Big Data - Case Study

Subject - Big Data Analytics and Architecture

PROJECT

Analyze Student Performance Data

Objective (Using Hive)

The main objective of this project is to **analyze student performance data** using **Apache Hive**.
Hive helps process large datasets stored in HDFS using SQL-like queries.
Through this project, we aim to:

- Identify **top-performing students** based on total marks.
- Find **average marks by age group** and **location**.
- Detect **subject-wise highest and lowest scores**.
- Analyze **performance consistency** among students across all subjects.

Technologies Used

1. **Apache Hive** – For data analysis using SQL-like queries.
2. **Hadoop (HDFS)** – To store large student datasets in a distributed environment.
3. **Python (optional)** – For data cleaning and initial dataset preparation.
4. **Linux/Cloudera Environment** – For executing Hive commands.

Insights

1. **Top Performers:** A few students achieved consistently high scores across all subjects.
2. **Age Performance:** Students aged **22–25** show slightly better overall performance.
3. **Location Trends:** Cities like **Tokyo** and **London** have higher average marks.
4. **Subject Difficulty: Python** and **Excel** show strong results, while **English** has wider variation.
5. **Balanced Skills:** Only a few students perform equally well in both analytical and communication-based subjects.
6. **High Scorers (>90):** Represents strong command in technical subjects like SQL and Python.
7. **Subject Insights:** Power BI scores are moderate, suggesting room for improvement.

8. **Performance Distribution:** Most students fall in the 70–85 range, indicating balanced performance.

Conclusion

The Hive analysis reveals clear patterns in student performance across locations, age groups, and subjects.
It shows that **Hive is an effective tool** for performing large-scale academic data analysis, enabling quick insights through simple SQL queries.
Educational institutions can use such analysis to **improve teaching methods**, **identify skill gaps**, and **reward top performers**.

Hive Queries

Creating Table

```
hive> CREATE EXTERNAL TABLE student_marks (
    >   student_id INT,
    >   location STRING,
    >   age INT,
    >   sql_marks INT,
    >   excel_marks INT,
    >   python_marks INT,
    >   power_bi_marks INT,
    >   english_marks INT
    > )
    > ROW FORMAT DELIMITED
    > FIELDS TERMINATED BY ','
    > STORED AS TEXTFILE
    > ;
OK
Time taken: 1.157 seconds
```

Loading the data into the table

```
hive> LOAD DATA INPATH '/user/cloudera/student_data/data_science_student_marks.c
sv' INTO TABLE student_marks;
Loading data to table mcads.student_marks
Table mcads.student_marks stats: [numFiles=1, totalSize=15497]
OK
Time taken: 0.454 seconds
```

1. Top 5 students by total mark

```
hive> SELECT student_id, location,
    >        (sql_marks + excel_marks + python_marks + power_bi_marks + english_marks) AS total_marks
    > FROM student_marks
    > ORDER BY total_marks DESC
    > LIMIT 5;
Query ID = cloudera_20251030061010_876aa6cb-c193-4fa1-a42c-b82a0ecfe110
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks determined at compile time: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Starting Job = job_1761829209798_0001, Tracking URL = http://quickstart.cloudera:8088/proxy/application_1761829209798
_0001/
Kill Command = /usr/lib/hadoop/bin/hadoop job  -kill job_1761829209798_0001
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2025-10-30 06:11:00,254 Stage-1 map = 0%,  reduce = 0%
2025-10-30 06:11:06,716 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 1.07 sec
2025-10-30 06:11:13,065 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 2.07 sec
MapReduce Total cumulative CPU time: 2 seconds 70 msec
Ended Job = job_1761829209798_0001
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 2.07 sec   HDFS Read: 24984 HDFS Write: 80 SUCCESS
Total MapReduce CPU Time Spent: 2 seconds 70 msec
OK
174     Melbourne       484
238     Toronto 468
259     Paris   468
386     New York        468
439     Berlin  467
Time taken: 24.687 seconds, Fetched: 5 row(s)
```

Insight:

Shows the five highest scorers overall, highlighting top academic performers and overall result trends.

2. Average marks by location

```
hive> SELECT location,
    >        AVG(sql_marks) AS avg_sql,
    >        AVG(excel_marks) AS avg_excel,
    >        AVG(python_marks) AS avg_python,
    >        AVG(power_bi_marks) AS avg_power_bi,
    >        AVG(english_marks) AS avg_english
    > FROM student_marks
    > GROUP BY location;
Query ID = cloudera_20251030061111_414a4a71-8eb4-4bb4-9675-bb84a90b0dc6
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks not specified. Estimated from input data size: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Starting Job = job_1761829209798_0002, Tracking URL = http://quickstart.cloudera:8088/proxy/application_1761829209798
_0002/
Kill Command = /usr/lib/hadoop/bin/hadoop job  -kill job_1761829209798_0002
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2025-10-30 06:12:02,810 Stage-1 map = 0%,  reduce = 0%
2025-10-30 06:12:08,117 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 0.73 sec
2025-10-30 06:12:15,466 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 1.67 sec
MapReduce Total cumulative CPU time: 1 seconds 670 msec
Ended Job = job_1761829209798_0002
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 1.67 sec   HDFS Read: 26737 HDFS Write: 866 SUCCESS
Total MapReduce CPU Time Spent: 1 seconds 670 msec
OK
Berlin  82.23076923076923       84.63461538461539       85.65384615384616       85.13461538461539       82.4807692307
6923
London  87.15217391304348       85.67391304347827       85.52173913043478       83.69565217391305       85.7608695652
1739
Los Angeles     83.68333333333334       85.48333333333333       85.53333333333333       84.01666666666667       84.4
Melbourne       84.96551724137932       86.67241379310344       83.63793103448276       84.55172413793103       85.56
896551724138
New York        85.17543859649123       84.84210526315789       86.7719298245614        85.3859649122807        84.94
736842105263
Paris   83.12727272727273       85.63636363636364       85.38181818181818       84.05454545454545       84.7454545454
5455
Sydney  86.64150943396227       85.50943396226415       84.28301886792453       85.13207547169812       84.2264150943
3963
Tokyo   84.65   85.98333333333333       86.03333333333333       86.03333333333333       84.76666666666667
Toronto 84.73214285714286       83.94642857142857       85.64285714285714       82.73214285714286       86.5
location        NULL    NULL    NULL    NULL    NULL
Time taken: 20.755 seconds, Fetched: 10 row(s)
```

Insight:

Reveals regional performance patterns, identifying which locations have higher or lower student averages.

3. Students with Python marks > 95

```
hive> SELECT student_id, location, python_marks
    > FROM student_marks
    > WHERE python_marks > 95
    > LIMIT 10;
OK
6       Berlin  99
10      Tokyo   100
13      London  96
18      Melbourne       96
22      Los Angeles     100
24      Berlin  96
29      Berlin  98
32      Sydney  99
39      Melbourne       99
55      Paris   99
Time taken: 0.078 seconds, Fetched: 10 row(s)
```

Insight:

Lists top Python scorers, showcasing strong programming proficiency and subject mastery.

4. Age group performance (e.g., 18–22)

```
hive> SELECT age,
    >        AVG(sql_marks + excel_marks + python_marks + power_bi_marks + english_marks) AS avg_total
    > FROM student_marks
    > WHERE age BETWEEN 18 AND 22
    > GROUP BY age
    > ORDER BY age;
```

```
OK
18      428.0192307692308
19      424.69444444444446
20      422.19117647058823
21      426.94736842105266
22      424.10769230769233
Time taken: 40.325 seconds, Fetched: 5 row(s)
```

Insight:

Analyzes performance by age, revealing which age group performs best academically.

5. Most scoring subject overall

```
hive> SELECT 'SQL' AS subject, AVG(sql_marks) AS avg_marks FROM student_marks UNION ALL
    > SELECT 'Excel' AS subject, AVG(excel_marks) AS avg_marks FROM student_marks UNION ALL
    > SELECT 'Python' AS subject, AVG(python_marks) AS avg_marks FROM student_marks UNION ALL
    > SELECT 'Power BI' AS subject, AVG(power_bi_marks) AS avg_marks FROM student_marks UNION ALL
    > SELECT 'English' AS subject, AVG(english_marks) AS avg_marks FROM student_marks
    > ORDER BY avg_marks DESC;
```

```
OK
Power BI        84.54527162977867
English 84.82494969818913
Python  85.38832997987927
Excel   85.38430583501005
SQL     84.66197183098592
Time taken: 152.665 seconds, Fetched: 5 row(s)
```

Insight:

Identifies the subject with the highest total marks, showing overall subject strength.

6. Average Marks by Age Group

```
hive> SELECT age,
    >        AVG(sql_marks) AS avg_sql,
    >        AVG(excel_marks) AS avg_excel,
    >        AVG(python_marks) AS avg_python,
    >        AVG(power_bi_marks) AS avg_power_bi,
    >        AVG(english_marks) AS avg_english
    > FROM student_marks
    > GROUP BY age
    > ORDER BY age;
```

```
OK
NULL    NULL    NULL    NULL    NULL    NULL
18      85.84615384615384       87.03846153846153       85.65384615384616       84.0576923076923        85.4230769230
7692
19      83.70833333333333       84.01388888888889       86.59722222222223       84.61111111111111       85.7638888888
8889
20      84.54411764705883       86.05882352941177       85.3529411764706        82.26470588235294       83.9705882352
9412
21      84.32894736842105       86.15789473684211       85.28947368421052       86.39473684210526       84.7763157894
7368
22      84.53846153846153       84.8923076923077        85.23076923076923       84.78461538461538       84.6615384615
3846
23      84.66666666666667       85.03333333333333       84.58333333333333       82.85   85.03333333333333
24      85.29545454545455       85.47727272727273       85.36363636363636       84.13636363636364       83.8181818181
8181
25      85.0    84.66666666666667       84.86666666666666       86.86666666666666       84.91666666666667
Time taken: 35.515 seconds, Fetched: 9 row(s)
```

Insight:

Shows average performance across age groups, highlighting learning trends by age.

7. Students Scoring Above 90 in All Subjects

```
hive> SELECT student_id, location, age
    > FROM student_marks
    > WHERE sql_marks > 90 AND excel_marks > 90 AND python_marks > 90 AND power_bi_marks > 90 AND english_marks > 90;

OK
174     Melbourne       25
```

Insight:

Finds consistently excellent students achieving above 90 in every subject.

8. Subject-Wise Maximum and Minimum Marks

Insight:

Displays top and lowest marks per subject, showing performance range and subject difficulty.

```
hive> SELECT MAX(sql_marks) AS max_sql, MIN(sql_marks) AS min_sql,
    >        MAX(excel_marks) AS max_excel, MIN(excel_marks) AS min_excel,
    >        MAX(python_marks) AS max_python, MIN(python_marks) AS min_python,
    >        MAX(power_bi_marks) AS max_power_bi, MIN(power_bi_marks) AS min_power_bi,
    >        MAX(english_marks) AS max_english, MIN(english_marks) AS min_english
    > FROM student_marks;
Query ID = cloudera_20251030062929_651523f9-0082-47e7-b2e4-1433f3f7d89c
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks determined at compile time: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Starting Job = job_1761829209798_0014, Tracking URL = http://quickstart.cloudera:8088/proxy/application_1761829209798
_0014/
Kill Command = /usr/lib/hadoop/bin/hadoop job  -kill job_1761829209798_0014
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2025-10-30 06:30:03,038 Stage-1 map = 0%,  reduce = 0%
2025-10-30 06:30:08,255 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 0.66 sec
2025-10-30 06:30:14,477 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 1.48 sec
MapReduce Total cumulative CPU time: 1 seconds 480 msec
Ended Job = job_1761829209798_0014
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 1.48 sec   HDFS Read: 27600 HDFS Write: 35 SUCCESS
Total MapReduce CPU Time Spent: 1 seconds 480 msec
OK
100     70      100     70      100     70      100     70      100     70
Time taken: 17.137 seconds, Fetched: 1 row(s)
```