**Applied Machine Learning**

**Lab Report 11**

**Hafiz Ahmad**

**19l-1316**

**Section-8A**

**INTRODUCTION:**

We investigate the concept of Principal Component Analysis (PCA) as a method for reducing dimensionality in this experiment. Dimensionality decrease strategies are utilized to diminish the quantity of elements or factors in a dataset while protecting fundamental data. PCA, a well-known method, uses a new coordinate system to find the most important features in the data.

**OBJECTIVES:**

The primary objective of this experiment is to apply Principal Component Analysis (PCA) to a provided dataset and examine how it effectively reduces the dimensionality of the data while retaining important information. By conducting this experiment, we seek to gain insights into the advantages and limitations of PCA in the context of feature selection and data visualization.

**Procedure:**

Modules and Libraries: Importing Importing various libraries and modules, including pandas, seaborn, MinMaxScaler, LabelEncoder, mutual\_info\_classif, VarianceThreshold, joblib, SFS, LinearRegression, pyplot, numpy, and PCA, is the first step in the experiment. Data preprocessing, visualization, feature scaling, encoding, feature selection, model persistence, linear regression, plotting, numerical computation, and principal component analysis are all features of these libraries and modules.

Preprocessing of Data: The dataset is imported, and explicit sections, specifically 'Value', 'Rooms', 'Washrooms', and 'Region', are switched over completely to drift information type. Column data types and other details about the DataFrame are displayed.

How to deal with missing values: The dataset is checked to see if there are any missing values. The missing values in some columns are filled in with either the mode (for categorical columns) or the mean (for numerical columns) after a copy of the DataFrame is made. Each column's count of missing values is looked at.

Encoding Absolute Segments: The LabelEncoder from scikit-learn is used to encode the categorical columns in the DataFrame. Each all out segment is fitted to the LabelEncoder, changing the section values into mathematical portrayals. After that, the encoded values are returned to their appropriate columns. The information of the DataFrame, including data types and non-null counts, is shown.

Missing Worth Proportion (MVR): The percentage of missing values in each column is represented by the missing value ratio. The names of columns whose MVR is less than or equal to 30% are saved in a "variable" list. The first few rows are displayed after the DataFrame is subdivided to include only the columns in the "variable" list.

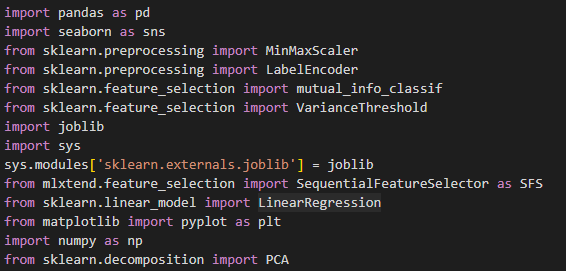
Filter for Low Variance: The dataset is processed with the low variance filter to find low variance features. After the 'Type' column is removed, the feature values are given to the variable X, and the target values from the 'Type' column are given to the variable Y. The estimator for the initialization of the Sequential Feature Selector (SFS) is the LinearRegression() model. SFS is arranged to choose 5 elements utilizing forward determination and the 'r2' scoring metric. The names of the selected features are retrieved after the feature selection process.

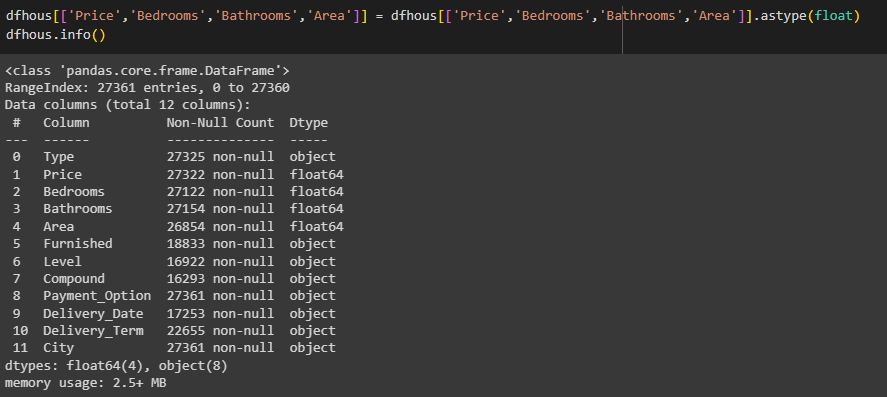
Reverse Propagation: The feature selection is further refined using backward propagation.

PCA: PCA is applied to decrease the dimensionality of the list of capabilities. The 'X' variable contains the element values, while the 'Y' variable holds the objective qualities. In order to reduce the feature set to three principal components, PCA is instantiated with "n\_components=3." The "fit\_transform" method is used to obtain the transformed PCA features. The components of the first and changed highlight sets are printed. Another DataFrame 'pca\_df' is made with the changed PCA highlights and the 'Type' values. The top of the 'pca\_df' DataFrame is shown.

Bar chart for the PCA: The PCA analysis's findings are represented by a bar graph.

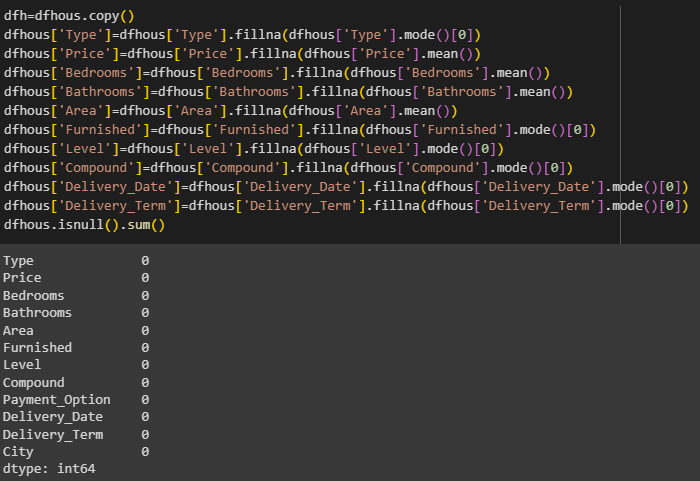
elemental analysis.



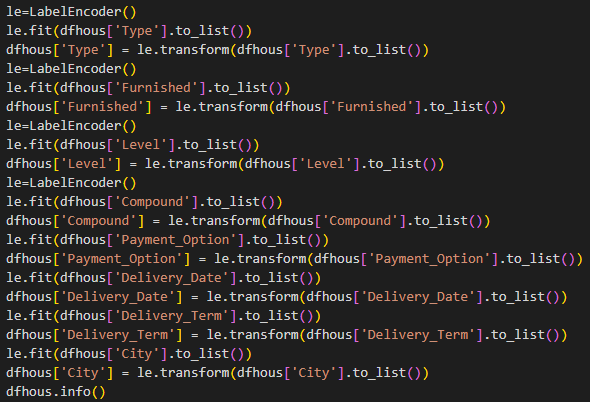


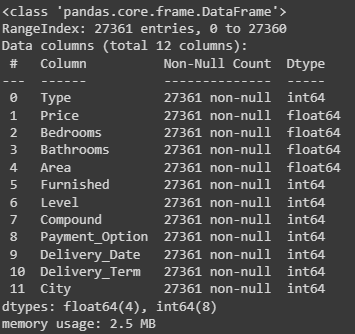
A screenshot of a computer

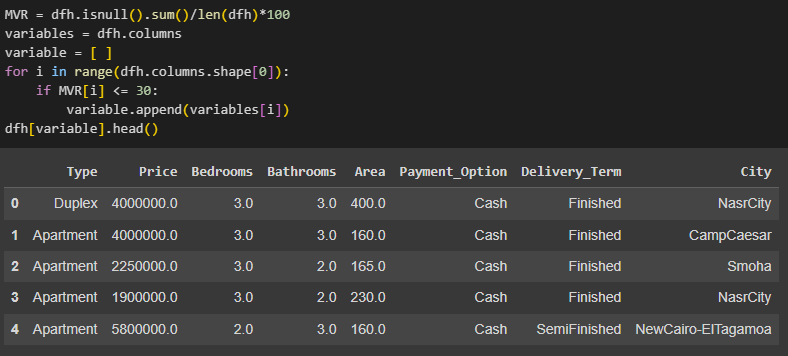
Description automatically generated with medium confidence

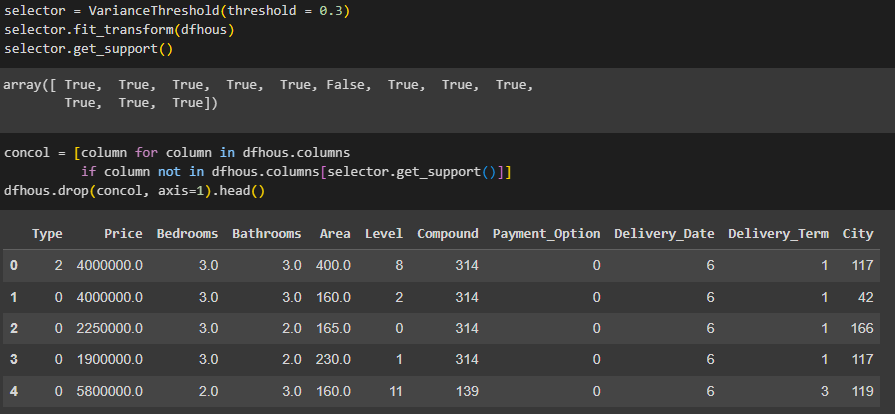
A screenshot of a computer

Description automatically generated







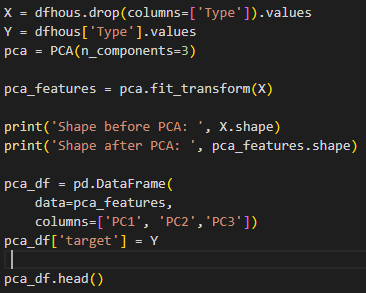


**A screen shot of a computer

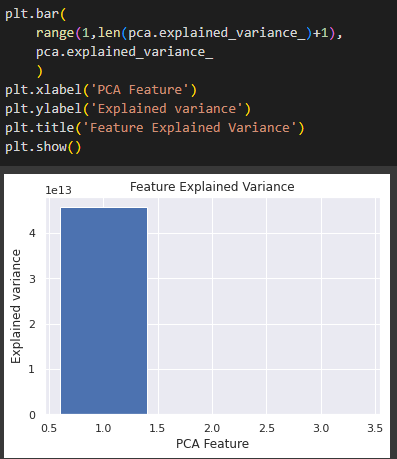
Description automatically generated with medium confidence**

A picture containing text, screenshot, font

Description automatically generated

A screenshot of a computer

Description automatically generated with medium confidence



**Application:**

PCA (Principal Component Analysis) offers several applications in data analysis, including:

1. Feature Selection: PCA aids in identifying the most important features in a dataset. By selecting the top principal components with the highest variances, PCA can help reduce the dimensionality of the data while retaining the most significant information. This feature selection process can lead to improved model performance, as irrelevant or redundant features are excluded, and computational efficiency is enhanced.

2. Data Visualization: PCA enables the visualization of high-dimensional data in a lower-dimensional space. By projecting the data onto a reduced number of principal components, which capture the most significant variations, complex data structures can be represented and understood visually. This allows for the identification of patterns, clusters, and relationships among data points, aiding in exploratory data analysis and decision-making processes.

3. Noise Reduction: PCA can effectively reduce noise or redundant information present in the dataset. By eliminating the principal components with low variances, which often correspond to noise or uninformative features, PCA produces a more concise and informative representation of the data. This noise reduction step can enhance the quality of the dataset and improve subsequent analysis or modeling tasks.

**Issues:**

No issue was found while performing in the lab.

**Conclusion:**

PCA is a powerful method for reducing dimensionality that makes it possible to effectively display and represent data. By choosing the most useful elements, it works on complex datasets while holding urgent examples and designs. PCA can boost performance, interpretability, and computational efficiency in a variety of data analysis tasks when understood and applied.

**Post lab:**A screenshot of a computer

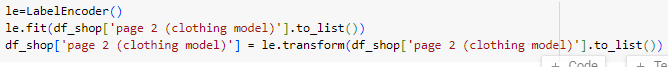
Description automatically generated with low confidence

A screenshot of a computer

Description automatically generated A screenshot of a computer

Description automatically generated with low confidence

A screenshot of a computer program

Description automatically generated with low confidence 

A screen shot of a computer code

Description automatically generated with medium confidence A screenshot of a computer

Description automatically generated with low confidence

A screenshot of a computer code

Description automatically generated with low confidence

A picture containing text, screenshot, rectangle, display

Description automatically generated