

Experiment–5

Aim:

Program to implement feature scaling, feature extraction and selection.

Code:

```
1 import pandas as pd
2 import numpy as np
3 import matplotlib.pyplot as plt
4 import seaborn as sns
5 from sklearn.datasets import load_iris
6 from sklearn.model_selection import train_test_split
7 from sklearn.preprocessing import StandardScaler, MinMaxScaler
8 from sklearn.decomposition import PCA
9 from sklearn.feature_selection import SelectKBest, f_classif, RFE
10 from sklearn.linear_model import LogisticRegression
11 from sklearn.metrics import accuracy_score
12
13 # Load Iris
14 iris = load_iris(as_frame=True)
15 X = iris.data
16 y = iris.target
17 feature_names = iris.feature_names
18 print("Original Features:", feature_names)
19 print("Shape:", X.shape)
20 X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3,
21     ↪ random_state=42)
22
23 # 1. No Scaling (Baseline)
24 model = LogisticRegression(max_iter=200)
25 model.fit(X_train, y_train)
26 acc0 = accuracy_score(y_test, model.predict(X_test))
27 print(f"\n1. No Scaling → Accuracy: {acc0:.4f}")
28
29 # 2. Standard Scaling (Zero mean, unit variance)
30 scaler_std = StandardScaler()
31 X_train_std = scaler_std.fit_transform(X_train)
32 X_test_std = scaler_std.transform(X_test)
33 model.fit(X_train_std, y_train)
34 acc1 = accuracy_score(y_test, model.predict(X_test_std))
35 print(f"\n2. StandardScaler → Accuracy: {acc1:.4f}")
36
37 # 3. MinMax Scaling (0 to 1)
38 scaler_mm = MinMaxScaler()
39 X_train_mm = scaler_mm.fit_transform(X_train)
40 X_test_mm = scaler_mm.transform(X_test)
41 model.fit(X_train_mm, y_train)
```