Final Project Report

Problem Statement: PCA-Based image recognition to classify Volcanoes on Venus: Use PCA-based features and a k-Nearest Neighbor classifier to classify the certainty that there is a volcano in the picture from the Volcanoes on UCI Venus image dataset.

Implementation:

1. **Downloading The Dataset:** I have used the dataset on https://www.kaggle.com/datasets/fmena14/volcanoesvenus

2. **Importing Libraries:**
    a. Numpy for calculations
    b. Pandas to read and import dataset
    c. Matplotlib to plot graphs
3. **Importing Downloaded Dataset:** By using pandas we have successfully imported the data and now ready to code
    a. We have combined both the train and test dataset for PCA

4. **Implementing Min-Max Normalization:**
    a. Min-Max Normalization is performed using the formula given: **X = (X - Xmin)/(Xmax-Xmin)**
5. **Implementing PCA:**
    a. **PCA** by using **Eigen Value Decomposition Method (EVD)**
    b. **Steps**
        i. **Find the mean X_mean**
        ii. **Subtract from original i.e Xm = X - X_mean**
        iii. **Now we find the covariance i.e Dot_product(Xm,Xm.T)/(len(Xm-1)**
        iv. **Now we find the eigenvectors and eigenvalues with help of linalg function**

6. **Plotting The Graph For Eigen Value and Variance:**
    a. We plot the curve for the descending sorted order curve for eigen value in order to get the components which covers certain variance of the data

7. **Implementation of Transformation:**
    a. **Steps**
        i. **Find Mean X_mean**
        ii. **Subtract from original i.e Xm = X - X_mean**
        iii. **We sort the eigen values in descending order**
        iv. **We take only the first selected K component corresponding eigenvectors**
        v. **Dot_product(Xm,vectors.T)**

8. **Splitting The Dataset:** We split the transformed dataset back to training and test.

9. **Implementation of KNN:** We have implemented the KNN model using euclidean distance and then finding the nearest index and label it to the given test data in order to predict the classification of the given particular data.

Result:

After performing the KNN on PCA converted Dataset we get 84.125% accuracy for the first component whereas we also have F-Score of 0, hence we then plot the graph for the total percent of variance covered by the number of components. So we conducted the experiment for variances in range {0.65,0.7,0.75,0.8,0.85,0.90} , for 3 nearest neighbors and we got the highest accuracy in 87.30 and F-score 0.43 , for 5 nearest neighbor we again got the highest accuracy for 86.68 and F-score 0.34. Below is the given table for different accuracies for different variances

Table 1.1
For 3 Nearest Neighbors

Variance	Accuracy	F-Score
0.65	84.125%	0.0
0.70	80.72%	0.0898
0.75	83.46%	0.28
0.80	87.3%	0.43
0.85	86.90%	0.37
0.90	85.66%	0.289

Table 1.2
For 5 Nearest Neighbors

Variance	Accuracy	F-Score
0.65	84.125%	0
0.70	82.47%	0.0439
0.75	84.41%	0.24
0.80	86.68%	0.3429
0.85	86.32%	0.329
0.90	85.47%	0.25

Observations:

So we can say from the above given tables the accuracies for the given variances slightly increases for first three variances in the list but decreases for the other remaining variances whereas the F-score is has been tremendously decreased for all the variances while we increase the number of neighbors from 3 to 5 in KNN implementation.