**CPSC 531-02**

**ADVANCED DATABASE MANAGEMENT**

PROJECT

**ANALYSIS OF CRICKET MATCH**

**USING SPARK AND GOOGLE CLOUD PLATFORM (GCP)**

**TEAM MEMBERS**

**SREEKAR KAIRIKA (834199887)**

**THARUN KUMAR REDDY TUMMALA (819831322)**

**INTRODUCTION**

Cricket, a sport cherished by millions worldwide, has seen remarkable growth in its strategies and methods of analysis. The influx of big data has revolutionized the way matches are understood and interpreted, offering vast amounts of information, including player statistics, match occurrences, and environmental variables. This abundance of data has paved the way for deeper analysis and meaningful discoveries.

This study leverages the power of big data to uncover underlying patterns, trends, and performance metrics. Conventional approaches often fail to encapsulate the intricate details of cricket, such as strategic nuances and dynamic gameplay. By employing advanced data analytics, this research provides a holistic view of team operations, player performance, and game outcomes.

**PROJECT ABSTRACT**

Cricket, a sport cherished by millions worldwide, has seen remarkable growth in its strategies and methods of analysis. The influx of big data has revolutionized the way matches are understood and interpreted, offering vast amounts of information, including player statistics, match occurrences, and environmental variables. This abundance of data has paved the way for deeper analysis and meaningful discoveries.

This study leverages the power of big data to uncover underlying patterns, trends, and performance metrics. Conventional approaches often fail to encapsulate the intricate details of cricket, such as strategic nuances and dynamic gameplay. By employing advanced data analytics, this research provides a holistic view of team operations, player performance, and game outcomes.

**TECHNOLOGY USED**

* **Operating Systems:** Windows and macOS.
* **Tools & Platforms**: Google Cloud Platform (GCP) and Apache Spark.
* **Programming Language:** Python (leveraging PySpark, SparkML, and Panda’s libraries).
* **Development Environment:** Jupyter Notebook, providing an interactive and collaborative platform for experimentation and analysis.

**FUNCTIONALITIES**

**1) Dataset Overview**

**A screenshot of a computer

Description automatically generated**

Fig. Match Overview

**A screenshot of a computer

Description automatically generated**

Fig. Ball by Ball Analysis

This project involves the collection of match data from Kaggle, covering the period from 2002 to 2023. The dataset comprises numerous instances, providing a substantial base for analysis. The primary objective is to extract valuable insights for assisting teams and players in making data-driven decisions regarding player selection and venue preferences.

To better understand the dataset’s structure and attributes, the dataset.info () function was utilized. This function offers a concise summary of the dataset's characteristics, including:

1. The total number of rows (entries) within the dataset.
2. The count and names of all columns.
3. Non-null item counts for each column.
4. Data types of columns, such as text, datetime, float, or integer.
5. Additional information about memory usage.

By employing dataset.info (), a detailed overview of the data’s structure was obtained, enabling efficient preprocessing and analysis.

**2) Data Cleaning**

Data cleaning refers to the process of detecting and resolving errors or inconsistencies within a dataset to improve its reliability, consistency, and overall quality. The following approaches were applied to the project dataset to ensure optimal usability:

**a. Data Imputation – Missing values**

Identifying missing data within the dataset is a critical initial step in the data pre-processing phase. The process begins by pinpointing columns in the dataset that are missing essential information, such as the season or venue. One method for addressing these gaps is to eliminate rows containing incomplete data. While this approach ensures that the dataset retains only complete records, it may also reduce the dataset's overall size.

Alternatively, missing values can be addressed through imputation methods, such as mean or median imputation, which substitute the missing entries with the average or central value of the available data. This approach maintains the dataset's integrity while ensuring it is comprehensive for further analysis. The decision to remove or impute missing values depends on the volume and significance of the missing data and its potential impact on subsequent research and interpretations.

Addressing missing data demands a thoughtful approach to maintain the dataset's reliability for analysis. While removing incomplete entries is simple, it can result in losing potentially significant information. Conversely, imputation ensures the dataset remains intact by minimizing data loss. The chosen strategy should align with the research goals, balancing accuracy and meaningful interpretation.

**b. Removing Duplicates**

The dataset was thoroughly reviewed to identify and address duplicate entries. Particular attention was given to the season, batting team, and bowling team to maintain data accuracy. Each record was carefully examined, and any duplicates were identified and removed. This process ensured that every unique combination of season, batting team, and bowling team appeared only once in the dataset. Eliminating duplicate entries was crucial for upholding the reliability and precision of the data, as duplicates could lead to biases and skew analytical results. As a result, the dataset was confirmed to be free of duplicate information.

**c. Consistent Formatting**

Recognizing the importance of consistent dataset presentation, we prioritized standardizing the formatting of columns such as venue, player of the match, and seasons. Each entry was carefully reviewed to address any discrepancies in abbreviations or capitalization. This standardization process eliminated inconsistencies, ensuring uniformity across the dataset. By enhancing data clarity, this step was critical in enabling precise analysis. Through a collaborative effort, we meticulously refined the formatting as necessary, resulting in a dataset with consistently structured elements. This improvement made it significantly easier to analyze the data and derive meaningful insights.

Maintaining uniformity in the dataset was essential for achieving reliable analysis. Addressing variations in formatting helped prevent potential errors and improved the dataset's overall quality. Standardizing key columns ensured that the data was clear and easy to interpret. This process laid the foundation for extracting valuable insights efficiently and accurately.

**d. Outliner Detection and Handling**

We recognized the importance of identifying and addressing outliers within the dataset. Through careful analysis, we determined whether these unusual values were valid data points or errors. When necessary, we minimized the influence of outliers on analytical outcomes by either removing them or making appropriate adjustments. Our collaborative efforts ensured that the dataset was free from distortions caused by outliers, enabling the derivation of more accurate and dependable insights from the data.

**e. Data Validation and Type Correction**

Ensuring data validation and maintaining the dataset's integrity were our primary objectives. Collaboratively, we performed validation checks to uphold accuracy and consistency throughout the dataset. For example, we cross-checked data across relevant columns to detect any anomalies and verified that records fell within acceptable ranges. Additionally, we carefully reviewed column data types to ensure numerical values were appropriately stored as numeric formats rather than text or categorical variables. Our collective efforts in conducting these checks and adjusting data types as needed enhanced the dataset's reliability, thereby improving the quality and precision of future analyses and interpretations.

Applying various data cleaning techniques to the dataset enhances its dependability, consistency, and overall quality. This ensures that research and insights related to cricket are more precise and reliable.

**3) Data Analysis**

In our research, we developed reliable visualization models using PySpark, a cutting-edge technology. By leveraging Spark's distributed computing capabilities, we efficiently processed and analyzed large volumes of data from the dataset. Additionally, we plan to utilize PySpark's machine learning components to implement various regression techniques. These models enable teams to gain valuable insights for enhancing their overall performance and development.

**4) Google Cloud Platform**

To enhance our data analysis for the project, we utilized Google Cloud Platform (GCP), specifically leveraging Cloud Dataproc. GCP provided a scalable and powerful infrastructure that allowed us to efficiently process and analyze cricket match data. Using Cloud Dataproc, we created single-node and three-node clusters to handle large datasets effectively. These clusters, equipped with Apache Spark, enabled distributed data processing and analysis. By distributing the workload across multiple nodes, we achieved faster execution times and enhanced performance.

The use of GCP allowed us to handle complex data processing tasks with ease. Cloud Dataproc’s flexibility enabled seamless creation and management of clusters tailored to our requirements. Apache Spark’s integration facilitated efficient distributed computing, ensuring optimized resource utilization. This approach significantly enhanced our ability to process large-scale cricket match data with speed and accuracy.

**5) Data Visualization and Exploration**

In our project, we utilized Google Cloud Platform (GCP) to facilitate data visualization and analysis for cricket matches.

Here's how we implemented it:

1. We stored the match information and match data datasets by uploading them to Google Cloud Storage, simplifying dataset accessibility and management.
2. To develop our Spark project, we used GCP's Dataproc service and set up a cluster with a single master node. This distributed computing environment enabled efficient processing of large datasets.
3. Within the cluster, we created an interactive PySpark Jupyter notebook to enable collaborative data processing and visualization. Dataproc connected seamlessly with the Hadoop Distributed File System (HDFS), allowing the dataset to be read directly from cloud storage.
4. Using PySpark SQL, we executed queries on the dataset to extract key insights related to cricket analysis. The resulting query outputs were saved back to the cloud for convenient access and further analysis.

* By implementing these methods, we successfully analyzed the data and derived valuable insights. Leveraging GCP, Dataproc, and PySpark SQL, we established a robust platform for data visualization, enabling us to effectively showcase and interpret trends, patterns, and statistics in a scalable manner. This visualization capability deepened our understanding of the dataset and provided players and teams with actionable insights for informed decision-making.

**ARCHITECTURE AND DESIGN**

To ensure reliable and consistent data for further analysis, we utilized PySpark and Spark to clean and process the dataset for our project.

We began by installing Spark and setting up a Spark Session, which serves as a gateway for utilizing the Dataset and Data Frame API in Spark programming. During this process, we learned to create Data Frames, register them as tables, and query the data using SQL. Using the builder pattern method builder () and the getOrCreate () function, we successfully established a Spark session. The dataset, formatted as a CSV file, was then loaded into a Data Frame for further processing.

The data cleaning process, a crucial step in ensuring the dataset's accuracy and consistency, was performed using PySpark. After installing PySpark with the pip install pyspark command, we leveraged its tools to remove errors, duplicates, and unnecessary data from the dataset. By creating Data Frames, registering them as tables, and executing SQL queries, we effectively cleaned the data. Using this refined dataset, we conducted several analyses, including generating Data Frames and extracting insights through SQL queries.

For our analysis, the cleaned dataset was uploaded to a Google Cloud Storage bucket. This facilitated secure storage and retrieval, enabling efficient cloud-based data processing.

We utilized the capabilities of PySpark and Spark to clean and analyze the dataset. These technologies allowed us to reduce inconsistencies, ensure data quality, and extract valuable insights into cricket trends. By integrating Spark, PySpark, and cloud storage, we achieved fast and scalable analysis of large datasets, enhancing the decision-making process significantly.

**A screenshot of a computer

Description automatically generated**

Fig. Cluster Creation

**A screenshot of a computer

Description automatically generated**

Fig. Cluster Configuration

**A screenshot of a computer

Description automatically generated**

Fig. VM Instances

**A screenshot of a computer

Description automatically generated**

Fig. Bucket Creation

**DEPLOYMENT INSTRUCTIONS**

In our project, we followed several steps to establish the infrastructure and tools needed for effective data visualization and analysis.

Our data processing and analysis workflow was built on Apache Spark, which we downloaded and installed from its official website. This distributed computing framework enabled us to efficiently manage large datasets.

Next, we installed the Python components and configured Jupyter Notebook to create an interactive and user-friendly development environment. This interface facilitated seamless integration of Spark and Python, simplifying data handling and analysis.

To leverage cloud computing, we set up a cluster in Google Cloud Platform (GCP) using Dataproc. This provided a scalable and efficient computing system to process our dataset.

Before beginning the analysis, we installed the required libraries and dependencies to support our code. Additionally, we created a Google Cloud Storage bucket to securely store the dataset.

Finally, we explored the data and uncovered valuable insights. Within the Jupyter Notebook environment, we executed SQL queries, analyzed data, and generated persistent visualizations for better understanding and presentation.

Throughout these stages, we built a robust foundation for data analysis and visualization, enabling us to make well-informed decisions and gain valuable insights.

**STEPS TO RUN**

**Step 1**: We launched the Google Cloud Platform (GCP) application to initiate our project. After gaining access to our GCP account, we set up a new project specifically designed for our study.

After successfully creating the project, we configured a virtual machine (VM) instance within the GCP framework. This VM instance served as our computational environment, providing the necessary resources and capabilities for data processing tasks. During the VM creation process, we specified various parameters, including the operating system, disk size, and machine type, tailored to meet the project's requirements and anticipated workload.

Once the VM instance was properly provisioned, we accessed it remotely, enabling us to install software, execute commands for data processing and analysis, and manage the virtual machine from a remote location.

**Step 2:** We adjusted the firewall settings and started the virtual machine instance in Google Cloud Platform (GCP) to ensure adequate network connectivity and security.

Initially, we navigated to the networking section within the GCP interface and located the firewall rules under the networking configuration. Necessary modifications were made to the firewall rules to meet the project’s requirements. After updating the rules, we initiated the boot-up process for the virtual machine instance. Once the instance became operational, we monitored its status to confirm a successful startup.

By configuring the firewall settings, we ensured that essential network access was granted while maintaining proper security measures. This step provided us with a fully functional computing environment to carry out our analytical tasks efficiently.

**Step 3:** We configured the development environment for our project by installing Jupyter Notebook and other required applications through an SSH terminal.

After successfully establishing an SSH connection to the virtual machine instance, we carried out the necessary steps to install Jupyter Notebook. This process involved using package managers such as pip or conda to install the latest version of Jupyter Notebook. Additionally, we installed supplementary packages and libraries essential for our data processing and analysis tasks, including NumPy, Pandas, Matplotlib, Seaborn, and other dependencies aligned with our project requirements.

To ensure the installations were completed successfully, we followed the package managers' instructions closely throughout the process. Installing Jupyter Notebook and the required applications within the virtual machine instance enabled us to create a dependable development environment. With the Jupyter Notebook interface, this setup facilitated efficient code development, execution, and collaboration, ensuring seamless integration.

Step 4: After setting up the virtual machine server, we launched Jupyter Notebook to execute our data visualization scripts. Using the SSH terminal, we entered the appropriate command, specifying the necessary port and additional settings, to start the Jupyter Notebook server. This enabled us to access the Jupyter Notebook interface through a web browser.

Once the interface was open, we navigated to the directory containing our data visualization code in a .py file. Jupyter Notebook provided an interactive and user-friendly platform for editing and running the code. By opening the .py file, we were able to generate meaningful plots, charts, and graphs using visualization libraries such as Matplotlib, Seaborn, or others.

Executing our data visualization scripts within Jupyter Notebook allowed us to create visual representations of the dataset. These visualizations helped us extract valuable insights and effectively communicate our findings.

**OUTPUT**

**A screenshot of a computer

Description automatically generated**

Fig. Cities in which match happened

**A screenshot of a computer

Description automatically generated**

Fig. Matches won by each team

**A screenshot of a computer

Description automatically generated**

Fig. Matches played in each city

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

Fig. Win Percentage based on Toss decision

**A graph with blue and white lines

Description automatically generated with medium confidence**

Fig. Team wins from 1990 to 2024

**A screenshot of a computer

Description automatically generated**

Fig. Top Player of Match Winners

**A screen shot of a graph

Description automatically generated**

Fig. Distribution of Matches Won by Run

**CHALLENGES**

We experienced various challenges during our project:

1. **Data quality and consistency:** By addressing issues such as missing data, inaccurate entries, and inconsistencies in formatting or labeling, we ensured the dataset's quality and uniformity. The data cleaning and verification process required meticulous attention to detail to uphold the study's reliability and accuracy.
2. **Managing enormous volumes of data:** The cricket dataset we worked on was extensive, encompassing data from multiple teams, venues, and seasons. Processing and analyzing such a large volume of data posed computational and resource challenges. To address these, we optimized our workflows and utilized technologies like Spark to enable efficient data processing.
3. **Data integration and compatibility**: The dataset comprised data from diverse sources and formats, necessitating data harmonization and integration. To ensure compatibility for analysis, we carefully converted, cleaned, and standardized the data, requiring meticulous handling and close attention to detail.
4. **Interpretation and communication**: Analyzing cricket data requires deriving meaningful insights and effectively conveying them to teams and players. To ensure our findings were relevant and impactful, we utilized data visualization tools and employed clear communication strategies to present complex analytical results in an understandable and actionable manner.

**CONCLUSION**

The primary objective of our project was to analyze cricket data and uncover trends in the sport using advanced technologies such as Spark, Google Cloud Platform, and machine learning algorithms. Through comprehensive data analysis and visualization, we extracted valuable insights from the dataset.

The outcomes of our initiative hold significant potential for improving player performance, strategic planning, and decision-making within the cricket industry. By implementing well-thought-out strategies, teams can enhance efficiency, make informed decisions, and reduce errors. This study illustrated how the integration of data analysis, machine learning, and data visualization can yield meaningful insights and reliable outcomes.

Future research efforts may focus on broadening the study's scope by incorporating additional data sources and exploring advanced techniques to refine planning and strengthen team dynamics. Overall, our experience highlighted the value of data-driven approaches in the cricket industry. By leveraging modern data analytics, we can pave the way for a future that is more efficient, profitable, and sustainable.

**FUTURE WORK**

The primary objective of our study was to leverage advanced analytics techniques to predict trends in cricket. Moving forward, several additional research areas could be explored to enhance our predictions and build on the findings.

To improve the accuracy of our forecasts, we could investigate machine learning models that go beyond traditional regression techniques. Advanced methods such as deep learning and ensemble approaches could capture complex patterns and nonlinear relationships within the dataset. This may also involve developing real-time analytics pipelines and incorporating streaming data to deliver up-to-date insights for decision-making.

Future research could expand to include sustainability considerations, integrating criteria that promote long-term benefits for the cricket industry. Upcoming efforts will primarily aim to refine our models, incorporate real-time data, address sustainability concerns, and encourage greater collaboration within the sports sector. By pursuing these directions, we can enhance our predictions, contribute to the growth of the cricket industry, and elevate the sport as a whole.