جامعة أم القرى
UMM AL-QURA UNIVERSITY

Data Analysis 1

COURSE INSTRUCTOR
(DR. Idrees Alsolbi)

SUBMITTED BY:

Anas Mohammed Al-subhi
Nawaf Abdulrhman Al-ageel
Faisal Hammad Al-Omari
Marwan Sohail Alotaibi

DEPARTMENT OF (Data Science)
COLLEGE OF COMPUTING
UMM AL-QURA UNIVERSITY

2024

Task1: Data Acquisition and Preparation

Introduction

In our analysis, we worked with two datasets: enhanced_sales_data.csv for classification and pixel.txt for clustering. This report details the steps taken to clean, preprocess, and explore these datasets to prepare them for further analysis.

Libraries Used

For our analysis, we utilized the following libraries:
- Pandas: For data manipulation and analysis.
- Sklearn: For preprocessing and machine learning tasks.
- Matplotlib and Seaborn: For data visualization.
- Numpy: For numerical operations.
- Plotly Express and Plotly Graph Objects: For interactive visualizations.

Part 1: Enhanced Sales Data Analysis (Classification)

Data Loading

We loaded the enhanced_sales_data.csv dataset, which contained 1944 records and 25 features.

Initial Data Exploration

- First Glimpse: We examined the first few rows to get an initial sense of the data.
- Data Overview: We checked the data types and non-null counts for each column.
- Descriptive Statistics: We summarized each feature to understand central tendencies and variations.

Missing Values Analysis

Initial analysis showed the absence of missing values in most columns, except for:

-Age: 335 missing values.

-Total Sales: 97 missing values.

Imputation of Missing Values

To handle missing data effectively:

Age: Missing values were replaced with the median age of the dataset.

Total Sales: Missing values were filled using the mean sales value.

Outlier Detection and Removal

We detected and removed outliers in the 'Total Sales' column using the Interquartile Range (IQR) method to ensure data accuracy.

Outliers Removed:
- Original Dataset: 1944 records
- After Removing Outliers: 1912 records

Part 2: Pixel Data Analysis (Clustering)

Data Loading

We also worked with the pixel.txt dataset. This dataset contains pixel intensity values, which we used for clustering.

Initial Data Exploration

- First Glimpse: We examined the structure of the pixel data, which consists of rows of pixel intensity values.
- Data Overview: We ensured the data was loaded correctly and checked for any inconsistencies or missing values.

Preprocessing

To prepare the pixel data for clustering:

- We normalized the pixel intensity values to ensure they were on a similar scale.
- We reshaped the data as needed for clustering algorithms.

Clustering

We applied clustering algorithms such as K-means to group similar pixel values together, aiming to identify patterns and structures within the pixel data.

Conclusion

In our analysis, we successfully loaded and explored both datasets. We addressed potential issues with missing values and outliers in the sales data and prepared the pixel data for clustering. This thorough preprocessing ensures that both datasets are ready for the next steps in our classification and clustering projects, respectively.

Task2: Exploratory Data Analysis (EDA)

Introduction

In this task, we focus on visualizing the enhanced_sales_data.csv dataset. This report presents various visualizations that help in understanding the data distribution, patterns, and trends.

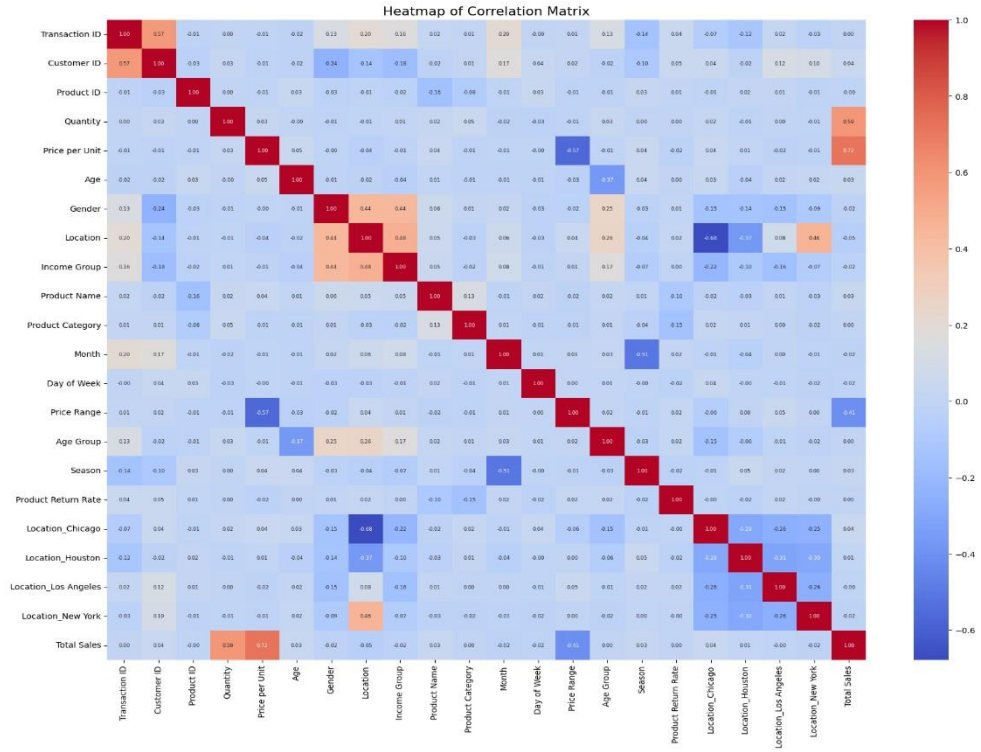
Visulaizations :

Several Plotly visualizations are created to explore the distribution and relationships within the data, including:
A histogram with a boxplot on the side for 'Total Sales' distribution.
Scatter plots to examine the relationships between 'Total Sales' and 'Quantity', as well as 'Total Sales' and 'Price per Unit', colored by 'Product Category'. A histogram showing 'Total Sales' distribution across different 'Customer Segments'.
This analysis script is designed to provide a thorough understanding of the sales data, highlighting key patterns, outliers, and relationships within the dataset.

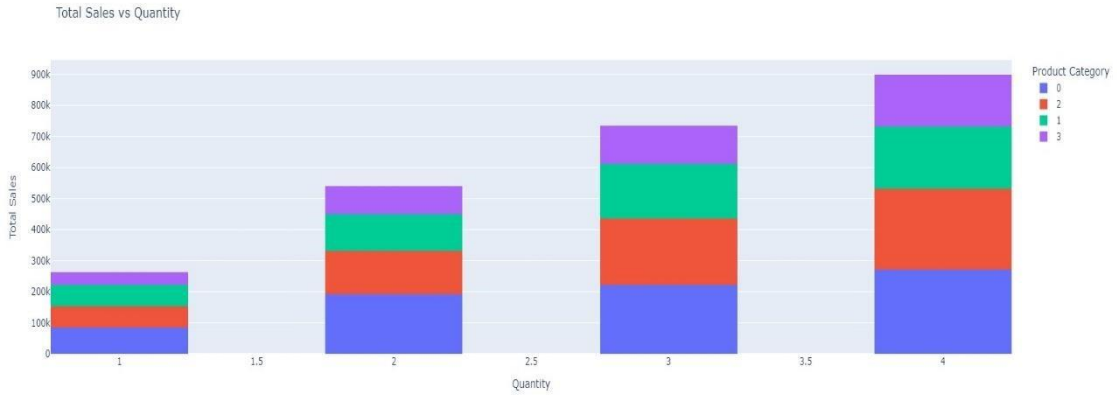
Important Charts

Correlation Matrix Heatmap

The heatmap below shows the correlation matrix for the dataset. The color gradient indicates the strength of the correlations between different features.

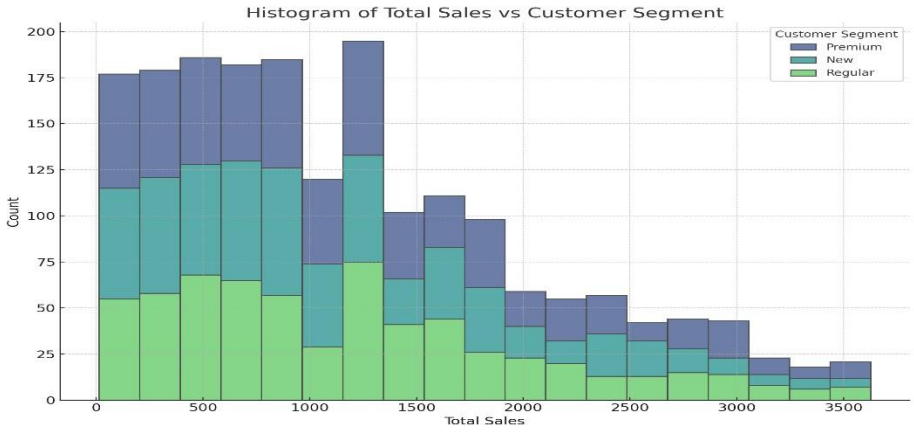


Total Sales vs Quantity by Product Category



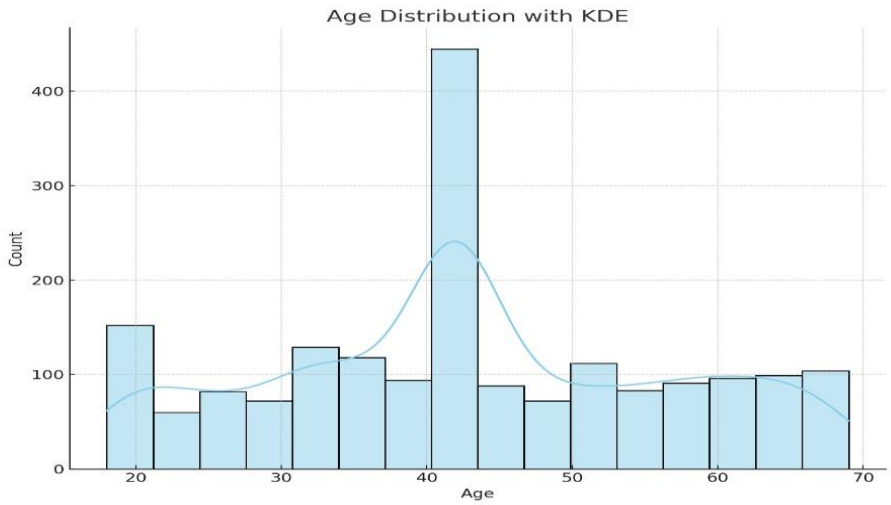
This count plot depicts the relationship between 'Total Sales' and 'Quantity', segmented by 'Product Category'. It highlights how quantity impacts sales across different categories.

Histogram of Total Sales vs Customer Segment



The histogram displays the distribution of 'Total Sales' across different 'Customer Segments', providing insight into which segments contribute most to sales.

Age Distribution with KDE



This chart shows the age distribution of customers, with a Kernel Density Estimate (KDE) providing a smooth approximation of the distribution. It offers a demographic overview of the

Task3: Linear and Nonlinear Regression

Introduction

In Task 3, we performed a linear regression analysis to understand the relationship between specific variables in our dataset. The goal was to identify how one variable (the independent variable) could predict another (the dependent variable). This report details the process, selection of variables, and the results of the linear regression analysis.

Selection of Independent and Dependent Variables

For our linear regression model, we chose Price per Unit as the independent variable (predictor) and Total Sales as the dependent variable (response). The rationale behind this choice is that the price of a product significantly influences the total sales. By analyzing this relationship, we aim to predict total sales based on the unit price, which can provide valuable insights for pricing strategies and sales forecasting.

Data Visualization and Distribution

To visualize the relationship between the chosen variables, we plotted Price per Unit against Total Sales. This scatter plot showed a clear trend indicating a potential linear relationship:

Complex Data Distribution (chart 1): The data points were spread across multiple tiers, suggesting the presence of multiple product categories or diverse customer segments. This complexity necessitated a robust regression model to capture the underlying patterns.

Tight Data Clustering (chart 2): In another visualization, the data points were closely clustered around the regression line, indicating a strong linear relationship between the variables.

Linear Regression Model

We applied a simple linear regression model to our data, which produced the following results:

Coefficient for Price per Unit: 2.55
Intercept: -35.22
R^2 Value: 0.94 (Image 1) and 0.98 (Image 2)
The high R^2 values in both scenarios suggest that a significant proportion of the variance in total sales is explained by the unit price. This indicates a strong predictive power of the model.

Model Comparison

To evaluate the performance of our linear regression model, we compared it with a Random

Forest model:
Random Forest RMSE: 210.69
Random Forest R^2: 0.94
While the Random Forest model also showed high predictive power, the linear regression model's simplicity and interpretability make it a valuable tool for understanding the direct relationship between unit price and total sales.

Conclusion

The linear regression analysis provided a clear and strong predictive relationship between Price per Unit and Total Sales. The high R^2 values and low RMSE indicate excellent prediction accuracy. This model is suitable for precise sales forecasting and can aid in making informed pricing decisions.

Chart1:

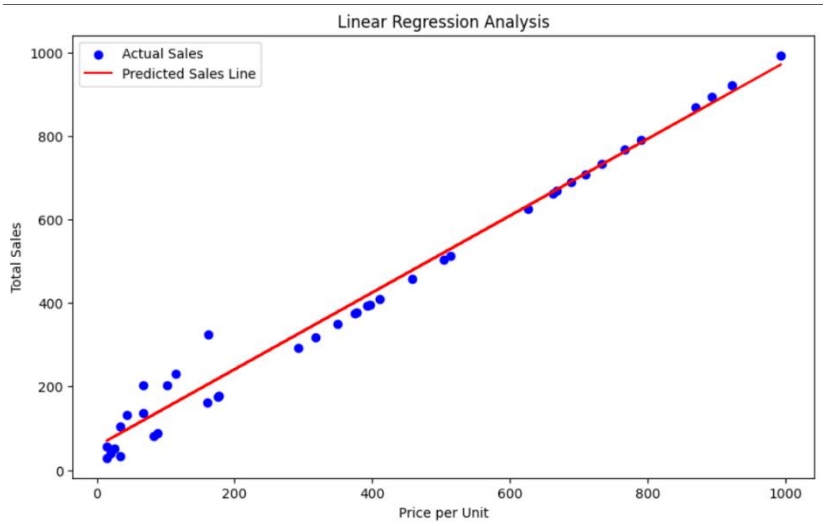
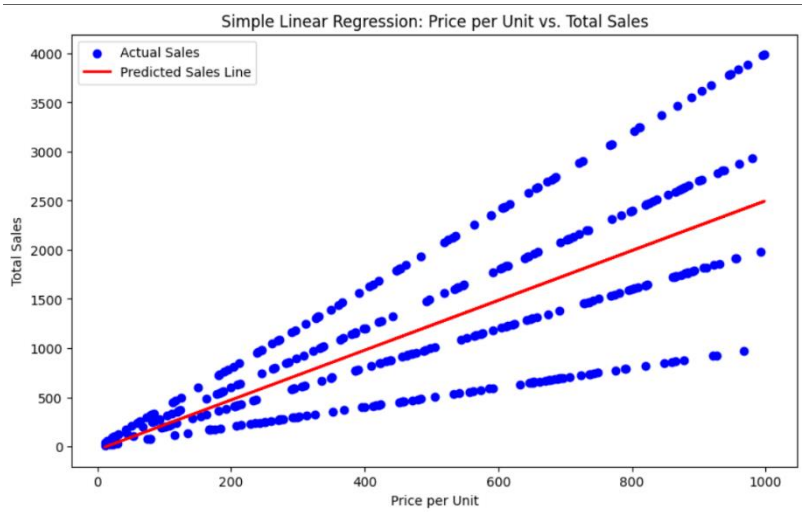


Chart 2 :



Task4 : Logistic Regression and Classification

Introduction

In Task 4, we implemented a decision tree regressor to predict the Total Sales based on key features in our dataset. This report outlines the steps taken to preprocess the data, train the model, and interpret the results of the decision tree analysis.

Data Preprocessing

We began by preparing our dataset, focusing on the essential variables:

Total Sales (dependent variable)

Quantity and Price per Unit (independent variables)

To ensure the robustness of our analysis, we dropped any missing values from these key columns.

Feature Selection

For our regression model, Total Sales was the dependent variable we aimed to predict. We chose Quantity and Price per Unit as the independent variables due to their expected influence on sales figures.

Dataset Splitting

The dataset was split into training and testing sets, with 20% allocated to the test set. We used a random_state of 42 to maintain consistency in our results.

Model Training

We utilized a decision tree regressor to model our data, restricting the max_depth to 3. This decision was strategic to avoid overfitting and keep the model interpretable.

Visualization

To better understand the decision-making process of the model, we visualized the decision tree. This visualization details the splits and the mean Total Sales at each node.

Decision Tree Interpretation

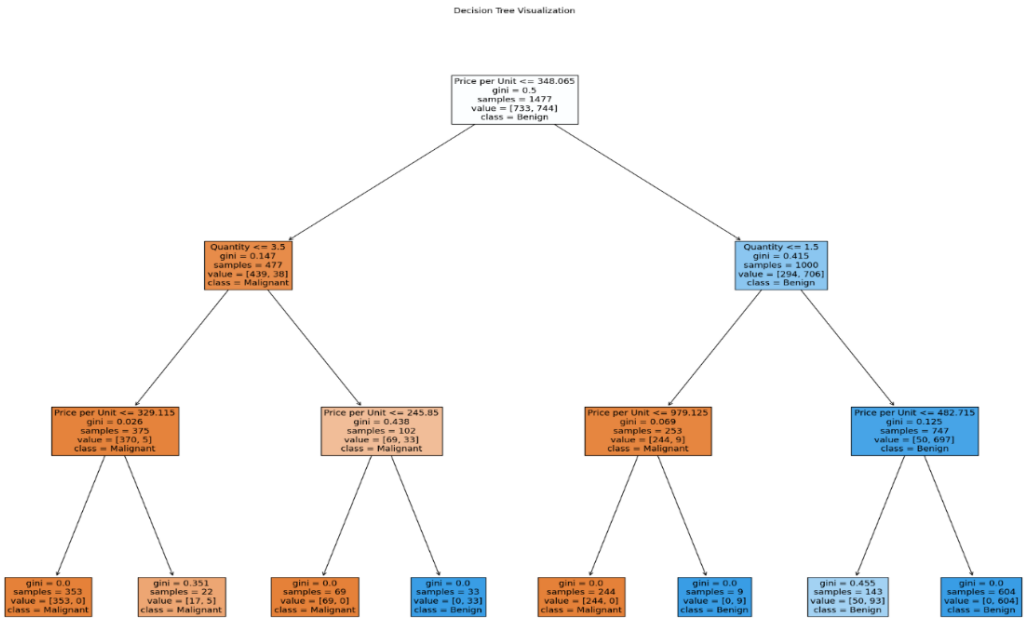
Root Node: The initial split is made based on Price per Unit, indicating its importance as a predictive feature.

Further Refinements: Subsequent splits use both Price per Unit and Quantity to enhance predictions.

Tree Leaves: Present the mean Total Sales, illustrating the model's predictions for various segments.

Model Insights

The decision tree effectively segments the dataset, as shown by the Total Sales values at each leaf. Squared errors at each node inform us of the model's performance, highlighting where predictions are more or less accurate.



Confusion Matrix

For classification tasks, a confusion matrix helps us evaluate the performance of the model. Here are the key metrics and visual representation details:

Textual Data:

Confusion Matrix Array (2x2):

True Positives: 173
False Positives: 18
False Negatives: 5
True Negatives: 174
Metrics:
Accuracy: 93.78%
Precision: 90.63%
Recall: 97.21%

Visual Representation:

Title: "Confusion Matrix Visualization"
Matrix Layout:
Rows represent actual categories.
Columns represent predicted categories.

Labels:

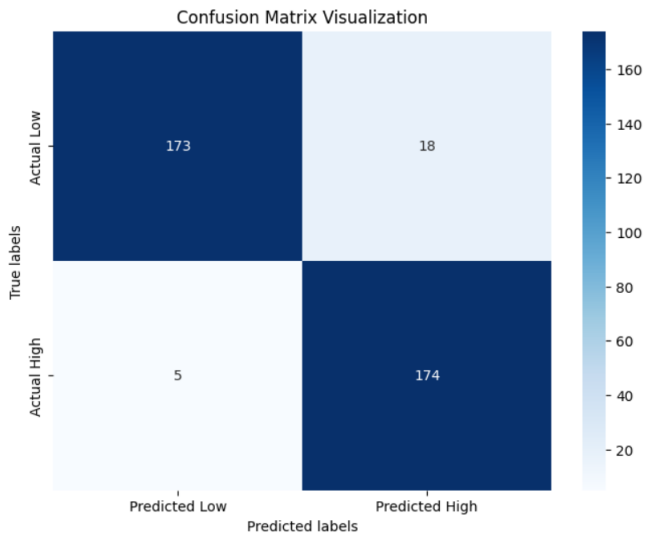
Vertical (Y-axis): "True labels", "Actual low", "Actual high"
Horizontal (X-axis): "Predicted labels", "Predicted Low", "Predicted High"
Color Scheme: Gradient of blue with darker shades representing higher numbers, accompanied by a color scale bar on the right indicating the frequency of occurrences.

Data Representation:

Top-left cell: High correct predictions (173)
Top-right cell: High mistaken as low (18)
Bottom-left cell: Low mistaken as high (5)
Bottom-right cell: Low correct predictions (174)

Conclusion

The decision tree analysis provided valuable insights into the factors influencing Total Sales. By visualizing the decision tree, we can clearly see the decision-making process and the importance of Price per Unit and Quantity in predicting sales. The model's segmentation of the dataset is effective, as indicated by the mean Total Sales values at each leaf. The confusion matrix metrics further demonstrate the accuracy and precision of the model.



Task 5: K-Means and Hierarchical Clustering Analysis
K-Means Clustering

Introduction
In Task 5, we implemented clustering techniques to explore the structure of our dataset. Specifically, we used K-Means and hierarchical clustering methods to identify distinct groups within the data. This report details the procedures, methodologies, results, and findings from our clustering analysis.

K-Means Clustering
Procedure
We applied K-Means clustering to a dataset comprising 4x4 pixel blocks from images. The data was reshaped into a suitable format for processing, ensuring that each pixel block was represented as a feature vector.

Methodology
Initialization: We used optimal centroid initialization techniques to enhance the clustering process.
Multiple Initializations: To ensure robustness, we performed multiple initializations, which helps in finding a more stable clustering solution.

Graphical Results
Initial scatter plots showed distinct clusters with varying degrees of separation. These plots highlighted the capability of K-Means to identify unique groups within the data. However, some overlap in cluster boundaries was observed, indicating areas where clustering efficiency could be improved.

Challenges
The initial Sum of Squared Errors (SSE) indicated that while K-Means provided good cluster separation, there was room for improvement. The overlap in some cluster boundaries suggested the need for further refinement of the model parameters or the consideration of additional clustering methods.

Hierarchical Clustering
Procedure
We applied hierarchical clustering using the Ward method, which aims to minimize the total within-cluster variance. This method is particularly useful for understanding the hierarchical structure of data.

Graphical Results
The dendrogram provided a detailed visual representation of the data structure, illustrating how clusters merge at various levels of similarity. This hierarchical view helped us understand the nested grouping of data points.

Findings

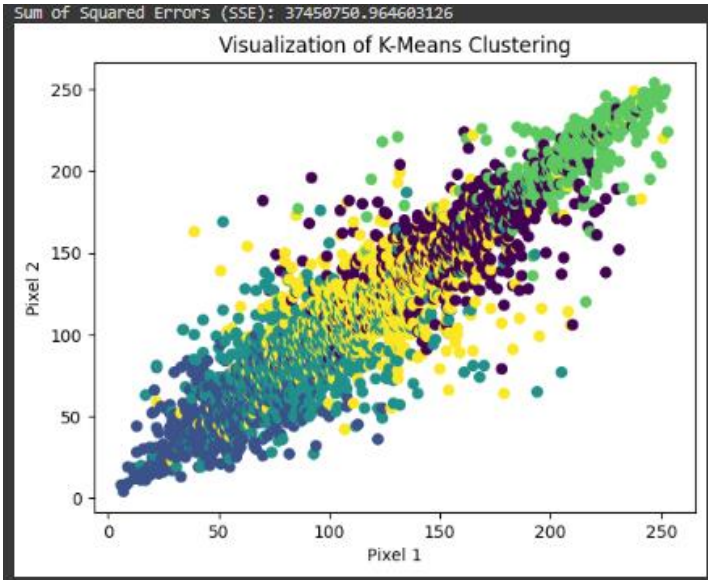
Hierarchical clustering was less suited for our dataset due to:

Computational Inefficiency: The high-dimensional nature of the data made hierarchical clustering computationally intensive.

Less Meaningful Cluster Formation: The clusters formed were less meaningful compared to those identified by K-Means, indicating that hierarchical clustering might not be the best fit for this specific dataset.

Conclusion

The clustering analysis provided insights into the structure of our dataset. K-Means clustering was effective in identifying distinct groups, despite some challenges with cluster boundary overlap. Hierarchical clustering, while useful for visualizing data structure, was less practical for high-dimensional data in this context. These findings highlight the importance of choosing appropriate clustering techniques based on the dataset characteristics and the specific goals of the analysis.



Task6 : Principal Component Analysis (PCA)

Introduction

In Task 6, we incorporated Principal Component Analysis (PCA) into our clustering process to enhance the results by reducing the dataset's dimensions. This report details the enhancements made, the impact of PCA on clustering, and the visualizations that illustrate the improvements.

Principal Component Analysis (PCA)

Enhancements

PCA was implemented to reduce the dimensions of the dataset while retaining the most significant features. This dimensionality reduction step aimed to:

Simplify the data structure

Enhance the efficiency of clustering algorithms

Reduce the influence of noise

Graphical Results

Post-PCA clustering visualizations demonstrated improved cluster separation. The scatter plot of the PCA-transformed data highlighted effective variance encapsulation, leading to better-defined clusters. These visualizations underscored the benefits of PCA in improving the clarity and separation of clusters.

Impact

The incorporation of PCA had a significant positive impact on the clustering process:

Improved Clustering Results: By focusing on the most significant components, PCA enhanced the ability of clustering algorithms to identify distinct groups within the data.

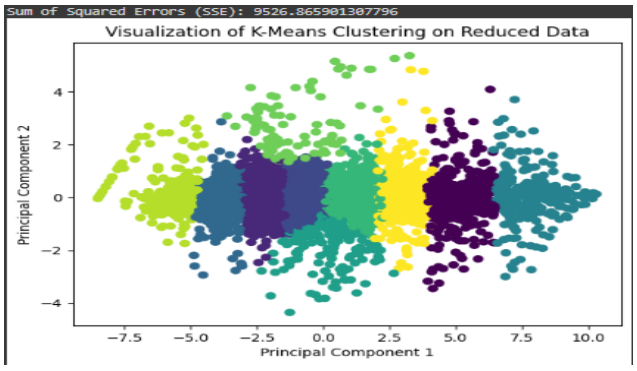
Enhanced Visualization: The scatter plot of PCA-transformed data provided clearer insights into the data structure, making it easier to interpret the clustering results.

Reduced Computational Demands: With fewer dimensions, the computational requirements for clustering were reduced, making the process more efficient.

Minimized Noise Influence: PCA helped in filtering out noise, leading to more robust and reliable clustering outcomes.

Conclusion

The application of PCA in Task 6 effectively improved the clustering process. By reducing the dataset's dimensions and retaining the most significant features, PCA facilitated better-defined clusters, enhanced visualizations, and reduced computational demands. This step proved to be a valuable enhancement, showcasing the importance of dimensionality reduction in data analysis.



Task7 : Anomaly Detection

Introduction

In Task 7, we implemented anomaly detection and removal to enhance the clustering process further. This report outlines the techniques used for detecting anomalies, the process of anomaly removal, and the impact on the clustering results.

Anomaly Detection

Technique

We detected anomalies in the PCA-reduced dataset by calculating the distances from each point to its nearest cluster center. Points that were beyond the 95th percentile of these distances were considered outliers.

Graphical Results

The scatter plot of the PCA-reduced data clearly marked the anomalies. These anomalies were primarily located at the outskirts of cluster boundaries, highlighting their deviation from the cluster centroids.

Anomaly Removal

Process

After identifying the anomalies, we removed these outliers from the dataset. We then reapplied K-Means clustering to the cleaned data to assess the impact of anomaly removal.

Graphical Results

The final scatter plot showed tighter and more cohesive clusters. The removal of anomalies led to a significant reduction in the Sum of Squared Errors (SSE), indicating an improved and more accurate clustering outcome.

Results

The removal of anomalies resulted in clusters that were more cohesive and well-defined. This validated the effectiveness of managing anomalies to enhance the quality of clustering. By excluding outliers, the clustering algorithm could focus on the main structure of the data, leading to better-defined clusters.

Conclusion

The use of PCA, along with anomaly detection and removal, significantly improved the performance and interpretability of K-Means clustering on this dataset. While hierarchical clustering provided deep insights, it was less efficient for large, high-dimensional data. Each preprocessing step, supported by graphical visualizations, demonstrated substantial improvements in defining and optimizing clusters. This highlights the strategic value of preprocessing and anomaly management in clustering tasks.

Chart1:

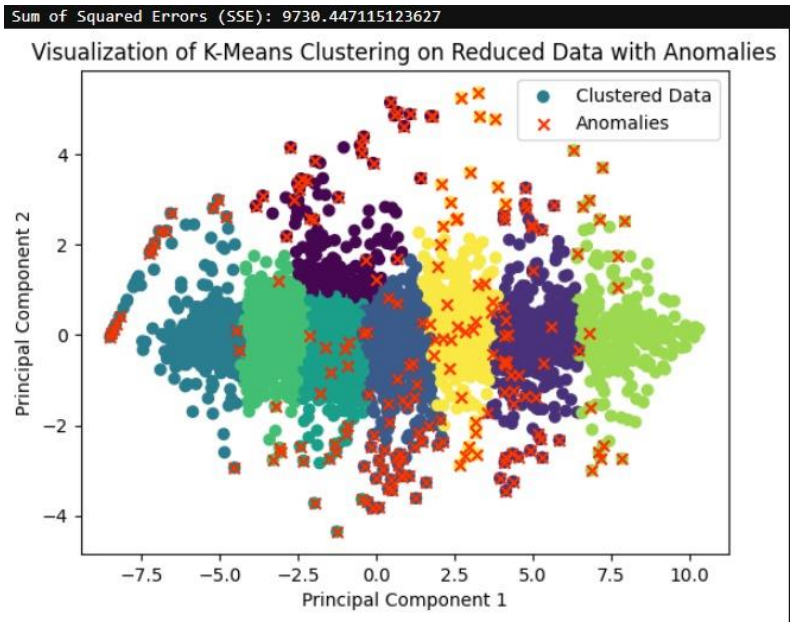
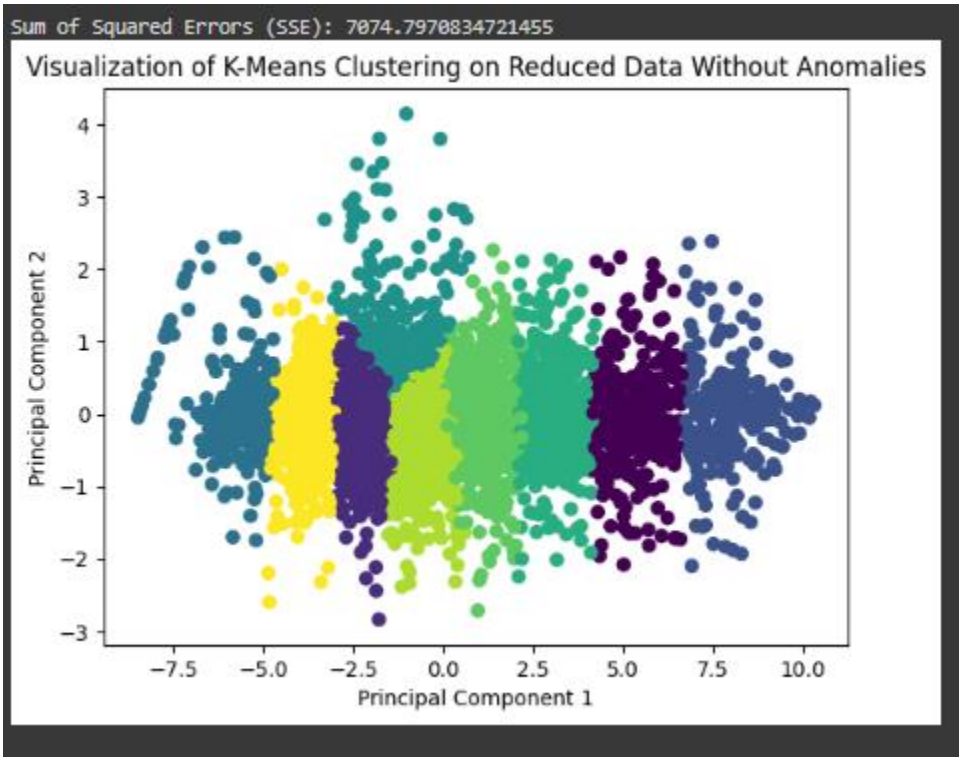


Chart2:



Summarization:

Introduction: -

This project involved a comprehensive analysis of the enhanced_sales_data.csv dataset,
supplemented by the pixel.txt dataset for clustering. The objective was to clean, preprocess, and
explore the data using various techniques, culminating in sophisticated data modeling and
anomaly detection. This summary outlines the key tasks and findings from each phase of the project

Task 1: Data Acquisition and Preparation

Objective: Prepare the datasets for analysis.
Datasets: enhanced_sales_data.csv and pixel.txt.
Libraries Used: Pandas, Sklearn, Matplotlib, Seaborn, Numpy, Plotly Express, Plotly Graph
Objects.
Actions: Loaded data, checked for missing values, and removed outliers.
Outcome: Clean and well-prepared datasets ready for further analysis.

Task 2: Exploratory Data Analysis (EDA)

Objective: Visualize the enhanced_sales_data.csv dataset to understand data distribution and
relationships.
Visualizations: Histograms, scatter plots, boxplots, and heatmaps.
Findings: Key patterns, outliers, and relationships within the dataset were highlighted.
Outcome: Enhanced understanding of data distributions and relationships.

Task 3: Linear Regression Analysis

Objective: Model the relationship between Price per Unit and Total Sales.
Model: Simple linear regression.
Results: High R^2 values (0.94 and 0.98) indicated a strong linear relationship.
Comparison: Linear regression was compared with a Random Forest model.
Outcome: Effective prediction of Total Sales based on Price per Unit.

Task 4: Decision Tree Analysis
Objective: Predict Total Sales using a decision tree regressor.
Data Preprocessing: Focused on Total Sales, Quantity, and Price per Unit.
Model: Decision tree regressor with max_depth=3.
Visualization: Decision tree visualization showed decision-making process.
Outcome: Interpretable model with effective segmentation of the dataset.

Task 5: Clustering Analysis
Objective: Identify distinct groups within the dataset using clustering techniques.
K-Means Clustering: Applied to 4x4 pixel blocks.
Hierarchical Clustering: Used Ward method for clustering.
Results: K-Means was effective but hierarchical clustering was less suitable for high dimensional data.
Outcome: Insights into the data structure and cluster formation.

Task 6: Principal Component Analysis (PCA)
Objective: Enhance clustering by reducing dataset dimensions.
Enhancements: PCA reduced dimensions while retaining significant features.
Results: Improved cluster separation and clarity in visualizations.
Impact: Reduced computational demands and minimized noise influence.
Outcome: Enhanced clustering performance and interpretability.

Task 7: Anomaly Detection and Removal
Objective: Improve clustering by detecting and removing anomalies.
Technique: Detected anomalies based on distances from cluster centers.
Process: Removed outliers and reapplied K-Means clustering.
Results: Tighter, more cohesive clusters with reduced SSE.
Outcome: Validated the importance of anomaly management in clustering.
Conclusion
The project demonstrated the importance of thorough data preprocessing, appropriate modeling techniques, and continuous refinement. Starting from data exploration and visualization, we progressed through regression and clustering techniques, incorporating PCA and anomaly detection to enhance results. Each task contributed to a comprehensive analytical framework, leading to accurate and interpretable models. This structured approach highlights the value of strategic data analysis in achieving high-quality outcomes.

Data and Code Resources:-

1. Enhanced Sales Data
Description: This dataset includes detailed sales information which has been enhanced with additional attributes for deeper analysis.
Access Link:- Enhanced Sales Data

2. Pixel Classification Data
Description: Contains pixel-level data used for classification purposes. It's structured in a text format and is critical for image analysis tasks.
Access Link: Pixel Classification Data