Experiment No. 7
Apply Dimensionality Reduction on Adult Census Income Dataset and analyze the performance of the model
Date of Performance:4/10/23
Date of Submission:11/10/23

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

# Assuming 'adult.csv' is the dataset file
data = pd.read_csv('adult.csv')
```

```
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder

# Handle missing values
data = data.dropna()

# Encode categorical features
categorical_cols = data.select_dtypes(include=['object']).columns
label_encoders = {}
for col in categorical_cols:
    le = LabelEncoder()
    data[col] = le.fit_transform(data[col])
    label_encoders[col] = le

# Split the data into X and y
X = data.drop('income', axis=1)
y = data['income']

# Split 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)
```

```
from sklearn.decomposition import PCA

# Specify the number of components (features) you want to keep
n_components = 10
pca = PCA(n_components=n_components)
X_train_pca = pca.fit_transform(X_train)
X_test_pca = pca.transform(X_test)
```

```
from sklearn.ensemble import RandomForestClassifier

# Create and train a classifier
clf = RandomForestClassifier(n_estimators=100, random_state=42)
clf.fit(X_train_pca, y_train)
```

```
▾        RandomForestClassifier
RandomForestClassifier(random_state=42)
```

```
from sklearn.metrics import classification_report, accuracy_score

# Make predictions on the test set
y_pred = clf.predict(X_test_pca)

# Evaluate the model
accuracy = accuracy_score(y_test, y_pred)
report = classification_report(y_test, y_pred)

print("Accuracy:", accuracy)
print("Classification Report:\n", report)
```

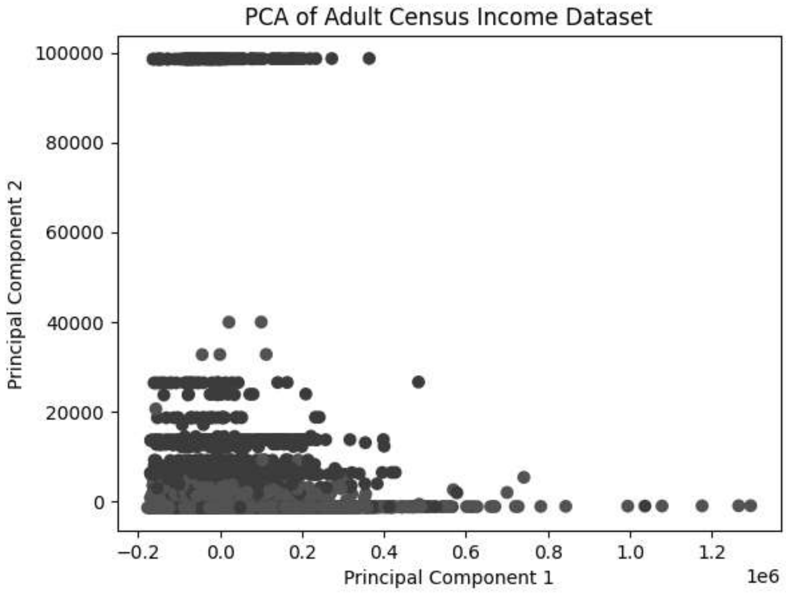
```
Accuracy: 0.8497191011235955
Classification Report:
               precision    recall  f1-score   support

           0       0.86      0.95      0.90      3722
           1       0.79      0.56      0.65      1262

    accuracy                           0.85      4984
   macro avg       0.82      0.75      0.78      4984
weighted avg       0.84      0.85      0.84      4984
```

```
import matplotlib.pyplot as plt

# Visualize the reduced data (first two principal components)
plt.scatter(X_train_pca[:, 0], X_train_pca[:, 1], c=y_train, cmap='coolwarm', marker='o')
plt.xlabel('Principal Component 1')
plt.ylabel('Principal Component 2')
plt.title('PCA of Adult Census Income Dataset')
plt.show()
```


PCA of Adult Census Income Dataset
Principal Component 2
100000
80000
60000
40000
20000
0
−0.2
0.0
0.2
0.4
0.6
0.8
1.0
1.2
1e6
Principal Component 1

Conclusion:

Comment on the impact of dimensionality reduction on the accuracy, precision, recall and F1 score.

Accuracy:

Positive Impact: Dimensionality reduction can often have a positive impact on accuracy. By reducing the number of features, you can reduce the risk of overfitting, which can lead to better generalization to unseen data.

Negative Impact: However, in some cases, if important information is lost during dimensionality reduction, accuracy may decrease. This can happen when the reduced feature space does not capture critical patterns in the data.

Precision:

Positive Impact: Dimensionality reduction can improve precision by reducing noise in the data. With fewer irrelevant features, the model may make fewer false positive predictions.

Negative Impact: If important discriminatory features are removed, precision may suffer because the model might not identify true positive cases as effectively.

Recall:

Positive Impact: Dimensionality reduction can improve recall if it helps the model focus on the most relevant features. Fewer irrelevant features can lead to better identification of true positive cases.

Negative Impact: If dimensionality reduction discards crucial information, recall may decrease because the model might miss true positive cases more often.

F1 Score:

The F1 score is the harmonic mean of precision and recall. Its impact will depend on how precision and recall change with dimensionality reduction.

If both precision and recall improve, the F1 score is likely to increase, indicating a better balance between precision and recall. However, if one metric improves while the other worsens, the impact on the F1 score can be less clear.