K-Nearest Neighbors (KNN) Report

1. About Dataset

The dataset contains 5000 entries with 12 columns, including features like various customer attributes, including demographics, financial details, and whether they have taken a personal loan. The target variables are `Personal Loan` (for classification) and `Income` (for regression). It is used for both classification (predicting loan approval) and regression (predicting income).

2. Preprocessing Steps

- Handled missing values by removing rows with NaNs.
- Dropped unnecessary columns (`ID`, `ZIP Code`).
- Encoded categorical variables using Label Encoding.
- Normalized the features using StandardScaler.
- Split the dataset into training (80%) and testing (20%) subsets.

3. Model Performance Results

Classification:

- Accuracy: The KNN classifier achieved a high accuracy score, indicating good performance in predicting whether a customer will accept a personal loan.
- Confusion matrix: The confusion matrix shows a high number of true negatives and true positives, with minimal misclassifications.
- Precision, Recall, and F1-score: These metrics confirm the model's ability to balance false positives and false negatives effectively.

Regression:

- MSE: Mean Squared Error is 1090.83, indicating the average squared difference between predicted and actual income values.
- RMSE: Root Mean Squared Error is 33.03, providing a more interpretable error measure in the same units as the target variable.
- R^2 score: The R^2 score of 0.4856 suggests that the model explains approximately 48.56% of the variance in the income data.

4. Observations on Performance Changes

- Number of Neighbors (n_neighbors): Increasing `n_neighbors` generally smoothens predictions but may reduce accuracy for classification and increase error for regression if set too high.
- Distance Metric: Using different distance metrics (e.g., Manhattan, Minkowski) can impact performance depending on the dataset's feature distribution.
- Feature Scaling: Normalization significantly improved performance, as KNN relies on distance calculations.
- Train-Test Split Ratio: A smaller test set may lead to overfitting, while a larger test set may reduce training data, impacting model performance.

Overall, KNN performed well for both tasks, but its sensitivity to parameter tuning and feature scaling highlights the importance of careful preprocessing and hyperparameter optimization.