ANALYSIS AND COMPARATIVE STUDY OF CHESS DATA INSIGHTS

GROUP NUMBER: 07

Member Details:

1. Name: Omkar Khaladkar (Team Leader)
   Student ID: 23262297
   Email: omkar.khaladkar2@mail.dcu.ie

2. Name: Shreya Ketkar
   Student ID: 23262829
   Email: shreya.ketkar2@mail.dcu.ie

3. Name: Stanley Johnson
   Student ID: 23260879
   Email: stanley.johnson7@mail.dcu.ie

4. Name: Vaishnavi Kulkarni
   Student ID: 23266282
   Email: vaishnavi.kulkarni2@mail.dcu.ie

Git Repository: https://gitlab.computing.dcu.ie/kulkarv2/group-7_cloud-assignment-02_chess-data-analytics-application

Dataset Link: https://www.kaggle.com/datasets/robikscube/this-week-in-chess-archive/data?select=twic_master.csv

Tableau Dashboard Link:
https://public.tableau.com/views/ChessDataAnalysis/ChessDataAnalysis?:language=en-GB&publish=yes&:display_count=n&:origin=viz_share_link

Video Link: https://drive.google.com/file/d/1vb-__-GIEEEvh0rOtlYU9aQRB_LqvtJe/view?usp=sharing

INTRODUCTION

Chess, an ancient game of strategy and skill, has transcended centuries, evolving into a deeply analytical and competitive sport. Originating over a millennium ago, this two-player game simulates warfare on a squared board, each participant controlling sixteen pieces with the goal of checkmating their opponent's king. Chess is not merely a test of tactical prowess but also a reflection of strategic depth, psychological endurance, and creative problem-solving.

Chess is a classic game of strategy that has coherently integrated with the technological age through data analytics. This fusion transform historical game records and player tactics into valuable insights, leveraging modern computing and statistical analysis. The intersection of chess with data science not only revolutionizes game strategy and training methods but also exemplifies the impactful blend of traditional sports with contemporary technology, offering a new perspective on this age-old game.

MOTIVATION

Developing a chess data analytics application that delves into strategic nuances presents an exciting opportunity for a multitude of reasons. It stands as a valuable strategic tool for players, enhancing their understanding and implementation of various openings and tactics, thus elevating their game. Such an application also serves as a platform, offering insights into chess's rich history and the evolution of its metagame.

Chess openings are the basics of a player's approach. By exploring various openings and their following win rates for both black and white pieces, we attempt to provide players with insights that could fundamentally improve their strategic approach. This understanding is not just about winning or losing; it is about understanding the deeper planned importance of each opening move.

Our project provides an extensive view that starts right from the first move which is the opening of the game that are sometimes less explored through data analysis. Our analysis offers a balanced perspective by presenting win rate data for both black and white pieces across different chess openings. This inclusive methodology ensures that players, irrespective of the color they play, have access to valuable strategic insights. By doing so, we aim to democratize the availability of high-level chess strategies, effectively equalizing opportunities for all players to enhance their game.

In the strategic landscape of chess, where each decision can alter the game's trajectory, the role of data analytics emerges as pivotal. Our project harnesses extensive datasets to analyze and reveal the success rates of various chess openings. This data-centric approach transcends traditional intuition-based strategies, providing a solid, evidence-

based foundation for players. Such insights empower them to make choices grounded in data, elevating their strategic planning from the very outset of the game.

TECHNOLOGY

For this data analytics application we have used the following technologies:

1. Google Cloud Platform (GCP)
2. Pyspark: For data-cleaning
3. Hive: For data-loading and analyzing
4. Tableau: For front end and visualization
5. Hadoop: For storing and data-processing
6. Apache Map reduce

RELATED WORK

1. **Chess.com Analytics:** It was launched in 2007, and is also one of the largest online chess platforms, it provides various analytical tools for all the players to review their previous play and learn from them as well. It also offers post-game analysis of a particular game and gives suggestions on how to improve on various factors like opening exploration, and tactical training all these will be based on the player's data. It also provides an analysis of every move played during the game and give suggestions on what could have been a particular move at that stage.

2. **AIM chess:** It is a creative platform that collects data from a user's online chess games to offer custom-made statistical insights. It inspects a player's game history and identifies various patterns, some common errors from the previous game, and various behavioral tendencies. This analysis is further used to make customized training recommendations and various strategic advice that aims to enhance areas of the player's chess skills. This finding helps players to improve the overall game by focusing on their unique playing style and past performance

3. **ARENA:** Arena is a free web-based application for chess, compatible with both windows and Linux. It provides game analysis which involves testing on various chess engines. It consists of about 250 chess engines of unique strengths. The web application gives an in-depth analysis of each chess engine process and has different tournament characteristics for engine competitions. It also contains big databases of the most popular games and tournaments played and it also provides online play, thus making it a thorough tool for beginners and masters.

OUR SOLUTION

The approach of our Chess Data Analytics project, focusing on the analysis of chess openings and their impact on game outcomes, offers a distinct perspective compared to tools like Chess.com Analytics, AIM chess, and Arena. While these platforms are geared towards providing a broad suite of services for playing, learning, and analyzing chess, encompassing a range of engines and databases, our project hones in on the strategic implications of various openings. It delves into the specifics of how different openings affect win, loss, and draw probabilities, furnishing players with targeted insights for selecting optimal strategies. This specialized focus on the statistical aspects of game openings and their effectiveness sets it apart from the more generalist approach of the aforementioned tools, which cater to a wider array of chess-related needs including gameplay and comprehensive training.

DATASET - DESCRIPTION OF THE DATASET

a. Source Of the Data:

This dataset is sourced from Kaggle, a renowned platform for data science and machine learning. This collection encompasses an exhaustive archive of two million premier chess games, featuring a structured tabular representation of all the games. The dataset is in CSV format and spans a decade, covering chess games from the years 2012 to 2022. This data is big data with a file size of 2GB.

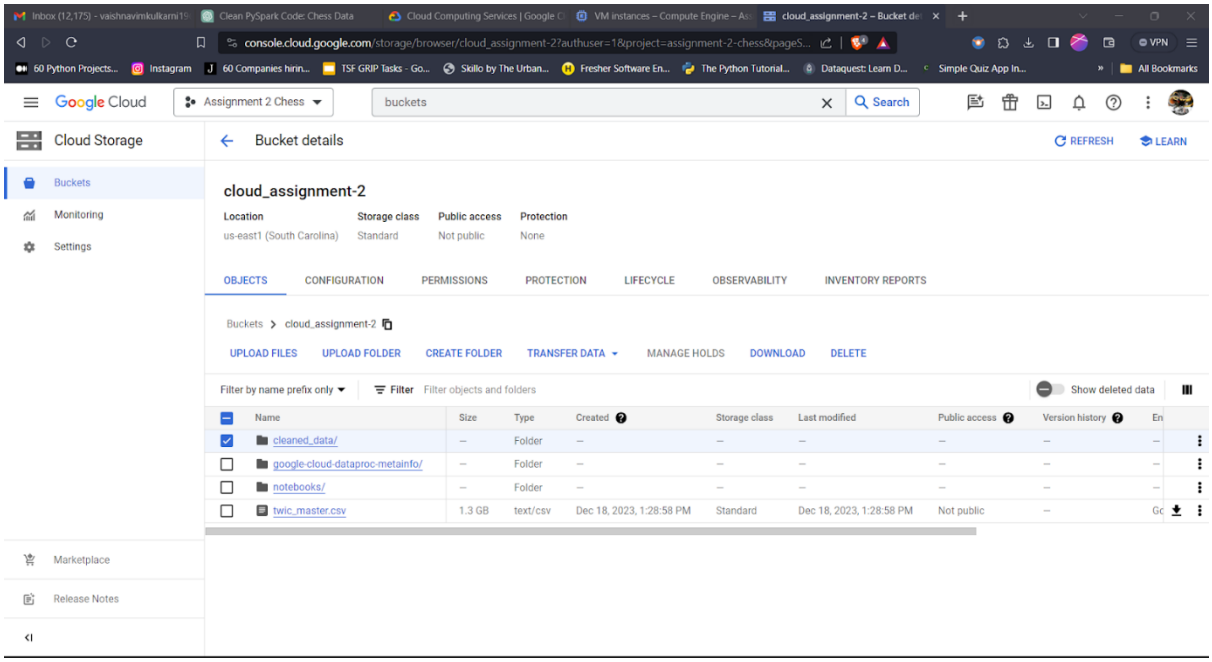
This dataset contained around 10,48,576 rows the dataset is structured to provide detailed information on each game, including player names for both white and black, game and event dates, event names, game results, and move sequences. Additionally, it encompasses a variety of other game-related details such as FIDE IDs, Elo ratings, opening names, event types, and specific chess960 setups.

Steps taken to acquire data: This dataset is publicly available on Kaggle, we downloaded the dataset from Kaggle and a zip folder that contained the file named “twic_master.csv”. Also, after carefully looking at the dataset and pointing out the findings and errors further steps were carried out.

b. Process of extraction/collection:

The CSV file of the dataset was downloaded and then we uploaded the dataset to the cloud storage i.e. bucket of the Google Cloud platform.

The dataset was uploaded to the bucket named **cloud_assignment-2.**



To execute this complete application, we need to create a cluster in the Google Cloud Platform (GCP) which will allow us to run various technologies like Hive, Hadoop, and Pyspark. The steps to create the cluster in GCP are as follows:

B1 - Setting Up a Hadoop Cluster with Google DataProc:

- We accessed the Google Cloud Console and navigated to the Dataproc section.
- Here, we initiated the creation of a new cluster, defining its name and selecting the appropriate Region/Zone.
- We chose a cluster configuration of a Single Master with 2 worker nodes.
- For the Image Type, we selected version 2.1, which includes Debian 11, Hadoop 3.3, and Spark 3.3, released on December 12, 2022.

B2 - Installing Key Hadoop Ecosystem Components:

After the cluster was set up, we navigated to the VM Instance and connected to the master node using SSH. We then proceeded with the installation of MapReduce, Pyspark, and Hive. We confirmed that MapReduce was already part of the standard Hadoop package. The initialization of Pyspark and Hive was carried out through basic command-line operations.

B3 – Ensuring Proper Configuration and Functionality:

- Our next step was to validate the correct functioning of both Pig and Hive within the Hadoop setup.
- Additionally, we checked the operation of all necessary bash commands to ensure everything was running smoothly.

- This describes the steps taken by our team to create and configure a Hadoop cluster using Google DataProc, including the installation and verification of key components like MapReduce, Pyspark, and Hive.

c. Data pre-processing – preparation and cleaning:

Data Cleaning using Pyspark:

We executed the pyspark command in the SSH terminal of GCP and after this pyspark terminal opened. Once we got inside the pyspark terminal the queries were run to achieve the cleaned dataset.

We started with importing the SparkSesssion module along with other necessary modules in pyspark. Once the module was imported then we initialized the spark session using the function below:

```
spark = SparkSession.builder \
    .appName("DatasetCleaningGCS") \
    .config("spark.jars.packages", "com.google.cloud.bigdataoss:gcs-connector:hadoop3-2.2.0") \
    .getOrCreate()
```

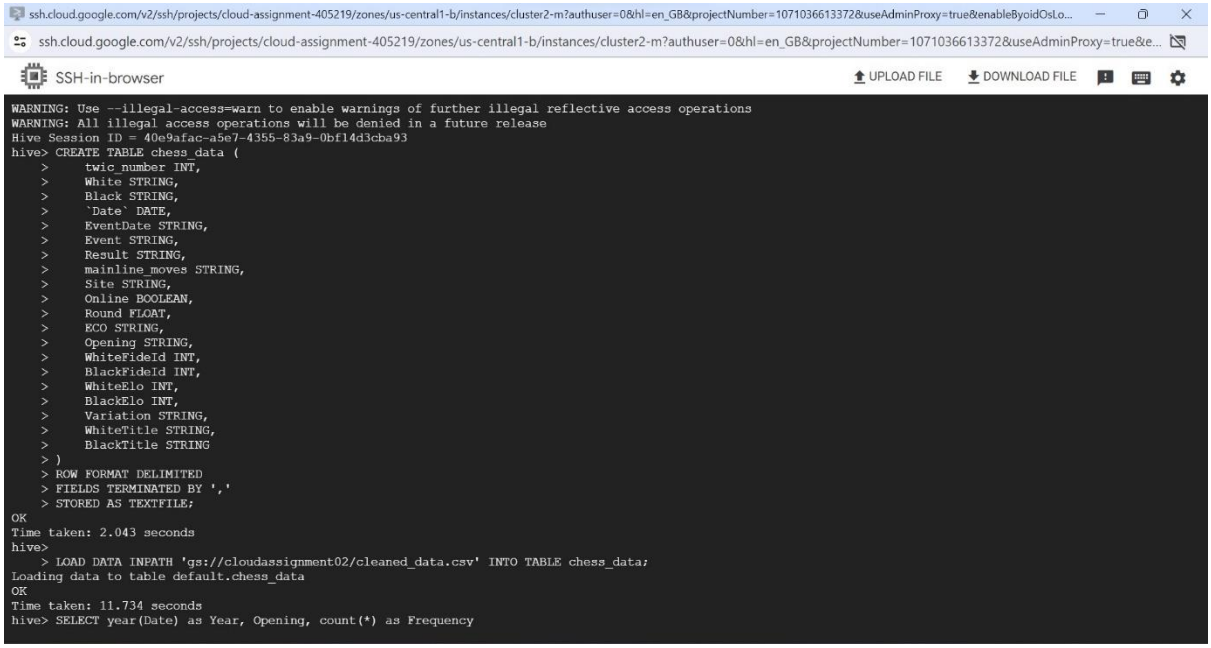
Next, we loaded the dataset from a specified file path at first, setting low_memory to False to handle huge datasets. Once the spark session was initialized then we gave our GCP bucket path to connect our dataset to pyspark. Once the bucket path was given, we first started by reading the dataset from the GCP bucket. As soon as this command was successfully executed, we split the column named ‘twic_number’ into multiple columns using the ‘split’ function in pyspark. The maximum number of splits identified in the data is then accommodated by the algorithm, which divides this column into additional columns depending on comma separators. Concatenating the new columns with the original dataset is done in a methodical manner, making sure that the original 'twic_number' column is not included. Once the column was split into different columns then we started selecting the required columns. The dataset's 'twic_number' column, which is probably a key identifier, is changed to a string data type. The cleaned and enlarged dataset is then shown as a preview of the first few rows, highlighting the altered data, and optionally saved to a new CSV file. Once the dataset was cleaned this dataset was saved in the google cloud storage bucket. Then finally we stopped the spark session and the cleaned dataset file was saved in the cloud storage bucket named *‘cleaned_data’. [from 3]*

DESCRIPTION OF DATA PROCESSING

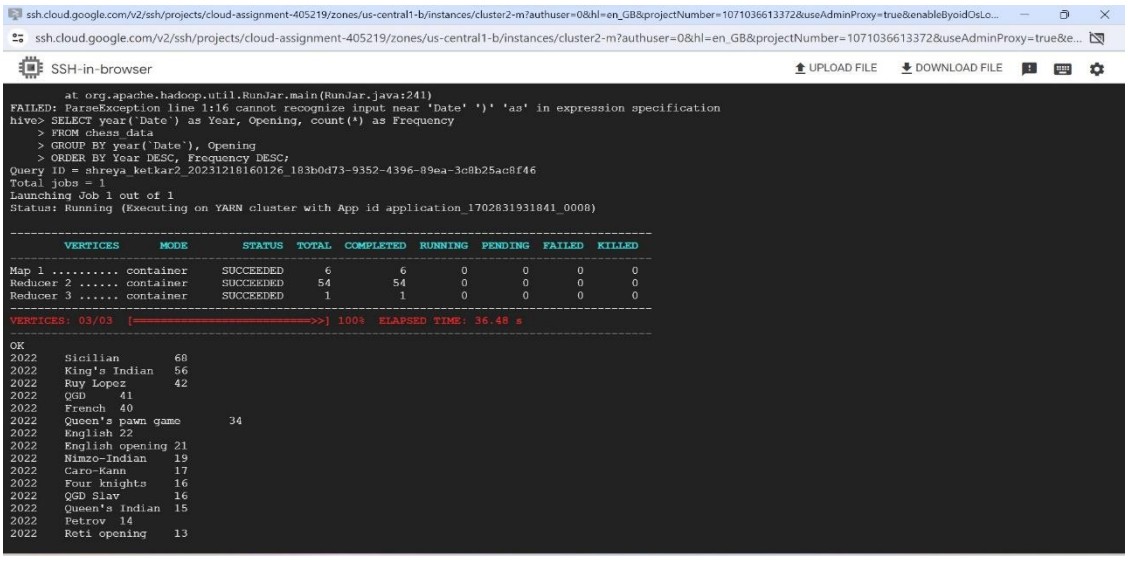
Data Processing using Hive:

Data processing is being done in the following steps:

Step 1: Once we got the cleaned dataset file in our cloud storage, we then executed the hive query to store the cleaned data in a table, for the using hive query language we created a new table called ‘chess_data’.



Step 2: Then we ran a hive query to display the year, name of the openings and the number openings for each year so that we can find the top openings from the entire cleaned dataset.



Step 3: The output of the above step was stored in a new table called *'top_openings'* to further visualise overall frequency of the openings in the dataset. The reason to create a new table again was to have an accurate data so that it would be easy for us to visualise it.

```
hive> CREATE TABLE top_openings (
    >     Year INT,
    >     Opening STRING,
    >     Frequency INT
    > );
OK
Time taken: 0.112 seconds
hive>
    > -- Inserting data into the table
    > INSERT INTO top_openings
    > SELECT year(`Date`) AS Year, Opening, count(*) AS Frequency
    > FROM chess_data
    > GROUP BY year(`Date`), Opening
    > ORDER BY Year DESC, Frequency DESC;
Query ID = shreya_ketkar2_20231219184027_9909ab83-4d4e-4c53-86e8-c256cf4d4f88
Total jobs = 1
Launching Job 1 out of 1
Tez session was closed. Reopening...
Session re-established.
Session re-established.
Status: Running (Executing on YARN cluster with App id application_1702831931841_0032)

--------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
--------------------------------------------------------------------------------
Map 1 .......... container     SUCCEEDED      6          6        0        0       0       0
Reducer 2 ...... container     SUCCEEDED     54         54        0        0       0       0
Reducer 3 ...... container     SUCCEEDED      1          1        0        0       0       0
--------------------------------------------------------------------------------
VERTICES: 03/03  [==========================>>] 100%  ELAPSED TIME: 39.51 s
--------------------------------------------------------------------------------
Loading data to table default.top_openings
OK
Time taken: 59.221 seconds
hive>
```

Step 4: Now to analyse new findings from the cleaned dataset we further executed another hive query to display the win rate of both black and white chess players. *[from 5]*

Step 5: The output of the step 4 was stored in a new table called 'effective_openings', to further visualise the number of wins and wins percentage for both black and white chess players and do a comparative study of the same. *[from 6]*

Step 6: Lastly, we converted the Hive tables named *'top_openings'* and *'effective_openings'* to CSV files using Hadoop processing commands to get a proper view and structure of the file. *[7]*

Code to processing using hive can be found in our git repo.

DEVELOPMENT OF THE APPLICATION PLATFORM

a. Data Extraction Tool

This process started with the acquisition of the dataset in CSV format, followed by its upload to the 'cloud_assignment-2' bucket on Google Cloud Platform. We then established a Hadoop cluster via Google DataProc, a critical step in preparing our computational environment. This setup was accomplished by configuring a cluster in the Dataproc section of the Google Cloud Console, with a specific focus on having a single master and two worker nodes. The installation phase involved establishing essential Hadoop components like MapReduce, PySpark, and Hive, ensuring the full functionality of our data processing system. Also, the cleaned data was extracted using Pyspark by executing the necessary code.

b. Data Processing Tool

Firstly, the cleaned dataset was retrieved from cloud storage, and then a new Hive table called 'chess_data' was created to hold the data. The year, name, and frequency of chess openings were then extracted from the information using a Hive query, allowing us to determine which openings have been the most common throughout time. The 'top_openings' database, created especially for the purpose of easily visualizing opening frequencies, contained the results of this query. The results of additional investigation on the victory rates of white and black players are kept in a separate table called "effective_openings."

c. Interface/Dashboard to view the results

We describe the intuitive and interactive visualizations created in **Tableau** using our chess dataset. We utilized a bubble chart to depict the top openings over the years, combining the aspects of popularity and time, thus offering a dynamic and comprehensive view of evolving game strategies.

The most popular or frequently used openings in chess were elegantly represented using a heat map, providing a vivid depiction of opening strategy preferences. We showcased the win rates of black and white players through a clear and informative bar graph, allowing for easy comparison. Additionally, a line chart was utilized to display the total number of games played over various years, highlighting trends and patterns in gameplay frequency. We created an interactive dashboard in tableau which consist of four buttons and these buttons will navigate through various graphs which were made using tableau. *[from 8 onwards]*

We further delved into the specifics of the dataset by illustrating the number of wins by black and white players each year, giving a detailed year-by-year breakdown. A comparative study was also presented, juxtaposing the total games played with the

wins achieved by white and black players, offering insights into game outcomes relative to player color.

CHALLENGES AND LESSONS LEARNED

We faced difficulties in computing results due to the separate Elo ratings for black and white players. This challenge taught us the importance of understanding and balancing complex data attributes to derive accurate insights. It highlighted the necessity of a nuanced approach to data analysis, especially in dealing with parameters that directly influence outcomes.

Initially opting for Pig for data cleaning, we encountered obstacles due to the extensive partitions in the data. This prompted us to switch to PySpark, which effectively managed the large datasets and offered greater flexibility. This experience underlined the importance of choosing the right tool for the job, being adaptable in our approach, and the value of PySpark in handling intricate data cleaning tasks.

Converting Hive query outputs into CSV format posed a significant hurdle. This process was crucial for our visualization phase, and learning to navigate Hive's export functionalities was a key takeaway. Furthermore, compatibility issues with Tableau, which initially did not accept CSV files, led us to convert these files into XLSX format. This taught us the importance of understanding the intricacies of file formats and the necessity of adaptability in data processing and visualization.

Initially considering Google Data Studio, we eventually settled on Tableau due to its advanced features. This decision-making process highlighted the importance of selecting the most suitable visualization tool that aligns with the project's needs, offering a lesson in evaluating and choosing software based on functionality and ease of use.

RESPONSIBILITY SECTION

Omkar Khaladkar

Task: Setting up the GCP, also handling front-end technologies like tableau, data processing using hive and git repository.

Description: Creation of buckets on GCP. Assisted in Frontend developments and visualizations using tableau, Choosing Technologies for frontend after analysis and creating a dashboard on tableau. I created the complete frontend dashboard where I showed various navigation in the dashboard. Along with that I helped in hive data processing queries to get the appropriate output which can further be used in visualisation. Also helped in designing a proper git repository.

Shreya Ketkar

Task: Assisted in backend code, data processing using hive, formulating question for analytics and documentation

Description: Firstly, I started with formulating the question for our analytics that would be the main finding of the entire assignment. I assisted in backend code to process the data using hive and executed the hive query to get top openings of all the years mentioned in the dataset. Along with that I helped in visualisation to perform various graph types which gave the appropriate finding. I also helped in creating the CSV file out of various hive outputs which we got after processing the data using hive queries. Lastly, I helped in documenting the complete report.

Stanley Johnson

Task: Dataset selection, choosing of technologies, helped in setting up GCP, assisted in data pre-processing and cleaning.

Description: I selected appropriate dataset on which we could do the analysis and comparative study. Once the dataset selection was done, I even analyzed the dataset and found out the important findings from it. Also, I helped in choosing the appropriate technologies and assisted my team mates in setting up GCP and data pre-processing and cleaning. I tried data cleaning with pig which was not successful and then we decided to move to pyspark for data cleaning.

Vaishnavi Kulkarni

Task: Dataset cleaning using pyspark, data integration, assisted in tableau visualization and documentation

Description: I successfully cleaned the dataset using pyspark by executing the appropriate code of the same. I executed the complete cleaning processing starting from uploading the dataset in a folder then executing it in pyspark. The cleaned dataset was then generated in GCP storage. I then helped in tableau visualization to get various graphs from the output which was generated using hive data processing. Also, I helped in data integration when cleaned dataset was generated in GCP bucket so the cleaned

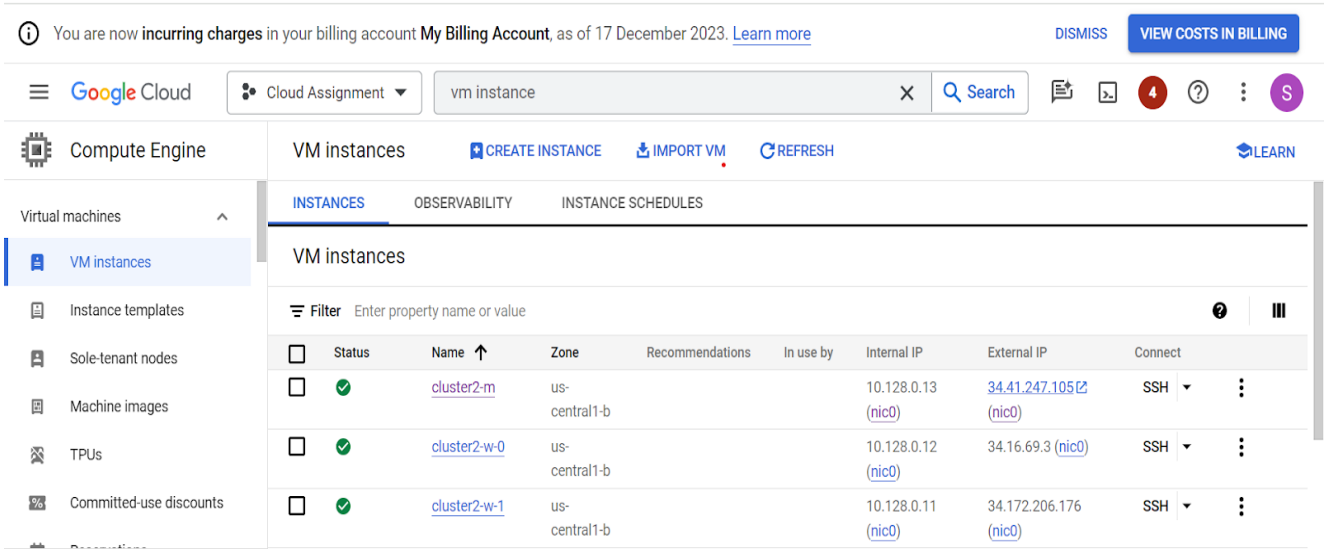
dataset had multiple files then using various pyspark code, I tried integrating the data into a single file so that the cleaned dataset can be used for data processing. Also, I carried out the complete documentation.

REFERENCES

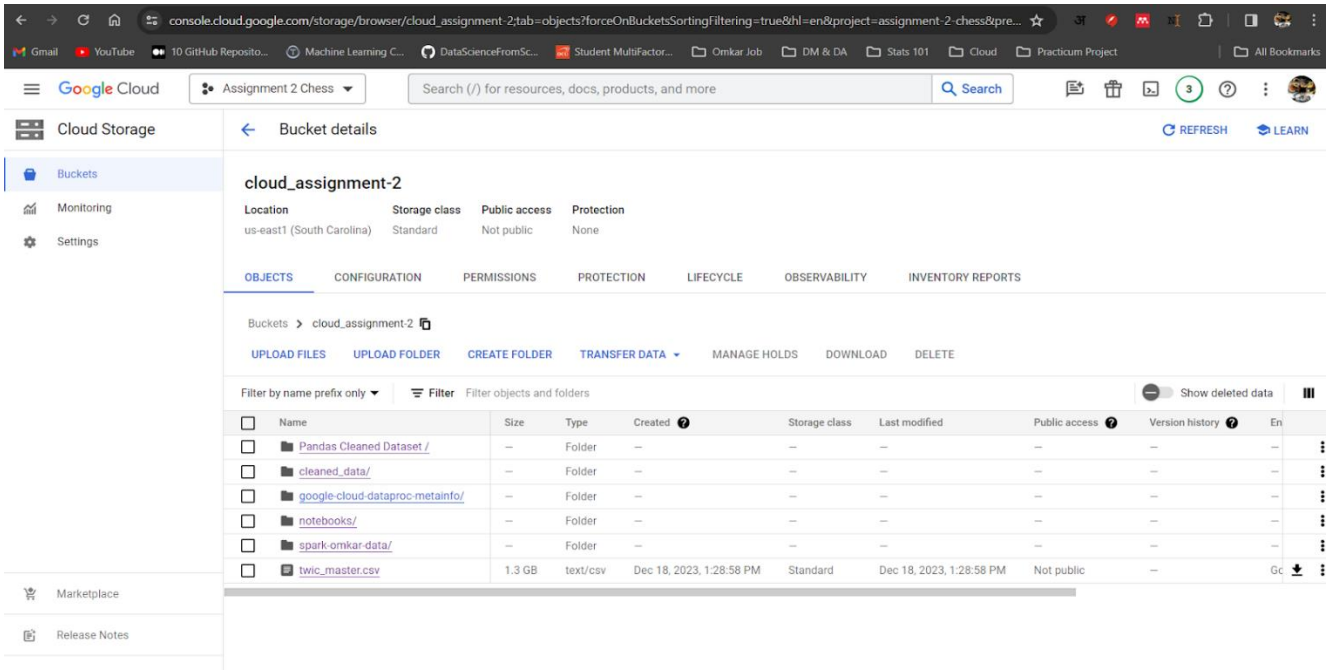
- **Apache Spark:** https://spark.apache.org/
- **Apache Hive:** https://www.geeksforgeeks.org/apache-hive/
- **GCP Basics:** https://www.geeksforgeeks.org/google-cloud-platform-tutorial/
- **Stack Overflow:** https://stackoverflow.com/

RELEVANT SCREENSHOTS

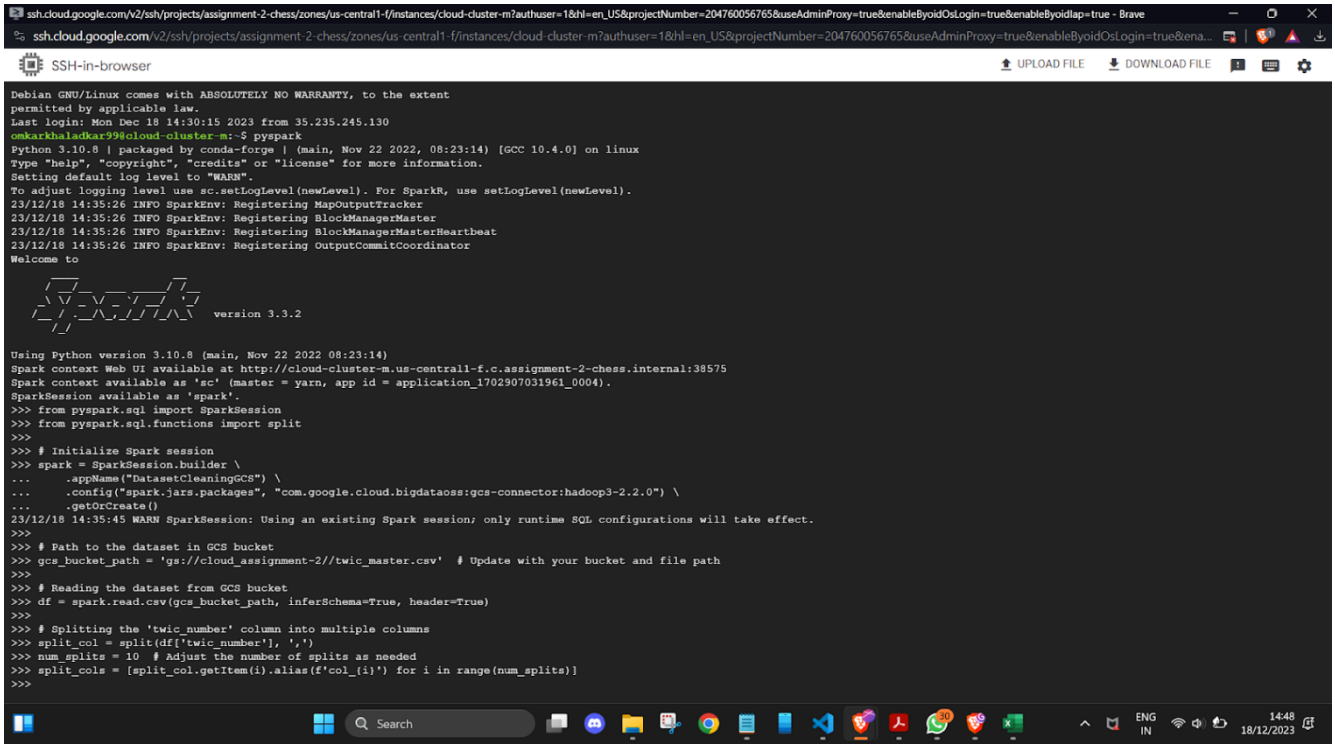
[1] Cluster created in GCP



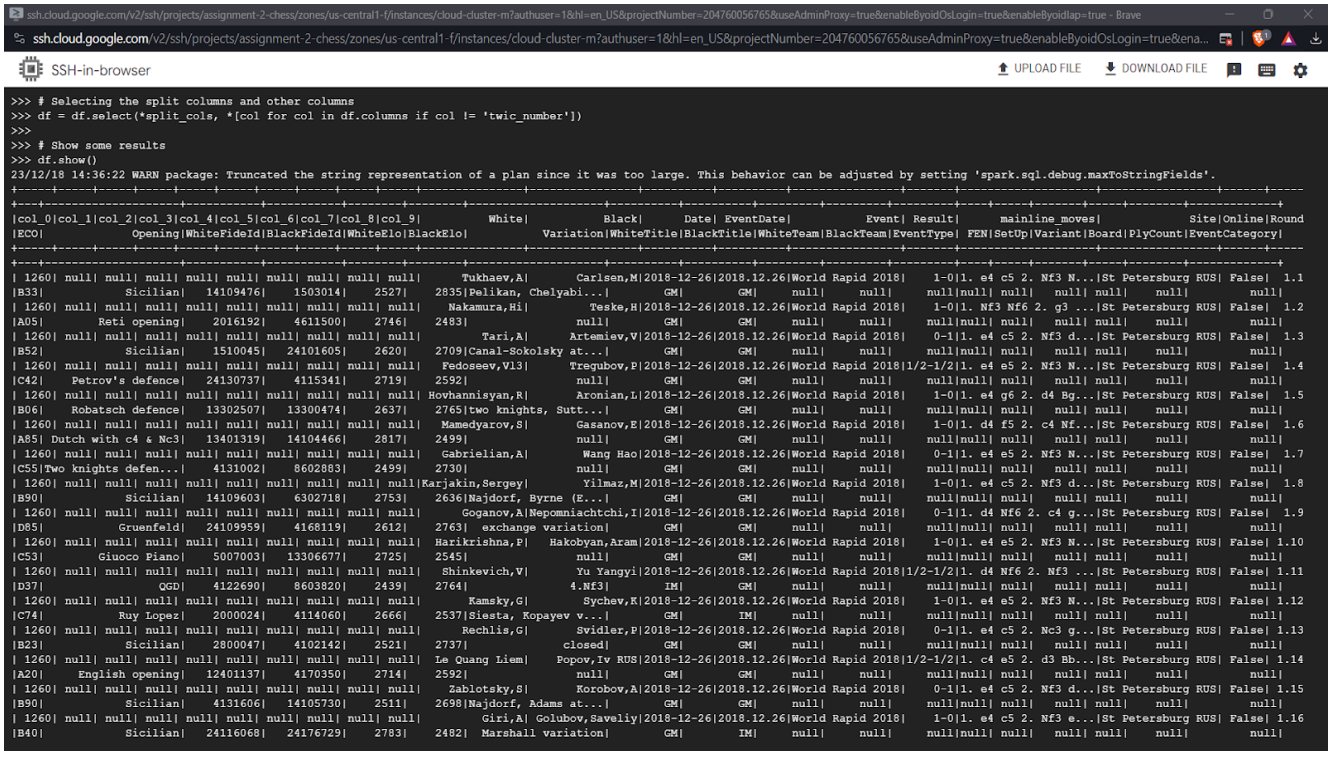
[2] Dataset uploaded in bucket



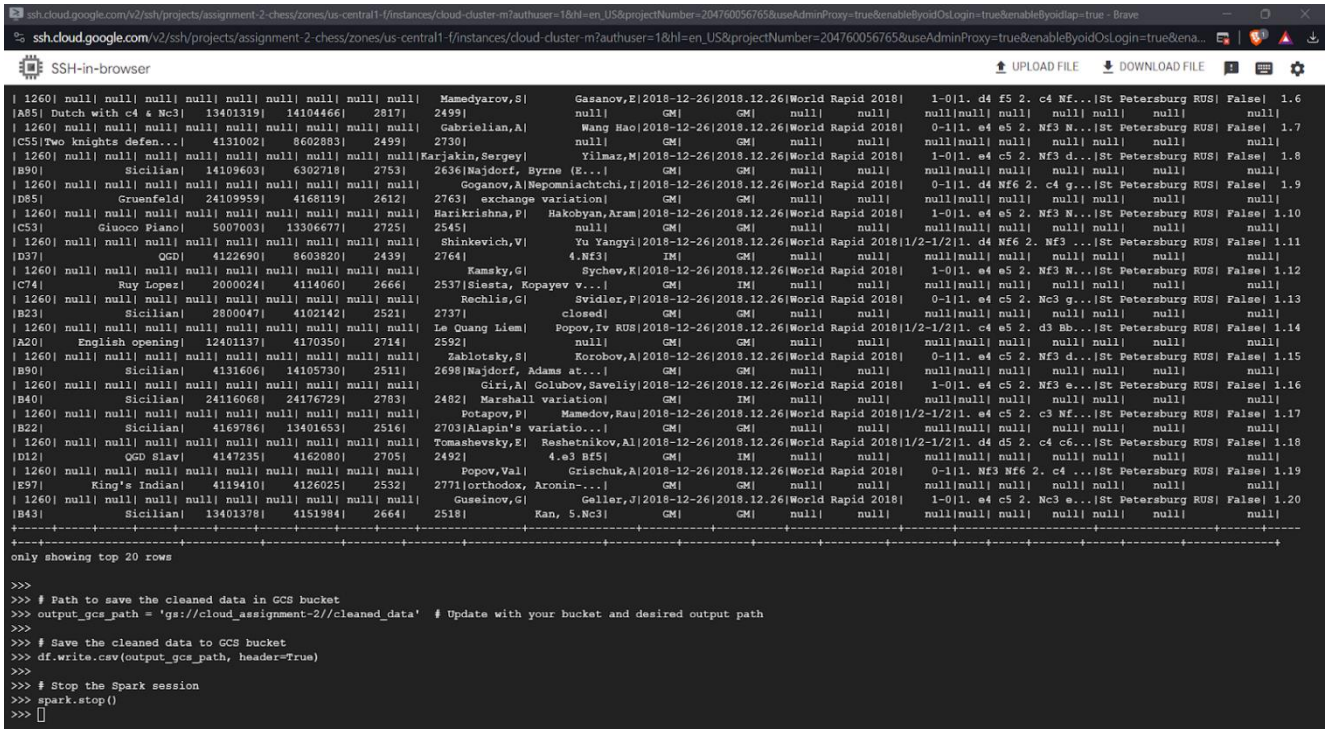
[3] Next few screenshots on Data Cleaning using pyspark

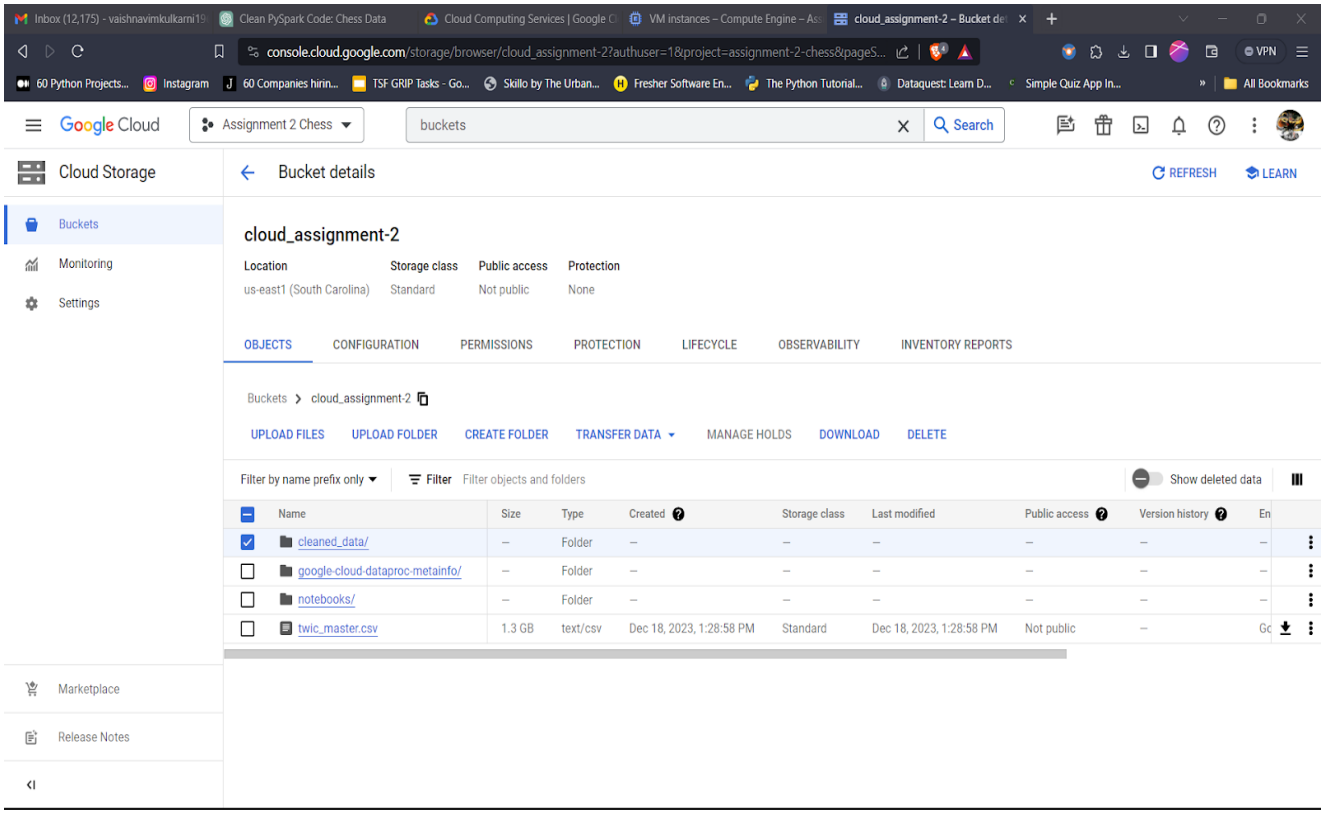


```
ssh.cloud.google.com/v2/ssh/projects/assignment-2-chess/zones/us-central1-f/instances/cloud-cluster-m?authuser=1&hl=en_US&projectNumber=204760056765&useAdminProxy=true&enableByoidOsLogin=true&enableByoidIap=true - Brave
SSH-in-browser   UPLOAD FILE   DOWNLOAD FILE

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Mon Dec 18 14:30:15 2023 from 35.235.245.130
omkarkhaladkar998@cloud-cluster-m:~$ pyspark
Python 3.10.8 | packaged by conda-forge | (main, Nov 22 2022, 08:23:14) [GCC 10.4.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
23/12/18 14:35:26 INFO SparkEnv: Registering MapOutputTracker
23/12/18 14:35:26 INFO SparkEnv: Registering BlockManagerMaster
23/12/18 14:35:26 INFO SparkEnv: Registering BlockManagerMasterHeartbeat
23/12/18 14:35:26 INFO SparkEnv: Registering OutputCommitCoordinator
Welcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /__ / .__/\_,_/_/ /_/\_\   version 3.3.2
      /_/

Using Python version 3.10.8 (main, Nov 22 2022 08:23:14)
Spark context Web UI available at http://cloud-cluster-m.us-central1-f.c.assignment-2-chess.internal:38575
Spark context available as 'sc' (master = yarn, app id = application_1702907031961_0004).
SparkSession available as 'spark'.
>>> from pyspark.sql import SparkSession
>>> from pyspark.sql.functions import split
>>>
>>> # Initialize Spark session
>>> spark = SparkSession.builder \
...     .appName("DatasetCleaningGCS") \
...     .config("spark.jars.packages", "com.google.cloud.bigdataoss:gcs-connector:hadoop3-2.2.0") \
...     .getOrCreate()
23/12/18 14:35:45 WARN SparkSession: Using an existing Spark session; only runtime SQL configurations will take effect.
>>>
>>> # Path to the dataset in GCS bucket
>>> gcs_bucket_path = 'gs://cloud_assignment-2//twic_master.csv'  # Update with your bucket and file path
>>>
>>> # Reading the dataset from GCS bucket
>>> df = spark.read.csv(gcs_bucket_path, inferSchema=True, header=True)
>>>
>>> # Splitting the 'twic_number' column into multiple columns
>>> split_col = split(df['twic_number'], ',')
>>> num_splits = 10  # Adjust the number of splits as needed
>>> split_cols = [split_col.getItem(i).alias(f'col_{i}') for i in range(num_splits)]
>>>
```

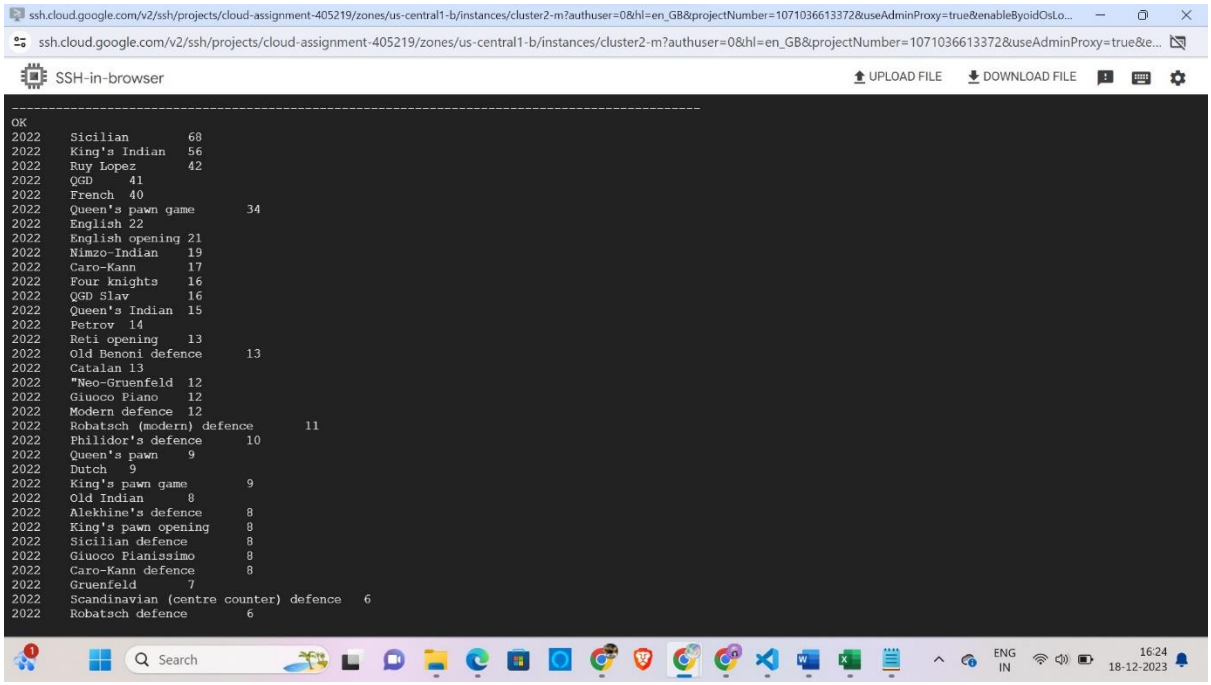


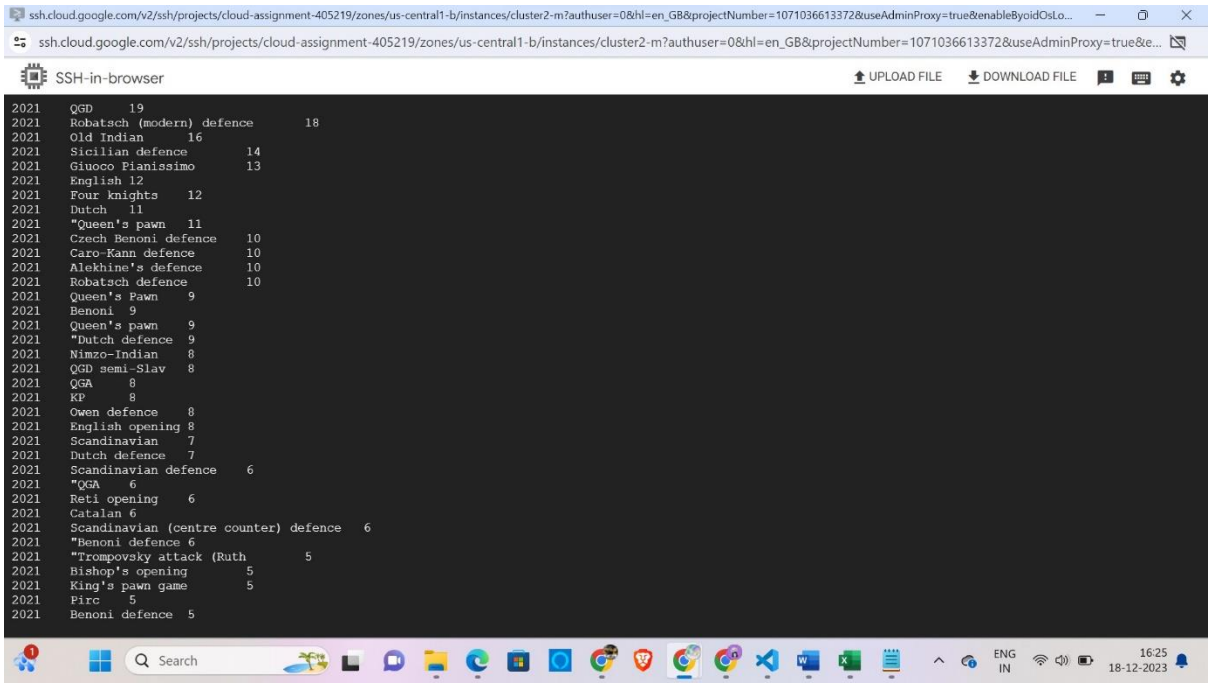
```
ssh.cloud.google.com/v2/ssh/projects/assignment-2-chess/zones/us-central1-f/instances/cloud-cluster-m?authuser=1&hl=en_US&projectNumber=204760056765&useAdminProxy=true&enableByoidOsLogin=true&enableByoidIap=true - Brave
SSH-in-browser   UPLOAD FILE   DOWNLOAD FILE

>>> # Selecting the split columns and other columns
>>> df = df.select(*split_cols, *[col for col in df.columns if col != 'twic_number'])
>>>
>>> # Show some results
>>> df.show()
23/12/18 14:36:22 WARN package: Truncated the string representation of a plan since it was too large. This behavior can be adjusted by setting 'spark.sql.debug.maxToStringFields'.
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+--------------------+--------------------+---------+---------+--------------------+-------+--------------------+--------------------+------+-----
+---+--------------------+----------+----------+-------+-------+--------------------+---------+---------+--------+--------+---------+----+-----+-------+-----+--------+-------------+
|col_0|col_1|col_2|col_3|col_4|col_5|col_6|col_7|col_8|col_9|          White|           Black|      Date| EventDate|           Event| Result|      mainline_moves|            Site|Online|Round
|ECO|             Opening|WhiteFideId|BlackFideId|WhiteElo|BlackElo|           Variation|WhiteTitle|BlackTitle|WhiteTeam|BlackTeam|EventType| FEN|SetUp|Variant|Board|PlyCount|EventCategory|
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+--------------------+--------------------+---------+---------+--------------------+-------+--------------------+--------------------+------+-----
+---+--------------------+----------+----------+-------+-------+--------------------+---------+---------+--------+--------+---------+----+-----+-------+-----+--------+-------------+
| 1260| null| null| null| null| null| null| null| null| null|      Tukhaev,A|       Carlsen,M|2018-12-26|2018.12.26|World Rapid 2018|    1-0|1. e4 c5 2. Nf3 N...|St Petersburg RUS| False|  1.1
|B33|           Sicilian|   14109476|    1503014|    2527|   2835|Pelikan, Chelyabi...|        GM|        GM|    null|    null|     null|null| null|   null| null|    null|         null|
| 1260| null| null| null| null| null| null| null| null| null|    Nakamura,Hi|         Teske,H|2018-12-26|2018.12.26|World Rapid 2018|    1-0|1. Nf3 Nf6 2. g3 ...|St Petersburg RUS| False|  1.2
|A05|       Reti opening|    2016192|    4611500|    2746|   2483|                null|        GM|        GM|    null|    null|     null|null| null|   null| null|    null|         null|
| 1260| null| null| null| null| null| null| null| null| null|         Tari,A|     Artemiev,V|2018-12-26|2018.12.26|World Rapid 2018|    0-1|1. e4 c5 2. Nf3 d...|St Petersburg RUS| False|  1.3
|B52|           Sicilian|    1510045|   24101605|    2620|   2709|Canal-Sokolsky at...|        GM|        GM|    null|    null|     null|null| null|   null| null|    null|         null|
| 1260| null| null| null| null| null| null| null| null| null|   Fedoseev,V13|      Tregubov,P|2018-12-26|2018.12.26|World Rapid 2018|1/2-1/2|1. e4 e5 2. Nf3 N...|St Petersburg RUS| False|  1.4
|C42|    Petrov's defence|   24130737|    4115341|    2719|   2592|                null|        GM|        GM|    null|    null|     null|null| null|   null| null|    null|         null|
| 1260| null| null| null| null| null| null| null| null| null|Hovhannisyan,R|      Aronian,L|2018-12-26|2018.12.26|World Rapid 2018|    1-0|1. e4 g6 2. d4 Bg...|St Petersburg RUS| False|  1.5
|B06|    Robatsch defence|   13302507|   13300474|    2637|   2765|two knights, Sutt...|        GM|        GM|    null|    null|     null|null| null|   null| null|    null|         null|
| 1260| null| null| null| null| null| null| null| null| null|   Mamedyarov,S|       Gasanov,E|2018-12-26|2018.12.26|World Rapid 2018|    1-0|1. d4 f5 2. c4 Nf...|St Petersburg RUS| False|  1.6
|A85| Dutch with c4 & Nc3|   13401319|   14104466|    2817|   2499|                null|        GM|        GM|    null|    null|     null|null| null|   null| null|    null|         null|
| 1260| null| null| null| null| null| null| null| null| null|   Gabrielian,A|       Wang Hao|2018-12-26|2018.12.26|World Rapid 2018|    0-1|1. e4 e5 2. Nf3 N...|St Petersburg RUS| False|  1.7
|C55|Two knights defen...|    4131002|    8602883|    2499|   2730|                null|        GM|        GM|    null|    null|     null|null| null|   null| null|    null|         null|
| 1260| null| null| null| null| null| null| null| null| null|Karjakin,Sergey|      Yilmaz,M|2018-12-26|2018.12.26|World Rapid 2018|    1-0|1. e4 c5 2. Nf3 d...|St Petersburg RUS| False|  1.8
|B90|           Sicilian|   14109603|    6302718|    2753|   2636|Najdorf, Byrne (E...|        GM|        GM|    null|    null|     null|null| null|   null| null|    null|         null|
| 1260| null| null| null| null| null| null| null| null| null|     Goganov,A|Nepomniachtchi,I|2018-12-26|2018.12.26|World Rapid 2018|    0-1|1. d4 Nf6 2. c4 g...|St Petersburg RUS| False|  1.9
|D85|          Gruenfeld|   24109959|    4168119|    2612|   2763|  exchange variation|        GM|        GM|    null|    null|     null|null| null|   null| null|    null|         null|
| 1260| null| null| null| null| null| null| null| null| null|  Harikrishna,P|  Hakobyan,Aram|2018-12-26|2018.12.26|World Rapid 2018|    1-0|1. e4 e5 2. Nf3 N...|St Petersburg RUS| False| 1.10
|C53|       Giuoco Piano|    5007003|   13306677|    2725|   2545|                null|        GM|        GM|    null|    null|     null|null| null|   null| null|    null|         null|
| 1260| null| null| null| null| null| null| null| null| null|  Shinkevich,V|       Yu Yangyi|2018-12-26|2018.12.26|World Rapid 2018|1/2-1/2|1. d4 Nf6 2. Nf3 ...|St Petersburg RUS| False| 1.11
|D37|             QGD|    4122690|    8603820|    2439|   2764|               4.Nf3|        IM|        GM|    null|    null|     null|null| null|   null| null|    null|         null|
| 1260| null| null| null| null| null| null| null| null| null|      Ramsky,G|      Sychev,K|2018-12-26|2018.12.26|World Rapid 2018|    1-0|1. e4 e5 2. Nf3 N...|St Petersburg RUS| False| 1.12
|C74|          Ruy Lopez|    2000024|    4114060|    2666|   2537|Siesta, Kopayev v...|        GM|        IM|    null|    null|     null|null| null|   null| null|    null|         null|
| 1260| null| null| null| null| null| null| null| null| null|     Rechlis,G|       Svidler,P|2018-12-26|2018.12.26|World Rapid 2018|    0-1|1. e4 c5 2. Nc3 g...|St Petersburg RUS| False| 1.13
|B23|           Sicilian|    2800047|    4102142|    2521|   2737|              closed|        GM|        GM|    null|    null|     null|null| null|   null| null|    null|         null|
| 1260| null| null| null| null| null| null| null| null| null|Le Quang Liem|   Popov,Iv RUS|2018-12-26|2018.12.26|World Rapid 2018|1/2-1/2|1. c4 e5 2. d3 Bb...|St Petersburg RUS| False| 1.14
|A20|     English opening|   12401137|    4170350|    2714|   2592|                null|        GM|        GM|    null|    null|     null|null| null|   null| null|    null|         null|
| 1260| null| null| null| null| null| null| null| null| null|   Zablotsky,S|      Korobov,A|2018-12-26|2018.12.26|World Rapid 2018|    0-1|1. e4 c5 2. Nf3 d...|St Petersburg RUS| False| 1.15
|B90|           Sicilian|    4131606|   14105730|    2511|   2698|Najdorf, Adams at...|        GM|        GM|    null|    null|     null|null| null|   null| null|    null|         null|
| 1260| null| null| null| null| null| null| null| null| null|       Giri,A| Golubov,Saveliy|2018-12-26|2018.12.26|World Rapid 2018|    1-0|1. e4 c5 2. Nf3 e...|St Petersburg RUS| False| 1.16
|B40|           Sicilian|   24116068|   24176729|    2783|   2482|  Marshall variation|        GM|        IM|    null|    null|     null|null| null|   null| null|    null|         null|
```

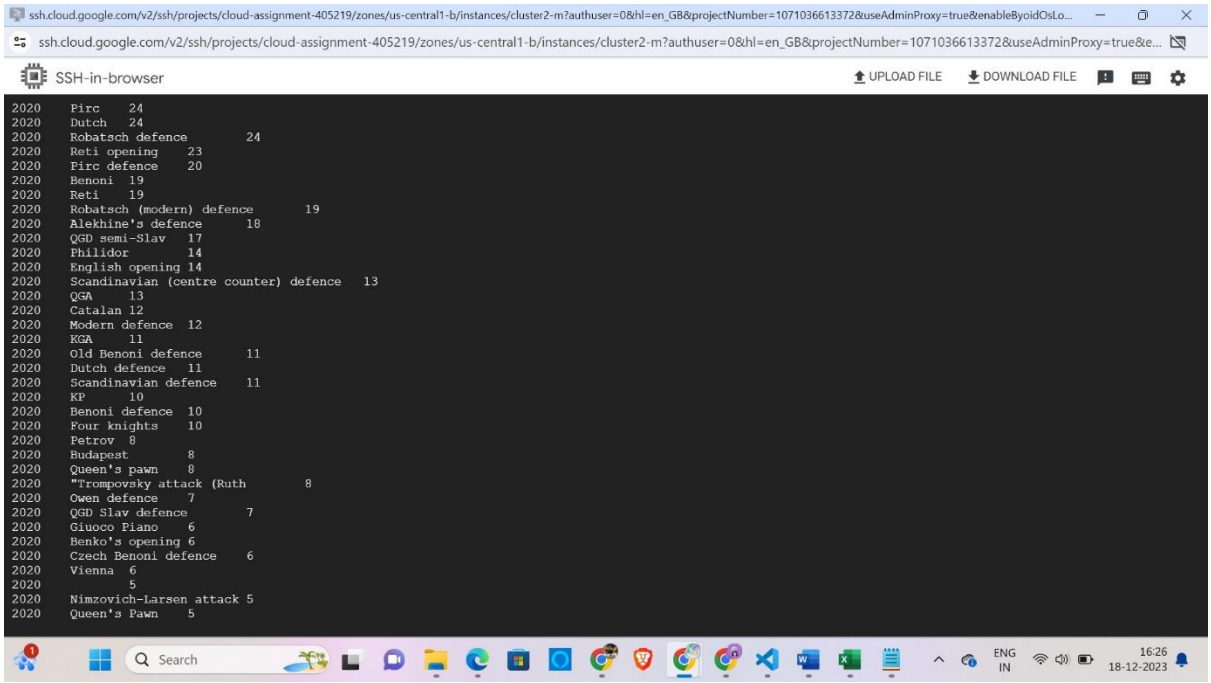

ssh.cloud.google.com/v2/ssh/projects/assignment-2-chess/zones/us-central1-f/instances/cloud-cluster-m?authuser=1&hl=en_US&projectNumber=204760056765&useAdminProxy=true&enableByoidOsLogin=true&enableByoidIap=true - Brave
SSH-in-browser
UPLOAD FILE
DOWNLOAD FILE
only showing top 20 rows
>>> # Path to save the cleaned data in GCS bucket
>>> output_gcs_path = 'gs://cloud_assignment-2//cleaned_data' # Update with your bucket and desired output path
>>> # Save the cleaned data to GCS bucket
>>> df.write.csv(output_gcs_path, header=True)
>>> # Stop the Spark session
>>> spark.stop()

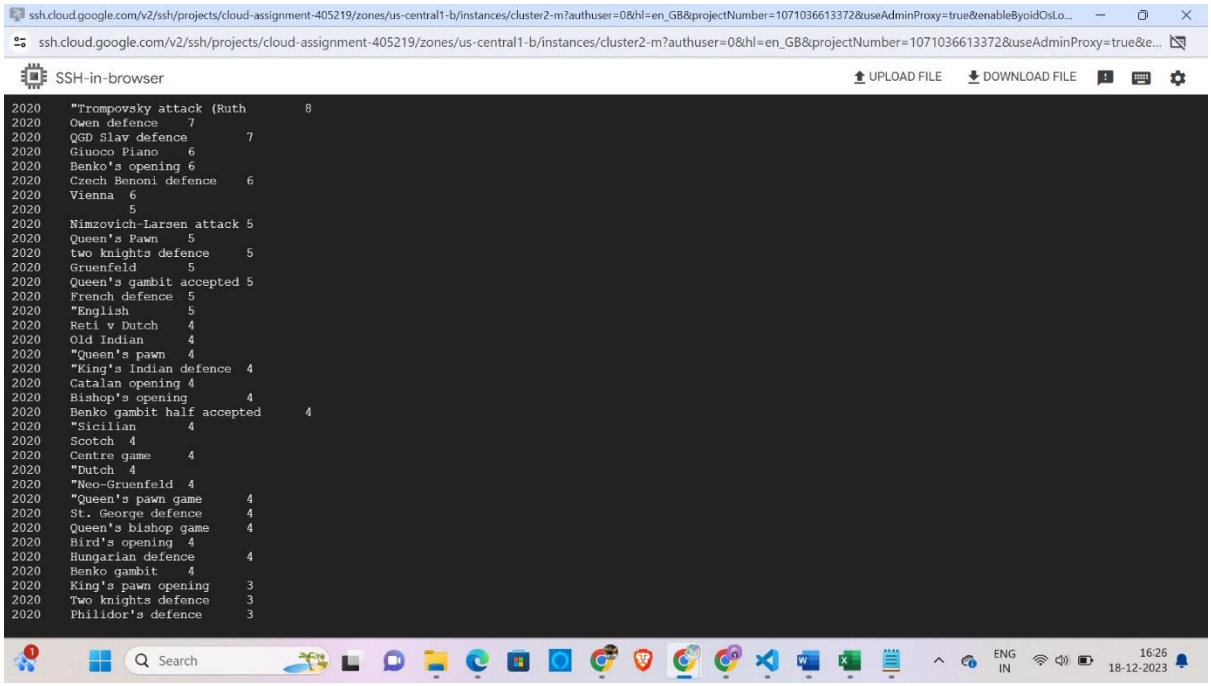

console.cloud.google.com/storage/browser/cloud_assignment-2?authuser=1&project=assignment-2-chess&pageS...
Google Cloud
Assignment 2 Chess
buckets
Search
Cloud Storage
Bucket details
REFRESH
LEARN
Buckets
Monitoring
Settings
cloud_assignment-2
Location
us-east1 (South Carolina)
Storage class
Standard
Public access
Not public
Protection
None
OBJECTS
CONFIGURATION
PERMISSIONS
PROTECTION
LIFECYCLE
OBSERVABILITY
INVENTORY REPORTS
Buckets › cloud_assignment-2
UPLOAD FILES
UPLOAD FOLDER
CREATE FOLDER
TRANSFER DATA
MANAGE HOLDS
DOWNLOAD
DELETE
Filter by name prefix only
Filter
Filter objects and folders
Show deleted data
Name
Size
Type
Created
Storage class
Last modified
Public access
Version history
cleaned_data/
google-cloud-dataproc-metainfo/
notebooks/
twic_master.csv
1.3 GB
text/csv
Dec 18, 2023, 1:28:58 PM
Standard
Dec 18, 2023, 1:28:58 PM
Not public
Marketplace
Release Notes

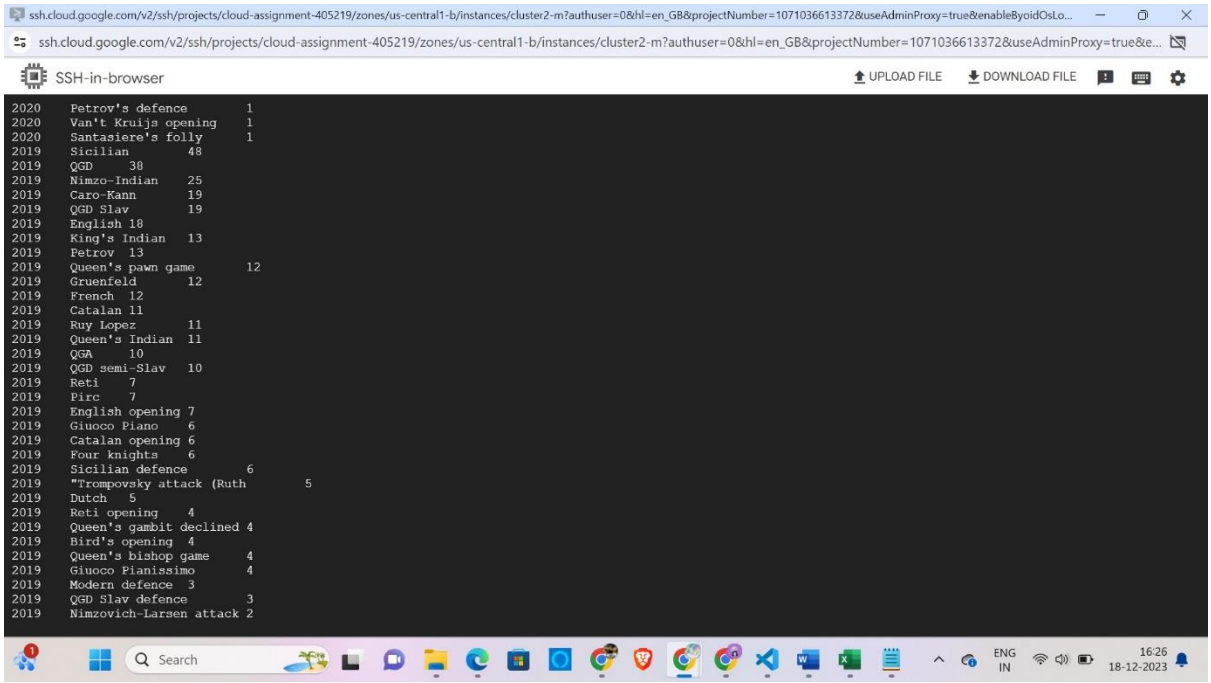
[4] Data Processing step 2 continuation

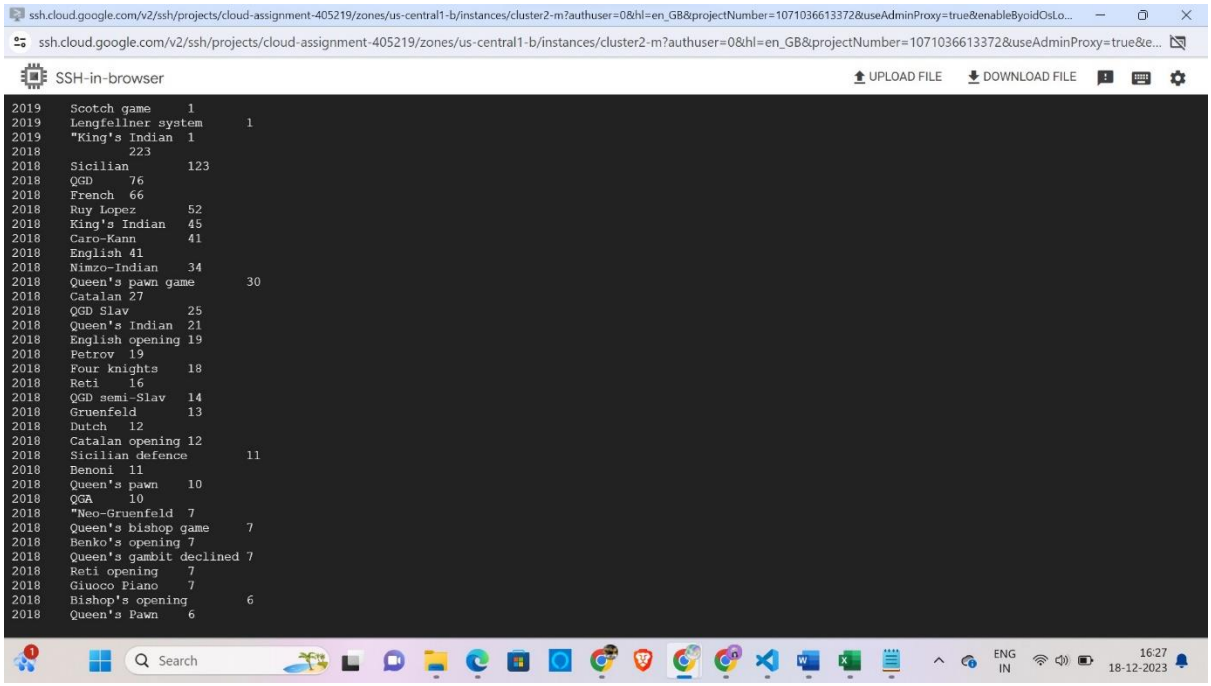


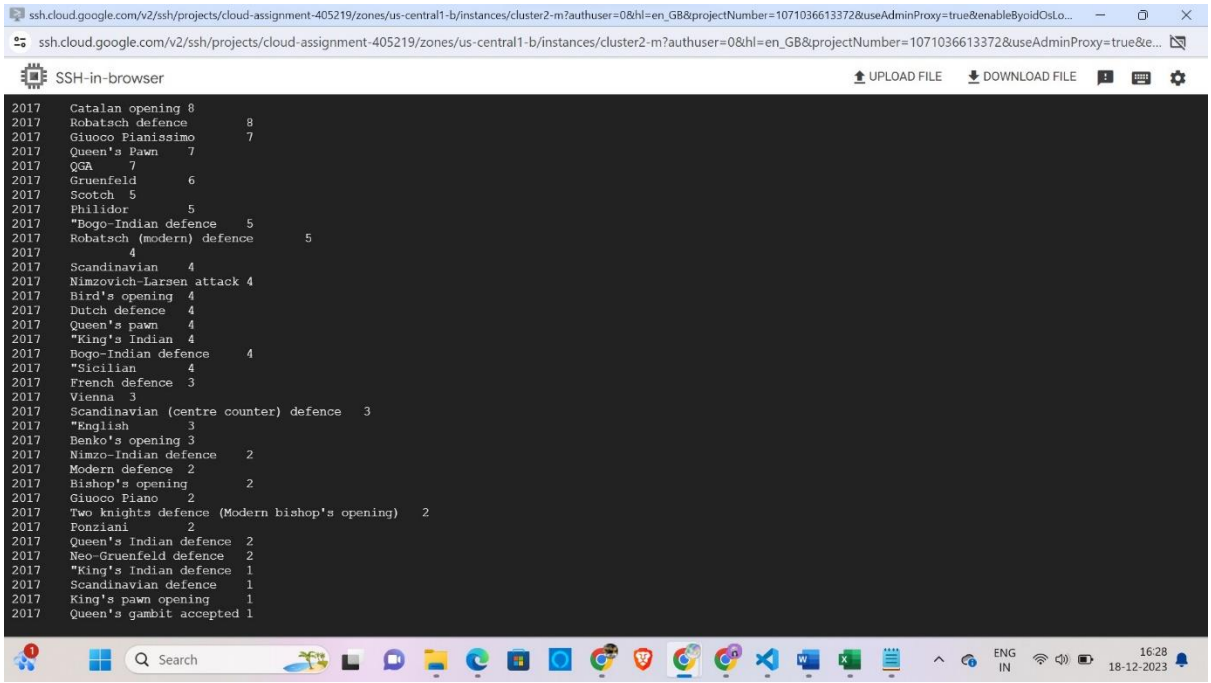


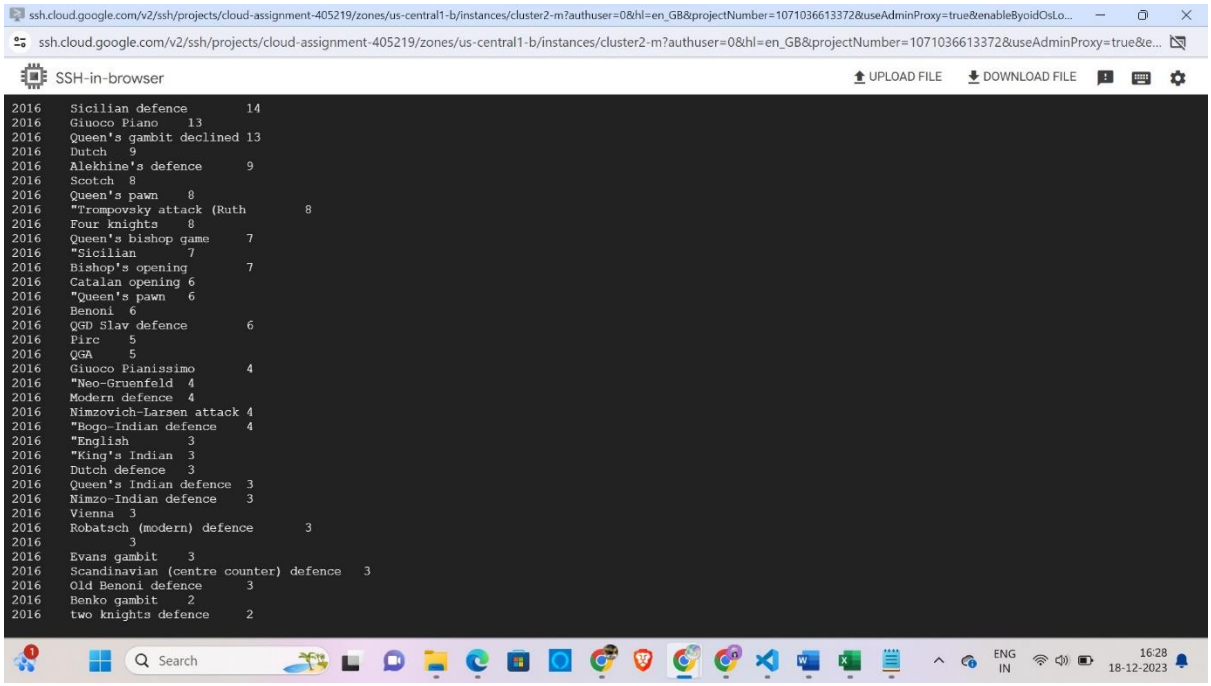

ssh.cloud.google.com/v2/ssh/projects/cloud-assignment-405219/zones/us-central1-b/instances/cluster2-m?authuser=0&hl=en_GB&projectNumber=1071036613372&useAdminProxy=true&enableByoidOsLo...
SSH-in-browser
UPLOAD FILE
DOWNLOAD FILE
2020 Pirc 24
2020 Dutch 24
2020 Robatsch defence 24
2020 Reti opening 23
2020 Pirc defence 20
2020 Benoni 19
2020 Reti 19
2020 Robatsch (modern) defence 19
2020 Alekhine's defence 18
2020 QGD semi-Slav 17
2020 Philidor 14
2020 English opening 14
2020 Scandinavian (centre counter) defence 13
2020 QGA 13
2020 Catalan 12
2020 Modern defence 12
2020 KGA 11
2020 Old Benoni defence 11
2020 Dutch defence 11
2020 Scandinavian defence 11
2020 KP 10
2020 Benoni defence 10
2020 Four knights 10
2020 Petrov 8
2020 Budapest 8
2020 Queen's pawn 8
2020 "Trompovsky attack (Ruth 8
2020 Owen defence 7
2020 QGD Slav defence 7
2020 Giuoco Piano 6
2020 Benko's opening 6
2020 Czech Benoni defence 6
2020 Vienna 6
2020 5
2020 Nimzovich-Larsen attack 5
2020 Queen's Pawn 5
Search
ENG
IN
16:26
18-12-2023

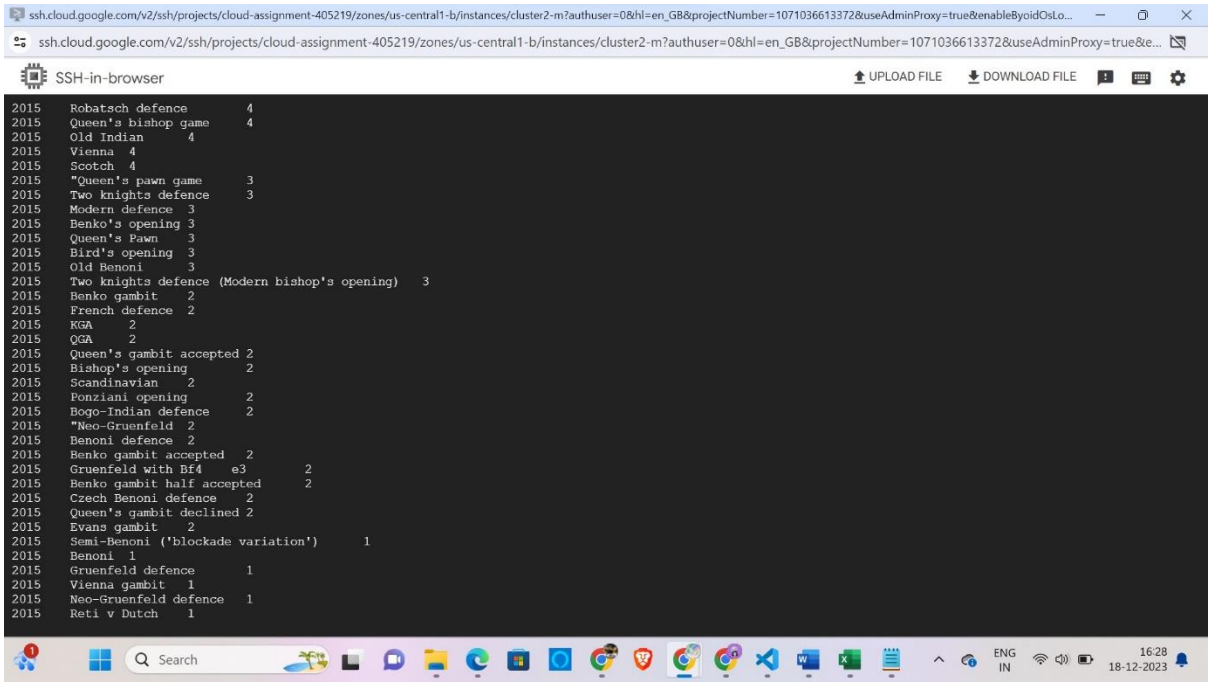

ssh.cloud.google.com/v2/ssh/projects/cloud-assignment-405219/zones/us-central1-b/instances/cluster2-m?authuser=0&hl=en_GB&projectNumber=1071036613372&useAdminProxy=true&enableByoidOsLo...
SSH-in-browser
UPLOAD FILE
DOWNLOAD FILE
2020 "Trompovsky attack (Ruth 8
2020 Owen defence 7
2020 QGD Slav defence 7
2020 Giuoco Piano 6
2020 Benko's opening 6
2020 Czech Benoni defence 6
2020 Vienna 6
2020 5
2020 Nimzovich-Larsen attack 5
2020 Queen's Pawn 5
2020 two knights defence 5
2020 Gruenfeld 5
2020 Queen's gambit accepted 5
2020 French defence 5
2020 "English 5
2020 Reti v Dutch 4
2020 Old Indian 4
2020 "Queen's pawn 4
2020 "King's Indian defence 4
2020 Catalan opening 4
2020 Bishop's opening 4
2020 Benko gambit half accepted 4
2020 "Sicilian 4
2020 Scotch 4
2020 Centre game 4
2020 "Dutch 4
2020 "Neo-Gruenfeld 4
2020 "Queen's pawn game 4
2020 St. George defence 4
2020 Queen's bishop game 4
2020 Bird's opening 4
2020 Hungarian defence 4
2020 Benko gambit 4
2020 King's pawn opening 3
2020 Two knights defence 3
2020 Philidor's defence 3
Search
ENG
IN
16:26
18-12-2023

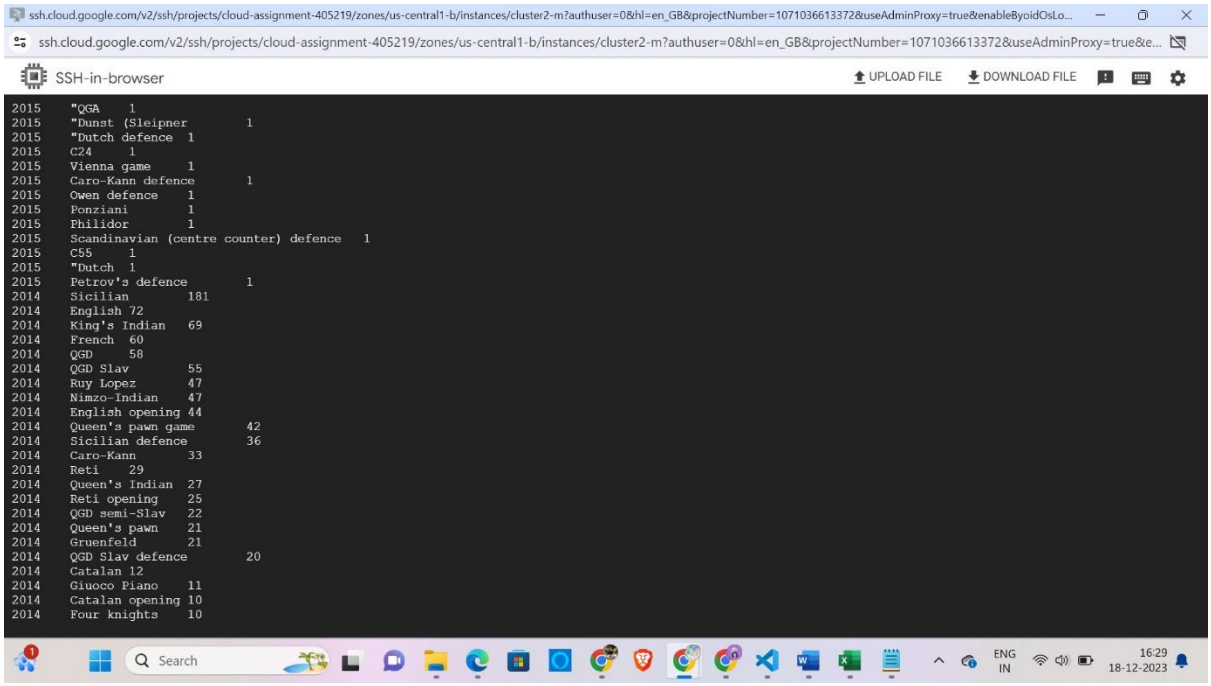

ssh.cloud.google.com/v2/ssh/projects/cloud-assignment-405219/zones/us-central1-b/instances/cluster2-m?authuser=0&hl=en_GB&projectNumber=1071036613372&useAdminProxy=true&enableByoidOsLo...
SSH-in-browser
UPLOAD FILE
DOWNLOAD FILE
2020 Petrov's defence 1
2020 Van't Kruijs opening 1
2020 Santasiere's folly 1
2019 Sicilian 48
2019 QGD 38
2019 Nimzo-Indian 25
2019 Caro-Kann 19
2019 QGD Slav 19
2019 English 18
2019 King's Indian 13
2019 Petrov 13
2019 Queen's pawn game 12
2019 Gruenfeld 12
2019 French 12
2019 Catalan 11
2019 Ruy Lopez 11
2019 Queen's Indian 11
2019 QGA 10
2019 QGD semi-Slav 10
2019 Reti 7
2019 Pirc 7
2019 English opening 7
2019 Giuoco Piano 6
2019 Catalan opening 6
2019 Four knights 6
2019 Sicilian defence 6
2019 "Trompovsky attack (Ruth 5
2019 Dutch 5
2019 Reti opening 4
2019 Queen's gambit declined 4
2019 Bird's opening 4
2019 Queen's bishop game 4
2019 Giuoco Pianissimo 4
2019 Modern defence 3
2019 QGD Slav defence 3
2019 Nimzovich-Larsen attack 2
Search
ENG
IN
16:26
18-12-2023

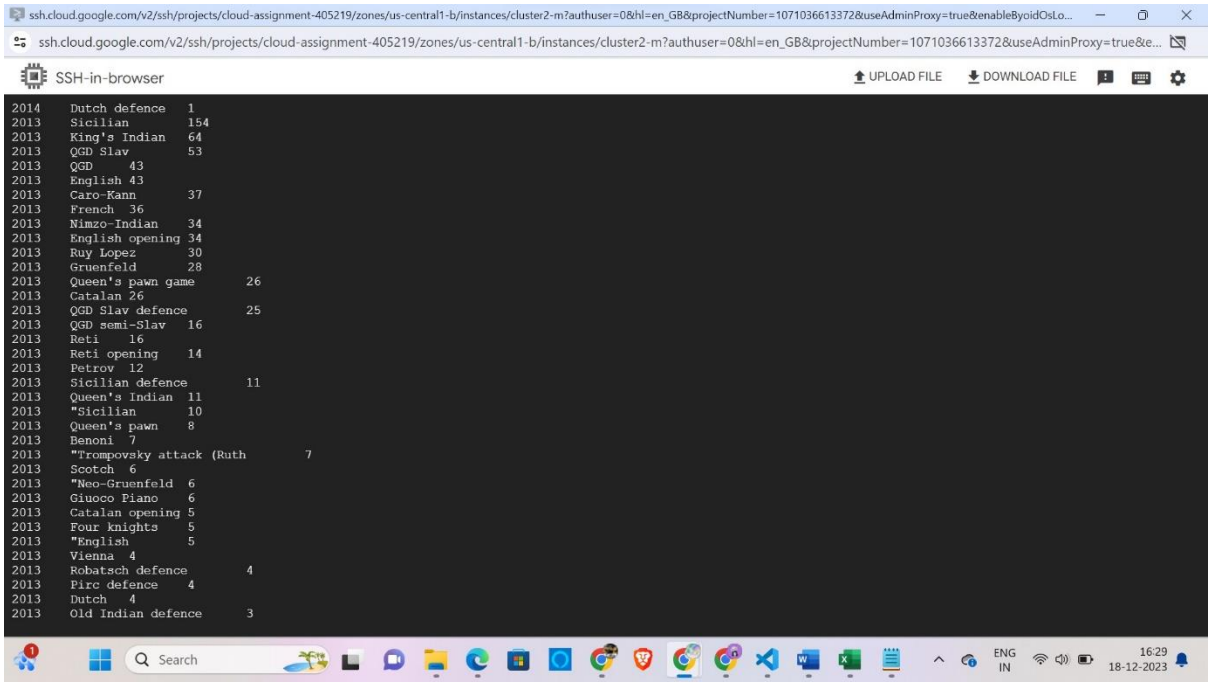

ssh.cloud.google.com/v2/ssh/projects/cloud-assignment-405219/zones/us-central1-b/instances/cluster2-m?authuser=0&hl=en_GB&projectNumber=1071036613372&useAdminProxy=true&enableByoidOsLo...
SSH-in-browser
UPLOAD FILE
DOWNLOAD FILE
2019 Scotch game 1
2019 Lengfellner system 1
2019 "King's Indian 1
2018 223
2018 Sicilian 123
2018 QGD 76
2018 French 66
2018 Ruy Lopez 52
2018 King's Indian 45
2018 Caro-Kann 41
2018 English 41
2018 Nimzo-Indian 34
2018 Queen's pawn game 30
2018 Catalan 27
2018 QGD Slav 25
2018 Queen's Indian 21
2018 English opening 19
2018 Petrov 19
2018 Four knights 18
2018 Reti 16
2018 QGD semi-Slav 14
2018 Gruenfeld 13
2018 Dutch 12
2018 Catalan opening 12
2018 Sicilian defence 11
2018 Benoni 11
2018 Queen's pawn 10
2018 QGA 10
2018 "Neo-Gruenfeld 7
2018 Queen's bishop game 7
2018 Benko's opening 7
2018 Queen's gambit declined 7
2018 Reti opening 7
2018 Giuoco Piano 7
2018 Bishop's opening 6
2018 Queen's Pawn 6
Search
ENG
IN
16:27
18-12-2023

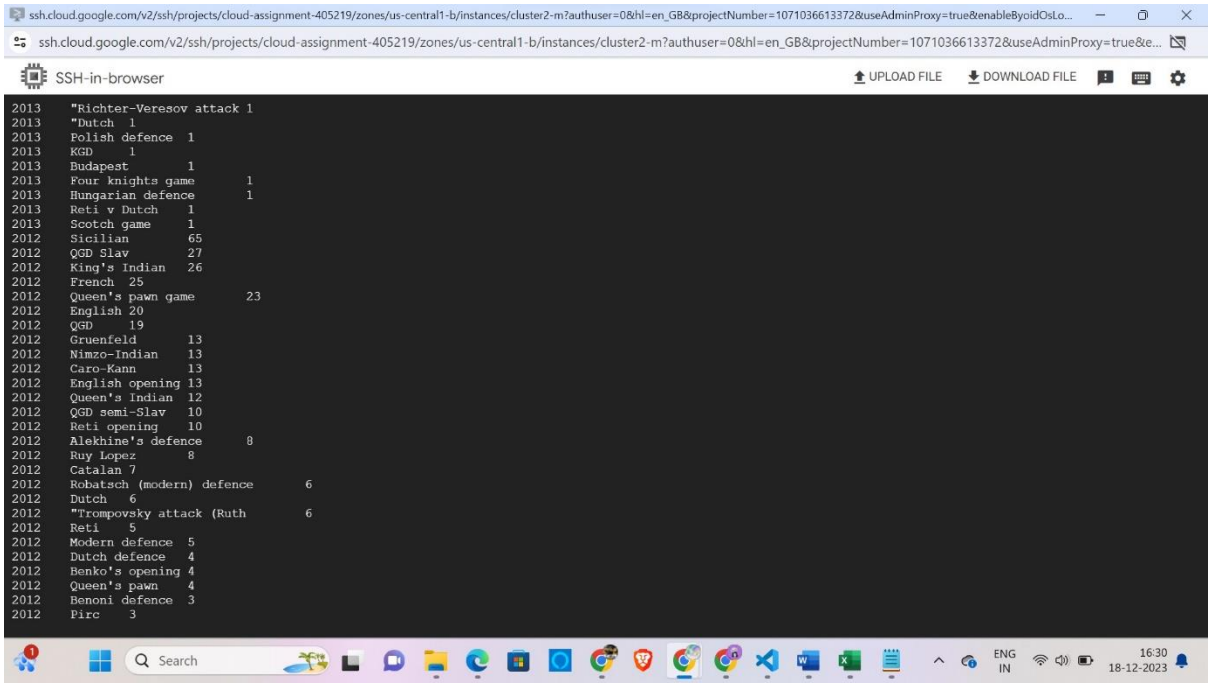

ssh.cloud.google.com/v2/ssh/projects/cloud-assignment-405219/zones/us-central1-b/instances/cluster2-m?authuser=0&hl=en_GB&projectNumber=1071036613372&useAdminProxy=true&enableByoidOsLo...
SSH-in-browser
UPLOAD FILE
DOWNLOAD FILE
2017 Catalan opening 8
2017 Robatsch defence 8
2017 Giuoco Pianissimo 7
2017 Queen's Pawn 7
2017 QGA 7
2017 Gruenfeld 6
2017 Scotch 5
2017 Philidor 5
2017 "Bogo-Indian defence 5
2017 Robatsch (modern) defence 5
2017 4
2017 Scandinavian 4
2017 Nimzovich-Larsen attack 4
2017 Bird's opening 4
2017 Dutch defence 4
2017 Queen's pawn 4
2017 "King's Indian 4
2017 Bogo-Indian defence 4
2017 "Sicilian 4
2017 French defence 3
2017 Vienna 3
2017 Scandinavian (centre counter) defence 3
2017 "English 3
2017 Benko's opening 3
2017 Nimzo-Indian defence 2
2017 Modern defence 2
2017 Bishop's opening 2
2017 Giuoco Piano 2
2017 Two knights defence (Modern bishop's opening) 2
2017 Ponziani 2
2017 Queen's Indian defence 2
2017 Neo-Gruenfeld defence 2
2017 "King's Indian defence 1
2017 Scandinavian defence 1
2017 King's pawn opening 1
2017 Queen's gambit accepted 1
Search
ENG
IN
16:28
18-12-2023


ssh.cloud.google.com/v2/ssh/projects/cloud-assignment-405219/zones/us-central1-b/instances/cluster2-m?authuser=0&hl=en_GB&projectNumber=1071036613372&useAdminProxy=true&enableByoidOsLo...
SSH-in-browser
UPLOAD FILE
DOWNLOAD FILE
2016 Sicilian defence 14
2016 Giuoco Piano 13
2016 Queen's gambit declined 13
2016 Dutch 9
2016 Alekhine's defence 9
2016 Scotch 8
2016 Queen's pawn 8
2016 "Trompovsky attack (Ruth 8
2016 Four knights 8
2016 Queen's bishop game 7
2016 "Sicilian 7
2016 Bishop's opening 7
2016 Catalan opening 6
2016 "Queen's pawn 6
2016 Benoni 6
2016 QGD Slav defence 6
2016 Pirc 5
2016 QGA 5
2016 Giuoco Pianissimo 4
2016 "Neo-Gruenfeld 4
2016 Modern defence 4
2016 Nimzovich-Larsen attack 4
2016 "Bogo-Indian defence 4
2016 "English 3
2016 "King's Indian 3
2016 Dutch defence 3
2016 Queen's Indian defence 3
2016 Nimzo-Indian defence 3
2016 Vienna 3
2016 Robatsch (modern) defence 3
2016 3
2016 Evans gambit 3
2016 Scandinavian (centre counter) defence 3
2016 Old Benoni defence 3
2016 Benko gambit 2
2016 two knights defence 2
Search
ENG
IN
16:28
18-12-2023


ssh.cloud.google.com/v2/ssh/projects/cloud-assignment-405219/zones/us-central1-b/instances/cluster2-m?authuser=0&hl=en_GB&projectNumber=1071036613372&useAdminProxy=true&enableByoidOsLo...
SSH-in-browser
UPLOAD FILE
DOWNLOAD FILE
2015 Robatsch defence 4
2015 Queen's bishop game 4
2015 Old Indian 4
2015 Vienna 4
2015 Scotch 4
2015 "Queen's pawn game 3
2015 Two knights defence 3
2015 Modern defence 3
2015 Benko's opening 3
2015 Queen's Pawn 3
2015 Bird's opening 3
2015 Old Benoni 3
2015 Two knights defence (Modern bishop's opening) 3
2015 Benko gambit 2
2015 French defence 2
2015 KGA 2
2015 QGA 2
2015 Queen's gambit accepted 2
2015 Bishop's opening 2
2015 Scandinavian 2
2015 Ponziani opening 2
2015 Bogo-Indian defence 2
2015 "Neo-Gruenfeld 2
2015 Benoni defence 2
2015 Benko gambit accepted 2
2015 Gruenfeld with Bf4 e3 2
2015 Benko gambit half accepted 2
2015 Czech Benoni defence 2
2015 Queen's gambit declined 2
2015 Evans gambit 2
2015 Semi-Benoni ('blockade variation') 1
2015 Benoni 1
2015 Gruenfeld defence 1
2015 Vienna gambit 1
2015 Neo-Gruenfeld defence 1
2015 Reti v Dutch 1
Search
ENG
IN
16:28
18-12-2023


ssh.cloud.google.com/v2/ssh/projects/cloud-assignment-405219/zones/us-central1-b/instances/cluster2-m?authuser=0&hl=en_GB&projectNumber=1071036613372&useAdminProxy=true&enableByoidOsLo...
SSH-in-browser
UPLOAD FILE
DOWNLOAD FILE
2015 "QGA 1
2015 "Dunst (Sleipner 1
2015 "Dutch defence 1
2015 C24 1
2015 Vienna game 1
2015 Caro-Kann defence 1
2015 Owen defence 1
2015 Ponziani 1
2015 Philidor 1
2015 Scandinavian (centre counter) defence 1
2015 C55 1
2015 "Dutch 1
2015 Petrov's defence 1
2014 Sicilian 181
2014 English 72
2014 King's Indian 69
2014 French 60
2014 QGD 58
2014 QGD Slav 55
2014 Ruy Lopez 47
2014 Nimzo-Indian 47
2014 English opening 44
2014 Queen's pawn game 42
2014 Sicilian defence 36
2014 Caro-Kann 33
2014 Reti 29
2014 Queen's Indian 27
2014 Reti opening 25
2014 QGD semi-Slav 22
2014 Queen's pawn 21
2014 Gruenfeld 21
2014 QGD Slav defence 20
2014 Catalan 12
2014 Giuoco Piano 11
2014 Catalan opening 10
2014 Four knights 10
Search
ENG
IN
16:29
18-12-2023


ssh.cloud.google.com/v2/ssh/projects/cloud-assignment-405219/zones/us-central1-b/instances/cluster2-m?authuser=0&hl=en_GB&projectNumber=1071036613372&useAdminProxy=true&enableByoidOsLo...
SSH-in-browser
UPLOAD FILE
DOWNLOAD FILE
2014 Dutch defence 1
2013 Sicilian 154
2013 King's Indian 64
2013 QGD Slav 53
2013 QGD 43
2013 English 43
2013 Caro-Kann 37
2013 French 36
2013 Nimzo-Indian 34
2013 English opening 34
2013 Ruy Lopez 30
2013 Gruenfeld 28
2013 Queen's pawn game 26
2013 Catalan 26
2013 QGD Slav defence 25
2013 QGD semi-Slav 16
2013 Reti 16
2013 Reti opening 14
2013 Petrov 12
2013 Sicilian defence 11
2013 Queen's Indian 11
2013 "Sicilian 10
2013 Queen's pawn 8
2013 Benoni 7
2013 "Trompovsky attack (Ruth 7
2013 Scotch 6
2013 "Neo-Gruenfeld 6
2013 Giuoco Piano 6
2013 Catalan opening 5
2013 Four knights 5
2013 "English 5
2013 Vienna 4
2013 Robatsch defence 4
2013 Pirc defence 4
2013 Dutch 4
2013 Old Indian defence 3
Search
ENG
IN
16:29
18-12-2023


ssh.cloud.google.com/v2/ssh/projects/cloud-assignment-405219/zones/us-central1-b/instances/cluster2-m?authuser=0&hl=en_GB&projectNumber=1071036613372&useAdminProxy=true&enableByoidOsLo...
SSH-in-browser
UPLOAD FILE
DOWNLOAD FILE
2013 "Richter-Veresov attack 1
2013 "Dutch 1
2013 Polish defence 1
2013 KGD 1
2013 Budapest 1
2013 Four knights game 1
2013 Hungarian defence 1
2013 Reti v Dutch 1
2013 Scotch game 1
2012 Sicilian 65
2012 QGD Slav 27
2012 King's Indian 26
2012 French 25
2012 Queen's pawn game 23
2012 English 20
2012 QGD 19
2012 Gruenfeld 13
2012 Nimzo-Indian 13
2012 Caro-Kann 13
2012 English opening 13
2012 Queen's Indian 12
2012 QGD semi-Slav 10
2012 Reti opening 10
2012 Alekhine's defence 8
2012 Ruy Lopez 8
2012 Catalan 7
2012 Robatsch (modern) defence 6
2012 Dutch 6
2012 "Trompovsky attack (Ruth 6
2012 Reti 5
2012 Modern defence 5
2012 Dutch defence 4
2012 Benko's opening 4
2012 Queen's pawn 4
2012 Benoni defence 3
2012 Pirc 3
Search
ENG
IN
16:30
18-12-2023

[5] Data Processing step 4

```
hive> SELECT
    >     year(`Date`) AS Year,
    >     Opening,
    >     COUNT(*) AS TotalGames,
    >     SUM(CASE WHEN Result = '1-0' THEN 1 ELSE 0 END) AS WhiteWins,
    >     SUM(CASE WHEN Result = '0-1' THEN 1 ELSE 0 END) AS BlackWins,
    >     SUM(CASE WHEN Result = '1-0' THEN 1 ELSE 0 END) / COUNT(*) AS WhiteWinRate,
    >     SUM(CASE WHEN Result = '0-1' THEN 1 ELSE 0 END) / COUNT(*) AS BlackWinRate
    > FROM
    >     chess_data
    > WHERE
    >     year(`Date`) BETWEEN 2012 AND 2022
    > GROUP BY
    >     year(`Date`), Opening;
Query ID = shreya_ketkar2_20231219184645_5b162955-e61d-43af-89a5-bf27b5cf8735
Total jobs = 1
Launching Job 1 out of 1
Tez session was closed. Reopening...
Session re-established.
Session re-established.
Status: Running (Executing on YARN cluster with App id application_1702831931841_0033)

--------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
--------------------------------------------------------------------------------
Map 1 .......... container     SUCCEEDED      6          6        0        0       0       0
Reducer 2 ...... container     SUCCEEDED     54         54        0        0       0       0
--------------------------------------------------------------------------------
VERTICES: 02/02  [==========================>>] 100%  ELAPSED TIME: 34.06 s
--------------------------------------------------------------------------------
OK
```

```
OK
2015    Giuoco Pianissimo       5       0       1       0.0     0.2
2015    Modern defence  3       2       0       0.6666666666666666      0.0
2015    Owen defence    1       0       1       0.0     1.0
2016    Giuoco Piano    13      6       3       0.46153846153846156     0.23076923076923078
2017    Catalan opening 8       1       1       0.125   0.125
2018    Pirc    4       0       3       0.0     0.75
2018    QGD semi-Slav   14      5       2       0.35714285714285715     0.14285714285714285
2018    Scandinavian defence    3       1       0       0.3333333333333333      0.0
2020            5       4       1       0.8     0.2
2020    French  90      29      5       0.32222222222222224     0.05555555555555555
2020    KP      10      4       2       0.4     0.2
2020    QGD     60      8       4       0.13333333333333333     0.06666666666666667
2020    Vienna  6       1       1       0.16666666666666666     0.16666666666666666
2021    Two knights defence     4       2       1       0.5     0.25
2022    Alekhine's defence      8       3       0       0.375   0.0
2022    Giuoco Piano    12      2       6       0.16666666666666666     0.5
2022    Vienna gambit   1       1       0       1.0     0.0
2012    Neo-Gruenfeld defence   2       2       0       1.0     0.0
2014    Blumenfeld counter-gambit accepted      1       1       0       1.0     0.0
2014    Catalan opening 10      2       1       0.2     0.1
2014    Scandinavian (centre counter) defence   5       1       2       0.2     0.4
2015    Neo-Gruenfeld defence   1       0       1       0.0     1.0
2017            4       2       1       0.5     0.25
2017    Old Benoni defence      1       0       0       0.0     0.0
2021    Bogo-Indian defence     1       0       1       0.0     1.0
2021    Damiano's defence       1       1       0       1.0     0.0
2021    Queen's Pawn    9       5       0       0.5555555555555556      0.0
2021    Scandinavian defence    6       3       1       0.5     0.16666666666666666
2022    Robatsch defence        6       2       0       0.3333333333333333      0.0
2013    Old Benoni defence      2       0       2       0.0     1.0
2015    "Neo-Gruenfeld  2       0       2       0.0     1.0
2015    Bogo-Indian defence     2       0       0       0.0     0.0
2015    Caro-Kann       24      7       7       0.2916666666666667      0.2916666666666667
```

[6] Data Processing Step 5: create table ‘effective_opening’

```
hive> CREATE TABLE effective_opening (
    >     Year INT,
    >     Opening STRING,
    >     TotalGames INT,
    >     WhiteWins INT,
    >     BlackWins INT,
    >     WhiteWinRate FLOAT,
    >     BlackWinRate FLOAT
    > );
OK
Time taken: 0.121 seconds
hive> INSERT INTO effective_opening
    > SELECT
    >     year(`Date`) AS Year,
    >     Opening,
    >     COUNT(*) AS TotalGames,
    >     SUM(CASE WHEN Result = '1-0' THEN 1 ELSE 0 END) AS WhiteWins,
    >     SUM(CASE WHEN Result = '0-1' THEN 1 ELSE 0 END) AS BlackWins,
    >     SUM(CASE WHEN Result = '1-0' THEN 1 ELSE 0 END) / COUNT(*) AS WhiteWinRate,
    >     SUM(CASE WHEN Result = '0-1' THEN 1 ELSE 0 END) / COUNT(*) AS BlackWinRate
    > FROM
    >     chess_data
    > WHERE
    >     year(`Date`) BETWEEN 2012 AND 2022
    > GROUP BY
    >     year(`Date`), Opening;
Query ID = shreya_ketkar2_20231219185444_2dbf26bd-a114-4acb-b6dd-f48f2cc920ec
Total jobs = 1
Launching Job 1 out of 1
```

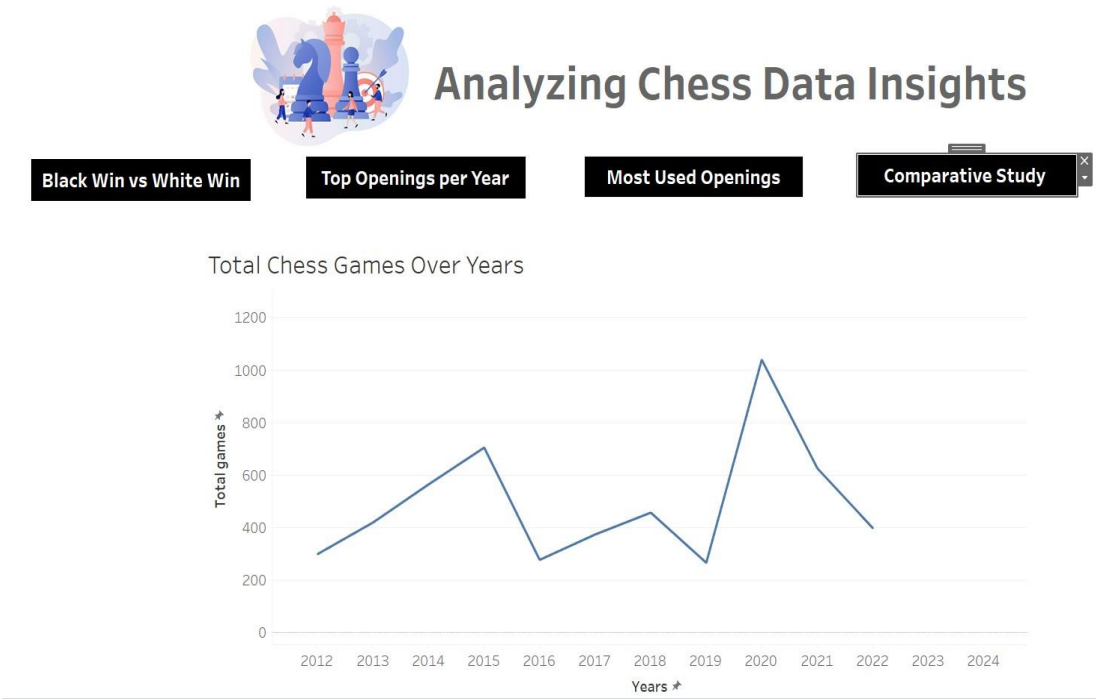
```
Query ID = shreya_ketkar2_20231219185444_2dbf26bd-a114-4acb-b6dd-f48f2cc920ec
Total jobs = 1
Launching Job 1 out of 1
Tez session was closed. Reopening...
Session re-established.
Session re-established.
Status: Running (Executing on YARN cluster with App id application_1702831931841_0034)

--------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
--------------------------------------------------------------------------------
Map 1 .......... container     SUCCEEDED      6          6        0        0       0       0
Reducer 2 ...... container     SUCCEEDED     54         54        0        0       0       0
Reducer 3 ...... container     SUCCEEDED      1          1        0        0       0       0
--------------------------------------------------------------------------------
VERTICES: 03/03  [==========================>>] 100%  ELAPSED TIME: 36.08 s
--------------------------------------------------------------------------------
Loading data to table default.effective_opening
OK
Time taken: 51.288 seconds
hive>
```

[7] Data Processing Step 6: Converting Hive output to CSV file

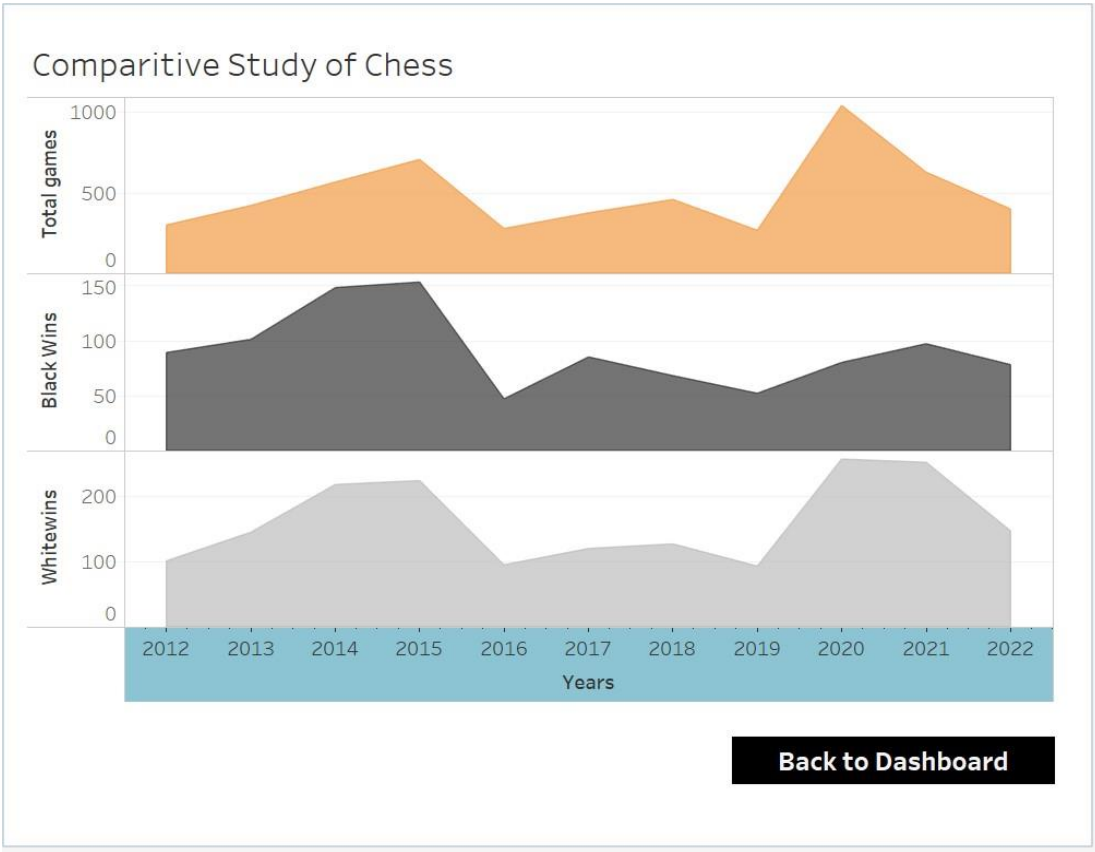
```
shreya_ketkar2@cluster2-m:~$ hadoop fs -cat hdfs://cluster2-m/user/hive/warehouse/effective_openings_export/* > ~/effective_openings.csv
shreya_ketkar2@cluster2-m:~$ ls
effective_openings.csv  pig_1702907788007.log
pig_1702907352406.log   top_openings.csv
```

```
shreya_ketkar2@cluster2-m:~$ hadoop fs -cat hdfs://cluster2-m/user/hive/warehouse/top_openings_export/* > ~/top_openings.csv
```

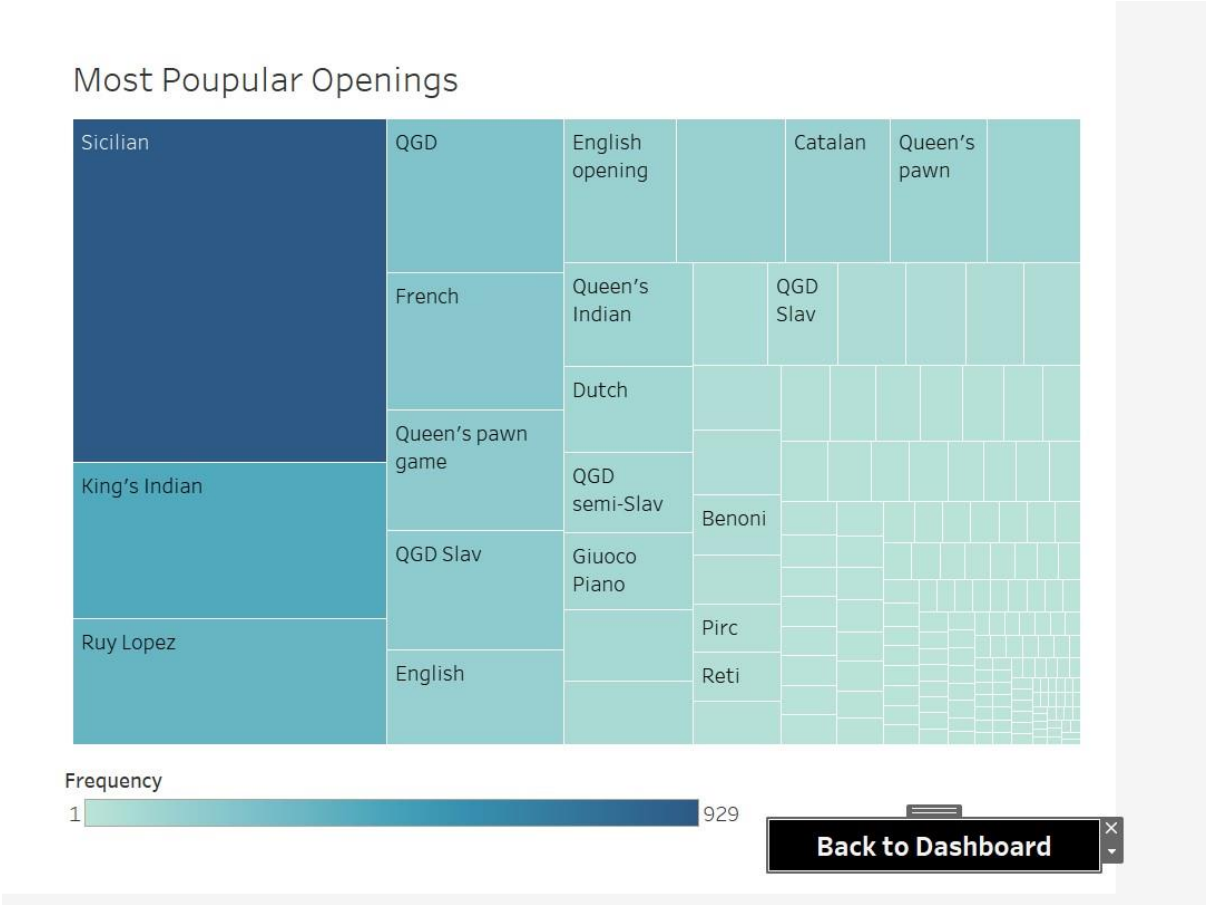
[8] Data Visualisation Using tableau – interactive front-end



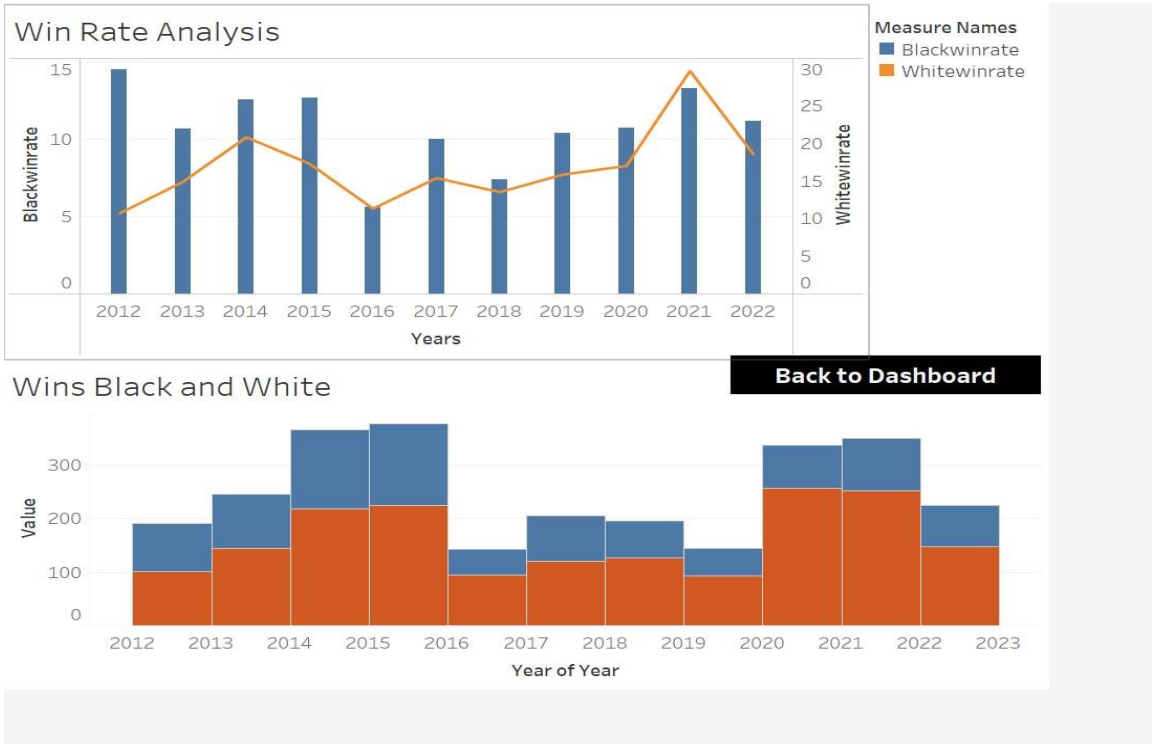
[9] Comparative study of total games, white wins and black wins



[10] Heatmap of most popular openings



[11] Win rate analysis of black and white chess players



[12] Top openings using bubble chart