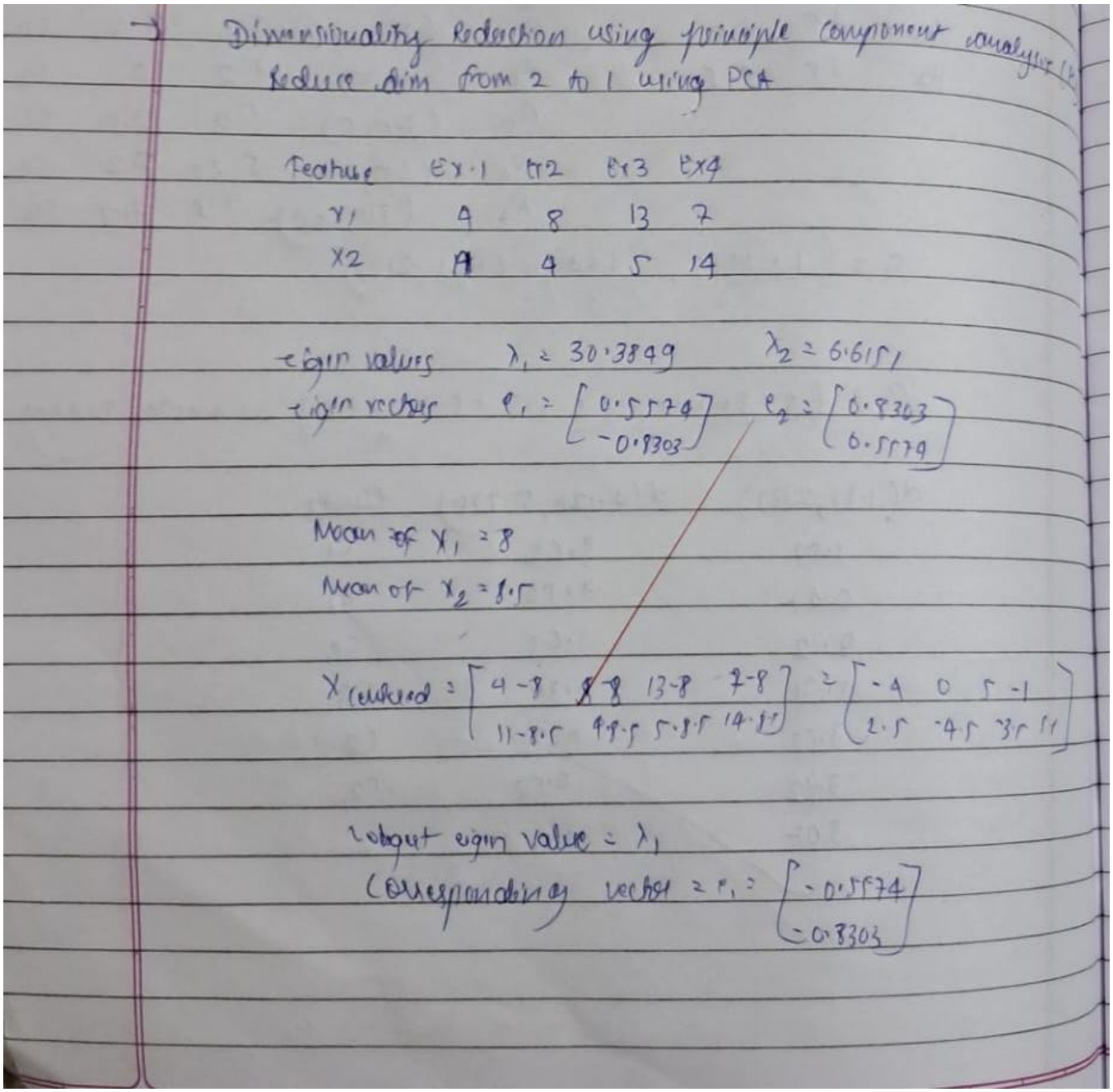
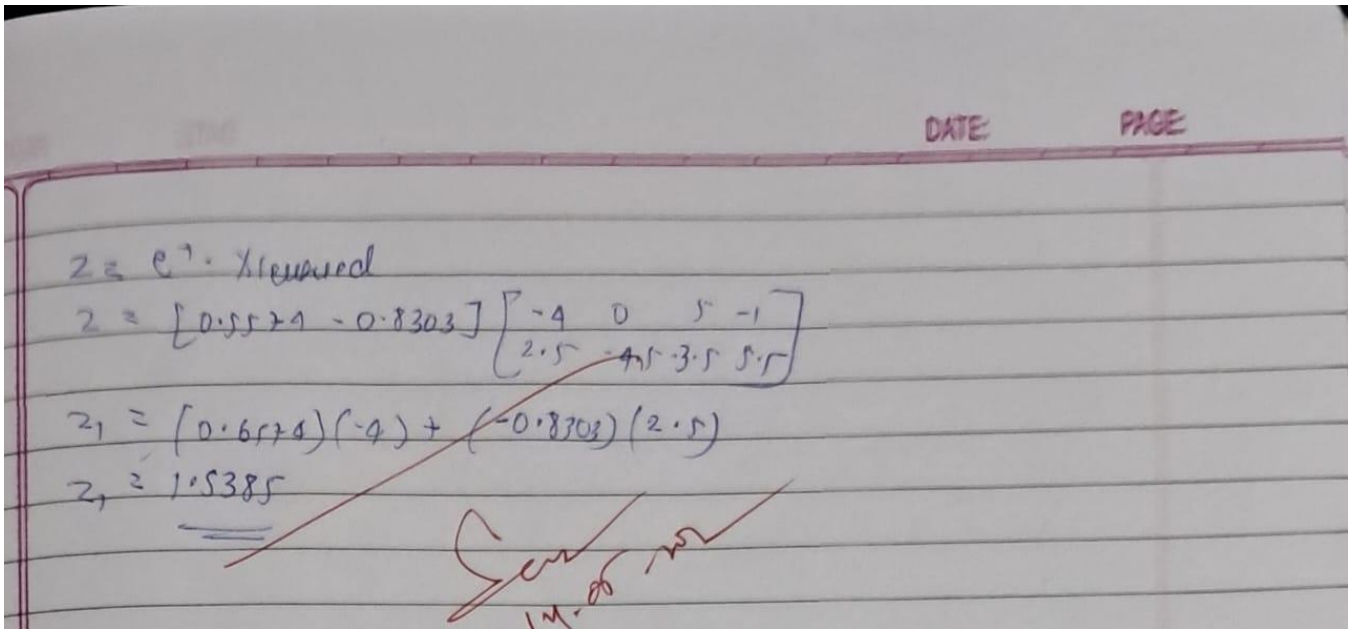
Program 11

Implement Dimensionality reduction using Principal Component Analysis (PCA) method

Screenshot:





Code:

```
import pandas as pd

from sklearn.model_selection import train_test_split

from sklearn.preprocessing import LabelEncoder, StandardScaler

from sklearn.svm import SVC

from sklearn.linear_model import LogisticRegression

from sklearn.ensemble import RandomForestClassifier

from sklearn.metrics import accuracy_score

from sklearn.decomposition import PCA


# Load the dataset

df = pd.read_csv('/content/heart (2).csv')


# Detect and encode categorical columns

text_cols = df.select_dtypes(include=['object']).columns.tolist()
```

```
for col in text_cols:
    le = LabelEncoder()
    df[col] = le.fit_transform(df[col].astype(str))

# Use correct target column
target_col = 'HeartDisease'
X = df.drop(columns=[target_col])
y = df[target_col]

# Scale the features
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

# Train-test split
X_train, X_test, y_train, y_test = train_test_split(X_scaled, y, test_size=0.2, random_state=42)

# Models
models = {
    'SVM': SVC(),
    'Logistic Regression': LogisticRegression(),
    'Random Forest': RandomForestClassifier()
}

# Evaluate without PCA
```

```
results = {}
for name, model in models.items():
    model.fit(X_train, y_train)
    y_pred = model.predict(X_test)
    results[name] = accuracy_score(y_test, y_pred)
    print(f"{name} Accuracy: {results[name]:.4f}")

# Apply PCA
pca = PCA(n_components=0.95)
X_train_pca = pca.fit_transform(X_train)
X_test_pca = pca.transform(X_test)

# Evaluate after PCA
pca_results = {}
for name, model in models.items():
    model.fit(X_train_pca, y_train)
    y_pred = model.predict(X_test_pca)
    pca_results[name] = accuracy_score(y_test, y_pred)
    print(f"{name} Accuracy after PCA: {pca_results[name]:.4f}")

# Final comparison
print("\nComparison (Original vs PCA):")
for name in models:
    print(f"{name}: Original = {results[name]:.4f}, PCA = {pca_results[name]:.4f}")
```