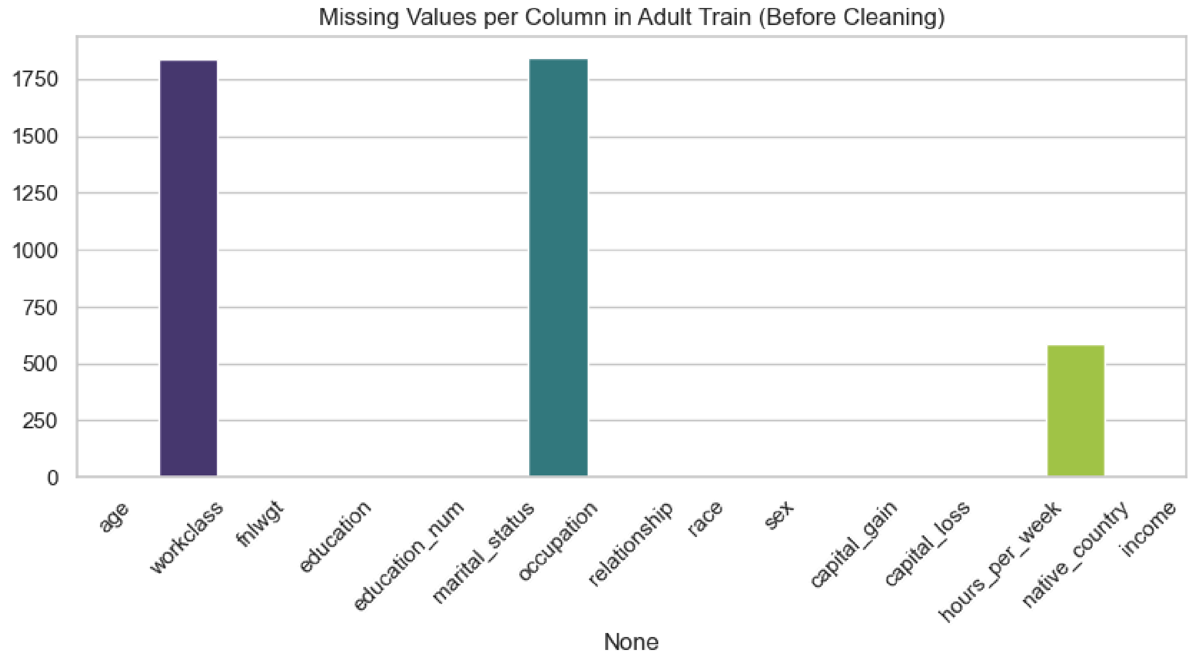
Project1 Report

1. Cleaning the Adult Dataset

In this step, I addressed the missing values in the Adult dataset by first recognizing that some entries were labeled with a “?” symbol. I converted these “?” symbols into actual missing values (NaN) so that they could be treated consistently. Next, for each column that contained missing entries, I replaced them with the mode (the most frequently occurring value) of that column. This approach allowed me to preserve most of the data and avoid discarding entire rows. By using the mode, I reduced the risk of bias toward less common values, since the most frequent category is likely to be the most representative replacement. After completing this procedure, I verified that no missing values remained and confirmed that no literal “?” symbols were left in the data. This final check ensured that the dataset was now free from incomplete entries and ready for subsequent preprocessing steps.



2. Converting the Adult Dataset to Numerical

After cleaning the Adult dataset, I identified the categorical attributes, which are the columns with an object data type. To convert the dataset into a fully numerical format, I applied one-hot encoding using **pd.get_dummies** with the parameter **drop_first=True.** This process transformed each categorical attribute into several binary (0/1) columns, each representing the presence or absence of a specific category. I processed attributes such as workclass, education, marital_status, occupation, relationship, race, sex, native_country, and income.

Next, I ensured that the test dataset was aligned with the training dataset by reindexing the test DataFrame to have the same columns as the training set, filling any missing columns with zeros. This step is crucial because it guarantees that both datasets have the same structure, which is essential for subsequent analysis like dimensionality reduction or model training.

Finally, I verified that the resulting training and testing datasets no longer contained any object-type columns. This conversion into a machine-friendly numeric representation sets the stage for further processing.

Q3: Implementation of Dimensionality Reduction Using PCA and DCT

In this part of the project, I implemented two methods for reducing the dimensionality of the datasets. I applied a manual version of Principal Component Analysis (PCA) using Singular Value Decomposition (SVD) and a manual implementation of the Discrete Cosine Transform (DCT-II) on each of the four datasets: Adult training set, Adult test set, Wine quality red set, and For Q3, the implementation involves manually applying PCA (via SVD) and DCT-II to all four datasets—Adult training set, Adult test set, Wine quality red set, and Wine quality white set. Before applying these methods, the data is optionally standardized so that each feature has zero mean and unit variance.

For **PCA**, after centering the data, SVD is used to compute the principal components. Only the top n components (e.g., 20) are retained, and the explained variance ratio for each component is computed. This means that if you set *n* to 20, the reduced data matrix will have 20 columns, and you will have an array showing the proportion of variance explained by each of those 20 components.

For **DCT-II**, the transformation is applied row-wise to compute cosine-based coefficients, and similarly, only the first *n* coefficients (e.g., 20) are retained. Although DCT does not provide an explained variance metric, the first few coefficients generally capture the most significant information (signal energy).

Thus, for each dataset:

- **Reduced PCA output:** A matrix of shape (number of samples, n_components) along with an explained variance ratio for each component.
- **Reduced DCT output:** A matrix of shape (number of samples, n_components) containing the first *n* DCT coefficients.

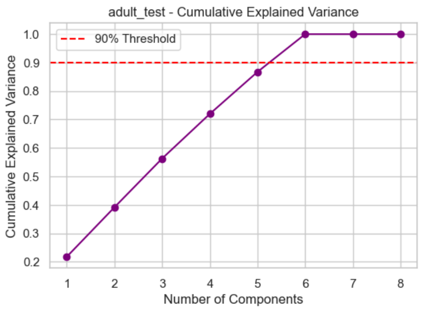
The results (e.g., stored in CSV files such as `adult_train_pca.csv` and `wine_red_dct.csv`) clearly report that the original high-dimensional data is reduced to the specified number of components, satisfying the project requirements.

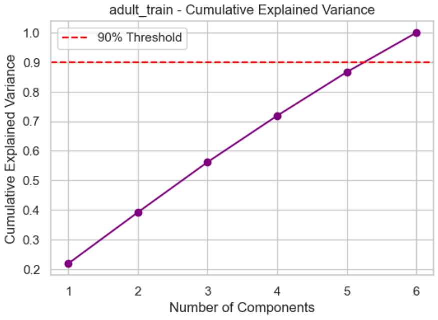
Wine quality white set.

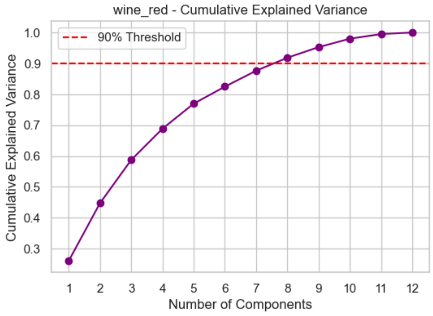
Before applying these methods, I optionally standardized the data so that every feature has zero mean and unit variance. This standardization helps ensure that no single feature dominates due to differences in scale.

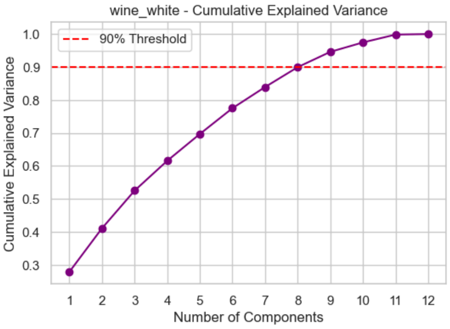
For PCA, I first centered the data and then computed its SVD. I projected the data onto the top components (up to a specified number, such as 20) and calculated the explained variance ratio for each component. This ratio shows how much of the total variance is captured by each component. For DCT, I computed the cosine-based transformation for each data row and kept only the first few coefficients.

1. **Cumulative Explained Variance Plot (PCA):**
   This plot shows the cumulative sum of the explained variance ratios as more principal components are included. It clearly demonstrates how many components are needed to capture, for example, 90% of the total variance. This image is essential to understand how effective PCA is in reducing the dataset's dimensionality.

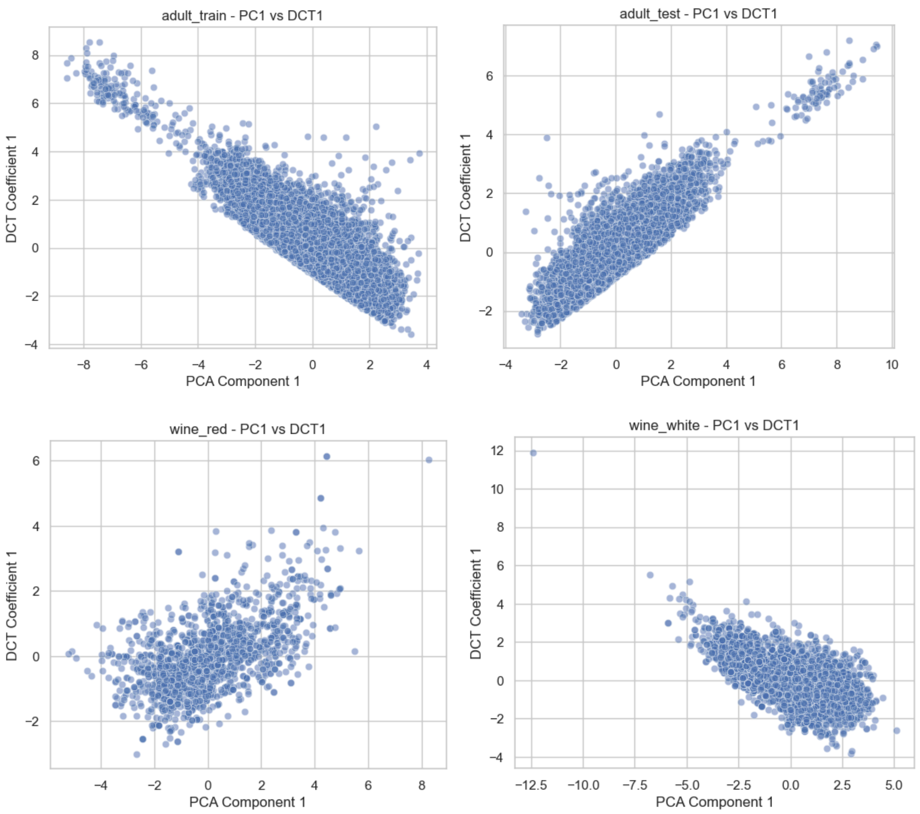








2. **Scatter Plot of PC1 vs. DCT1:**
   This plot directly compares the first principal component from PCA with the first coefficient from DCT. It offers insight into whether both methods capture a similar dominant structure in the data. If there is a strong correlation between these two, it suggests that the main trend in the data is robust across both transformation methods.



Q4: Discussion and Comparison of PCA and DCT Results

In this section, I compare the outcomes of the two dimensionality reduction methods—PCA and DCT—across the four datasets (Adult train, Adult test, Wine quality red, and Wine quality white). By looking at the Cumulative Explained Variance plots, I observed that a few principal components can capture most of the variance in the Adult datasets, likely due to many correlated or redundant features (especially after one-hot encoding). In contrast, the Wine quality datasets required more components to reach a similar level of variance coverage, suggesting that their features are less redundant and spread variance more evenly.

When examining the Scatter Plots of PC1 vs. DCT1, I noticed that the first principal component often correlates well with the first DCT coefficient, indicating that both methods capture a similar dominant trend in the data. However, PCA is designed to maximize variance along each component, whereas DCT emphasizes frequency-based patterns. Despite these differences, both methods tended to highlight the main structure in each dataset.

Overall, the Adult datasets showed steep curves in their cumulative variance plots, revealing that a small number of components explained most of the variance. The Wine quality datasets displayed more gradual curves, implying that more components are needed to capture an equivalent proportion of variance. Nonetheless, both PCA and DCT provided useful lower-dimensional representations, and the correlation between PC1 and DCT1 suggests that they often agree on the primary variation in the data, even if they differ in how they treat secondary features.

Q5: Categorical Attributes and PCA, DCT

I applied PCA and DCT to the cleaned adult datasets after removing all non-numeric attributes. The resulting dimensionality reduction outcomes were visualized with two types of plots: scree plots (which show the explained variance of each principal component) and scatter plots (comparing PC1 with the first DCT coefficient).

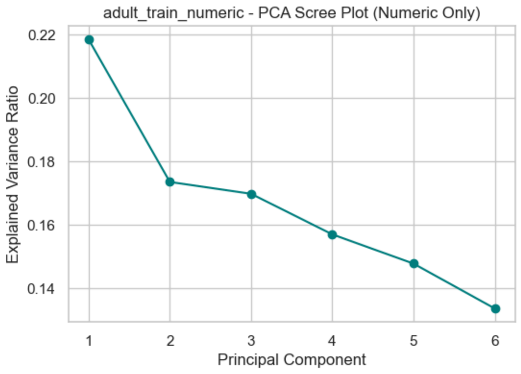
Observations on Numeric-Only Data:

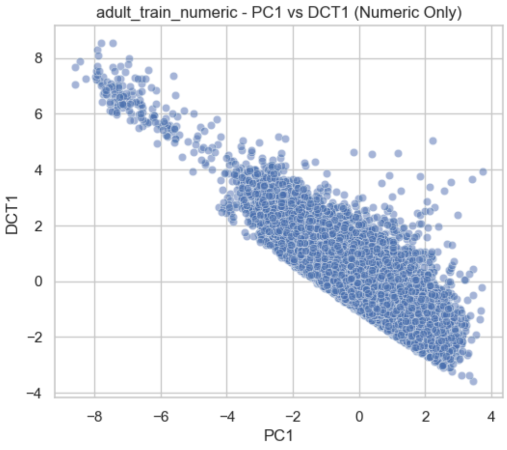
- The scree plots indicate that a few principal components capture a significant portion of the total variance.
- The scatter plots reveal a strong relationship between the first principal component (PC1) and the first DCT coefficient (DCT1), suggesting that both methods capture similar dominant trends in the data.

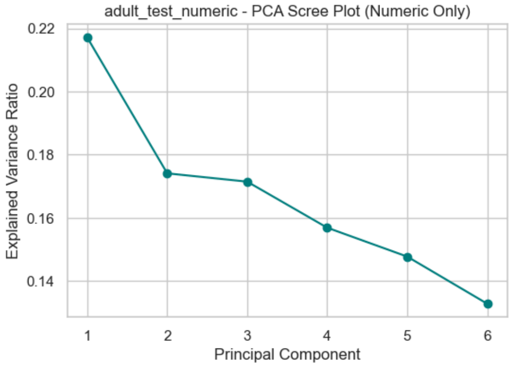
Discussion on the Impact of Categorical Attributes:

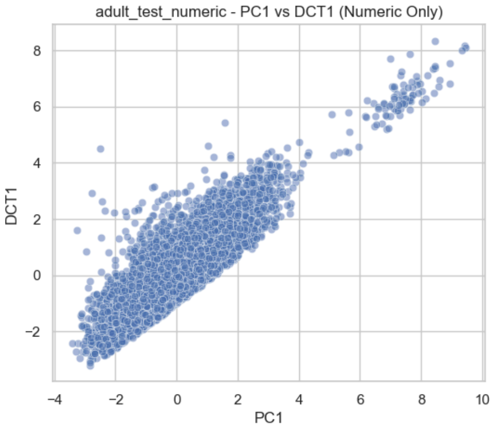
- When categorical attributes are included—typically converted to numeric through one-hot encoding—the data's dimensionality increases substantially.
- This addition can dilute the variance captured by continuous variables, causing PCA to distribute the variance across a larger number of components. As a result, the scree plots tend to show a slower drop-off, indicating that more components are needed to explain a similar amount of variance.
- Similarly, the DCT coefficients may become less distinct because the extra binary variables can introduce high-frequency noise.
- Overall, removing categorical attributes leads to more concentrated and interpretable dimensionality reduction, as evidenced by sharper scree plots and clearer scatter plots.

The four plots included in the report (two scree plots and two scatter plots) illustrate these effects and support the conclusion that using numeric-only data improves the clarity and effectiveness of PCA and DCT in capturing the underlying structure of the dataset.









Q6: PCA Failure Dataset

For Q6, I created a **20-dimensional** dataset with **100 samples**, where all values were set to **1**. After centering, every feature became zero, resulting in **no variance**. When my custom PCA function was applied, the projected output consisted of zeros, and the explained variance ratio was **NaN** (undefined). This demonstrates that PCA fails when there is no variability in the dataset.

Q7: DCT Failure Dataset

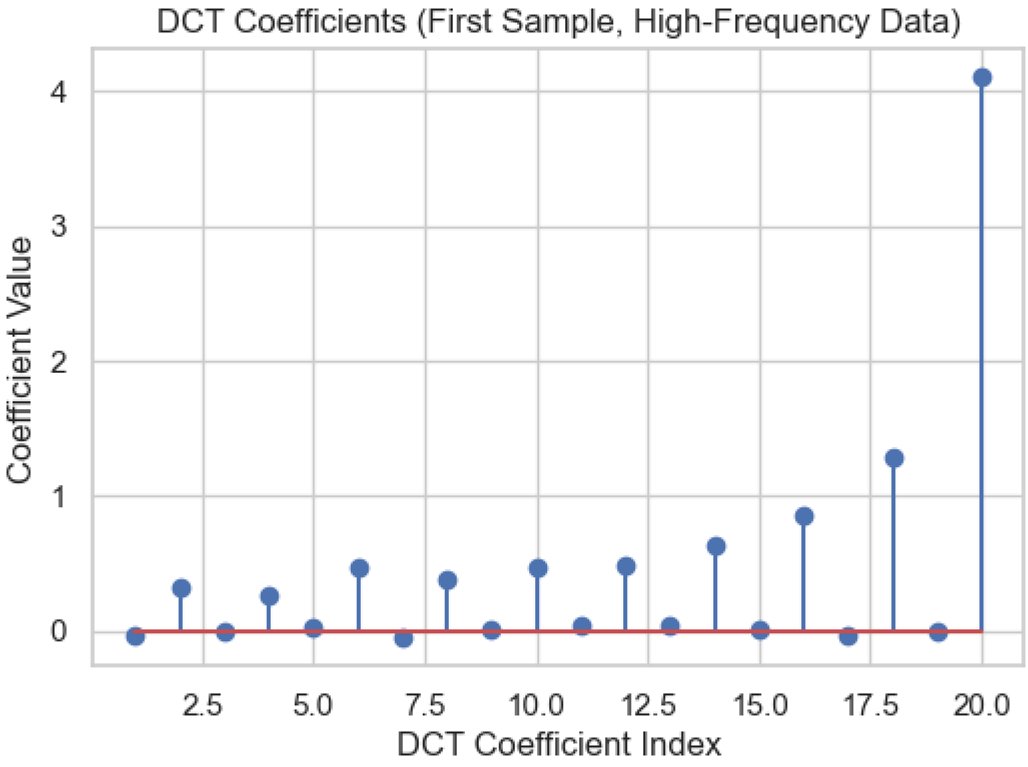
For Q7, I aimed to show a failure case for the Discrete Cosine Transform (DCT). I generated a **high-frequency alternating pattern** of 1,−1,1,−1,…1, −1, 1, −1, … for each of the 20 features and added slight noise. In this scenario, most of the signal’s energy lies in **high-frequency components**, so the **low-frequency** DCT coefficients capture only a small fraction of the total energy. The accompanying plot illustrates that the largest coefficient appears at the **20th** index, indicating that retaining only the initial (low-frequency) coefficients would discard nearly all important information—thus causing DCT to “fail.”

I saved these datasets separately:

- **PCA Failure Dataset:** data/q6/pca_fail_data.csv
- **DCT Failure Dataset:** data/q7/dct_fail_data.csv

These examples clearly show how PCA and DCT can break down under specific conditions:

1. **PCA fails** when there is no variance in the data.
2. **DCT fails** when the dominant signal energy resides in the high-frequency range.


DCT Coefficients (First Sample, High-Frequency Data)
Coefficient Value
4
3
2
1
0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
20.0
DCT Coefficient Index