Department of Computer Science Engineering (AI)

ARTIFICIAL INTELLIGENCE

Project Report

On

Classify Students Based on Study Methods

By

ANSHUL ANAND

202401100300056

DataSet Used:student_methods.csv

TABLE OF CONTENTS

1.1 INTRODUCTION

Identifying the learning style of a student—whether **visual**, **auditory**, or **kinesthetic**—can significantly improve educational outcomes. Each student has a preferred method of absorbing information, and by classifying students accordingly, educators can better personalize their teaching methods.

In this project, we utilize a **K-Nearest Neighbors (KNN)** classifier to predict the learning style based on three input features: `visual_score`, `auditory_score`, and `kinesthetic_score`. The classification is performed using machine learning techniques with proper data visualization to understand the distribution and correlation of the data.

NATIONAL INSTITUTIONAL RANKING FRAMEWORK
NIRF-2023
Engineering Rank Band (151-200)
Pharmacy Rank - 88
Innovation Rank Band (51-100)
www.kiet.edu
Delhi-NCR, Ghaziabad
KIET GROUP OF INSTITUTIONS
Connecting Life with Learning
CELEBRATING 25 YEARS OF EXCELLENCE
A+ NAAC
Accredited by
NAAC with Grade "A+"
IQAC

1.2 Methodology

Data Preprocessing:

- **Loaded the dataset (`student_methods.csv`) containing student scores across three sensory modalities.**
- **Checked for missing values and confirmed dataset integrity.**
- **Encoded the `learning_style` column using `LabelEncoder`.**
- **Standardized feature values using `StandardScaler` for optimal KNN performance.**

Visualization:

- **Created box plots to analyze the distribution of scores across learning styles.**
- **Plotted a correlation heatmap between the scores.**
- **Used PCA for dimensionality reduction and visualized the dataset in a 2D plane.**

Model Selection:

- **Chose K-Nearest Neighbors (KNN) classifier due to its simplicity and effectiveness in multi-class classification problems.**

Model Training:

- **Data split into 80% training and 20% testing sets using `train_test_split()`.**

- **KNN model trained on scaled training data.**

Evaluation:

- **Used `classification_report` and `accuracy_score` to evaluate predictions.**

- **Displayed a confusion matrix to assess performance visually.**

CODE

```python
# Import required libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

# Machine learning tools
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder, StandardScaler
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import classification_report, accuracy_score, confusion_matrix
from sklearn.decomposition import PCA

# Load the dataset
df = pd.read_csv('/content/student_methods.csv')
```

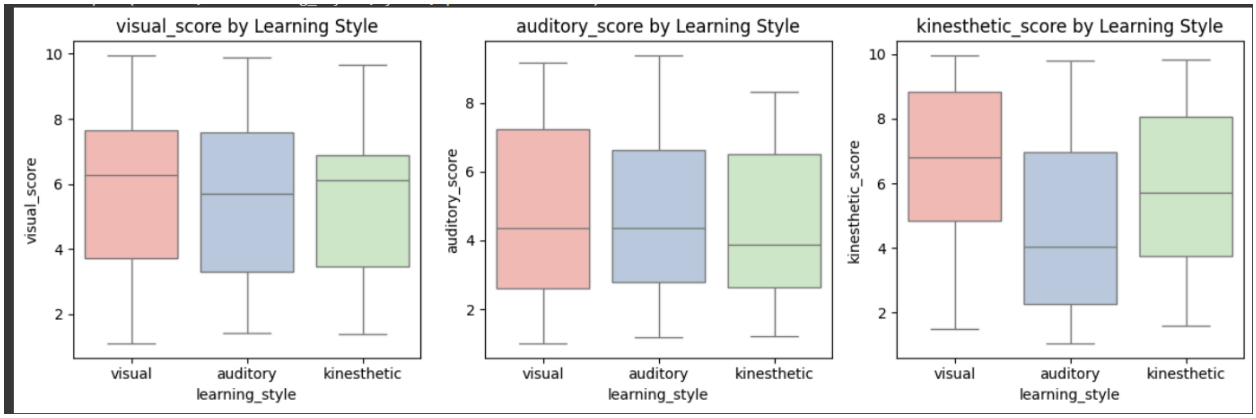
```
print("Dataset preview:")
print(df.head())

# Check for missing values
print("\nMissing values:")
print(df.isnull().sum())

# Encode the target variable
le = LabelEncoder()
df['learning_style_encoded'] = le.fit_transform(df['learning_style'])

# Box plots of each score by learning style
plt.figure(figsize=(12, 4))
for idx, col in enumerate(['visual_score', 'auditory_score', 'kinesthetic_score']):
    plt.subplot(1, 3, idx + 1)
    sns.boxplot(data=df, x='learning_style', y=col, palette='Pastel1')
    plt.title(f'{col} by Learning Style')
plt.tight_layout()
plt.show()
```

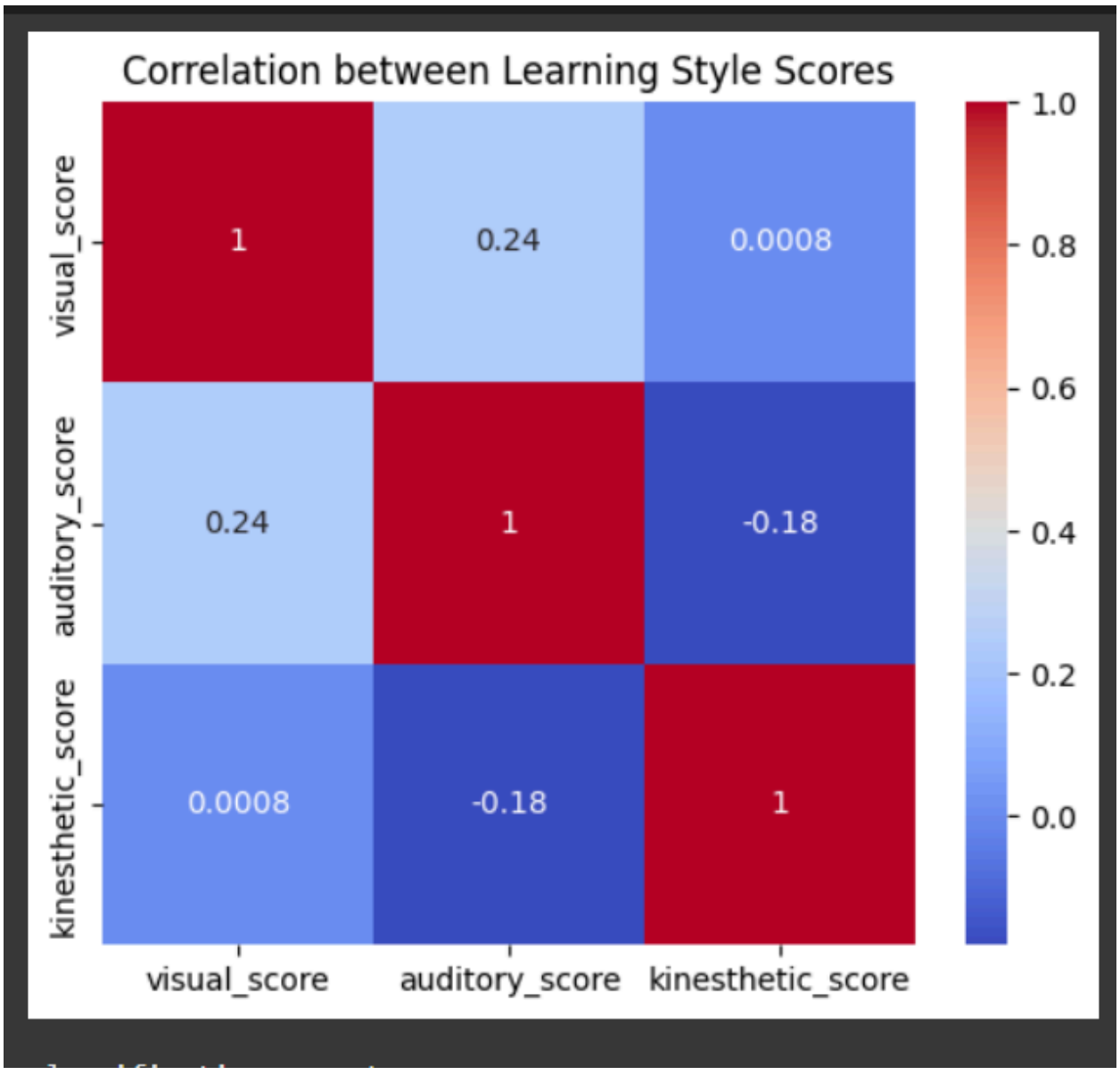
```
# Correlation heatmap
plt.figure(figsize=(6, 5))
sns.heatmap(df[['visual_score', 'auditory_score', 'kinesthetic_score']].corr(), annot=True, cmap='coolwarm'
plt.title("Correlation between Learning Style Scores")
plt.show()

# Define features and target
X = df[['visual_score', 'auditory_score', 'kinesthetic_score']]
y = df['learning_style_encoded']

# Standardize features for KNN
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

# Split dataset
X_train, X_test, y_train, y_test = train_test_split(X_scaled, y, test_size=0.2, random_state=42)

# Initialize and train KNN model
knn = KNeighborsClassifier(n_neighbors=5)
knn.fit(X_train, y_train)

# Predict and evaluate
y_pred = knn.predict(X_test)

print("\nClassification Report:")
print(classification_report(y_test, y_pred, target_names=le.classes_))

print("Accuracy Score:", accuracy_score(y_test, y_pred))
```

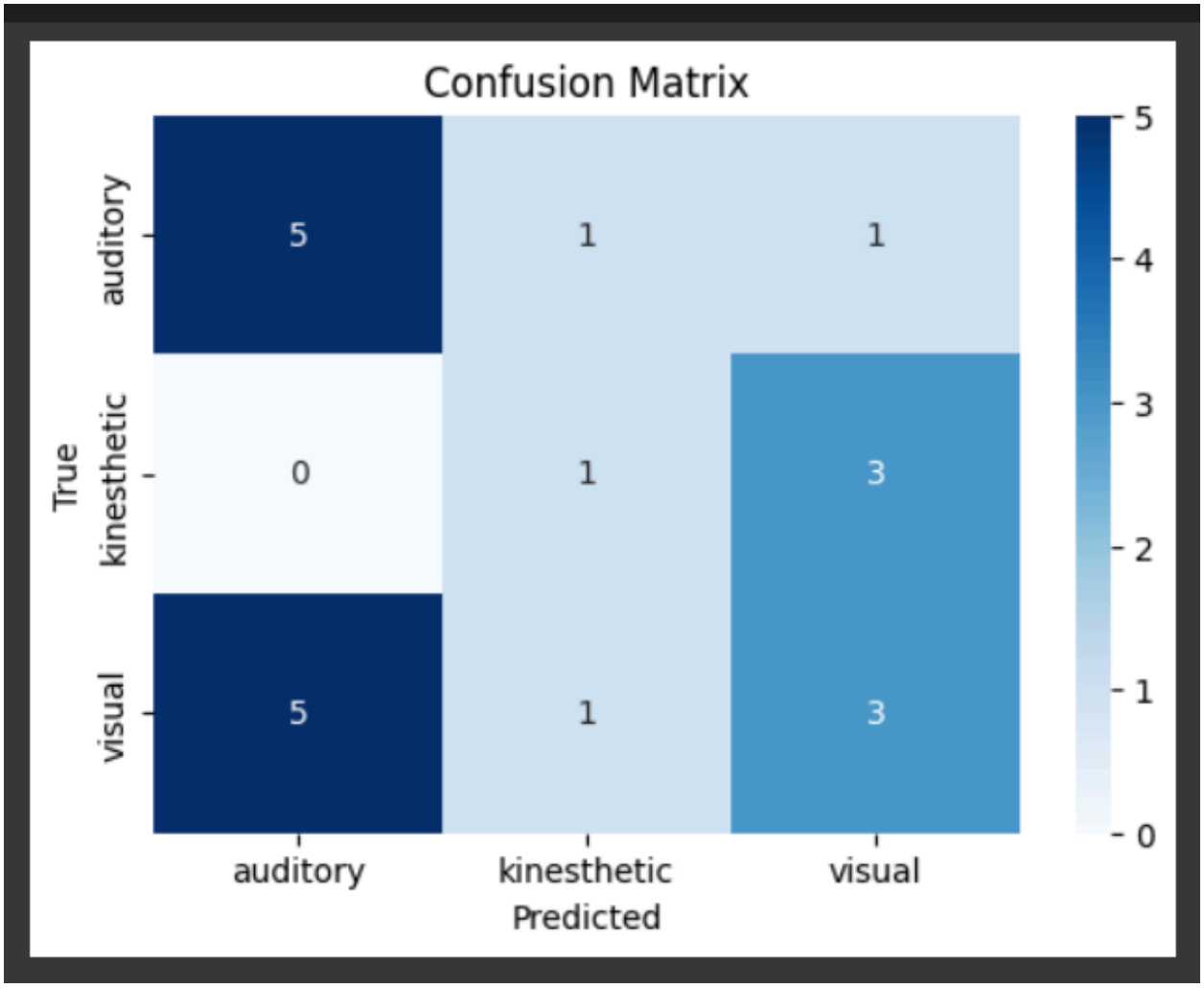
```python
# Confusion matrix heatmap
conf = confusion_matrix(y_test, y_pred)
plt.figure(figsize=(6, 4))
sns.heatmap(conf, annot=True, fmt='d', cmap='Blues',
            xticklabels=le.classes_, yticklabels=le.classes_)
plt.title("Confusion Matrix")
plt.xlabel("Predicted")
plt.ylabel("True")
plt.show()
```

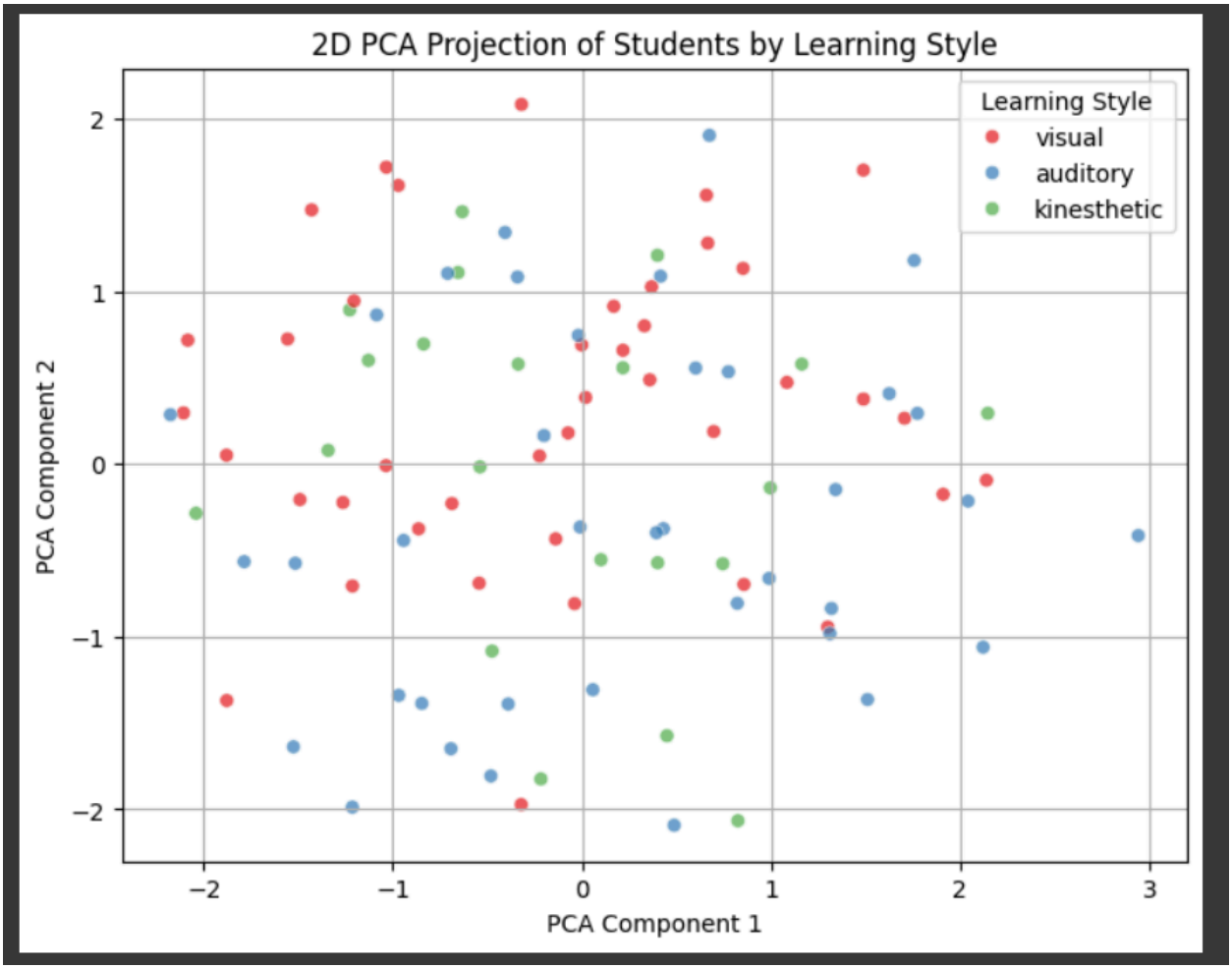
```python
# PCA for 2D visualization
pca = PCA(n_components=2)
X_pca = pca.fit_transform(X_scaled)

# Create DataFrame for PCA plot
pca_df = pd.DataFrame({
    'PCA1': X_pca[:, 0],
    'PCA2': X_pca[:, 1],
    'learning_style': df['learning_style']
})

# Plot PCA scatterplot
plt.figure(figsize=(8, 6))
sns.scatterplot(data=pca_df, x='PCA1', y='PCA2', hue='learning_style', palette='Set1', alpha=0.7)
plt.title("2D PCA Projection of Students by Learning Style")
plt.xlabel("PCA Component 1")
plt.ylabel("PCA Component 2")
plt.grid(True)
plt.legend(title='Learning Style')
plt.show()
```

OUTPUT


visual_score by Learning Style
auditory_score by Learning Style
kinesthetic_score by Learning Style
visual_score
auditory_score
kinesthetic_score
visual
auditory
kinesthetic
learning_style


Correlation between Learning Style Scores
visual_score
auditory_score
kinesthetic_score
1
0.24
0.0008
0.24
1
-0.18
0.0008
-0.18
1
1.0
0.8
0.6
0.4
0.2
0.0


Confusion Matrix
auditory
kinesthetic
visual
True
5
1
1
0
1
3
5
1
3
auditory
kinesthetic
visual
Predicted
5
4
3
2
1
0



References/Credits:

- **Accuracy Score:** Reported on the test set.

- **Classification Report:** Precision, recall, and F1-score metrics for each class.

- **Confusion Matrix:** Displayed using a heatmap.

- **PCA Visualization:** Demonstrated that KNN can reasonably distinguish between student groups in reduced dimensions.

These results confirm that learning styles can be effectively predicted using KNN with a relatively small and interpretable feature set.

1.5 CREDITS

Dataset: Provided internally for academic use.

Libraries Used:

- `Pandas`, `NumPy` – for data handling

- `Scikit-learn` – for modeling, preprocessing, evaluation

- `Matplotlib`, `Seaborn` – for visualization

Model Used: K-Nearest Neighbors (KNN)

Tools Used: Google Colab

Special Thanks: Course instructors and open-source community