0.61	

178 rows × 13 columns

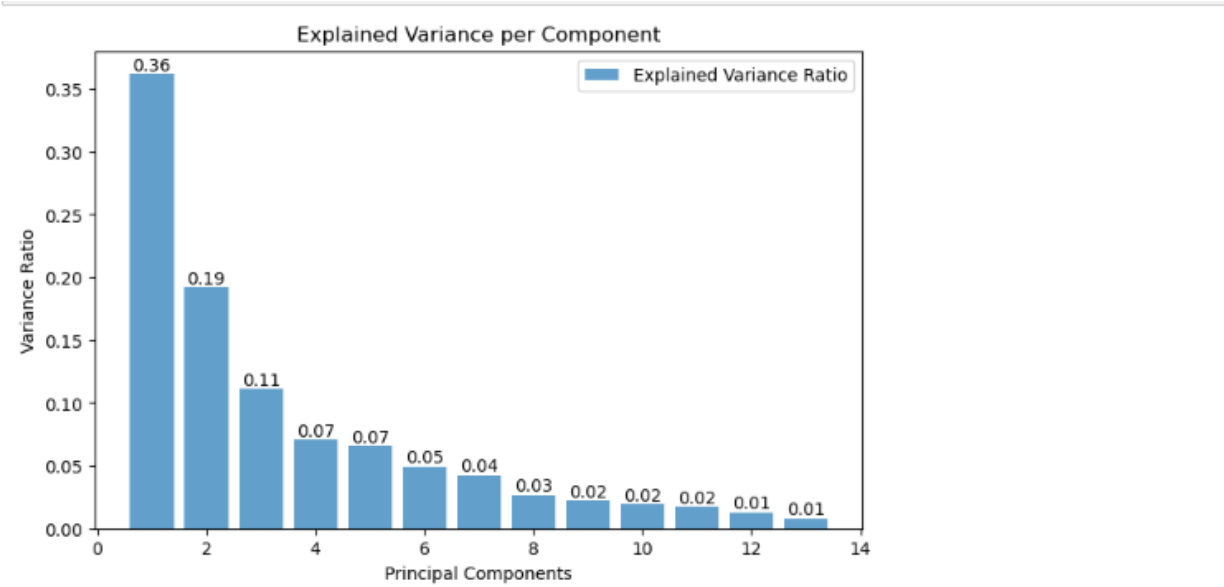
We used the following tools to carry out the experiments:

- Python for implementation and overall analysis.
- NumPy for performing mathematical operations like Singular Value Decomposition (SVD) for PCA.
- Scikit-learn for data preprocessing, PCA computation, model training, and accuracy evaluation.
- Matplotlib and Seaborn for visualizing data and results through scatterplots, pairplots, and heatmaps.
- The dataset was normalized using StandardScaler, which scales each feature to have a mean of zero and a standard deviation of one. Normalization was necessary because PCA relies on variance, and features with larger scales would dominate the process if left unscaled. PCA was implemented using SVD to extract principal components, which capture the most important patterns in the data.

The contribution of each principal component was measured using the explained variance ratio, which shows how much variance each component explains. The cumulative explained variance helped determine how many components (k) were needed to retain 95% of the total variance. Classification performance was tested using Logistic Regression and Random Forest models, and their accuracy was measured with and without PCA. A hyperparameter study was conducted by varying the number of principal components (k) from 2 to 10 to see how this affected accuracy.

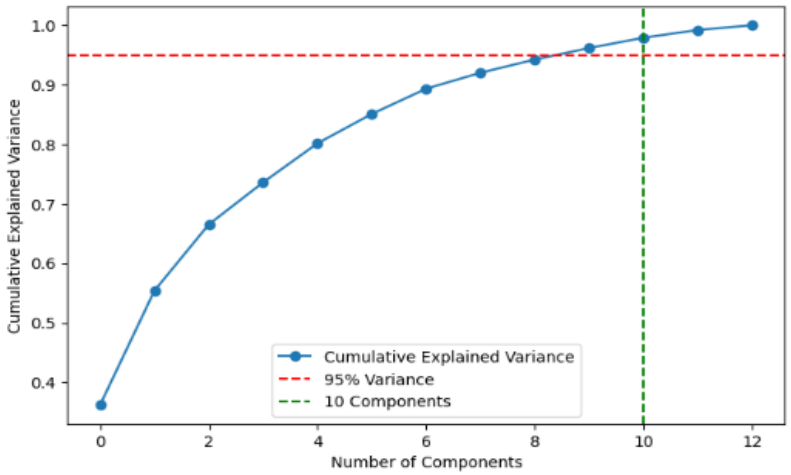
Results

The PCA analysis revealed that the first ten principal components captured over 95% of the total variance, meaning that reducing the dataset to ten dimensions retained nearly all the important information. The cumulative explained variance plot showed that after the tenth component, the additional variance explained by each component was minimal, confirming that ten components were enough to represent the dataset effectively.



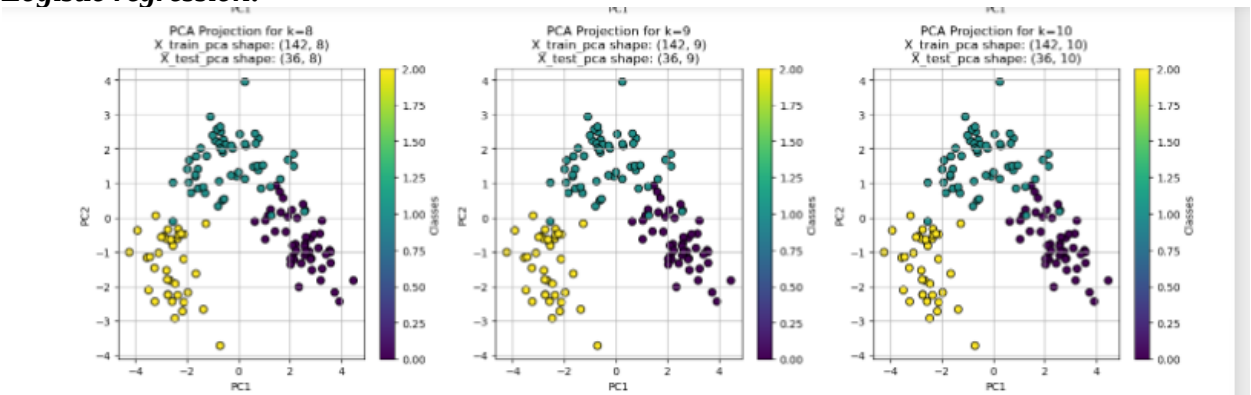

```
Cumulative Explained Variance: [0.36198848 0.55406338 0.66529969 0.73598999 0.80162293 0.85098116
 0.89336795 0.92017544 0.94239698 0.96169717 0.97906553 0.99204785
 1.        ]

Number of components chosen (95% variance): 10
```



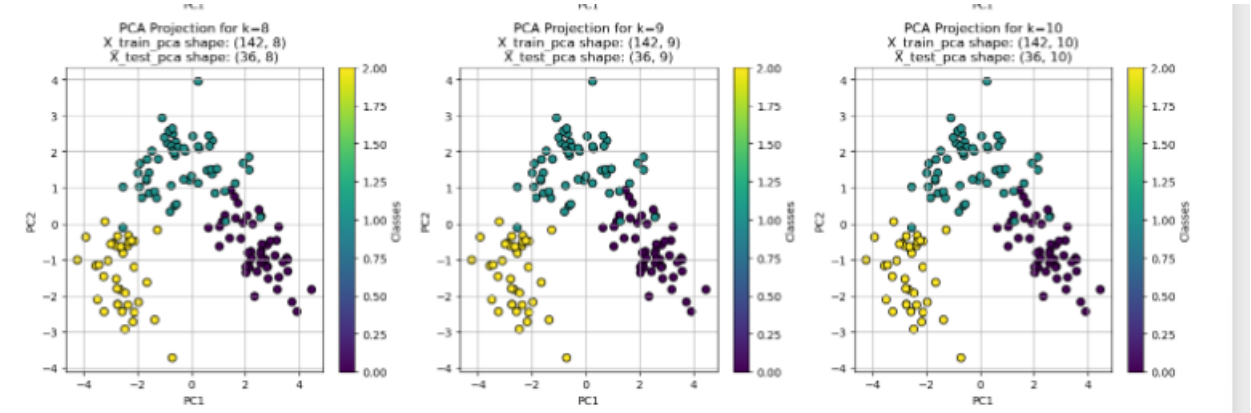
For classification, Logistic Regression achieved an accuracy of about 94-100% on the dataset reduced with PCA, compared to 98% accuracy on the original dataset. Similarly, the Random Forest classifier maintained an accuracy of 97-100% with PCA and 100% without PCA. These results showed that even though the number of features was reduced, the models still performed very well, as PCA kept the key information intact.

Logistic regression:



	k	Accuracy Without PCA	Accuracy With PCA
0	2	0.972222	1.000000
1	3	0.972222	0.972222
2	4	0.972222	0.972222
3	5	0.972222	0.972222
4	6	0.972222	0.944444
5	7	0.972222	0.944444
6	8	0.972222	0.944444
7	9	0.972222	0.972222
8	10	0.972222	0.972222

Random Forest Classifier:



	k	Accuracy Without PCA	Accuracy With PCA
0	2	1.0	0.972222
1	3	1.0	0.972222
2	4	1.0	0.972222
3	5	1.0	1.000000
4	6	1.0	0.972222
5	7	1.0	1.000000
6	8	1.0	0.972222
7	9	1.0	1.000000
8	10	1.0	1.000000

The hyperparameter study, where the number of principal components (k) varied from 2 to 10, showed that accuracy is varying across each k value as in the less components sometimes enough to demonstrate the darta, and sometimes more components adds outliers and forms redundant data.. This aligned with the explained variance results, which indicated that most of the dataset's information was captured within these 10 components. Scatterplots of the PCA-reduced data with different values of k showed better separation between wine classes as k increased, but the scatter plots has been shiwcased between first two principal components.

Another important result was the reduction in computational time. PCA helped reduce the training time for Random Forest, which is computationally expensive for datasets with many features. Overall, the experiments showed that PCA is a practical and effective tool for simplifying datasets while maintaining their structure and classification performance.

Conclusion:

This study highlights the value of PCA as a dimensionality reduction technique for simplifying the datasets, improving visualization, and reducing computational costs. By retaining the most significant components, PCA preserves the structure and variance of the data while discarding noise and redundancy. The classification results confirmed that PCA maintains high accuracy even with reduced dimensions, making it a powerful tool for preprocessing high-dimensional datasets. Future work could involve applying PCA to more complex datasets or exploring its integration with advanced machine learning pipelines.