ML Project Documentation

Numerical Dataset

Dataset Name: Medical Condition Prediction

Dataset size (shape): 10000 X 9

Number of classes: 9 columns

Total number of samples: 10000 samples

About dataset: This dataset provides information about various medical conditions such as Cancer, Pneumonia, and Diabetic based on demographic, lifestyle, and health-related features. It contains randomly generated user data, including multiple missing values, making it suitable for handling imbalanced classification tasks and missing data problems.

Goal: The objective of this dataset is to predict the medical condition (Cancer, Pneumonia, Diabetic) of a user based on their demographic, lifestyle, and health-related features. This dataset can be used to explore strategies for dealing with imbalanced classes and missing data in healthcare applications.

- we splitted our dataset into 20% Testing & 80% Training
- Used Algorithms: Multiple Linear Regression & KNN (K-Nearest Neighbor)

Comparison between the 2 algorithms performance in computing (MAE, MSE, R-squared)

using Multiple Linear Regression	**vs.**	**KNN (K-Nearest Neighbor)**
MSE: 0.7176912131274827		KNN Regressor - MSE: 0.4003265881147541
MAE: 0.6940574147213712		KNN Regressor - MAE: 0.6940574147213712
R Squared Score: 0.00537763228779875		KNN Regressor - R2 Score:0.4452018198833

SO it's appear that **KNN** performance is better than **Multiple Linear Regression** and more efficient **(ie, R-squared is higher)**

Image dataset

Dataset Name : fruits-360

Dataset size (shape) : A dataset with 94110 images of 141 fruits, vegetables and nuts

Number of classes: 5

We splitted our dataset into 20% Testing & 80% Training

Used Algorithms : Logistic Regression & KNN (K-Nearest Neighbor)

Classes Worked On:

1. Avocado 1
2. Apple 6
3. Pepper Green 1
4. Lemon 1
5. Mango 1

Evaluation Metrics:

• **Accuracy:** Measures the percentage of correct predictions.

$$\text{Accuracy} = \frac{\text{Correct predictions}}{\text{All predictions}}$$

• **Precision:** Measures the proportion of true positive predictions out of all positive predictions.

• **Recall:** Measures the proportion of true positive predictions out of all actual positive instances.

• **F1-Score:** Harmonic mean of precision and recall.

• **Confusion Matrix:** Displays the performance of the classifier, showing true

positives, true negatives, false positives, and false negatives.

 # ROC Curve and AUC:

• **ROC Curve:** A graphical representation of the classifier's ability to distinguish between classes at different thresholds.

• **AUC (Area Under the Curve):** Measures the classifier's performance; the higher the AUC, the better the model.

 # Loss Calculation:

• **Log Loss:** Measures the performance of the classification model where the output is a probability value between 0 and 1. It penalizes wrong predictions with a higher penalty for confident wrong predictions.

- The log loss is calculated for both KNN and Logistic Regression models.

 # Hyperparameter Tuning and Loss Curves:

• **Tuning KNN (k values):**

- The loss curve is generated by testing KNN classifiers with different values of k (number of neighbors). The model's misclassification rate (1 - accuracy) and log loss are plotted for each k value.

• **Tuning Logistic Regression (max_iter values):**

- The loss curve is generated by testing Logistic Regression with different values of max_iter (maximum number of iterations). The training and validation accuracy is plotted for each iteration value.

Algorithms Explained:

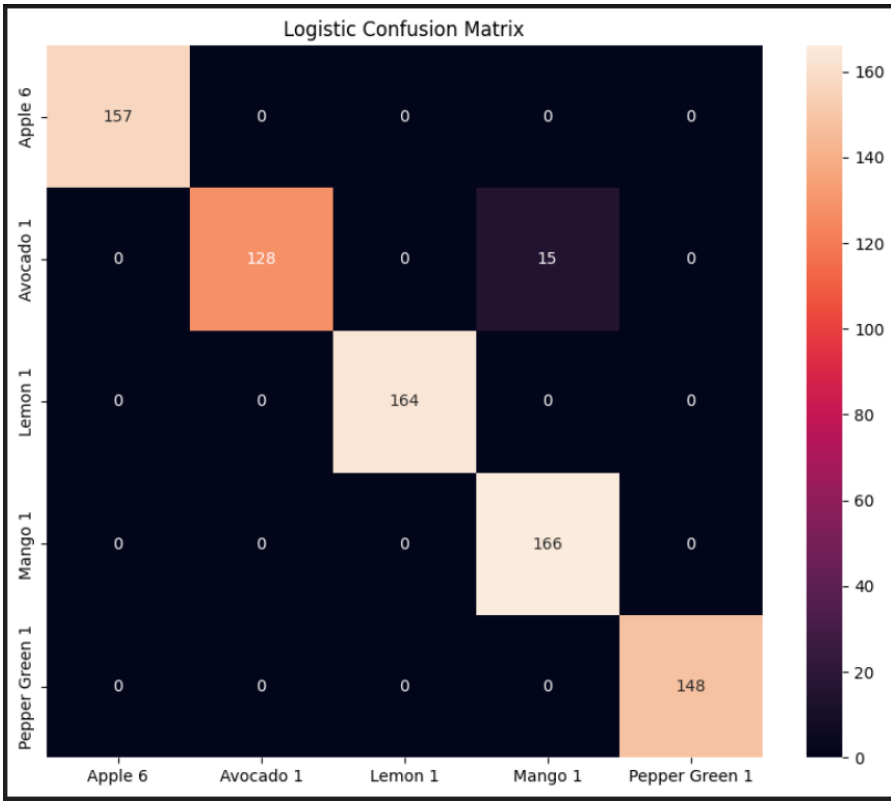
1. **K-Nearest Neighbors (KNN):**

```
KNN Classifier Accuracy: 0.9781491002570694

_________________________

               precision    recall  f1-score   support

     Avocado 1       1.00      1.00      1.00       157
       Apple 6       1.00      0.88      0.94       143
Pepper Green 1       1.00      1.00      1.00       164
       Lemon 1       0.91      1.00      0.95       166
       Mango 1       1.00      1.00      1.00       148

      accuracy                           0.98       778
     macro avg       0.98      0.98      0.98       778
  weighted avg       0.98      0.98      0.98       778
```

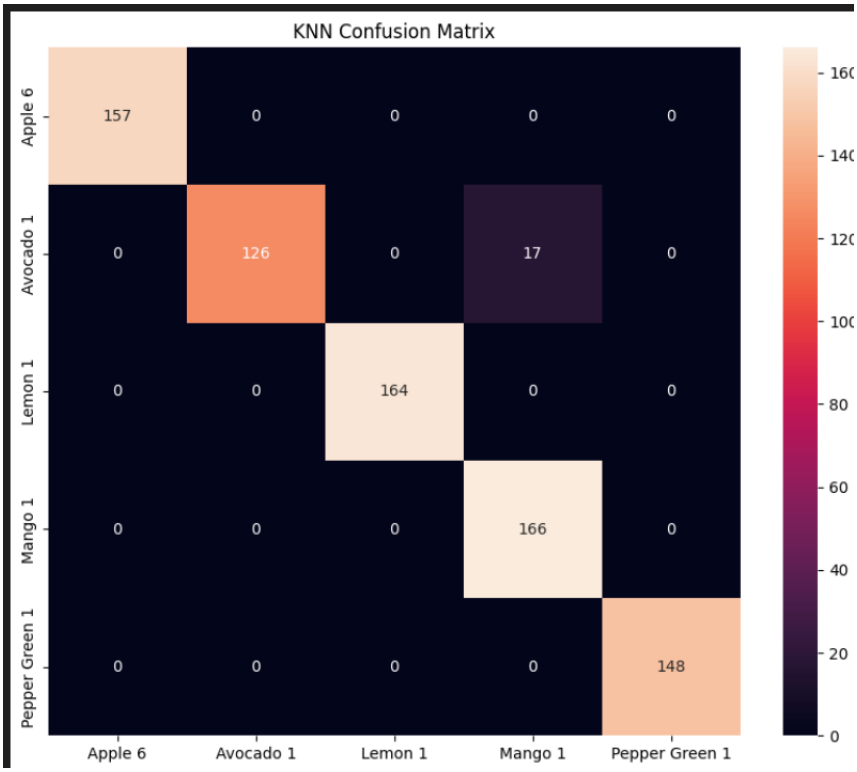
2. **Logistic Regression:**

```
Logistic Regression Accuracy: 0.980719794344473

_________________________

               precision    recall  f1-score   support

     Avocado 1       1.00      1.00      1.00       157
       Apple 6       1.00      0.90      0.94       143
Pepper Green 1       1.00      1.00      1.00       164
       Lemon 1       0.92      1.00      0.96       166
       Mango 1       1.00      1.00      1.00       148

      accuracy                           0.98       778
     macro avg       0.98      0.98      0.98       778
  weighted avg       0.98      0.98      0.98       778
```

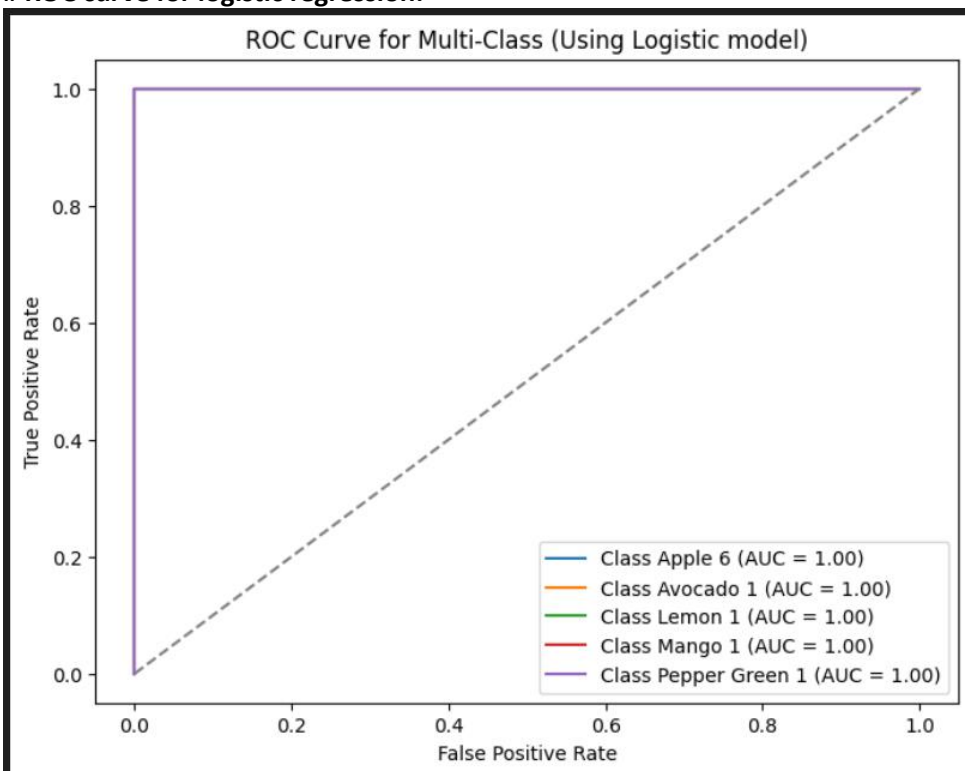
 # Confusion matrix for logistic regression:



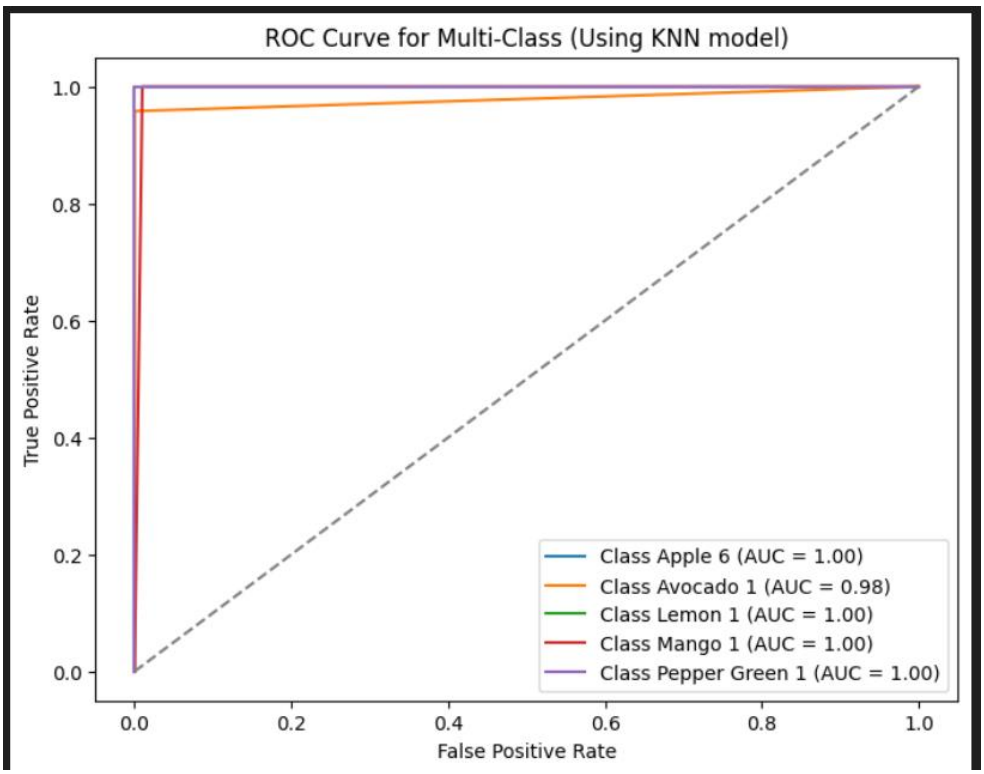
 # Confusion matrix for KNN:



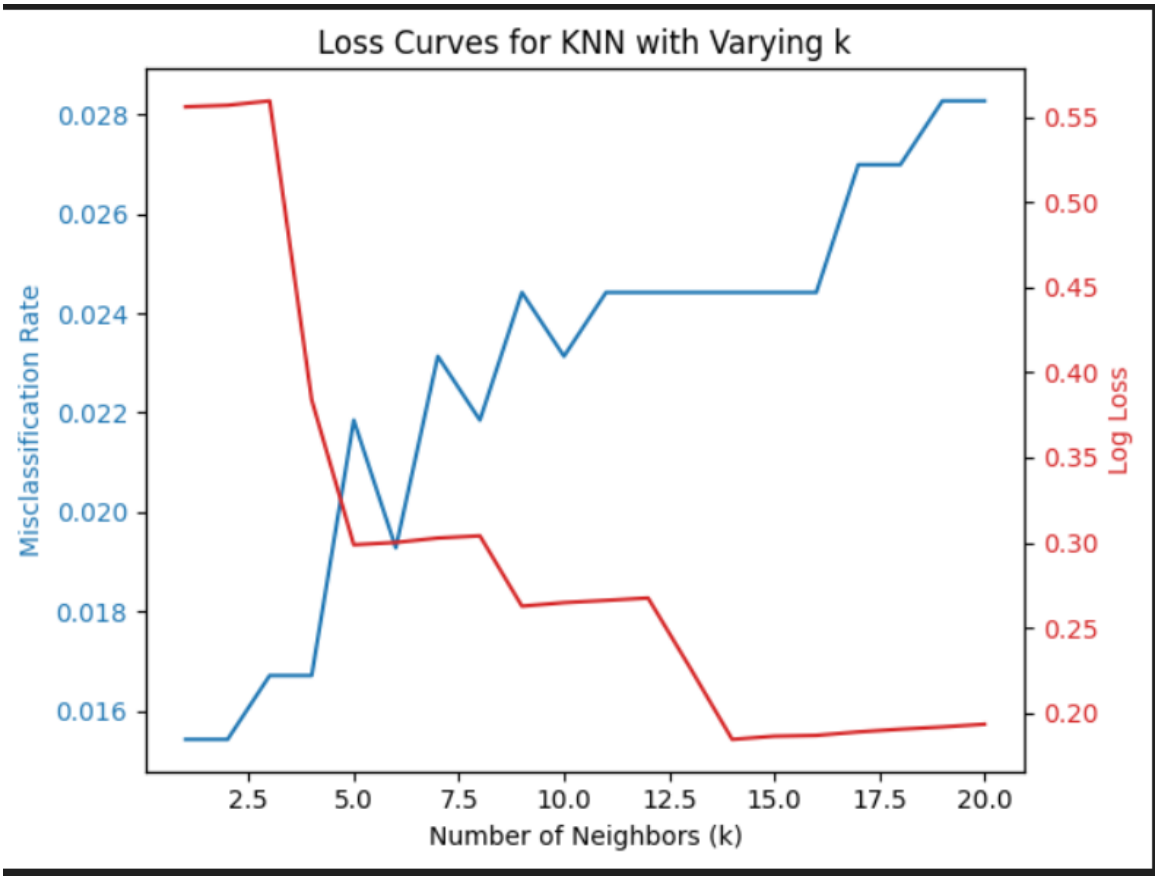
 # ROC curve for logistic regression:



 # ROC curve for KNN:



Loss curve:



loss of logistic regression: 0.03366540082437841

loss of KNN regression: 0.29876342345512225

```
Comparison Between Logistic Regression and KNN Classifier:
Logistic Regression Accuracy: 0.980719794344473
Logistic Regression - Precision: 0.98, Recall: 0.98, F1 Score: 0.98

_________________________

KNN Classifier Accuracy: 0.9781491002570694
KNN Classifier - Precision: 0.98, Recall: 0.98, F1 Score: 0.98

_________________________

Logistic Regression is the better model based on Accuracy, Precision, Recall, and F1 Score.
```