|  |  |  |  |
| --- | --- | --- | --- |
| A picture containing drawing, stop, room  Description automatically generated | Machine Learning  Practical # 7 | | |
|  |  |  |  |
|  |  |  |  |
| **Name** |  | **Roll Number** |  |
| **Subject/Course:** | Machine Learning | **Class** | M.Sc. IT – Sem III |
| **Topic** |  | **Batch** |  |
|  |  |  |  |
| **Topic:** **Feature Selection** | | | |
| **Aim:** **Data loading, feature scoring and ranking, feature selection (principal component analysis)**  **Description:**  **Principal Component Analysis:-**        **Code and output**  import numpy as np  import pandas as pd  from sklearn.model\_selection import train\_test\_split  from sklearn.preprocessing import StandardScaler  from sklearn.decomposition import PCA  from sklearn.ensemble import RandomForestClassifier  from sklearn.metrics import confusion\_matrix, accuracy\_score    url = "https://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.data"  names = ['sepal-length', 'sepal-width', 'petal-length', 'petal-width', 'Class']  dataset = pd.read\_csv(url, names=names)  dataset.head()    # Store the feature sets into X variable and the series of corresponding variables in y  x = dataset.drop('Class', axis=1)  y = dataset['Class']  x.head()  y.head()    x\_train, x\_test, y\_train, y\_test = train\_test\_split(x, y, test\_size=0.2, random\_state=0)    sc = StandardScaler()  x\_train1 = sc.fit\_transform(x\_train)  x\_test1 = sc.transform(x\_test)    y\_train1 = y\_train  y\_test1 = y\_test    pca = PCA()  x\_train1 = pca.fit\_transform(x\_train1)  x\_test1 = pca.transform(x\_test1)    explained\_variance = pca.explained\_variance\_ratio\_  print(explained\_variance)    pca = PCA(n\_components=1)  x\_train1 = pca.fit\_transform(x\_train1)  x\_test1 = pca.transform(x\_test1)    classifier = RandomForestClassifier(max\_depth=2, random\_state=0)  classifier.fit(x\_train1, y\_train1)  y\_pred = classifier.predict(x\_test1)    cm = confusion\_matrix(y\_test, y\_pred)  print(cm)  print('Accuracy:', accuracy\_score(y\_test, y\_pred))  **Learnings:** | | | |
|  | | | |
|  | | | |