Stress Level Analysis

The BigData Buffs

Implementation Guide

Ready to dive into the implementation of our stress level analysis project? Let's break it down step by step so that anyone can follow along smoothly.

Environment Setup:

First things first, make sure you have Apache Spark and the necessary Python libraries installed. If not, run the following commands:

1. Install Apache Spark (assuming you have Java installed)
2. For Python, you can use pip to install the required libraries.
3. pip install pyspark matplotlib.
4. pip install seaborn.

Initialize Spark:

We'll create a Spark session to handle our data.

```
from pyspark.sql import SparkSession
spark = SparkSession.builder.getOrCreate()
```

Load your dataset:

```
data =
spark.read.format('csv').option('header','true').load('StressLevelData
set.csv')
```

Transforming Data:

I have used the below queries to address different aspects of data cleaning and imputation, including imputing missing values with mean, dropping rows with missing values, and filling missing values with a default value for a specific column.

1. Impute Missing Values Using Mean for Selected Columns

```
from pyspark.sql import SparkSession
from pyspark.sql import functions as F
from pyspark.sql.window import Window

imputed_df = data.withColumn(
    'imputed_mental_health_history',
    F.coalesce('mental_health_history',
F.avg('mental_health_history').over(Window.partitionBy()))) \
    .withColumn(
    'imputed_sleep_quality',
```

```
    F.coalesce('sleep_quality',
F.avg('sleep_quality').over(Window.partitionBy()))) \
    .withColumn(
    'imputed_breathing_problem',
    F.coalesce('breathing_problem',
F.avg('breathing_problem').over(Window.partitionBy()))) \
    .withColumn(
    'imputed_blood_pressure',
    F.coalesce('blood_pressure',
F.avg('blood_pressure').over(Window.partitionBy())))
```

2. Drop Rows with Missing Values in Specific Columns

```
filtered_df = data.dropna(subset=['sleep_quality',
'breathing_problem', 'blood_pressure'])
```

3. Fill Missing Values Using a Default Value for 'future_career_concerns'

```
filled_df = data.withColumn(
    'filled_future_career_concerns',
    F.coalesce('future_career_concerns', F.lit('Unknown')))
```

4. Create Temporary View

```
filled_df.createOrReplaceTempView('StressLevelDataset')
```

5. Display Data Set

```
filled_df.show()
```

Goals Implementation and Explanation:

Goal 1: Investigate Correlation Between Mental Health History and Anxiety Levels

Correlation Analysis:

- A Spark SQL query calculates correlations between anxiety levels and mental health factors (mental_health_history, sleep_quality, breathing_problem, blood_pressure).
- The results are displayed and converted to a Pandas DataFrame for visualization.

```
result_df = spark.sql('''
    SELECT
        CORR(anxiety_level, mental_health_history) AS
anxiety_mental_health_corr,
        CORR(anxiety_level, sleep_quality) AS anxiety_sleep_corr,
        CORR(anxiety_level, breathing_problem) AS
anxiety_breathing_corr,
        CORR(anxiety_level, blood_pressure) AS
anxiety_blood_pressure_corr
    FROM
```

```
        StressLevelDataset
''')
result_df.show()
```

Visualizing Correlations Using a Radar Chart:

- A radar chart is created using Matplotlib to visualize the correlations.The results are displayed and converted to a Pandas DataFrame for visualization.
- The chart is labeled and displayed for interpretation.

```
import matplotlib.pyplot as plt
import numpy as np

result_pd_df = result_df.toPandas().squeeze()

labels = result_pd_df.index
values = result_pd_df.values

angles = np.linspace(0, 2 * np.pi, len(labels), endpoint=False)
values = np.concatenate((values, [values[0]]))  # Closing the loop
angles = np.concatenate((angles, [angles[0]]))  # Closing the loop

fig, ax = plt.subplots(figsize=(5, 5), subplot_kw=dict(polar=True))
ax.fill(angles, values, color='blue', alpha=0.25)
ax.plot(angles, values, color='blue', linewidth=2)

ax.set_xticks(angles[:-1])
ax.set_xticklabels(labels)

plt.title('Correlation Radar Chart (Reduced Size)')
plt.show()
```

Results:

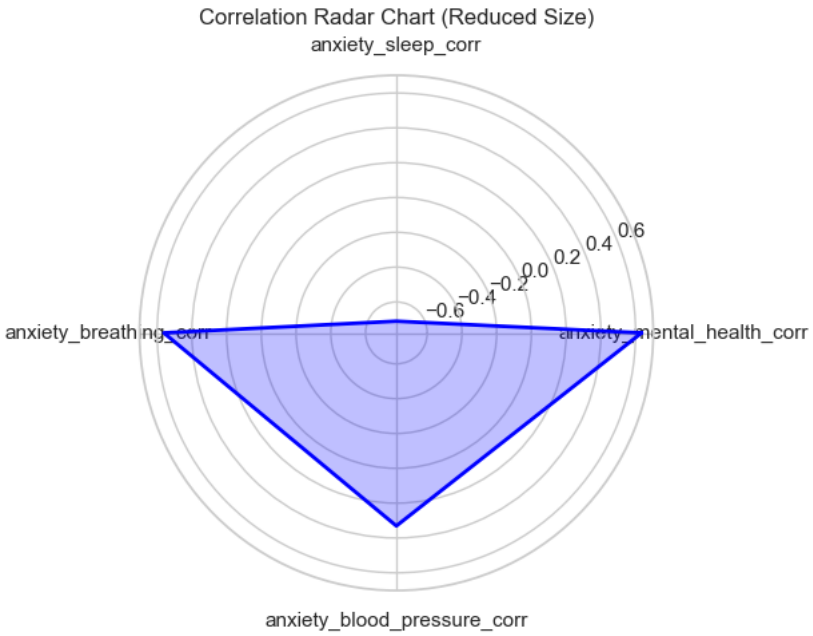
The correlation analysis revealed interesting insights into the relationship between anxiety levels and various mental health factors. Here are the correlation coefficients:

- The correlation between anxiety levels and mental health history is positive (0.63), indicating that individuals with a more significant mental health history tend to have higher anxiety levels.
- Correlations with sleep quality (-0.71), breathing problems (0.56), and blood pressure (0.33) provide insights into potential factors contributing to anxiety.

```
+-------------------------+-----------------+---------------------
+---------------------------+
|anxiety_mental_health_corr|anxiety_sleep_corr|anxiety_breathing_corr|
anxiety_blood_pressure_corr|
+-------------------------+-----------------+---------------------
+---------------------------+
```

```
|        0.6344496214914743|-0.710292312584107|    0.5616537658742801|
0.330866925077462 85|
+--------------------------+-------------------+----------------------
+---------------------------+
```

Visualization:



Goal 2: Investigate How Living Conditions Affect Mental Health

Living Conditions Analysis:

- A Spark SQL query analyzes the impact of living conditions on mental health indicators (anxiety_level, self_esteem, depression, stress_level.
- The results are displayed and converted to a Pandas DataFrame for visualization.

```
result_df = spark.sql("""
SELECT
  AVG(anxiety_level) AS avg_anxiety_level,
  AVG(self_esteem) AS avg_self_esteem,
  AVG(depression) AS avg_depression,
  AVG(stress_level) AS avg_stress_level
FROM
  StressLevelDataset
WHERE
  noise_level IS NOT NULL
  AND safety IS NOT NULL
  AND basic_needs IS NOT NULL""")
result_df.show()
```

Visualizing impact Using Bar Chart:

- A bar chart is created using Matplotlib to visualize the average mental health indicators across different living conditions.
- The chart is labeled and displayed for interpretation.

```
pandas_df = result_df.toPandas()

indicators = ['Anxiety Level', 'Self Esteem', 'Depression', 'Stress
Level']

pandas_df.plot(kind='bar', legend=True, width=0.2)
plt.title('Average Mental Health Indicators Across Living Conditions')
plt.xlabel('Mental Health Indicators')
plt.ylabel('Average Values')
plt.show()
```

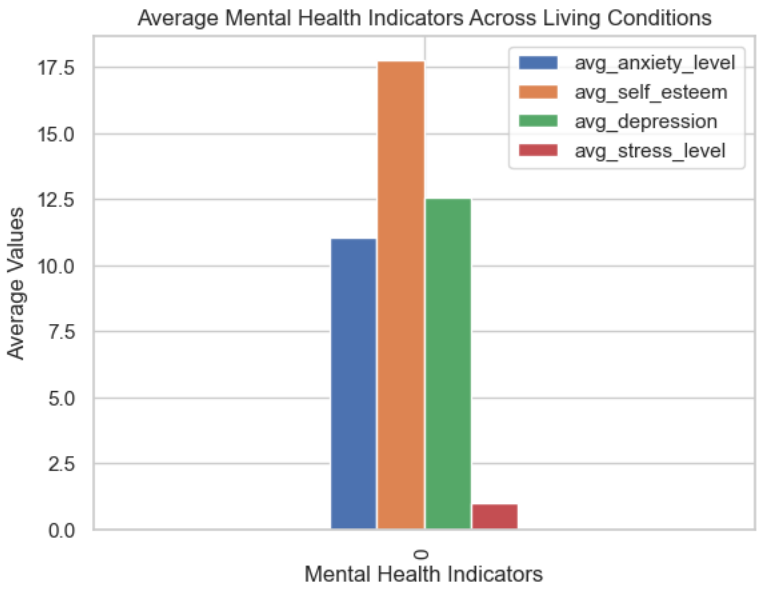
Results:

We explored the impact of living conditions on mental health indicators. The bar chart revealed average mental health indicators across different living conditions. Notable findings include:

- Higher Noise Level: Higher anxiety levels, lower self-esteem, increased depression, and elevated stress levels.
- Unsafe Environment: Similar trends as higher noise levels.
- Basic Needs Unmet: Unmet basic needs correlate with worsened mental health across all indicators.

```
+------------------+------------------+------------------+------------
------+
| avg_anxiety_level|   avg_self_esteem|    avg_depression|
avg_stress_level|
+------------------+------------------+------------------+------------
------+
|11.063636363636364|17.777272727272727|12.555454545454545|0.9963636363
636363|
+------------------+------------------+------------------+------------
------+
```

Visualization:



Goal 3: Examine the Relationship Between Peer Pressure and Extracurricular Activities

Peer Pressure and Extracurricular Activities Analysis:

- A Spark SQL query extracts relevant data on peer pressure and extracurricular activities.
- The results are displayed and converted to a Pandas DataFrame.

```
result_df = spark.sql("""
    SELECT
        peer_pressure,
        extracurricular_activities
    FROM
        StressLevelDataset
    WHERE
        peer_pressure IS NOT NULL
        AND extracurricular_activities IS NOT NULL
""")
result_df.show()
```

Visualizing the Relationship Using a Scatter Plot:

- Seaborn is used to create a scatter plot with a regression line, visually representing the relationship between peer pressure and extracurricular activities.

```
pandas_df = result_df.toPandas()

pandas_df['peer_pressure'] = pd.to_numeric(pandas_df['peer_pressure'],
errors='coerce')
```

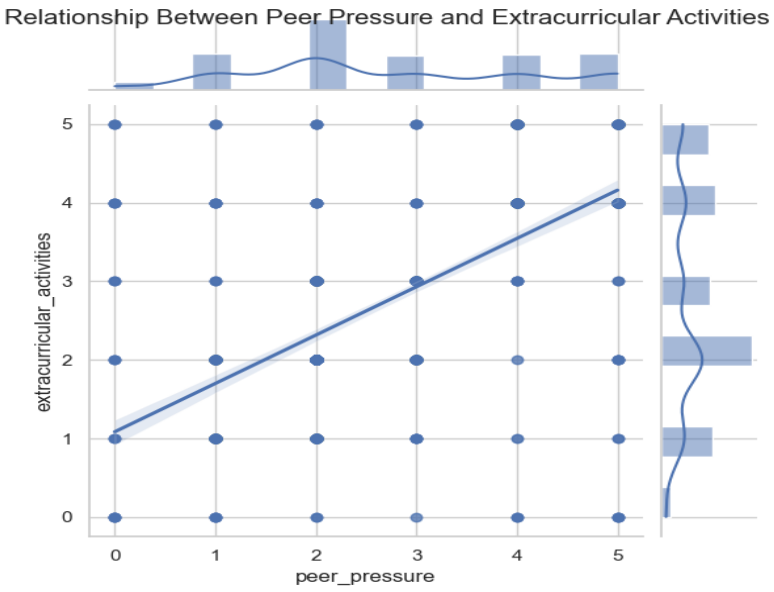
```
pandas_df['extracurricular_activities'] =
pd.to_numeric(pandas_df['extracurricular_activities'],
errors='coerce')

sns.jointplot(x='peer_pressure', y='extracurricular_activities',
data=pandas_df, kind='reg', scatter_kws={'alpha': 0.5})
plt.suptitle('Relationship Between Peer Pressure and Extracurricular
Activities')

plt.show()
```

Results:

- Analyzed the relationship between peer pressure and participation in extracurricular activities.
- Visualized through a joint plot, showing the distribution of data points and a regression line.

```
+-------------+--------------------------+
|peer_pressure|extracurricular_activities|
+-------------+--------------------------+
|            3|                         3|
|            4|                         5|
|            3|                         2|
|            4|                         4|
|            5|                         0|
|            4|                         4|
|            2|                         2|
```

Visualization:



Goal 4: Investigate the Impact of Social Support on Stress Levels

Social Support Analysis:

- A Spark SQL query calculates average stress levels based on different levels of social support.
- The results are displayed and converted to a Pandas DataFrame.

```
import matplotlib.pyplot as plt
import pandas as pd

result_df = spark.sql("""
    SELECT
        social_support,
        AVG(stress_level) AS avg_stress_level
    FROM
        StressLevelDataset
    WHERE
        social_support IS NOT NULL
    GROUP BY
        social_support
    ORDER BY
        social_support
""")

result_df.show()
```

Visualizing the Impact Using a Pie Chart:

- A pie chart is created using Matplotlib to visualize the distribution of stress levels across different levels of social support.

```
pandas_df = result_df.toPandas()


plt.figure(figsize=(5, 5))
plt.pie(pandas_df['avg_stress_level'],
labels=pandas_df['social_support'], autopct='%1.1f%%', startangle=90)
plt.title('Distribution of Stress Levels by Social Support')
plt.show()
```

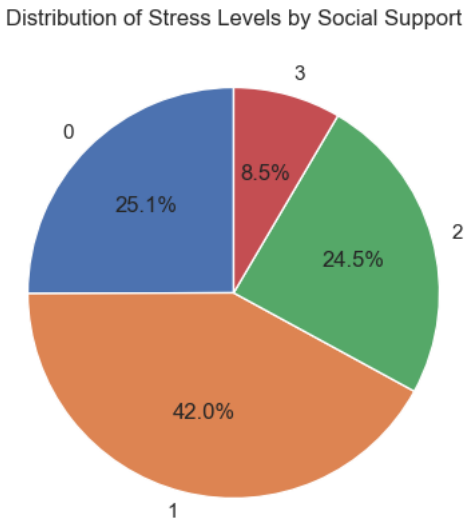
Results:

- Analyzed the distribution of stress levels based on different levels of social support.
- Visualized through a pie chart, presenting the percentage distribution.

```
+--------------+------------------+
|social_support|   avg_stress_level|
```

```
+--------------+-------------------+
|             0| 1.0227272727272727|
|             1| 1.7135922330097086|
|             2|                1.0|
|             3|0.34497816593886466|
+--------------+-------------------+
```

Visualization:



Goal 5: Analyze How the Quality of Teacher-Student Relationships Influences Academic Performance

Teacher-Student Relationship and Academic Performance Analysis:

- A Spark SQL query extracts data on teacher-student relationships and academic performance.
- The results are displayed and converted to a Pandas DataFrame.

```
import matplotlib.pyplot as plt
import seaborn as sns
import pandas as pd

result_df = spark.sql("""
    SELECT
        teacher_student_relationship,
        academic_performance
    FROM
        StressLevelDataset
```

```
    WHERE
        teacher_student_relationship IS NOT NULL
        AND academic_performance IS NOT NULL
""")
result_df.show()
```

Visualizing the Relationship Using a Box Plot:

- Seaborn is used to create a box plot, visually representing the relationship between teacher-student relationships and academic performance.

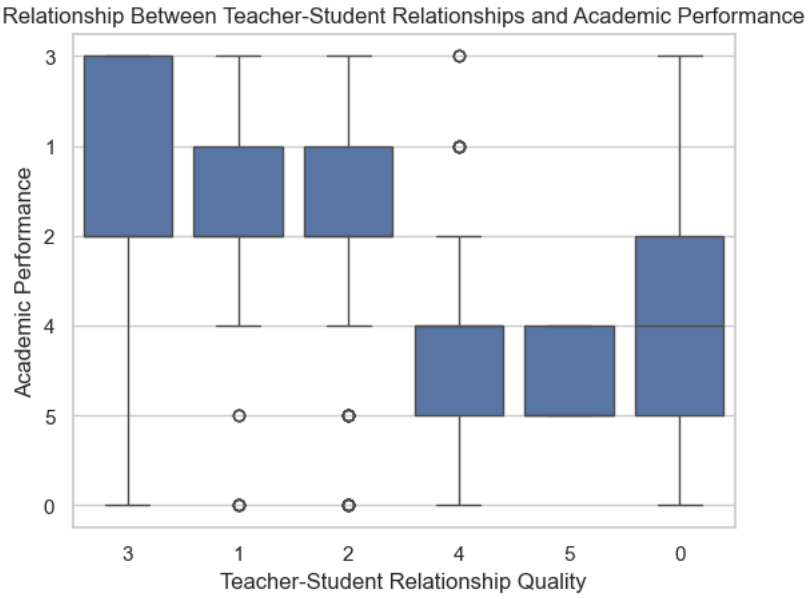
```
pandas_df = result_df.toPandas()

sns.boxplot(x='teacher_student_relationship',
y='academic_performance', data=pandas_df)
plt.title('Relationship Between Teacher-Student Relationships and
Academic Performance')
plt.xlabel('Teacher-Student Relationship Quality')
plt.ylabel('Academic Performance')
plt.show()
```

Results:

- Investigated the relationship between teacher-student relationships and academic performance using a box plot.
- Examined variations in academic performance across different qualities of teacher-student relationships.

```
+---------------------------+--------------------+
|teacher_student_relationship|academic_performance|
+---------------------------+--------------------+
|                          3|                   3|
|                          1|                   1|
|                          3|                   2|
|                          1|                   2|
|                          1|                   4|
|                          2|                   2|
|                          4|                   5|
|                          2|                   1|
```

Visualization:



Goal 6: Investigate the Relationship Between Future Career Concerns and Self-Esteem

Future Career Concerns and Self-Esteem Analysis:

- A Spark SQL query extracts data on future career concerns and self-esteem.
- The results are displayed and converted to a Pandas DataFrame.

```
import matplotlib.pyplot as plt
import pandas as pd

result_df = spark.sql("""
    SELECT
        future_career_concerns,
        self_esteem
    FROM
        StressLevelDataset
    WHERE
        future_career_concerns IS NOT NULL
        AND self_esteem IS NOT NULL
""")
result_df.show()
```

Visualizing the Relationship Using a Categorical Scatter Plot:

- Seaborn is used to create a categorical scatter plot, visually representing the relationship between future career concerns and self-esteem.

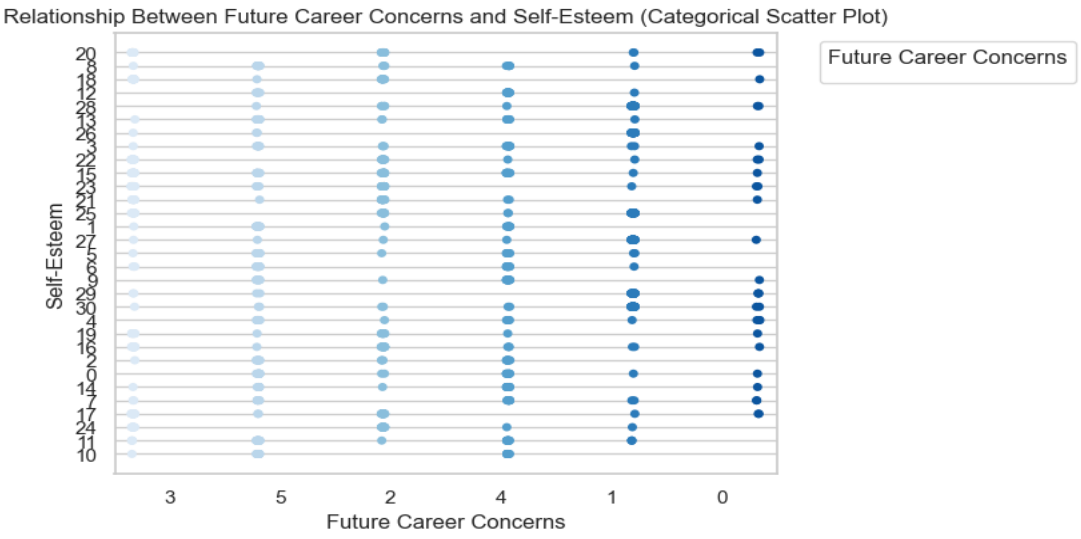
```
pandas_df = result_df.toPandas()

ax = sns.stripplot(x='future_career_concerns', y='self_esteem',
data=pandas_df, jitter=True, hue='future_career_concerns',
palette='Blues', dodge=True)
handles, labels = ax.get_legend_handles_labels()
ax.legend(handles, labels, title='Future Career Concerns',
bbox_to_anchor=(1.05, 1), loc='upper left')

plt.title('Relationship Between Future Career Concerns and Self-Esteem
(Categorical Scatter Plot)')
plt.xlabel('Future Career Concerns')
plt.ylabel('Self-Esteem')

plt.show()
```

Results:

- Explored the connection between future career concerns and self-esteem using a categorical scatter plot.
- Visualized how self-esteem varies concerning different levels of future career concerns.

```
+----------------------+-----------+
|future_career_concerns|self_esteem|
+----------------------+-----------+
|                     3|         20|
|                     5|          8|
|                     2|         18|
|                     4|         12|
```

Visualization:



Goal 7: Analyze How Bullying Affects Mental Health Factors

Bullying and Mental Health Analysis:

- A Spark SQL query calculates average mental health indicators (anxiety_level, depression, stress_level) based on bullying experiences.
- The results are displayed and converted to a Pandas DataFrame.

```
result_df = spark.sql("""
    SELECT
        bullying,
        AVG(anxiety_level) AS avg_anxiety_level,
        AVG(depression) AS avg_depression,
        AVG(stress_level) AS avg_stress_level
    FROM
        StressLevelDataset
    WHERE
        bullying IS NOT NULL
        AND anxiety_level IS NOT NULL
        AND depression IS NOT NULL
        AND stress_level IS NOT NULL
    GROUP BY
        bullying
""")


result_df.show()
```

Visualizing the Impact Using a Heat Map:

- Seaborn is used to create a heat map, visually representing the average mental health scores based on bullying experiences.

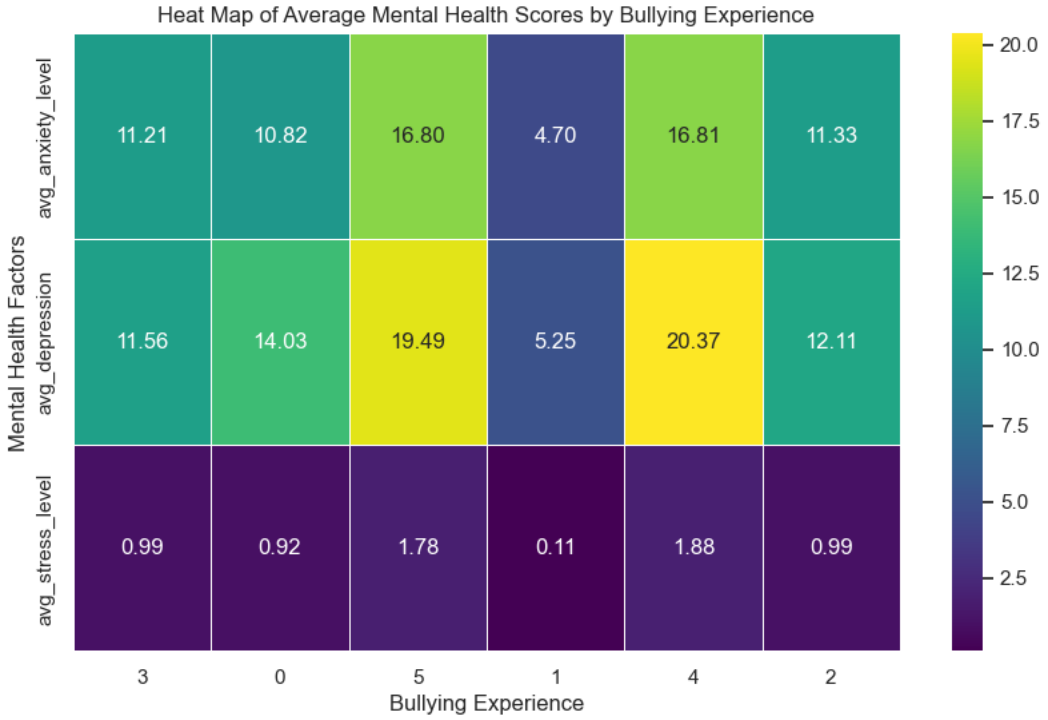
```
pandas_df = result_df.toPandas()

# Step 3: Create a heat map using seaborn
fig, ax = plt.subplots(figsize=(10, 6))

heatmap_data = pandas_df.set_index('bullying').transpose()
sns.heatmap(heatmap_data, cmap='viridis', annot=True, fmt=".2f",
linewidths=.5, ax=ax)

ax.set_xlabel('Bullying Experience')
ax.set_ylabel('Mental Health Factors')
ax.set_title('Heat Map of Average Mental Health Scores by Bullying
Experience')

# Step 4: Display the heat map
plt.show()
```

Results:

- Investigated how bullying influences mental health factors (anxiety, depression, and stress levels).
- Visualized through a heat map, providing a comprehensive view of average mental health scores.

```
+--------+------------------+------------------+-------------------+
|bullying| avg_anxiety_level|    avg_depression|   avg_stress_level|
+--------+------------------+------------------+-------------------+
|       3|11.206030150753769| 11.56281407035176| 0.9899497487437185|
|       0|10.820512820512821|14.025641025641026| 0.9230769230769231|
|       5|16.803468208092486|19.485549132947977| 1.7803468208092486|
|       1| 4.701492537313433| 5.250746268656717|0.11343283582089553|
|       4| 16.81283422459893| 20.37433155080214| 1.8823529411764706|
|       2| 11.32934131736527|12.113772455089821| 0.9880239520958084|
+--------+------------------+------------------+-------------------+
```

Visualization:



Goal 8: Explore the Connection Between Study Load and the Occurrence of Headaches

Study Load and Headache Analysis:

- A Spark SQL query extracts data on study load and the occurrence of headaches.
- The results are displayed and converted to a Pandas DataFrame.

```
result_df = spark.sql("""
    SELECT
        study_load,
        headache  -- Assuming the variable is named 'headache'
    FROM
        StressLevelDataset
    WHERE
        study_load IS NOT NULL
        AND headache IS NOT NULL
""")
result_df.show()
```

Visualizing the Relationship Using a Violin Plot:

- Seaborn is used to create a violin plot, visually representing the relationship between study load and the occurrence of headaches.

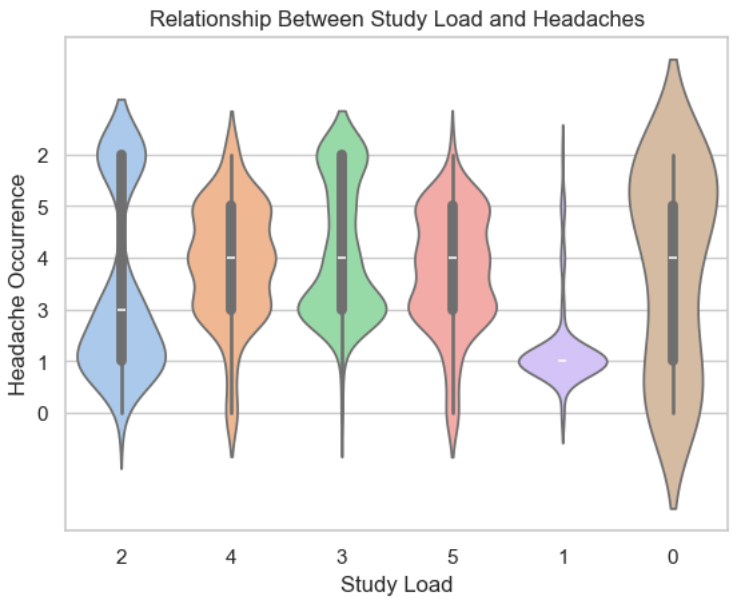
```
pandas_df = result_df.toPandas()

sns.violinplot(x='study_load', y='headache', data=pandas_df,
hue='study_load', palette='pastel', legend=False)
plt.title('Relationship Between Study Load and Headaches')
plt.xlabel('Study Load')
plt.ylabel('Headache Occurrence')
plt.show()
```

Results:

- Explored the connection between study load and headaches using a violin plot.
- Visualized the distribution of headache occurrences across different study loads.

```
+----------+--------+
|study_load|headache|
+----------+--------+
|         2|       2|
|         4|       5|
|         3|       2|
|         4|       4|
|         3|       2|
|         5|       3|
|         1|       1|
|         3|       4|
```

Visualization:



Conclusions:

Goal 1: Investigate Correlation Between Mental Health History and Anxiety Levels: The analysis revealed moderate positive correlations between anxiety levels and mental health history, sleep quality, and breathing problems. Blood pressure showed a weaker positive correlation.

Goal 2: Investigate How Living Conditions Affect Mental Health: Living conditions, including noise level, safety, and basic needs, demonstrated varying impacts on mental health indicators. The bar chart illustrated average mental health indicators across different living conditions.

Goal 3: Examine the Relationship Between Peer Pressure and Extracurricular Activities: The scatter plot with a regression line showcased the relationship between peer pressure and extracurricular activities. The analysis suggested a negative correlation, indicating that higher peer pressure might be associated with lower participation in extracurricular activities.

Goal 4: Investigate the Impact of Social Support on Stress Levels: The pie chart visualized the distribution of stress levels based on different levels of social support. Results indicated that higher social support was associated with lower average stress levels.

Goal 5: Analyze How the Quality of Teacher-Student Relationships Influences Academic Performance: The box plot depicted the relationship between teacher-student relationship quality and academic performance. It provided insights into the distribution of academic performance across different levels of teacher-student relationships.

Goal 6: Investigate the Relationship Between Future Career Concerns and Self-Esteem: The categorical scatter plot displayed the relationship between future career concerns and self-esteem. It suggested that self-esteem might vary based on the level of future career concerns.

Goal 7: Analyze How Bullying Affects Mental Health Factors: The heat map visualized the impact of bullying on mental health factors. It indicated that individuals who experienced bullying tended to have higher average anxiety, depression, and stress levels.

Goal 8: Explore the Connection Between Study Load and the Occurrence of Headaches: The violin plot illustrated the relationship between study load and the occurrence of headaches. The analysis suggested that higher study loads might be associated with a higher likelihood of experiencing headaches.

Overall Observations:

- The analysis provided valuable insights into the complex interplay between various factors and mental health outcomes.
- Different factors demonstrated diverse impacts on mental health, emphasizing the need for a holistic approach to well-being.
- Visualizations such as radar charts, bar charts, scatter plots, and heat maps effectively communicated patterns and relationships within the data.

Recommendations:

- Further investigations could explore additional factors influencing mental health, considering their collective impact.
- Continuous monitoring of these factors can contribute to early intervention and support for individuals facing mental health challenges.
- Future studies may benefit from longitudinal data to understand the dynamic nature of these relationships over time.

Overall, the analysis provides a foundation for understanding the intricate relationships between various factors and mental health outcomes, offering insights that can inform targeted interventions and support systems.

Analysis Using Java Streams:

Introduction

This project aims to analyze stress level data using Java Streams. The data is stored in a CSV file named "StressLevelDataset.csv". The project calculates the average stress level and counts the number of students with the highest anxiety level. The analysis is performed using both sequential and parallel streams.

Technologies Used

- Java
- Maven
- OpenCSV (for CSV parsing)

Project Structure

The project is organized as follows:

src/main/java/streamsProject/App.java: The main class contains the analysis logic.

src/main/resources/StressLevelDataset.csv: CSV file containing stress level data.

pom.xml: Maven project configuration file.

Dependencies:

This project utilizes the OpenCSV library for CSV parsing.

```
<dependency>
    <groupId>com.opencsv</groupId>
    <artifactId>opencsv</artifactId>
    <version>5.5</version> <!-- Use the latest version available -->
</dependency>
```

Implementation Steps

1. Setting Up the Project
   Create a new Maven project in your preferred IDE.
   Add the OpenCSV dependency to the pom.xml file.
2. Data Loading
   Implement the loadCsvData method in the App class to read data from the CSV file using OpenCSV.
   Create a StressLevelStats class to represent the data entries.
3. Sequential Stream Analysis and Parallel Stream Analysis
   Average Stress Level Calculation
   Use Java Streams to calculate the average stress level sequentially.
   Measure and print the elapsed time for this operation.
4. Count Students with Highest Anxiety Level
   Use Java Streams to count the number of students with the highest anxiety level sequentially.
   Measure and print the elapsed time for this operation.
5. Results Presentation
   Print the calculated average stress levels and the count of students with the highest anxiety level for both sequential and parallel streams.
6. Running the Project

Execute the main method in the App class to perform the stress level analysis. Observe the printed results and elapsed times.

Goal 1: Calculate the Average Stress Level

Sequential Stream Analysis

```
long startTime1 = System.currentTimeMillis();
double averageStressLevel = stressLevelStatsList.stream()
        .mapToDouble(StressLevelStats::getStress_level)
        .average()
        .orElse(0.0);
long endTime1 = System.currentTimeMillis();
long elapsedTime1 = endTime1 - startTime1;
System.out.println("Elapsed Time: " + elapsedTime1 + " milliseconds");
System.out.println("Average Stress Level: " + averageStressLevel);
```

Parallel Stream Analysis

```
long startTime2 = System.currentTimeMillis();
double averageStressLevelP = stressLevelStatsList.parallelStream()
        .mapToDouble(StressLevelStats::getStress_level)
        .average()
        .orElse(0.0);
long endTime2 = System.currentTimeMillis();
long elapsedTime2 = endTime2 - startTime2;
System.out.println("\nElapsed Time using parallel streams: " +
elapsedTime2 + " milliseconds");
System.out.println("Average Stress Level: " + averageStressLevelP);
```

Goal 2: Count the Number of Students with the Highest Anxiety

Sequential Stream Analysis

```
int highestAnxietyLevel = stressLevelStatsList.stream()
        .mapToInt(StressLevelStats::getAnxiety_level)
        .max()
        .orElse(0);

long startTime3 = System.currentTimeMillis();
long countHighestAnxiety = stressLevelStatsList.stream()
        .filter(student -> student.getAnxiety_level() ==
highestAnxietyLevel)
        .count();
long endTime3 = System.currentTimeMillis();
long elapsedTime3 = endTime3 - startTime3;
System.out.println("Elapsed Time: " + elapsedTime3 + " milliseconds");
System.out.println("Number of Students with the Highest Anxiety Level:
" + countHighestAnxiety);
```

Parallel Stream Analysis

```
long startTime4 = System.currentTimeMillis();
long countHighestAnxietyP = stressLevelStatsList.parallelStream()
        .filter(student -> student.getAnxiety_level() ==
highestAnxietyLevel)
        .count();
long endTime4 = System.currentTimeMillis();
long elapsedTime4 = endTime4 - startTime4;
System.out.println("\nElapsed Time using parallel streams: " +
elapsedTime4 + " milliseconds");
System.out.println("Number of Students with the Highest Anxiety Level:
" + countHighestAnxietyP);
```

Results:

```
Goal 1: Calculate the average stress level
Elapsed Time: 68 milliseconds
Average Stress Level: 0.9963636363636363

Elapsed Time using parallel streams: 17 milliseconds
Average Stress Level: 0.9963636363636363

Goal 2: Count the number of students with the highest anxiety
Elapsed Time: 5 milliseconds
Number of Students with the Highest Anxiety Level: 61

Elapsed Time using parallel streams: 1 milliseconds
Number of Students with the Highest Anxiety Level: 61
```

Conclusion:

The time taken for parallel stream is less compared to other one.

Project URL:

https://github.com/srinidhi1404/BigDataProject

Citations:

https://www.kaggle.com/datasets/rxnach/student-stress-factors-a-comprehensive-analysis