



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



**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 dimensionality 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.



## Vidyavardhini's College of Engineering & Technology

### Department of Computer Engineering

---

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, Trinidad&Tobago, Peru, Hong, Holand-Netherlands.



Code:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler, LabelEncoder
from sklearn.decomposition import PCA
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import classification_report, confusion_matrix,
accuracy_score
df = pd.read_csv('adult.csv', header=None)

# Assign column names
columns = ['age', 'workclass', 'fnlwgt', 'education', 'education.num',
          'marital.status', 'occupation', 'relationship', 'race',
          'sex', 'capital.gain', 'capital.loss', 'hours.per.week',
          'native.country', 'income']
df.columns = columns
# Handle missing values (if any) - removing or filling them

df = df.replace(' ?', np.nan)
df.dropna(inplace=True)
# Encode categorical features
label_encoders = {}
for column in ['workclass', 'education', 'marital.status',
               'occupation',
               'relationship', 'race', 'sex', 'native.country',
               'income']:
    le = LabelEncoder()
    df[column] = le.fit_transform(df[column])
    label_encoders[column] = le
X = df.drop('income', axis=1)
y = df['income']
imputer = SimpleImputer(strategy='mean')
X_imputed = imputer.fit_transform(X)
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X_imputed)
pca = PCA(n_components=5)
```



```
X_pca = pca.fit_transform(X_scaled)
X_train, X_test, y_train, y_test = train_test_split(X_pca, y,
test_size=0.3, random_state=42)
```

```
model = LogisticRegression()
model.fit(X_train, y_train)
```

```
LogisticRegression
LogisticRegression()
```

```
y_pred = model.predict(X_test)
print("Accuracy Score:", accuracy_score(y_test, y_pred))
print("Confusion Matrix:")
print(confusion_matrix(y_test, y_pred))
print("Classification Report:")
print(classification_report(y_test, y_pred))
20]
```

0s

```
print("Accuracy Score:", accuracy_score(y_test, y_pred))
print("Confusion Matrix:")
print(confusion_matrix(y_test, y_pred))
print("Classification Report:")
print(classification_report(y_test, y_pred))
Accuracy Score: 0.7984440577336472
Confusion Matrix:
[[6944  470]
 [1499  856]]
Classification Report:
```

	precision	recall	f1-score	support
0	0.82	0.94	0.88	7414
1	0.65	0.36	0.47	2355
accuracy			0.80	9769
macro avg	0.73	0.65	0.67	9769
weighted avg	0.78	0.80	0.78	9769



#### Conclusion:

Applying dimensionality reduction using PCA on the Adult Census Income dataset effectively streamlined the feature set, transforming it into five principal components. The logistic regression model, trained on this reduced dataset, achieved an accuracy of approximately 79.8%. However, while the model demonstrated strong performance on the majority class (class 0), with a recall of 94% and an F1-score of 88%, it struggled with the minority class (class 1), yielding a recall of only 36% and an F1-score of 47%. This indicates that although PCA helped in reducing dimensionality and maintaining computational efficiency, further efforts are needed to enhance the model's ability to identify the minority class, potentially through techniques such as class balancing or alternative modeling approaches.