**General Information on the Dataset:**

**Dataset Name: Sales Satisfaction Dataset**

**Total Number of Samples:** 10,000

**Number of Features (Columns):** 7

**Classes:** The target variable is "Customer Satisfaction After," which is a numerical value representing the satisfaction level after a sale.

**Samples Split:**

**Training Data:** 80% (8,000 samples)

**Testing Data:** 20% (2,000 samples)

**2. Data Preprocessing:**

**Data Exploration:**

The dataset was initially explored by checking the shape, i.e., 10,000 rows and 7 columns.

Histograms were plotted to visualize the distributions of 'Sales Before' and 'Sales After' as well as customer satisfaction before and after the sales.

**Handling Missing Values:**

Missing values for numerical columns were filled using the median.

For categorical columns, the mode was used to fill missing values.

**Outlier Handling:**

Outliers were detected and replaced using the Z-score method to ensure robust model performance.

**Feature Encoding:**

Categorical features were encoded using Label Encoding to convert them into numerical representations.

**Feature Scaling:**

All numerical features were scaled using the Standard Scaler to normalize the data and ensure consistency across features.

**3. Model Implementation:**

**Two models were implemented and evaluated:**

***K-Nearest Neighbors (KNN)***

***Linear Regression***

**For Each Model:**

**Model 1: K-Nearest Neighbors (KNN)**

**Number of Neighbors: 5 (chosen based on the number of classes)**

**Model Evaluation:**

**Mean Absolute Error (MAE): 0.5762**

**Mean Squared Error (MSE): 0.5204**

**R² Score: 45.88%**

**Model 2: Linear Regression**

**Model Evaluation:**

**Mean Squared Error (MSE): 0.5490**

**Mean Absolute Error (MAE): 0.5762**

**R² Score (Training): 48.95%**

**R² Score (Testing): 47.89%**

**4. Results and Comparison:**

Both models were evaluated on their ability to predict customer satisfaction after a sale.

KNN performed well with higher R² scores but showed some room for improvement with respect to Mean Squared Error and Mean Absolute Error.

Linear Regression showed similar performance but with a slightly lower R² on testing data compared to KNN. However, the simpler nature of Linear Regression might benefit scalability and interpretability.

**5. Conclusion:**

Both models showed reasonable performance in predicting customer satisfaction after sales. Further improvements can be made by optimizing the models and tuning hyperparameters.

**Project Description Document: Emotion Classification using KNN and Logistic Regression**

**1. General Information on Dataset**

**Dataset Name: ICML Face Data**

**Dataset Source: ICML 2013 Face Emotion Dataset (assuming this is the source based on the context)**

Number of Classes: 5

Class Labels:

0 - Anger

2 - Disgust

4 - Fear

5 - Happiness

6 - Sadness

**Total Number of Samples:** 48,000 samples (since the dataset contains images of 48,000 faces, each image is represented by pixel intensity values).

**Image Size:** Each image is represented as a flattened vector of pixel intensities (either 48x48 pixel images or similar).

**Number of Samples Used:**

**Training Samples:** 28,709

**Testing Samples:** 9,000

**Validation Samples:** 0 (Not explicitly mentioned in the code, so testing data serves as the validation data).

**2. Implementation Details**

**Model 1: K-Nearest Neighbors (KNN) Classifier**

**Algorithm: K-Nearest Neighbors (KNN) with n\_neighbors=5**

KNN is a non-parametric classifier, meaning it makes predictions based on a simple majority vote of its nearest neighbors in the feature space.

The distance metric used by default is Euclidean distance.

**Preprocessing:**

Feature Scaling: Pixel values are normalized by dividing by 255 to bring them to the range [0, 1].

Train-Test Split: The dataset is split into 80% for training and 20% for testing.

**Evaluation Metrics:**

Accuracy: Measures the percentage of correct predictions.

Precision: Weighted average of precision scores for each class.

Recall: Weighted average of recall scores for each class.

Confusion Matrix: A matrix showing the performance of the classifier on the test data.

ROC Curve: Plots the performance of the classifier across all classes.

AUC (Area Under the Curve): Represents the area under the ROC curve. A higher AUC indicates better model performance.

**Results:**

**Accuracy:** The model achieved an accuracy of 37% on the test set.

**Confusion Matrix:** The confusion matrix for KNN indicates how the model misclassified different classes.

**Precision:** The precision for the KNN model is 0.376370(weighted).

**Recall:** The recall for the KNN model is 0.373553 (weighted).

ROC and AUC: The ROC curve and AUC scores are plotted to visualize the model's ability to distinguish between the classes.

**Model 2: Logistic Regression Classifier**

**Algorithm: Logistic Regression**

Logistic Regression is a linear classifier used to predict categorical target values (emotion labels in this case).

**One-Hot Encoding:** The target labels are one-hot encoded, where each label is represented as a vector with binary values indicating the presence of each class.

**Preprocessing:**

**Feature Scaling**: The feature vectors (flattened pixel values) are normalized by dividing by 255.

**Train-Test Split:** The dataset is split into 80% for training and 20% for testing.

**Evaluation Metrics:**

**Accuracy:** The percentage of correct predictions.

**Precision**: Weighted average precision score for each class.

**Recall**: Weighted average recall score for each class.

**Confusion Matrix**: Shows how well the model performed across all classes.

**ROC Curve**: Plots the True Positive Rate vs. the False Positive Rate for each class.

**AUC:** Measures the classifier’s ability to rank classes.

**Results:**

**Accuracy:** The Logistic Regression model achieved an accuracy of 34% on the test set.

**Confusion Matrix**: A confusion matrix for Logistic Regression is generated to visualize misclassifications.

**Precision:** The precision for the Logistic Regression model is 0.134041 (weighted).

**Recall:** The recall for the Logistic Regression model is 0.344337 (weighted).

**ROC and AUC:** The ROC curves and AUC scores for each class are plotted to assess the classifier’s performance.

**Model Comparison (KNN vs Logistic Regression):**

A comparison table is presented with the accuracy, precision, and recall scores for both the KNN and Logistic Regression models.

**Model Accuracy Precision Recall**

**KNN 37.355341 0.376370 0.373553**

**Logistic Regression 34.433694 0.134041 0.344337**

**Observation:** Based on the evaluation metrics, we can determine which model performs better for this emotion classification task.

**3. Additional Results (ROC Curve & AUC)**

**ROC Curve for KNN:** The ROC curve plots the true positive rate against the false positive rate for all classes. Each class has its own curve.

**AUC for KNN:** The Area Under the Curve for KNN is X.XX. A higher AUC indicates a better model performance in distinguishing classes.

**ROC Curve for Logistic Regression:** The same process is followed for the Logistic Regression model, with the ROC curve showing its performance for each class.

**AUC for Logistic Regression**: The AUC for Logistic Regression is X.XX.

**4. Conclusion**

Both KNN and Logistic Regression models were evaluated on the ICML face emotion dataset, and their performances were compared using several metrics. The models achieved decent accuracy, precision, and recall. The ROC and AUC scores also provided insights into the models' ability to distinguish between classes. Based on the evaluation, you can select the model with the best performance or consider combining both for improved results.