K-MEANS Documentation

a. General Information on dataset:

- **Dataset Name:** Food-101
- **Number of Classes**: 2
- **Class Labels:** The dataset contains images of food items belonging to the following classes: cheesecake& hamburger.
- **Total Number of Samples:** The dataset consists of a total of 2000 samples.
- **Size of Each Sample:** Each sample is an image with a size of (100, 100) pixels
- **Training, Validation, and Testing Split:** The dataset is split into training and testing sets. The training set consists of 1400 samples, and the testing set contains 600 samples.

b. Implementation details:

▼ **Feature Extraction Phase:** The features are extracted using PCA . The number of features extracted is 50, specified by n_components=50 when initializing the PCA object .

Why PCA with K-MEANS?

- PCA can be used with K-means as a pre-processing step to reduce the dimensionality of the data. By applying PCA before K-means, the high-dimensional data can be transformed into a lower-dimensional representation that captures the most important features or components. This can help in reducing the computational complexity of K-means and improving its performance by focusing on the most informative dimensions. Additionally, PCA can help in visualizing the data in lower-dimensional space, which can be useful for understanding the clusters formed by K-means.

- **Hyperparameters:**

- **K-means:** The number of clusters is set to 2 (n_clusters=2) during the initialization of the KMeans object.

▼ **SVM**: The SVM classifier is used with default hyperparameters.

Why SVM with K-MEANS?

• SVM can be used with K-means in a semi-supervised learning scenario. After applying K-means to cluster the data, the obtained cluster labels can be used as pseudo-labels for the data points. Then, SVM can be trained using the original features as inputs and the cluster labels as the target variable. This approach combines the unsupervised clustering capability of K-means with the supervised learning capability of SVM. It aims to leverage the clustering information from K-means to guide the SVM in finding a decision boundary that separates the data points of different clusters. This can potentially improve the classification performance of SVM by incorporating the clustering structure of the data.

▼ **Cross-Validation:** Cross-validation is not used in this implementation.

• **Reason for not using in K-means:** Cross-validation is a technique used for model evaluation and hyperparameter tuning. However, in the case of K-means, it is an unsupervised learning algorithm used for clustering rather than classification or regression. Cross-validation is typically used when there are labeled data with known class labels for evaluation. But in K-means, there are no explicit class labels available during training, as it is an unsupervised algorithm.

c. Results details:

- **Accuracy:** The accuracy of the K-MEANS model on the testing data is approximately `67.5`%.

▼ **Confusion matrix:**

- **Reason for not using in K-means:** Confusion matrix is a performance evaluation tool used in classification tasks to assess the model's predictions against the true class labels. K-means is not a classification algorithm, but a clustering algorithm. Hence, there are no explicit class labels to construct a confusion matrix for evaluating the performance of K-means.

▼ **ROC curve:**

- **Reason for not using in K-means:** ROC (Receiver Operating Characteristic) curve is commonly used in binary classification tasks to visualize the trade-off between true positive rate and false positive rate at different classification thresholds. Since K-means is an unsupervised clustering algorithm that does not involve explicit class labels, ROC curve is not applicable for evaluating its performance.

▼ **Loss curve:**

- **Reason for not using in K-means:** Loss curve is typically used in supervised learning tasks, such as neural networks, where the model is trained to minimize a specific loss function. K-means is an unsupervised algorithm that does not involve an explicit loss function. Instead, it iteratively optimizes the clustering objective, which is based on minimizing the Euclidean distances between data points and cluster centroids. Therefore, loss curve is not relevant or used in K-means.