

Project Documentation: Personalized Student Recommendations

Table of Contents

1. Introduction
 2. Objectives
 3. Data Description
 4. Approach and Methodology
 5. Features
 6. Setup Instructions
 7. Results and Insights
 8. Future Work
 9. References
-

Introduction

This project focuses on analysing student quiz performance to generate personalized recommendations. By leveraging quiz data, the solution provides actionable insights to improve preparation for competitive exams such as NEET. The project utilizes current and historical quiz datasets to identify patterns, highlight performance gaps, and propose targeted learning strategies.

Objectives

1. Analyse quiz data to uncover performance trends by topics, difficulty levels, and response accuracy.
2. Provide students with personalized recommendations for improvement.
3. Highlight individual strengths and weaknesses to guide study efforts.
4. Bonus: Predict a student's potential NEET rank based on historical performance and past exam results.

Data Description

1. Current Quiz Data

- Contains user-specific details from the latest quiz submission.
- Includes information on questions, topics, responses, and accuracy.
- Endpoint: Quiz Submission Data API.

2. Historical Quiz Data

- Includes performance data for the last 5 quizzes per user.
- Provides scores and a response map (Key: Question ID, Value: Selected Option ID).
- Endpoint: Historical Quiz Data API.

Approach and Methodology

Step 1: Data Analysis

- **Exploration:** Load and explore the data to understand schema, patterns, and key metrics.
- **Analysis:** Analyse performance by topic, difficulty, and response accuracy.

Step 2: Insights Generation

- Identify weak areas and improvement trends for each user.
- Highlight performance gaps and create visualizations for better understanding.

Step 3: Recommendations Engine

- Develop a system to generate personalized feedback based on performance metrics.
- Suggest topics, question types, and difficulty levels for focused preparation.

Step 4: Bonus Tasks

- Define a student persona using patterns in quiz performance.
- Build a probabilistic model to predict NEET rank based on historical data and prior results.

Features

1. **Personalized Recommendations:** Actionable feedback tailored to each student's performance.
2. **Visual Insights:** Graphical representations of strengths, weaknesses, and trends.
3. **Student Persona:** Creative labelling of user-specific attributes to guide improvement.
4. **NEET Rank Prediction:** Probabilistic model predicting rank based on performance metrics

1 Analyses of performance by topic based upon difficulty

In [790...

```
topic_distribution = df['quiz.topic'].value_counts()
difficulty_distribution = df['quiz.difficulty_level'].value_counts()

print("\nTopic Distribution:")
print(topic_distribution)
print("\nDifficulty Level Distribution:")
print(difficulty_distribution)
```

```
Topic Distribution:
quiz.topic
Body Fluids And Circulation      6
Reproductive Health              3
Structural Organisation In Animals 1
Human Reproduction               1
Principles Of Inheritance And Variation 1
Microbes In Human Welfare        1
Human Health And Disease         1
Respiration And Gas Exchange     1
Name: count, dtype: int64

Difficulty Level Distribution:
quiz.difficulty_level
Unknown      15
Name: count, dtype: int64
```

2 Average Performance By Topic wise -

In [796...

```
##Performance
df.loc[:, 'quiz.topic'] = df['quiz.topic'].str.strip().str.title()

# Now, group again and calculate the mean
topic_performance = df.groupby('quiz.topic')[['score', 'accuracy']].mean()
print("\nAverage Performance by Topic:")
print(topic_performance)
```

```
Average Performance by Topic:
```

	score	accuracy
quiz.topic		
Body Fluids And Circulation	70.666667	7.616667e+11
Human Health And Disease	112.000000	9.300000e+11
Human Reproduction	40.000000	3.800000e+11
Microbes In Human Welfare	76.000000	1.000000e+12
Principles Of Inheritance And Variation	12.000000	3.000000e+11
Reproductive Health	52.000000	7.566667e+11
Respiration And Gas Exchange	24.000000	6.600000e+11
Structural Organisation In Animals	32.000000	8.000000e+11

3 – Performance By Topic and Weak areas Where User need Improvements

```
Counts of is_correct (True/False):
is_correct
0      9
1      6
Name: count, dtype: int64
Unique Topics in quiz.topic:
['Structural Organisation In Animals' 'Body Fluids And Circulation'
 'Human Reproduction' 'Principles Of Inheritance And Variation'
 'Microbes In Human Welfare' 'Reproductive Health'
 'Human Health And Disease' 'Respiration And Gas Exchange']
```

```
Performance by Topic (in %):
quiz.topic
Body Fluids And Circulation      50.000000
Human Health And Disease        100.000000
Human Reproduction               0.000000
Microbes In Human Welfare       100.000000
Principles Of Inheritance And Variation  0.000000
Reproductive Health             33.333333
Respiration And Gas Exchange    0.000000
Structural Organisation In Animals  0.000000
Name: is_correct, dtype: float64
```

Average performance: 40.0%

```
Weak Areas (where user performance is lower than average):
quiz.topic
Human Reproduction      0.000000
Principles Of Inheritance And Variation  0.000000
Reproductive Health     33.333333
Respiration And Gas Exchange  0.000000
Structural Organisation In Animals  0.000000
Name: is_correct, dtype: float64
```

4 – Performance Gap

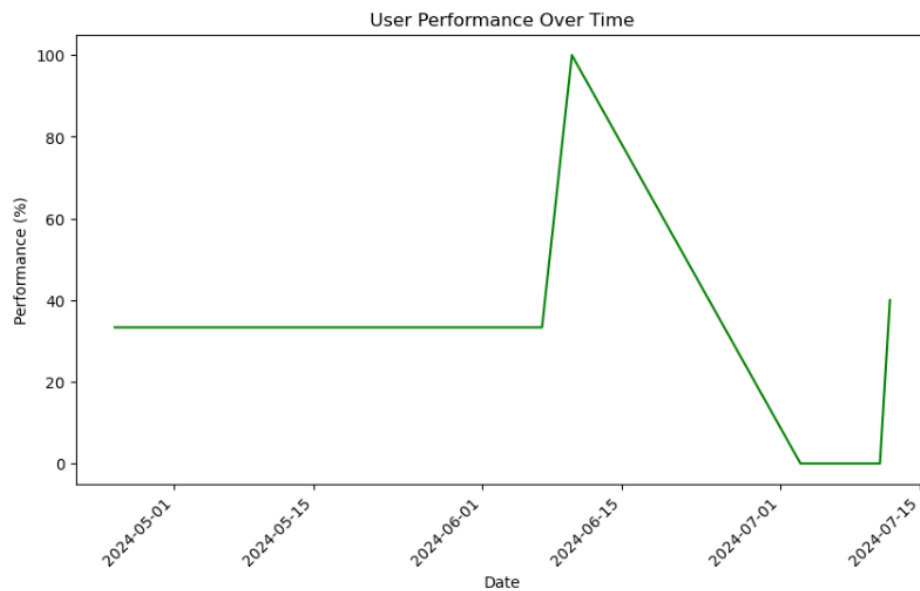
```
In [923...  ##Calculate performance gaps by topic
performance_gap = performance_by_topic - overall_performance
print("\nPerformance Gap by Topic (relative to overall average):")
print(performance_gap)
```

```
Performance Gap by Topic (relative to overall average):
quiz.topic
Body Fluids And Circulation      50.000000
Human Health And Disease        100.000000
Human Reproduction               0.000000
Microbes In Human Welfare       100.000000
Principles Of Inheritance And Variation  0.000000
Reproductive Health             33.333333
Respiration And Gas Exchange    0.000000
Structural Organisation In Animals  0.000000
Name: is_correct, dtype: float64
```

5 Improvement Trends

```
In [927]: ## Improvement Trends (if quiz_date is available)
if 'quiz_date' in df.columns:
    df['quiz_date'] = pd.to_datetime(df['quiz_date']) # Ensure it's in datetime format
    performance_time_trend = df.groupby('quiz_date')['is_correct'].mean() * 100

    # Plotting the trend over time
    performance_time_trend.plot(kind='line', figsize=(10, 6), color='green')
    plt.title('User Performance Over Time')
    plt.xlabel('Date')
    plt.ylabel('Performance (%)')
    plt.xticks(rotation=45)
    plt.show()
```



6 - Complete Data Insights and Recommendations

Overall Performance: 40.0%

Performance by Topic (in %):

quiz.topic	
Body Fluids And Circulation	50.000000
Human Health And Disease	100.000000
Human Reproduction	0.000000
Microbes In Human Welfare	100.000000
Principles Of Inheritance And Variation	0.000000
Reproductive Health	33.333333
Respiration And Gas Exchange	0.000000
Structural Organisation In Animals	0.000000

Name: is_correct, dtype: float64

Weak Areas (where performance is lower than average):

quiz.topic	
Human Reproduction	0.000000
Principles Of Inheritance And Variation	0.000000
Reproductive Health	33.333333
Respiration And Gas Exchange	0.000000
Structural Organisation In Animals	0.000000

Name: is_correct, dtype: float64

Suggested Topics to Focus On:

- Human Reproduction
- Principles Of Inheritance And Variation
- Reproductive Health
- Respiration And Gas Exchange
- Structural Organisation In Animals

No difficulty level data available.

Study Plan:

1. Focus on the following topics where you need improvement:
 - Human Reproduction, Principles Of Inheritance And Variation, Reproductive Health, Respiration And Gas Exchange, Structural Organisation In Animals
 2. Start practicing questions at the easy difficulty level to build confidence.
 3. Gradually increase the difficulty level after mastering the basics.
 4. Track progress regularly by checking performance on each topic.
 5. Consider practicing different question types (e.g., multiple choice, short answer) to improve speed and accuracy.
-

7 -Probabilistic model that predicts the student's NEET rank based on their quiz performance based upon the data

```
model.fit(X_train, y_train)

# Predictions
y_pred = model.predict(X_test)

# Evaluate the model
mae = mean_absolute_error(y_test, y_pred)

# Feature importance
feature_importances = model.feature_importances_

# Clean Output
print("Model Evaluation Results:")
print(f"Mean Absolute Error: {mae:.2f}\n")

print("Feature Importances:")
for feature, importance in zip(features, feature_importances):
    print(f"{feature.capitalize()}: {importance:.4f}")

# Predicting for new data (Example)
new_data = pd.DataFrame({'score': [80], 'accuracy': [95]})
predicted_rank = model.predict(new_data)
print(f"\nPredicted NEET Rank for score 80 and accuracy 95: {predicted_rank[0]:.2f}")
```

Model Evaluation Results:
Mean Absolute Error: 2132.50

Feature Importances:
Score: 0.6706
Accuracy: 0.3294

Predicted NEET Rank for score 80 and accuracy 95: 3517.14

Thank You

Syed Asad

Code Base GitHub - <https://github.com/SyedAsad777/Personalized-Student-Recommendations-/blob/main/Assessment.ipynb>