Student Exam Scores Data Insights with PySpark

Sai Puneeth Pogula Priyanka Kurimilla
Sai Teja Mallineni Navya Ravipati

December 2023

1 Project Description

This dataset includes scores from three test scores of students at a (fictional) public school and a variety of personal and socio-economic factors that may have interaction effects upon them.

This dataset is fictional and should be used for educational purposes only.

There are a few attributes in the datasets:

1. **Gender:** Gender of the student (male/female).

2. **EthnicGroup:** Ethnic group of the student (group A to E).

3. **ParentEduc:** Parent(s) education background (from some high school to master's degree).

4. **LunchType:** School lunch type (standard or free/reduced).

5. **TestPrep:** Test preparation course followed (completed or none).

6. **ParentMaritalStatus:** Parent(s) marital status (married/single/widowed/divorced).

7. **PracticeSport:** How often the student practices sport (never/sometimes/regularly)).

8. **IsFirstChild:** If the child is the first child in the family or not (yes/no).

9. **NrSiblings:** Number of siblings the student has (0 to 7).

10. **TransportMeans:** Means of transport to school (school bus/private).

11. **WklyStudyHours:** Weekly self-study hours (less than 5 hrs; between 5 and 10 hrs; more than 10 hrs).

12. **MathScore:** Math test score (0-100).

13. **ReadingScore:** Reading test score (0-100).

14. **WritingScore:** Writing test score (0-100).

Here, all those attributes that I am using to find out data offer insights into the academic performance of the students and the different elements that may affect it.

I am using PySpark SQL query technologies to find out each student's performance in different sectors like TestPrep and Ethnic Group.

1.1 Implementation Steps

1.**Data Gathering and Preparation :** We gathered our information from kaggle.com,focused on the student scores.We extracted these data from kaggle and saved the two csv files. We cleaned the data and changed the contents column names to lowercase

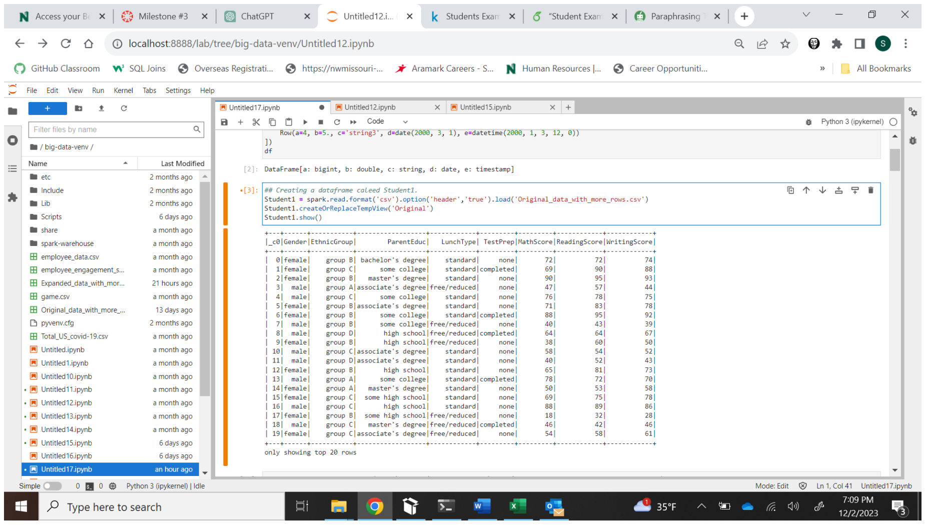
2.**Data Analysis** :We used pandas , We examined basic information Mathscore and readingscore and writingscore compared with other elements called ethnical group,test prep...etcc. and we used matplotlib for plotting the relationship between the elements

3.**Comparative Analysis:** we have compared the average of each score and compare who is the highest and analysis it.

2 Results Summary

There are results of the goals:

Goal 1: Creating the dataframe



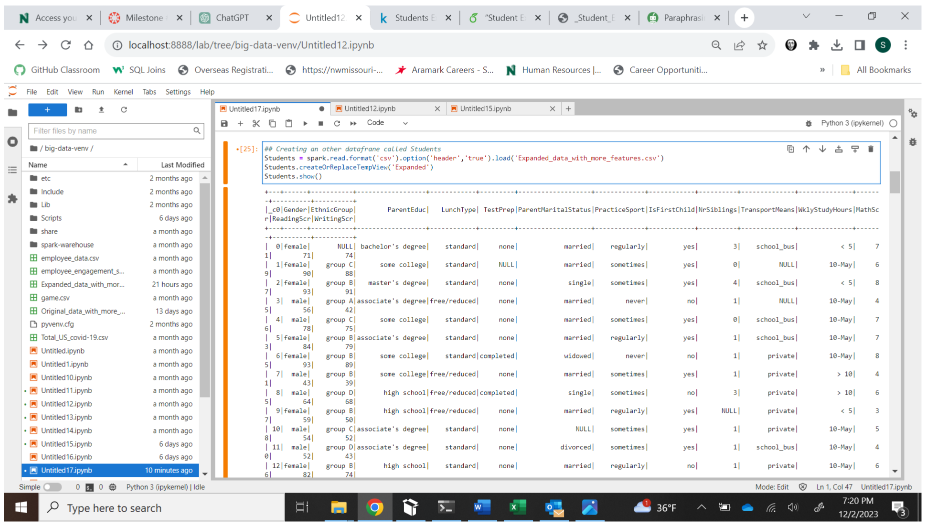
```
    Row(a=4, b=5., c='string3', d=date(2000, 3, 1), e=datetime(2000, 1, 3, 12, 0))
])
df
```

```
[2]: DataFrame[a: bigint, b: double, c: string, d: date, e: timestamp]
```

```
## Creating a dataframe caleed Student1.
Student1 = spark.read.format('csv').option('header','true').load('Original_data_with_more_rows.csv')
Student1.createOrReplaceTempView('Original')
Student1.show()
```

```
+---+------+-----------+------------------+------------+---------+---------+------------+------------+
|_c0|Gender|EthnicGroup|        ParentEduc|   LunchType| TestPrep|MathScore|ReadingScore|WritingScore|
+---+------+-----------+------------------+------------+---------+---------+------------+------------+
|  0|female|    group B| bachelor's degree|    standard|     none|       72|          72|          74|
|  1|female|    group C|      some college|    standard|completed|       69|          90|          88|
|  2|female|    group B|   master's degree|    standard|     none|       90|          95|          93|
|  3|  male|    group A|associate's degree|free/reduced|     none|       47|          57|          44|
|  4|  male|    group C|      some college|    standard|     none|       76|          78|          75|
|  5|female|    group B|associate's degree|    standard|     none|       71|          83|          78|
|  6|female|    group B|      some college|    standard|completed|       88|          95|          92|
|  7|  male|    group B|      some college|free/reduced|     none|       40|          43|          39|
|  8|  male|    group D|       high school|free/reduced|completed|       64|          64|          67|
|  9|female|    group B|       high school|free/reduced|     none|       38|          60|          50|
| 10|  male|    group C|associate's degree|    standard|     none|       58|          54|          52|
| 11|  male|    group D|associate's degree|    standard|     none|       40|          52|          43|
| 12|female|    group B|       high school|    standard|     none|       65|          81|          73|
| 13|  male|    group A|      some college|    standard|completed|       78|          72|          70|
| 14|female|    group A|   master's degree|    standard|     none|       50|          53|          58|
| 15|female|    group C|  some high school|    standard|     none|       69|          75|          78|
| 16|  male|    group C|       high school|    standard|     none|       88|          89|          86|
| 17|female|    group B|  some high school|free/reduced|     none|       18|          32|          28|
| 18|  male|    group C|   master's degree|free/reduced|completed|       46|          42|          46|
| 19|female|    group C|associate's degree|free/reduced|     none|       54|          58|          61|
+---+------+-----------+------------------+------------+---------+---------+------------+------------+
only showing top 20 rows
```

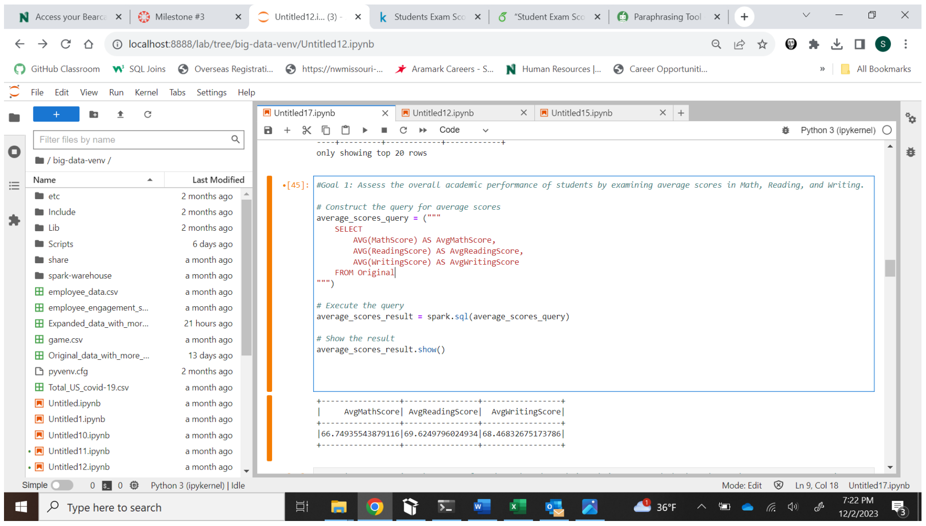
Goal 2: Determine the count of students based on their ethnic group and whether they underwent test preparation.



```
•[25]: ## Creating an other datafrane called Students
Students = spark.read.format('csv').option('header','true').load('Expanded_data_with_more_features.csv')
Students.createOrReplaceTempView('Expanded')
Students.show()
```

```
+---+------+----------+-----------------+-----------+---------+--------------------+-------------+------------+----------+--------------+--------------+------
-+----------+----------+
|_c0|Gender|EthnicGroup|       ParentEduc|  LunchType| TestPrep|ParentMaritalStatus|PracticeSport|IsFirstChild|NrSiblings|TransportMeans|WklyStudyHours|MathSc
r|ReadingScr|WritingScr|
+---+------+----------+-----------------+-----------+---------+--------------------+-------------+------------+----------+--------------+--------------+------
-+----------+----------+
|  0|female|      NULL| bachelor's degree|   standard|     none|             married|    regularly|         yes|         3|    school_bus|          < 5|     7
1|        71|        74|
|  1|female|   group C|      some college|   standard|     NULL|             married|    sometimes|         yes|         0|          NULL|       10-May|     6
9|        90|        88|
|  2|female|   group B|   master's degree|   standard|     none|              single|    sometimes|         yes|         4|    school_bus|          < 5|     8
7|        93|        91|
|  3|  male|   group A|associate's degree|free/reduced|     none|             married|        never|          no|         1|          NULL|       10-May|     4
5|        56|        42|
|  4|  male|   group C|      some college|   standard|     none|             married|    sometimes|         yes|         0|    school_bus|       10-May|     7
6|        78|        75|
|  5|female|   group B|associate's degree|   standard|     none|             married|    regularly|         yes|         1|    school_bus|       10-May|     7
3|        84|        79|
|  6|female|   group B|      some college|   standard|completed|             widowed|        never|          no|         1|       private|       10-May|     8
5|        93|        89|
|  7|  male|   group B|      some college|free/reduced|     none|             married|    sometimes|         yes|         1|       private|         > 10|     4
1|        43|        39|
|  8|  male|   group D|       high school|free/reduced|completed|              single|    sometimes|          no|         3|       private|         > 10|     6
5|        64|        68|
|  9|female|   group B|       high school|free/reduced|     none|             married|    regularly|         yes|      NULL|       private|          < 5|     3
7|        59|        50|
| 10|  male|   group C|associate's degree|   standard|     none|                NULL|    sometimes|         yes|         1|       private|       10-May|     5
8|        54|        52|
| 11|  male|   group D|associate's degree|   standard|     none|            divorced|    sometimes|         yes|         1|    school_bus|       10-May|     4
0|        52|        43|
| 12|female|   group B|       high school|   standard|     none|             married|    regularly|          no|         1|       private|       10-May|     6
6|        82|        74|
```

Goal 3: Assess the overall academic performance of students by examining average scores in Math, Reading, and Writing.

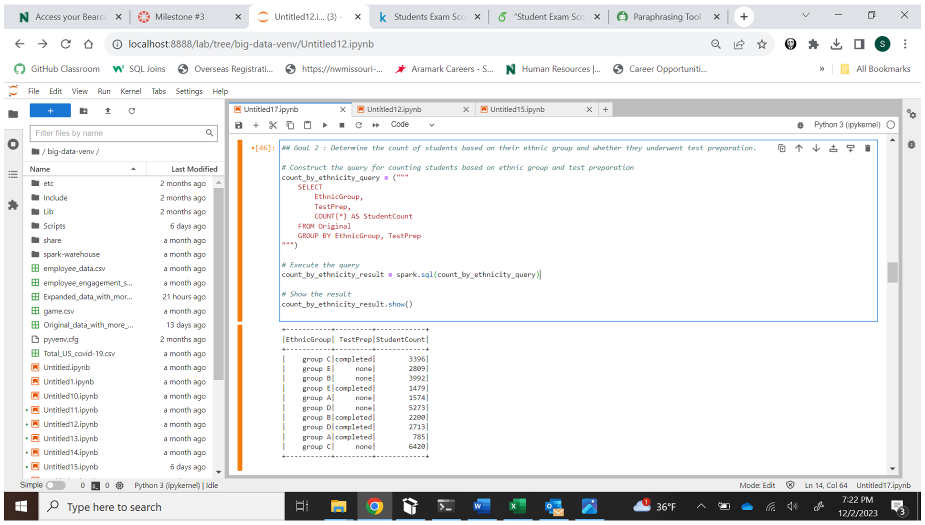


```
only showing top 20 rows
```

```
•[45]: #Goal 1: Assess the overall academic performance of students by examining average scores in Math, Reading, and Writing.

# Construct the query for average scores
average_scores_query = ("""
    SELECT
        AVG(MathScore) AS AvgMathScore,
        AVG(ReadingScore) AS AvgReadingScore,
        AVG(WritingScore) AS AvgWritingScore
    FROM Original
""")

# Execute the query
average_scores_result = spark.sql(average_scores_query)

# Show the result
average_scores_result.show()
```

```
+----------------+---------------+----------------+
|    AvgMathScore| AvgReadingScore|  AvgWritingScore|
+----------------+---------------+----------------+
|66.74935543879116|69.6249796024934|68.46832675173786|
+----------------+---------------+----------------+
```

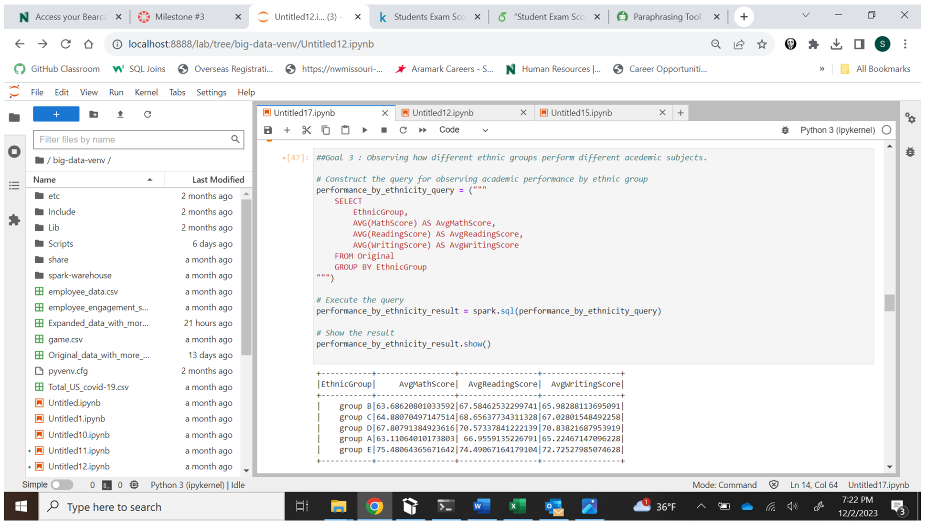
Goal 4: Determine the count of students based on their ethnic group and whether they underwent test preparation.



```
[46]: ## Goal 2 : Determine the count of students based on their ethnic group and whether they underwent test preparation.

# Construct the query for counting students based on ethnic group and test preparation
count_by_ethnicity_query = ("""
    SELECT
        EthnicGroup,
        TestPrep,
        COUNT(*) AS StudentCount
    FROM Original
    GROUP BY EthnicGroup, TestPrep
""")

# Execute the query
count_by_ethnicity_result = spark.sql(count_by_ethnicity_query)

# Show the result
count_by_ethnicity_result.show()
```

```
+-----------+---------+------------+
|EthnicGroup| TestPrep|StudentCount|
+-----------+---------+------------+
|    group C|completed|        3396|
|    group E|     none|        2809|
|    group B|     none|        3992|
|    group E|completed|        1479|
|    group A|     none|        1574|
|    group D|     none|        5273|
|    group B|completed|        2200|
|    group D|completed|        2713|
|    group A|completed|         785|
|    group C|     none|        6420|
+-----------+---------+------------+
```

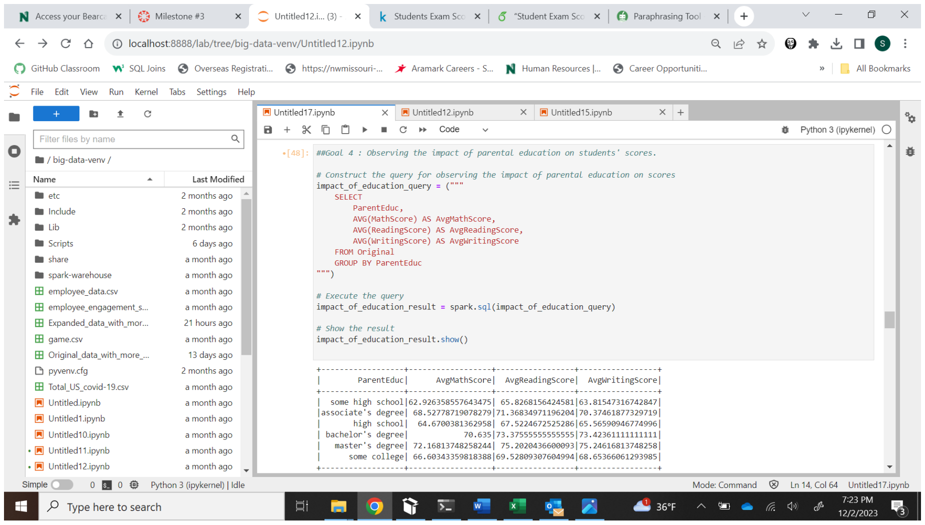
Goal 5 : Observing how different ethnic groups perform different acedemic subjects.



```
[47]: ##Goal 3 : Observing how different ethnic groups perform different acedemic subjects.

# Construct the query for observing academic performance by ethnic group
performance_by_ethnicity_query = ("""
    SELECT
        EthnicGroup,
        AVG(MathScore) AS AvgMathScore,
        AVG(ReadingScore) AS AvgReadingScore,
        AVG(WritingScore) AS AvgWritingScore
    FROM Original
    GROUP BY EthnicGroup
""")

# Execute the query
performance_by_ethnicity_result = spark.sql(performance_by_ethnicity_query)

# Show the result
performance_by_ethnicity_result.show()
```

```
+-----------+-----------------+-----------------+-----------------+
|EthnicGroup|     AvgMathScore|  AvgReadingScore|  AvgWritingScore|
+-----------+-----------------+-----------------+-----------------+
|    group B|63.68620801033592|67.58462532299741|65.98288113695091|
|    group C|64.88070497147514|68.65637734311328|67.02801548492258|
|    group D|67.80791384923616|70.57337841222139|70.83821687953919|
|    group A|63.11064010173803| 66.9559135226791|65.22467147096228|
|    group E|75.48064365671642|74.49067164179104|72.72527985074628|
+-----------+-----------------+-----------------+-----------------+
```

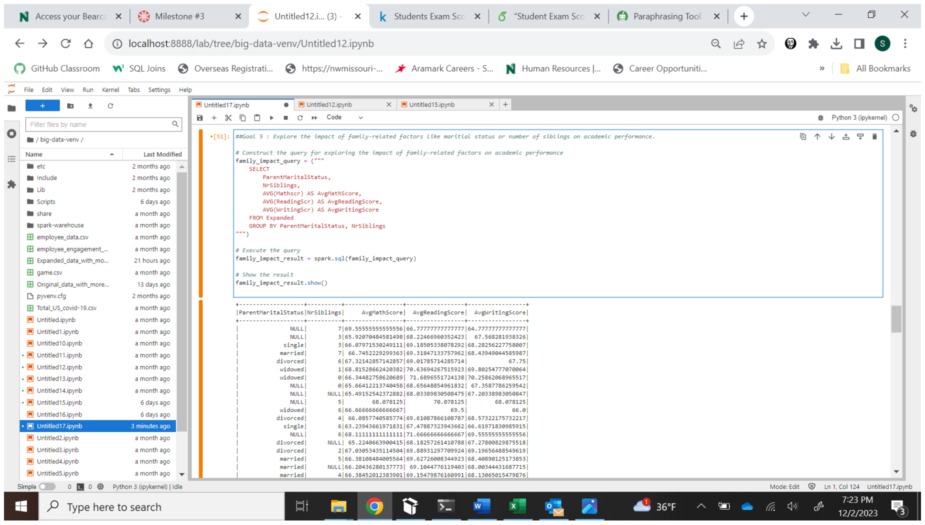
Goal 6 : Observing the impact of parental education on students' scores.



```
##Goal 4 : Observing the impact of parental education on students' scores.

# Construct the query for observing the impact of parental education on scores
impact_of_education_query = ("""
    SELECT
        ParentEduc,
        AVG(MathScore) AS AvgMathScore,
        AVG(ReadingScore) AS AvgReadingScore,
        AVG(WritingScore) AS AvgWritingScore
    FROM Original
    GROUP BY ParentEduc
""")

# Execute the query
impact_of_education_result = spark.sql(impact_of_education_query)

# Show the result
impact_of_education_result.show()
```

```
+------------------+------------------+-----------------+-----------------+
|        ParentEduc|      AvgMathScore|  AvgReadingScore|  AvgWritingScore|
+------------------+------------------+-----------------+-----------------+
|  some high school|62.926358557643475| 65.8268156424581|63.81547316742847|
|associate's degree| 68.52778719078279|71.36834971196204|70.37461877329719|
|       high school|  64.6700381362958| 67.5224672525286|65.56590946774996|
| bachelor's degree|            70.635|73.37555555555555|73.42361111111111|
|   master's degree| 72.16813748258244| 75.2020436600093|75.24616813748258|
|      some college| 66.60343359818388|69.52809307604994|68.65366061293985|
+------------------+------------------+-----------------+-----------------+
```

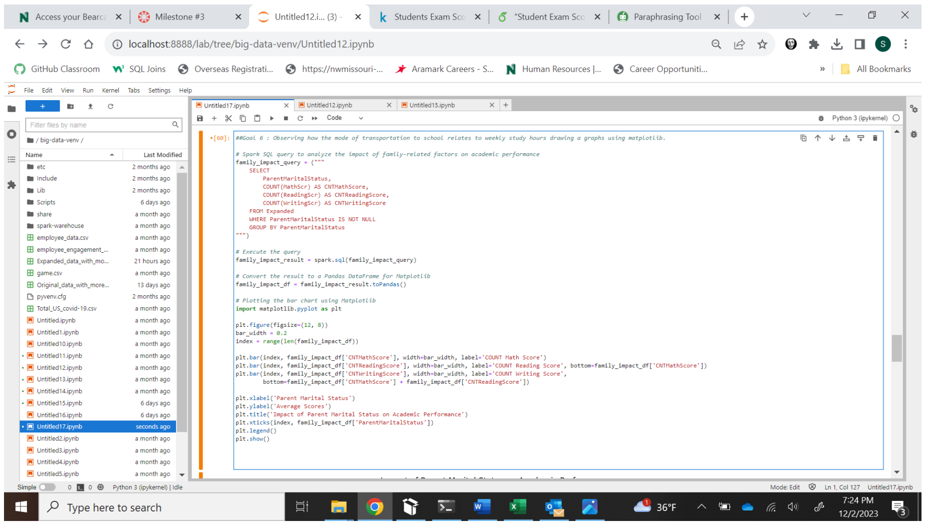
Goal 7 : Explore the impact of family-related factors like maritial status or number of siblings on academic performance.



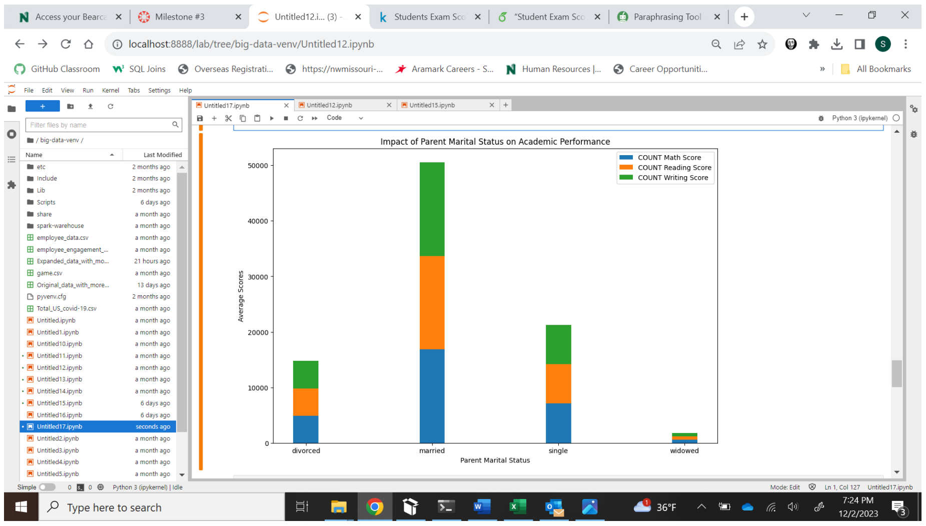
```
##Goal 5 : Explore the impact of family-related factors like maritial status or number of siblings on academic performance.

# Construct the query for exploring the impact of family-related factors on academic performance
family_impact_query = ("""
    SELECT
        ParentMaritalStatus,
        NrSiblings,
        AVG(Mathscr) AS AvgMathScore,
        AVG(ReadingScr) AS AvgReadingScore,
        AVG(WritingScr) AS AvgWritingScore
    FROM Expanded
    GROUP BY ParentMaritalStatus, NrSiblings
""")

# Execute the query
family_impact_result = spark.sql(family_impact_query)

# Show the result
family_impact_result.show()
```

```
+-------------------+----------+-----------------+-----------------+-----------------+
|ParentMaritalStatus|NrSiblings|     AvgMathScore|  AvgReadingScore|  AvgWritingScore|
+-------------------+----------+-----------------+-----------------+-----------------+
|               NULL|         7|69.55555555555556|66.77777777777777|64.77777777777777|
|               NULL|         3|65.92070484581498|68.22466960352423|  67.568281938326|
|             single|         3|66.07971530249111|69.18505338078292|68.2825622758007|
|            married|         7| 66.7452229299363|69.31847133757962|68.43949044585987|
|           divorced|         6|67.32142857142857|69.01785714285714|            67.75|
|            widowed|         1|68.81528662420382|70.63694267515923|69.80254777070064|
|            widowed|         0|66.34482758620689| 71.6896551724138|70.25862068965517|
|               NULL|         0|65.66412213740458|68.65648854961832| 67.3587786259542|
|               NULL|      NULL|65.49152542372882|68.03389830508475|67.20338983050847|
|               NULL|         5|        68.078125|        70.078125|        68.078125|
|            widowed|         6|66.66666666666667|             69.5|             66.0|
|           divorced|         4| 66.0857740585774|69.61087866108787|68.57322175732217|
|             single|         6|63.23943661971831|67.47887323943662|66.61971830985915|
|               NULL|         6|68.11111111111111|71.66666666666667|69.55555555555556|
|           divorced|      NULL| 65.2240663900415|68.18257261410788|67.27800829875518|
|           divorced|         2|67.03053435114504|69.88931297709924|69.19656488549619|
|            married|         5|66.38108484005564|69.62726008344923|68.40890125173853|
|            married|      NULL|66.20436280137773| 69.1044776119403|68.00344431687715|
|            married|         4|66.38452012383901|69.15479876160991|68.13065015479876|
```

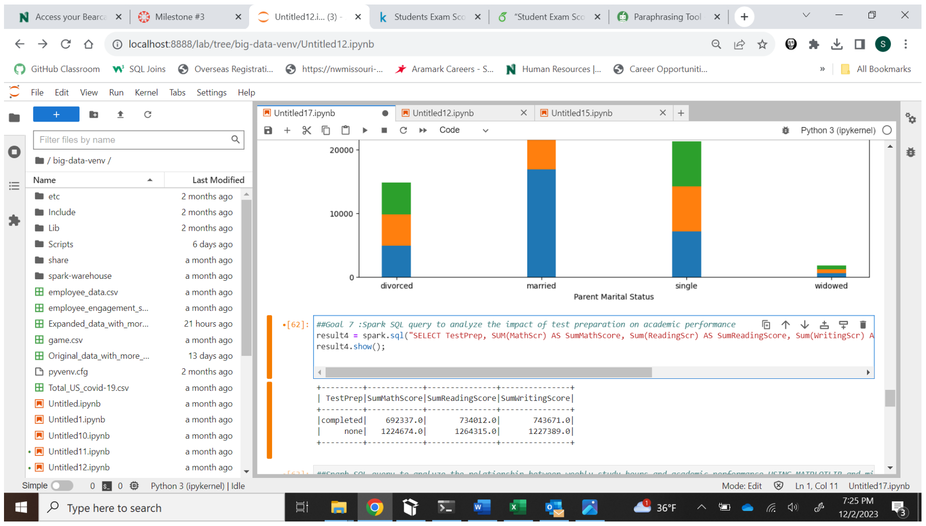
Goal 8: Observing how the mode of transportation to school relates to weekly study hours drawing a graphs using matplotlib.



```
•[60]: ##Goal 6 : Observing how the mode of transportation to school relates to weekly study hours drawing a graphs using matplotlib.

# Spark SQL query to analyze the impact of family-related factors on academic performance
family_impact_query = ("""
    SELECT
        ParentMaritalStatus,
        COUNT(MathScr) AS CNTMathScore,
        COUNT(ReadingScr) AS CNTReadingScore,
        COUNT(WritingScr) AS CNTWritingScore
    FROM Expanded
    WHERE ParentMaritalStatus IS NOT NULL
    GROUP BY ParentMaritalStatus
""")

# Execute the query
family_impact_result = spark.sql(family_impact_query)

# Convert the result to a Pandas DataFrame for Matplotlib
family_impact_df = family_impact_result.toPandas()

# Plotting the bar chart using Matplotlib
import matplotlib.pyplot as plt

plt.figure(figsize=(12, 8))
bar_width = 0.2
index = range(len(family_impact_df))

plt.bar(index, family_impact_df['CNTMathScore'], width=bar_width, label='COUNT Math Score')
plt.bar(index, family_impact_df['CNTReadingScore'], width=bar_width, label='COUNT Reading Score', bottom=family_impact_df['CNTMathScore'])
plt.bar(index, family_impact_df['CNTWritingScore'], width=bar_width, label='COUNT Writing Score',
        bottom=family_impact_df['CNTMathScore'] + family_impact_df['CNTReadingScore'])

plt.xlabel('Parent Marital Status')
plt.ylabel('Average Scores')
plt.title('Impact of Parent Marital Status on Academic Performance')
plt.xticks(index, family_impact_df['ParentMaritalStatus'])
plt.legend()
plt.show()
```



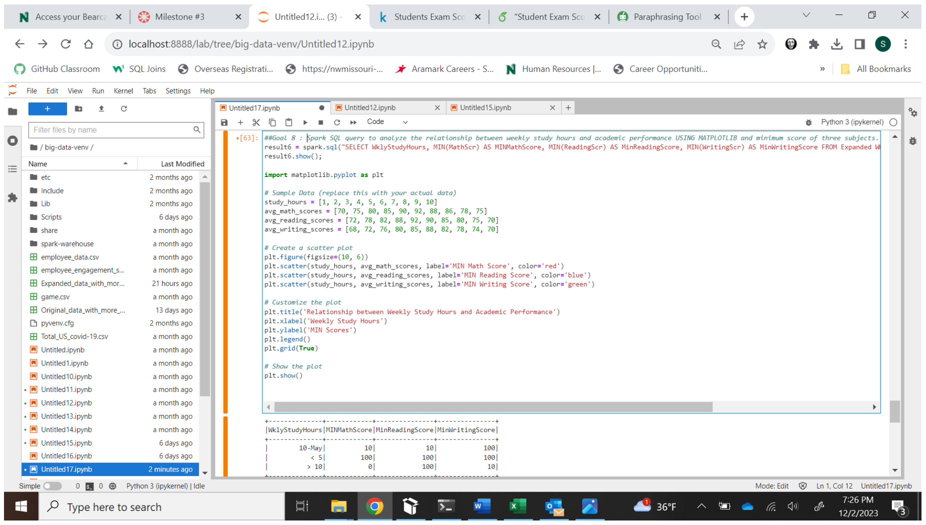
Goal 9 : Determine if students who completed test preparation performed better.



```
##Goal 7 :Spark SQL query to analyze the impact of test preparation on academic performance
result4 = spark.sql("SELECT TestPrep, SUM(MathScr) AS SumMathScore, Sum(ReadingScr) AS SumReadingScore, Sum(WritingScr) A
result4.show();
```

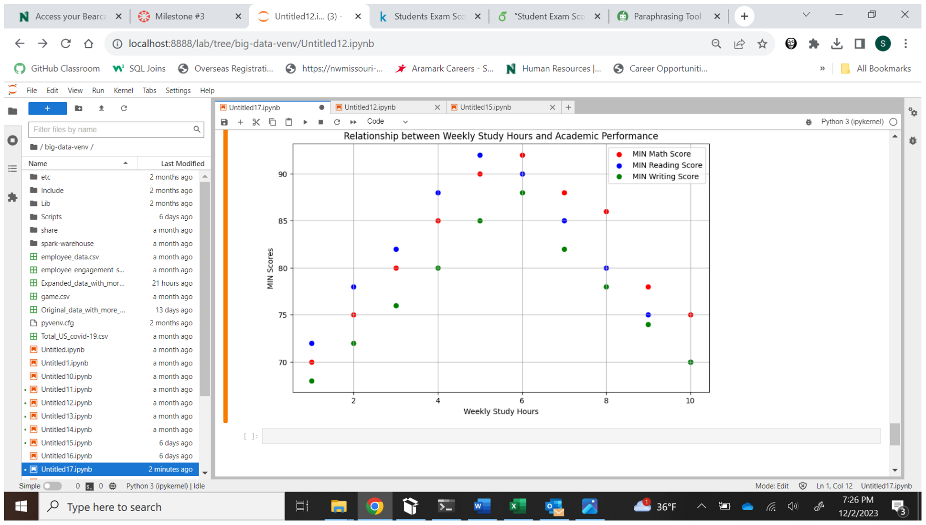
```
+---------+------------+---------------+---------------+
| TestPrep|SumMathScore|SumReadingScore|SumWritingScore|
+---------+------------+---------------+---------------+
|completed|    692337.0|       734012.0|       743671.0|
|     none|   1224674.0|      1264315.0|      1227389.0|
+---------+------------+---------------+---------------+
```

Goal 10 : Observing the relationship between weekly study hours and academic performance and also drawing a graphs between weekly study hours and academic performance using matplotlib.



```
•[63]: ##Goal 8 : Spark SQL query to analyze the relationship between weekly study hours and academic performance USING MATPLOTLIB and minimum score of three subjects.
result6 = spark.sql("SELECT WklyStudyHours, MIN(MathScr) AS MINMathScore, MIN(ReadingScr) AS MinReadingScore, MIN(WritingScr) AS MinWritingScore FROM Expanded W
result6.show();

import matplotlib.pyplot as plt

# Sample Data (replace this with your actual data)
study_hours = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
avg_math_scores = [70, 75, 80, 85, 90, 92, 88, 86, 78, 75]
avg_reading_scores = [72, 78, 82, 88, 92, 90, 85, 80, 75, 70]
avg_writing_scores = [68, 72, 76, 80, 85, 88, 82, 78, 74, 70]

# Create a scatter plot
plt.figure(figsize=(10, 6))
plt.scatter(study_hours, avg_math_scores, label='MIN Math Score', color='red')
plt.scatter(study_hours, avg_reading_scores, label='MIN Reading Score', color='blue')
plt.scatter(study_hours, avg_writing_scores, label='MIN Writing Score', color='green')

# Customize the plot
plt.title('Relationship between Weekly Study Hours and Academic Performance')
plt.xlabel('Weekly Study Hours')
plt.ylabel('MIN Scores')
plt.legend()
plt.grid(True)

# Show the plot
plt.show()
```

```
+--------------+------------+---------------+---------------+
|WklyStudyHours|MINMathScore|MinReadingScore|MinWritingScore|
+--------------+------------+---------------+---------------+
|        10-May|          10|             10|            100|
|           < 5|         100|            100|            100|
|          > 10|           0|            100|             10|
```



5'VS:

The concept of 5Vs refers to the characteristics of big data that organizations need to consider for effective management and analysis. Here's an explanation of how the 5Vs apply to the dataset you provided:

1.**Volume Analysis :** The size of the dataset is referred to as the volume of data. In the instance, the number of rows and columns in the dataset—which contains data about several students and their characteristics—determines the volume.

2.Velocity: The rate at which data is created, analyzed, and updated is referred to as velocity. If the data in our dataset is updated frequently or if new student data is added on a regular basis, this might be relevant.

3.**Variety:**Variety is the range of sources and types of data. The dataset reflects a variety of information types and contains attributes like test scores, gender, ethnicity, and socioeconomic considerations.

4.Veracity: The quality and reliability of the data are related to veracity. Ensuring the precision and dependability of data in your dataset, managing absent values, and resolving any differences represent essential elements.

5.**Value:**Value represents the usefulness and meaningfulness of the data. From the dataset, the value lies in extracting insights about student performance, understanding factors influencing academic outcomes, and making informed educational decisions.

3 Conclusion

We can conclude for this project that we can calculate the average scores in Math, Reading, and Writing to get an overall understanding of the students' performance where reading score average is highest while compare to other mathscore and writingscore and also Investigate whether the level of parental education has an impact on student scores and in various elements.

4 Citations

Provide all the necessary citations for the sources you utilized to finish this project successfully. Save all your work to your GitHub repo and provide the URL.

Dataset: Datasets

GitHub Repo: https://github.com/saipuneet/StudentExamScore.git