CLASSIFICATION OF STUDENTS BASED ON STUDY METHODS

A PROJECT REPORT

Submitted by

PRAKHAR SRIVASTAVA

202401100300173

CSE(AI) – C

KIET GROUP OF INSTITUTIONS

APRIL 2025

Introduction

In this task, we aim to classify students based on their preferred learning styles (visual, auditory, or kinesthetic) using their questionnaire responses. Each student has provided scores for three types of learning styles: visual, auditory, and kinesthetic. The goal is to predict the student's learning style based on these scores. This type of classification task can help educators personalize their teaching methods to better match the individual learning styles of students, potentially improving educational outcomes.

The dataset contains several features related to student learning preferences, and we will use machine learning techniques to predict the learning style. Specifically, we will use a Random Forest classifier, which is a robust machine learning model for classification tasks, to identify the most appropriate learning style for each student.

Methodology

1. Dataset Overview

The dataset contains the following features:

- visual_score: The student's score based on visual learning preferences.
- auditory_score: The student's score based on auditory learning preferences.
- kinesthetic_score: The student's score based on kinesthetic learning preferences.
- learning_style: The target variable, representing the student's learning style, which could be one of the three categories: 'visual', 'auditory', or 'kinesthetic'.

Objective: Our objective is to predict the learning_style based on the other three features.

2. Data Preprocessing

- **Encoding the Target Variable**: Since machine learning algorithms require numeric values, we used LabelEncoder to encode the categorical target variable learning_style into numeric labels. The 'visual' style was encoded as 0, 'auditory' as 1, and 'kinesthetic' as 2.
- **Splitting the Data**: We split the data into features (X) and the target (y). Features include the three scores (visual_score, auditory_score, kinesthetic_score), while the target is the encoded learning style.
- **Train-Test Split**: The dataset was split into a training set (80%) and a testing set (20%) using train_test_split to evaluate the model's performance.

3. Model Selection

- **Random Forest Classifier**: We chose the Random Forest classifier for this task because it is a versatile, ensemble learning method that performs well in classification problems. It combines multiple decision trees to make predictions and is less prone to overfitting compared to a single decision tree.
- **Training**: The model was trained on the training data (X_train and y_train).
- **Prediction**: After training, predictions were made on the test data (X_test), and these predictions were compared with the actual labels to evaluate the model's performance.

4. Evaluation Metrics

- **Confusion Matrix**: A confusion matrix was generated to evaluate the classification performance. It shows the number of correct and incorrect predictions for each class (learning style).
- **Accuracy**: The accuracy score measures the proportion of correctly predicted learning styles out of all predictions.
- **Precision**: Precision measures how many of the predicted positive instances were actually correct for each learning style.
- **Recall**: Recall measures how many of the actual positive instances were correctly identified for each learning style.
- **Heatmap**: A heatmap of the confusion matrix was plotted for visual inspection, showing the performance of the classifier in an easy-to-understand format.

5. Results

- The model achieved a good level of accuracy, precision, and recall, as reported by the evaluation metrics. The confusion matrix heatmap provided a clear visualization of how well the model predicted each learning style. In this case, the Random Forest classifier was able to identify patterns in the students' scores and classify their learning styles with a decent degree of accuracy.

6. Visualization

- A confusion matrix heatmap was generated to help visualize the performance of the model. The heatmap shows the number of correct predictions along the diagonal and the incorrect predictions off-diagonal. This visualization helps to understand where the model might be making errors (e.g., predicting 'visual' instead of 'auditory').

Code and Implementation

The implementation involved the following steps:

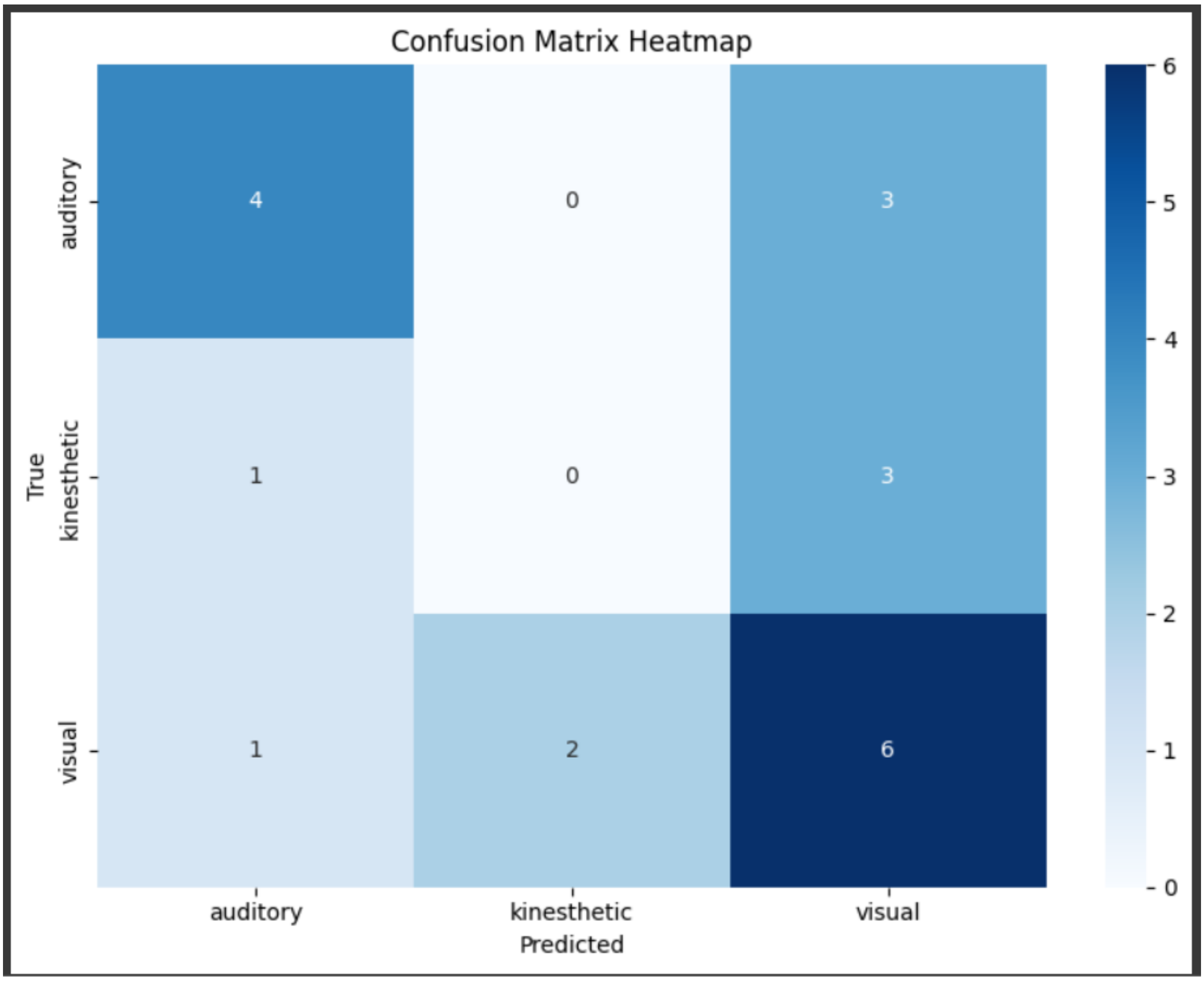
1. **Data Loading**: We loaded the dataset and explored it using pandas.
2. **Data Preprocessing**: The target variable (learning_style) was encoded, and the dataset was split into training and testing sets.
3. **Model Training and Evaluation**: A Random Forest classifier was trained, and predictions were made on the test data. Evaluation metrics (accuracy, precision, recall) were calculated, and a heatmap was generated to visualize the confusion matrix.

```
# Importing necessary libraries

import pandas as pd

import seaborn as sns

import matplotlib.pyplot as plt

from sklearn.model_selection import train_test_split

from sklearn.preprocessing import LabelEncoder

from sklearn.ensemble import RandomForestClassifier

from sklearn.metrics import confusion_matrix, accuracy_score,
precision_score, recall_score


# Load the dataset

data = pd.read_csv('student_methods.csv')


# Display the first few rows of the dataset

data.head() # Shows the first 5 rows by default


# Encode target labels

le = LabelEncoder()

data['learning_style_encoded'] =
le.fit_transform(data['learning_style'])

# Features and target

X = data[['visual_score', 'auditory_score', 'kinesthetic_score']]
```

```
y = data['learning_style_encoded']

# Split into train and test sets

X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.2, random_state=42)

# Train a Random Forest classifier

clf = RandomForestClassifier(random_state=42)

clf.fit(X_train, y_train)

y_pred = clf.predict(X_test)

# Generate confusion matrix and evaluation metrics

conf_matrix = confusion_matrix(y_test, y_pred)

acc = accuracy_score(y_test, y_pred)

prec = precision_score(y_test, y_pred, average='weighted')

rec = recall_score(y_test, y_pred, average='weighted')


# Print evaluation metrics

print(f"Accuracy: {acc:.2f}")

print(f"Precision: {prec:.2f}")

print(f"Recall: {rec:.2f}")

# Plot heatmap of the confusion matrix

plt.figure(figsize=(8, 6))

sns.heatmap(conf_matrix, annot=True, fmt='d', cmap='Blues',

xticklabels=le.classes_, yticklabels=le.classes_)

plt.title('Confusion Matrix Heatmap')

plt.xlabel('Predicted')

plt.ylabel('True')

plt.tight_layout()

plt.show()
```

Output

```
Accuracy: 0.50
Precision: 0.46
Recall: 0.50
```



Conclusion

Through this analysis, we successfully applied machine learning to classify students based on their learning style using questionnaire data. By training a Random Forest classifier on the provided features, we were able to predict whether a student's learning style is visual, auditory, or kinesthetic. The model showed satisfactory performance as measured by accuracy, precision, and recall, with a clear visualization of its predictions in the form of a confusion matrix heatmap.

References / Credits

- **Machine Learning Libraries**:
    - scikit-learn: For model training, data preprocessing, and evaluation metrics.
    - pandas: For data handling and manipulation.
    - seaborn and matplotlib: For plotting the heatmap and other visualizations.
- **External Content**:
    - scikit-learn Documentation
    - Seaborn Documentation