|  |
| --- |
| Experiment No. 7 |
| Apply Dimensionality/ Reduction on Adult Census Income Dataset and analyze the performance of the model |
| Date of Performance: 11/09/2023 |
| Date of Submission: 25/09/2023 |

**Aim:** Apply Dimensionality Reduction on Adult Census Income Dataset and analyze the performance of the model.

**Objective:** Able to perform various feature engineering tasks, perform dimetionality reduction on the given dataset and maximize the accuracy, Precision, Recall, F1 score.

**Theory:**

In machine learning classification problems, there are often too many factors on the basis of which the final classification is done. These factors are basically variables called features. The higher the number of features, the harder it gets to visualize the training set and then work on it. Sometimes, most of these features are correlated, and hence redundant. This is where dimensionality reduction algorithms come into play. Dimensionality reduction is the process of reducing the number of random variables under consideration, by obtaining a set of principal variables. It can be divided into feature selection and feature extraction.

**Dataset:**

Predict whether income exceeds $50K/yr based on census data. Also known as "Adult" dataset.

Attribute Information:

Listing of attributes:

>50K, <=50K.

age: continuous.

workclass: Private, Self-emp-not-inc, Self-emp-inc, Federal-gov, Local-gov, State-gov, Without-pay, Never-worked.

fnlwgt: continuous.

education: Bachelors, Some-college, 11th, HS-grad, Prof-school, Assoc-acdm, Assoc-voc, 9th, 7th-8th, 12th, Masters, 1st-4th, 10th, Doctorate, 5th-6th, Preschool.

education-num: continuous.

marital-status: Married-civ-spouse, Divorced, Never-married, Separated, Widowed, Married-spouse-absent, Married-AF-spouse.

occupation: Tech-support, Craft-repair, Other-service, Sales, Exec-managerial, Prof-specialty, Handlers-cleaners, Machine-op-inspct, Adm-clerical, Farming-fishing, Transport-moving, Priv-house-serv, Protective-serv, Armed-Forces.

relationship: Wife, Own-child, Husband, Not-in-family, Other-relative, Unmarried.

race: White, Asian-Pac-Islander, Amer-Indian-Eskimo, Other, Black.

sex: Female, Male.

capital-gain: continuous.

capital-loss: continuous.

hours-per-week: continuous.

native-country: United-States, Cambodia, England, Puerto-Rico, Canada, Germany, Outlying-US(Guam-USVI-etc), India, Japan, Greece, South, China, Cuba, Iran, Honduras, Philippines, Italy, Poland, Jamaica, Vietnam, Mexico, Portugal, Ireland, France, Dominican-Republic, Laos, Ecuador, Taiwan, Haiti, Columbia, Hungary, Guatemala, Nicaragua, Scotland, Thailand, Yugoslavia, El-Salvador, Trinadad & Tobago, Peru, Hong, Holand-Netherlands.

**Code:**

import pandas as pd

import numpy as np

from sklearn.preprocessing import LabelEncoder, StandardScaler

from sklearn.model\_selection import train\_test\_split

from sklearn.decomposition import PCA

from sklearn.ensemble import RandomForestClassifier

from sklearn.metrics import accuracy\_score, precision\_score, recall\_score, f1\_score

# Load the dataset

# data = pd.read\_csv("adult.csv")  # Make sure to load your dataset

# Encode categorical features

categorical\_features = data.select\_dtypes(include=['object']).columns

for feature in categorical\_features:

    data[feature] = LabelEncoder().fit\_transform(data[feature])

# Split the data into features (X) and target (y)

X = data.drop('>50K', axis=1)

y = data['>50K']

# Standardize the features

scaler = StandardScaler()

X = scaler.fit\_transform(X)

# Split the data into training and testing sets

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

# Perform dimensionality reduction using PCA

n\_components = 10  # Adjust the number of components as needed

pca = PCA(n\_components=n\_components)

X\_train\_pca = pca.fit\_transform(X\_train)

X\_test\_pca = pca.transform(X\_test)

# Train a classifier (Random Forest, for example)

clf = RandomForestClassifier(n\_estimators=100, random\_state=42)

clf.fit(X\_train\_pca, y\_train)

# Make predictions on the test set

y\_pred = clf.predict(X\_test\_pca)

# Evaluate the model

accuracy = accuracy\_score(y\_test, y\_pred)

precision = precision\_score(y\_test, y\_pred)

recall = recall\_score(y\_test, y\_pred)

f1 = f1\_score(y\_test, y\_pred)

print("Accuracy: {:.2f}".format(accuracy))

print("Precision: {:.2f}".format(precision))

print("Recall: {:.2f}".format(recall))

print("F1 Score: {:.2f}".format(f1))

# Optional: Visualize explained variance ratio

explained\_variance\_ratio = pca.explained\_variance\_ratio\_

print("Explained Variance Ratio for Each Principal Component:")

print(explained\_variance\_ratio)

**Output:**

Accuracy: 0.85

Precision: 0.72

Recall: 0.62

F1 Score: 0.66

Explained Variance Ratio for Each Principal Component:

[0.15518513 0.10236402 0.09369864 0.08605513 0.08026009 0.07491667

0.07026711 0.06332068 0.06128732 0.04822278]

**Conclusion:**

The process of reducing dimensionality in machine learning models can yield a spectrum of outcomes. On the positive side, it can effectively curb overfitting, enhance the model's capacity to make accurate predictions on new data, and significantly optimize computational efficiency. However, it's crucial to recognize that there can be potential drawbacks associated with dimensionality reduction. These include the risk of losing valuable information, potential decreases in model precision and recall, and the potential difficulty in handling noisy data effectively. It's a trade-off that requires careful consideration in the context of each specific machine learning problem.