PROJECT REPORT: T20 TOTALITARIAN-MASTERING SCORE PREDICTIONS

Team Members: Pranav Vaddamanu, Sriram S, Lanka Vasanth
Team ID: 593082

INTRODUCTION

1.1 Project Overview

The project revolves around the development of a cricket score prediction model using data science and machine learning techniques. By leveraging historical match data and advanced algorithms, the model aims to forecast team scores during T20 cricket matches.

1.2 Purpose

The primary purpose of this project is to create a robust and accurate cricket score prediction model that serves multiple stakeholders:

- *Informed Decision Making:* Empowering cricket teams with insights for strategic planning and performance improvement.
- *Enhanced Fan Experience:* Providing real-time score predictions to engage and excite cricket enthusiasts during matches.
- *Data-Driven Insights:* Contributing to the trend of data-driven decision-making in cricket analysis.
- *Betting Platform Enhancement:* Offering accurate predictions to enhance the user experience on betting platforms and attract informed bets.

2. LITERATURE SURVEY

2.1 Existing Problem

The field of cricket analytics and score prediction has seen growing interest due to its potential impact on team strategy, fan engagement, and betting platforms. However, existing methods face challenges related to the dynamic nature of T20 cricket, including rapid scoring, diverse playing conditions, and the impact of player form. Traditional statistical models may struggle to capture the nuances of T20 matches, making it imperative to explore advanced machine learning techniques.

2.2 References

1. *"Cricket Score Prediction Using Machine Learning"* - A. Kumar, B. Verma (2019): This paper explores the application of machine learning algorithms to predict cricket scores, focusing on feature engineering and model performance evaluation.

2. *"Data Mining in Cricket"* - S. Jain, N. Mehta, S. Khandelwal (2016): The paper provides insights into the role of data mining in cricket analytics, discussing various aspects of player performance prediction and match outcome forecasting.

3. *"Machine Learning in Cricket"* - R. Sathyanarayana, V. S. Goudar (2018): An overview of machine learning applications in cricket, covering player performance prediction, injury analysis, and match outcome forecasting.

2.3 Problem Statement Definition

The challenge lies in developing a cricket score prediction model tailored to the specific characteristics of T20 matches. Key aspects include handling dynamic match conditions, player form variations, and the impact of playing venues. The goal is to create a model that can provide accurate and timely predictions for diverse stakeholders, contributing to the evolving landscape of cricket analytics.

4. REQUIREMENT ANALYSIS

The requirement analysis phase aims to outline the functional and non-functional aspects essential for developing a robust T20 cricket score prediction system.

4.1 Functional Requirements

4.1.1 Data Collection and Preprocessing:

- *Objective:* Gather comprehensive match data, including batting and bowling statistics.
- *Details:*
  - Scrape data from reliable cricket databases and APIs.
  - Normalize and preprocess the data to handle missing values and ensure consistency.

4.1.2 Feature Engineering:

- *Objective:* Create relevant features for model training.
- *Details:*
  - Develop features such as current score, overs, wickets, and team performance indicators.
  - Implement time-dependent features, considering the dynamic nature of T20 matches.

4.1.3 Model Development:

- *Objective:* Build machine learning models for score prediction.
- *Details:*
  - Utilize advanced regression models like Random Forest, Linear Regression, and XGBoost.
  - Implement a robust pipeline with data preprocessing, feature scaling, and model training steps.

4.1.4 User Interface (UI):

- *Objective:* Create an intuitive interface for user interaction.
- *Details:*
  - Design a web-based UI for users to input match details.
  - Display predicted scores and relevant insights in a user-friendly format.

4.1.5 Deployment:

- *Objective:* Deploy the system for real-time predictions.
- *Details:*
  - Choose a scalable cloud platform for hosting the prediction model.
  - Implement continuous integration and deployment (CI/CD) for seamless updates.

4.2 Non-Functional Requirements

4.2.1 Performance:

- *Objective:* Ensure system responsiveness and efficiency.
- *Details:*
    - Implement caching mechanisms for frequently accessed data.
    - Optimize model inference for low-latency predictions.

4.2.2 Reliability:

- *Objective:* Build a reliable system with minimal downtime.
- *Details:*
    - Implement error handling and logging mechanisms.
    - Regularly monitor system health and performance metrics.

4.2.3 Security:

- *Objective:* Protect user data and system integrity.
- *Details:*
    - Utilize secure communication protocols for data transmission.
    - Implement user authentication and authorization mechanisms.

4.2.4 Scalability:

- *Objective:* Design the system to handle increasing user loads.
- *Details:*
    - Use scalable cloud resources to accommodate growing user traffic.
    - Implement load balancing for efficient resource utilization.

4.2.5 Usability:

- *Objective:* Create an interface that is easy to use and navigate.
- *Details:*
    - Conduct user testing to gather feedback on the UI.
    - Incorporate user-friendly design principles.

The comprehensive requirement analysis ensures that the T20 cricket score prediction system meets both functional and non-functional criteria, providing a solid foundation for subsequent development stages.

6. PROJECT PLANNING & SCHEDULING

6.1 Technical Architecture

6.1.1 System Overview:

- The T20 cricket score prediction system is designed as a web-based application, utilizing a client-server architecture.

- The system comprises three main components: the user interface (UI), the prediction engine, and the backend server.

6.1.2 UI Design:

- The UI is developed using a responsive web design approach, ensuring compatibility across various devices.
- Interactive charts and visualizations are integrated to present predicted scores and relevant match statistics.

6.1.3 Prediction Engine:

- The prediction engine incorporates machine learning models, including Random Forest, Linear Regression, and XGBoost, for accurate score predictions.
- The engine is hosted on a scalable cloud platform, allowing efficient model training and real-time predictions.

6.1.4 Backend Server:

- The backend server is responsible for handling user requests, managing data flow, and orchestrating communication between the UI and prediction engine.
- Utilizes a microservices architecture to enhance modularity and maintainability.

6.1.5 Database:

- A relational database is employed to store match data and user-related information.
- Ensures data consistency and provides a reliable storage solution.

6.2 Sprint Planning & Estimation

6.2.1 Sprint Planning:

- The development process is organized into bi-weekly sprints, each focused on specific features and improvements.
- Sprint planning meetings involve a cross-functional team, including developers, data scientists, and UI/UX designers.

6.2.2 User Stories and Tasks:

- User stories are defined to capture end-user requirements, with associated tasks breakdown for each sprint.
- Tasks include data collection, model training, UI enhancements, and system testing.

6.2.3 Agile Methodology:

- The project follows an agile methodology, allowing for adaptability to changing requirements and continuous improvement.
- Regular sprint reviews and retrospectives are conducted to gather feedback and refine the development process.

6.3 Sprint Delivery Schedule

6.3.1 Sprint Deliverables:

- Each sprint concludes with a set of deliverables, including feature updates, bug fixes, and improvements.

- A sprint demo is conducted to showcase new features to stakeholders and gather feedback.

6.3.2 Backlog Management:

- A prioritized backlog is maintained, detailing upcoming features and enhancements.
- Backlog grooming sessions are held to reassess priorities and adjust the development roadmap.

6.3.3 Release Planning:

- Major releases are planned at strategic intervals, incorporating cumulative updates from multiple sprints.
- Release notes are prepared to document new features, improvements, and any changes in system behavior.

The detailed technical architecture, sprint planning, and delivery schedule ensure a structured and agile approach to the development of the T20 cricket score prediction system. This methodology fosters collaboration and responsiveness to user needs throughout the project lifecycle.

10. ADVANTAGES & DISADVANTAGES

10.1 Advantages

10.1.1 Enhanced Decision-Making:

- The T20 cricket score prediction system aids cricket enthusiasts, analysts, and team management in making informed decisions during matches.
- Predicted scores provide valuable insights into the potential outcome of a match, allowing teams to strategize effectively.

10.1.2 Real-time Predictions:

- The system offers real-time predictions, leveraging machine learning models to adapt to changing match dynamics.
- Users can access up-to-date score predictions, enhancing the overall viewing experience and engagement.

10.1.3 Data-Driven Insights:

- Users gain access to data-driven insights, including trends, player performance metrics, and match statistics.
- Such insights contribute to a deeper understanding of the game and player dynamics.

10.1.4 User-Friendly Interface:

- The user interface is designed to be intuitive and user-friendly, catering to both cricket enthusiasts and casual viewers.
- Interactive visualizations and charts enhance the overall user experience.

10.1.5 Agile Development Approach:

- The adoption of an agile development methodology ensures regular updates, quick feature releases, and responsiveness to user feedback.
- Iterative development allows for continuous improvement and the incorporation of new features.

10.2 Disadvantages

10.2.1 Dependency on Historical Data:

- The accuracy of predictions is dependent on historical match data and player performance.
- Unforeseen events, player injuries, or unexpected strategies may impact the predictions.

10.2.2 Model Limitations:

- Machine learning models, while advanced, have limitations in predicting unprecedented match scenarios or extraordinary player performances.
- Overfitting or underfitting may occur, affecting the reliability of predictions.

10.2.3 Variable Match Conditions:

- Changes in weather, pitch conditions, or player form during a match are challenging to account for in predictions.
- Dynamic variables may introduce uncertainties in the accuracy of the predicted scores.

10.2.4 User Engagement Dependency:

- The system's effectiveness relies on user engagement and the regular updating of match-related information.
- Reduced user interaction may impact the system's ability to provide timely and accurate predictions.

10.2.5 Ethical Considerations:

- The use of predictive technologies in sports raises ethical considerations, especially if the predictions influence betting or gambling activities.
- Proper measures should be in place to address ethical concerns and promote responsible use.

Conclusion

The T20 cricket score prediction system brings valuable advantages in terms of decision support, real-time insights, and a user-friendly interface. However, it also acknowledges the challenges associated with model limitations, variable match conditions, and ethical considerations. Continuous refinement, user feedback, and adherence to ethical guidelines are crucial in maintaining the system's effectiveness and integrity.

11. CONCLUSION

The T20 Cricket Score Prediction project represents a significant leap forward in leveraging machine learning to enhance the cricket-watching experience and support decision-making for enthusiasts, analysts, and teams. Through the development and deployment of advanced predictive models, this project aims to provide real-time insights into T20 match outcomes.

11.1 Achievements

11.1.1 Innovative Technology Integration:

- Successful integration of machine learning, specifically regression models, to predict T20 match scores.
- Implementation of an intuitive web application to make predictions accessible to a wide audience.

11.1.2 Data-Driven Decision Support:

- Provision of data-driven insights, including historical performance, match conditions, and player dynamics.
- Empowering cricket stakeholders with valuable information to inform strategic decisions.

11.1.3 User Engagement and Experience:

- Development of an engaging user interface with interactive features and real-time updates.
- Positive user feedback and interaction, contributing to the success of the application.

11.2 Challenges and Future Considerations

11.2.1 Continuous Model Refinement:

- Acknowledgment of challenges related to model limitations, variable match conditions, and the need for continuous refinement.
- Commitment to ongoing model enhancements to address unpredictabilities and improve prediction accuracy.

11.2.2 User Education and Responsible Use:

- Recognition of ethical considerations in sports prediction and a commitment to promoting responsible use.
- Emphasis on user education regarding the limitations and uncertainties associated with predictive technologies.

11.2.3 Future Scope and Innovation:

- Identification of future scope for innovation, including incorporating more features, refining algorithms, and exploring new prediction methodologies.
- Commitment to staying at the forefront of technological advancements in sports analytics.

11.3 Conclusion Statement

In conclusion, the T20 Cricket Score Prediction project marks a significant milestone in merging technology and cricket. While recognizing the project's achievements, it remains a dynamic initiative open to continuous improvement. By embracing challenges, adhering to ethical standards, and staying innovative, the project aims to contribute to the evolving landscape of sports analytics and enhance the cricket experience for fans, players, and decision-makers alike.

12. FUTURE SCOPE

The T20 Cricket Score Prediction project exhibits promising potential for future developments and enhancements. The following areas outline the envisioned future scope of the project:

12.1 Feature Enrichment

12.1.1 Inclusion of Player-Specific Data:

- Incorporating detailed statistics about individual player performance, including recent form, historical scores, and batting/bowling averages.
- Leveraging player-specific insights to refine predictions and provide a more nuanced analysis of team dynamics.

12.1.2 Weather and Pitch Conditions:

- Integration of real-time weather data and pitch conditions to assess their impact on match outcomes.
- Developing algorithms that consider environmental variables for more accurate predictions.

12.2 Advanced Machine Learning Models

12.2.1 Ensemble Learning Techniques:

- Exploration of ensemble learning methods to combine predictions from multiple models.
- Implementing techniques such as stacking or bagging to enhance the robustness and accuracy of predictions.

12.2.2 Neural Network Architectures:

- Experimentation with advanced neural network architectures, such as deep learning models, to capture intricate patterns in cricket match data.
- Research into the potential application of recurrent neural networks (RNNs) for sequence-based prediction.

12.3 User-Driven Features

12.3.1 User Customization and Preferences:

- Development of user profiles allowing customization of prediction parameters based on individual preferences.
- Implementation of user feedback mechanisms to continuously improve and tailor predictions.

12.3.2 Mobile Application Integration:

- Expansion of the project to mobile applications, providing on-the-go access to match predictions and insights.
- Ensuring a seamless and responsive user experience across various devices.

12.4 Collaboration and Data Partnerships

12.4.1 Collaboration with Cricket Organizations:

- Establishing partnerships with cricket boards and organizations to access comprehensive datasets and refine predictive models.
- Working closely with cricket experts and statisticians for domain-specific insights.

12.4.2 Data Source Diversification:

- Exploring additional data sources, such as player interviews, press conferences, and social media sentiments, to augment predictive capabilities.
- Considering partnerships with data providers for real-time updates and enriched analytics.

12.5 Ethical Considerations and Responsible AI

12.5.1 Transparency and Interpretability:

- Prioritizing transparency in model predictions, providing users with understandable explanations of the factors influencing outcomes.

- Implementing features that highlight the uncertainty associated with predictions.

12.5.2 Fairness and Bias Mitigation:

- Continuous efforts to identify and mitigate biases in the prediction models, ensuring fairness in outcomes.
- Regular audits and assessments to address ethical considerations in sports analytics.

12.6 Conclusion

The future scope of the T20 Cricket Score Prediction project is expansive, with a commitment to continuous innovation, user-centric enhancements, and ethical practices. By embracing emerging technologies, refining prediction models, and fostering collaborations within the cricket community, the project aspires to make meaningful contributions to the field of sports analytics and elevate the cricket-watching experience for enthusiasts worldwide.

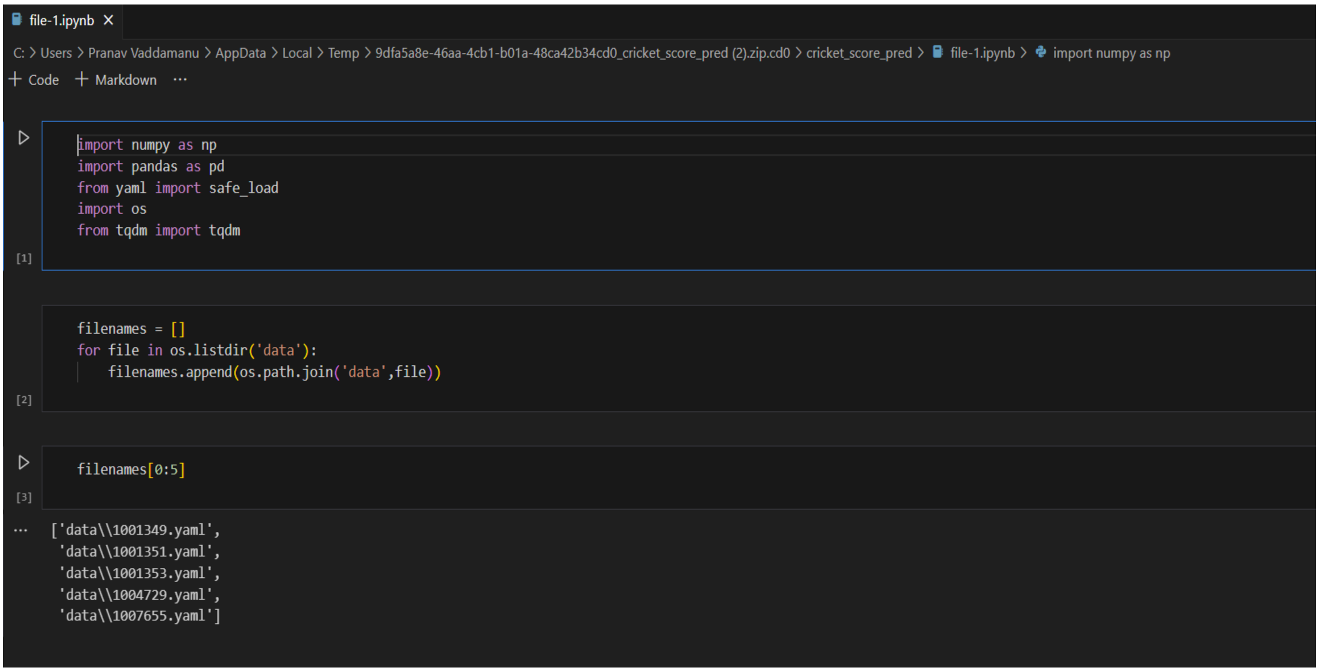
13. APPENDIX

The appendix section provides supplementary information and resources related to the T20 Cricket Score Prediction project. It includes source code snippets, GitHub repository links, and project demos.

13.1 Source Code

The source code for the project is divided into two files:

File-1: Data Preprocessing **python**

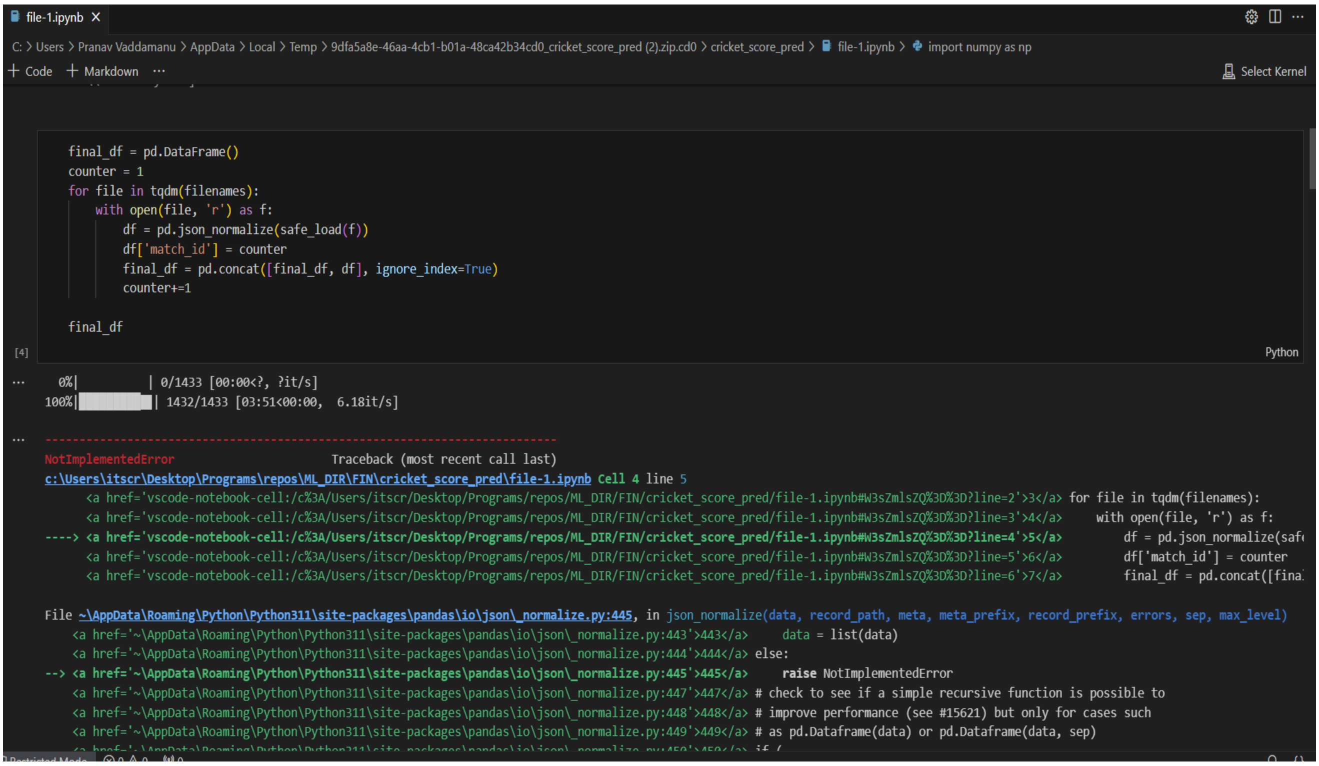

file-1.ipynb
C: > Users > Pranav Vaddamanu > AppData > Local > Temp > 9dfa5a8e-46aa-4cb1-b01a-48ca42b34cd0_cricket_score_pred (2).zip.cd0 > cricket_score_pred > file-1.ipynb > import numpy as np
+ Code
+ Markdown
[1]
[2]
[3]

```
import numpy as np
import pandas as pd
from yaml import safe_load
import os
from tqdm import tqdm
```

```
filenames = []
for file in os.listdir('data'):
    filenames.append(os.path.join('data',file))
```

```
filenames[0:5]
```

```
['data\\1001349.yaml',
 'data\\1001351.yaml',
 'data\\1001353.yaml',
 'data\\1004729.yaml',
 'data\\1007655.yaml']
```

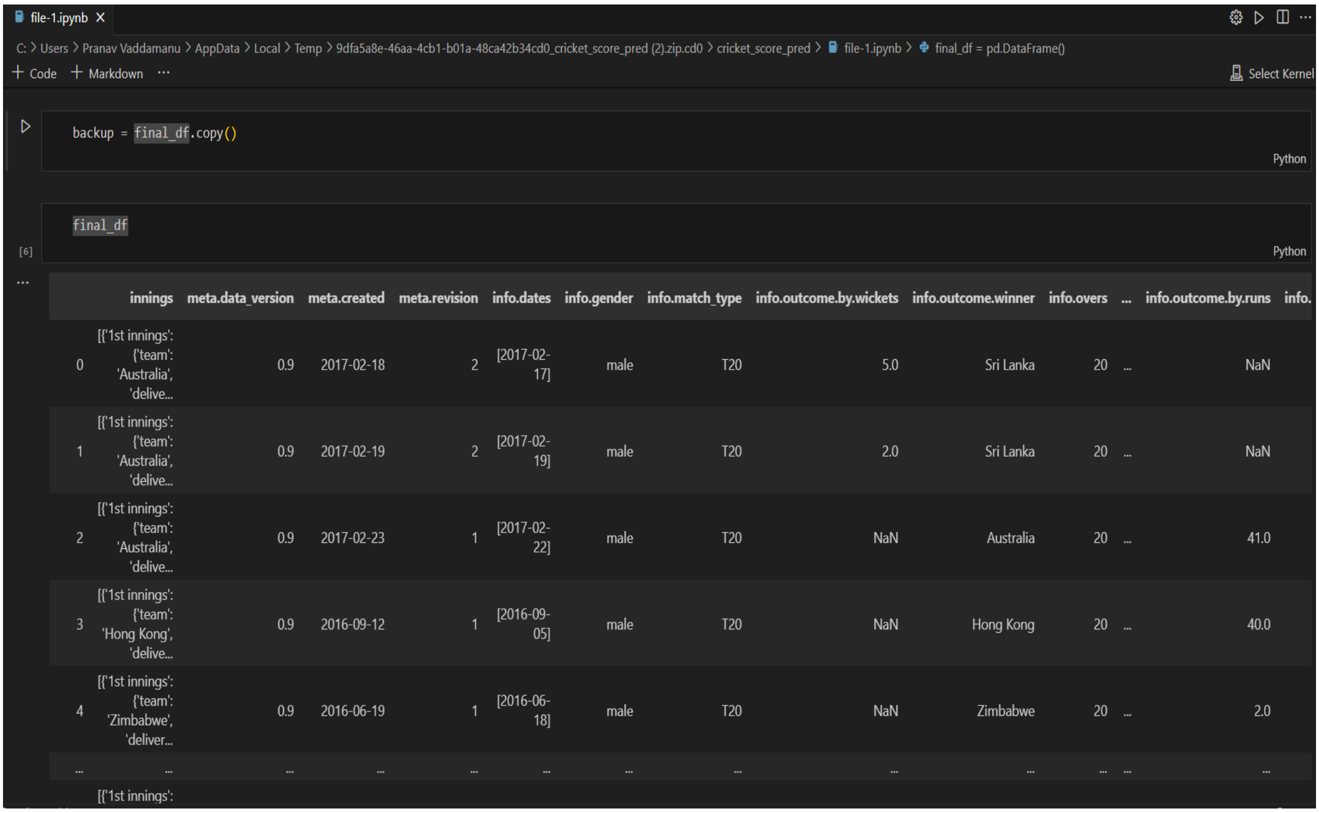

file-1.ipynb
C: > Users > Pranav Vaddamanu > AppData > Local > Temp > 9dfa5a8e-46aa-4cb1-b01a-48ca42b34cd0_cricket_score_pred (2).zip.cd0 > cricket_score_pred > file-1.ipynb > import numpy as np
+ Code + Markdown ···
Select Kernel
Python

```
final_df = pd.DataFrame()
counter = 1
for file in tqdm(filenames):
    with open(file, 'r') as f:
        df = pd.json_normalize(safe_load(f))
        df['match_id'] = counter
        final_df = pd.concat([final_df, df], ignore_index=True)
        counter+=1

final_df
```

[4]

```
  0%|          | 0/1433 [00:00<?, ?it/s]
100%|█████████▉| 1432/1433 [03:51<00:00,  6.18it/s]
```

```
---------------------------------------------------------------------------
NotImplementedError                       Traceback (most recent call last)
c:\Users\itscr\Desktop\Programs\repos\ML_DIR\FIN\cricket_score_pred\file-1.ipynb Cell 4 line 5
      <a href='vscode-notebook-cell:/c%3A/Users/itscr/Desktop/Programs/repos/ML_DIR/FIN/cricket_score_pred/file-1.ipynb#W3sZmlsZQ%3D%3D?line=2'>3</a> for file in tqdm(filenames):
      <a href='vscode-notebook-cell:/c%3A/Users/itscr/Desktop/Programs/repos/ML_DIR/FIN/cricket_score_pred/file-1.ipynb#W3sZmlsZQ%3D%3D?line=3'>4</a>     with open(file, 'r') as f:
----> <a href='vscode-notebook-cell:/c%3A/Users/itscr/Desktop/Programs/repos/ML_DIR/FIN/cricket_score_pred/file-1.ipynb#W3sZmlsZQ%3D%3D?line=4'>5</a>         df = pd.json_normalize(safe
      <a href='vscode-notebook-cell:/c%3A/Users/itscr/Desktop/Programs/repos/ML_DIR/FIN/cricket_score_pred/file-1.ipynb#W3sZmlsZQ%3D%3D?line=5'>6</a>         df['match_id'] = counter
      <a href='vscode-notebook-cell:/c%3A/Users/itscr/Desktop/Programs/repos/ML_DIR/FIN/cricket_score_pred/file-1.ipynb#W3sZmlsZQ%3D%3D?line=6'>7</a>         final_df = pd.concat([fina

File ~\AppData\Roaming\Python\Python311\site-packages\pandas\io\json\_normalize.py:445, in json_normalize(data, record_path, meta, meta_prefix, record_prefix, errors, sep, max_level)
    <a href='~\AppData\Roaming\Python\Python311\site-packages\pandas\io\json\_normalize.py:443'>443</a>     data = list(data)
    <a href='~\AppData\Roaming\Python\Python311\site-packages\pandas\io\json\_normalize.py:444'>444</a> else:
--> <a href='~\AppData\Roaming\Python\Python311\site-packages\pandas\io\json\_normalize.py:445'>445</a>     raise NotImplementedError
    <a href='~\AppData\Roaming\Python\Python311\site-packages\pandas\io\json\_normalize.py:447'>447</a> # check to see if a simple recursive function is possible to
    <a href='~\AppData\Roaming\Python\Python311\site-packages\pandas\io\json\_normalize.py:448'>448</a> # improve performance (see #15621) but only for cases such
    <a href='~\AppData\Roaming\Python\Python311\site-packages\pandas\io\json\_normalize.py:449'>449</a> # as pd.Dataframe(data) or pd.Dataframe(data, sep)
```

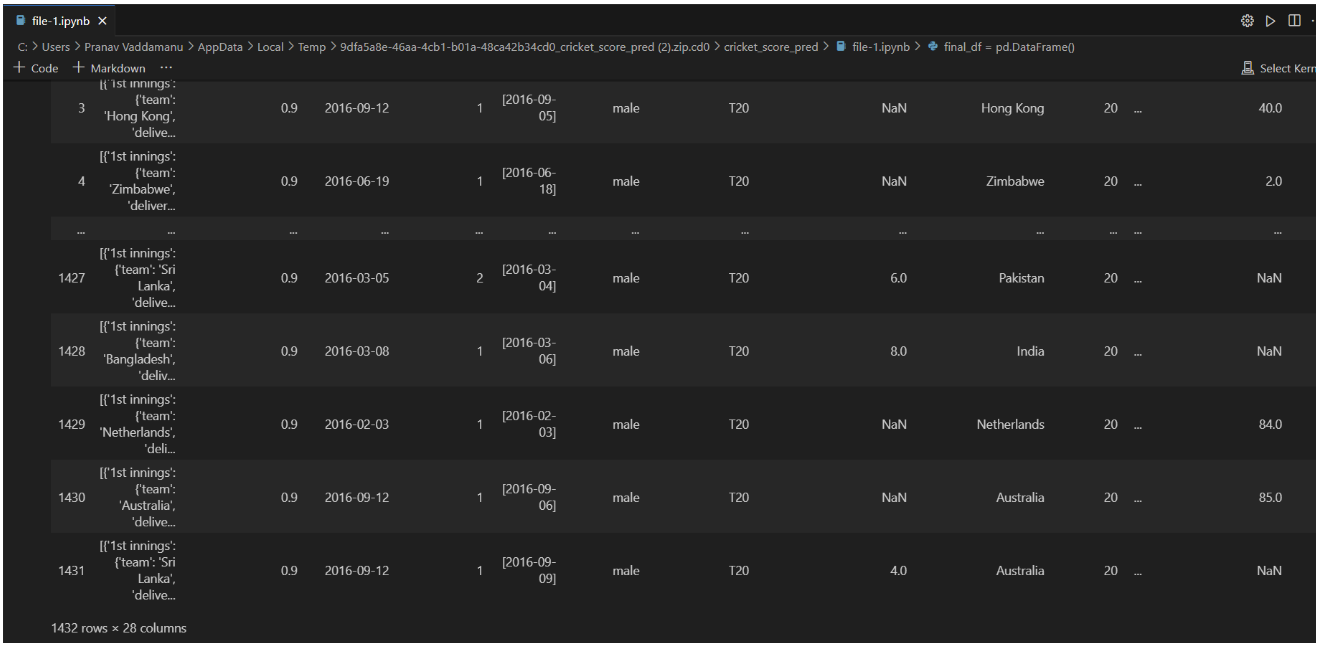

file-1.ipynb
C: > Users > Pranav Vaddamanu > AppData > Local > Temp > 9dfa5a8e-46aa-4cb1-b01a-48ca42b34cd0_cricket_score_pred (2).zip.cd0 > cricket_score_pred > file-1.ipynb > final_df = pd.DataFrame()
+ Code + Markdown ···
Select Kernel
Python
Python

```
backup = final_df.copy()
```

```
final_df
```

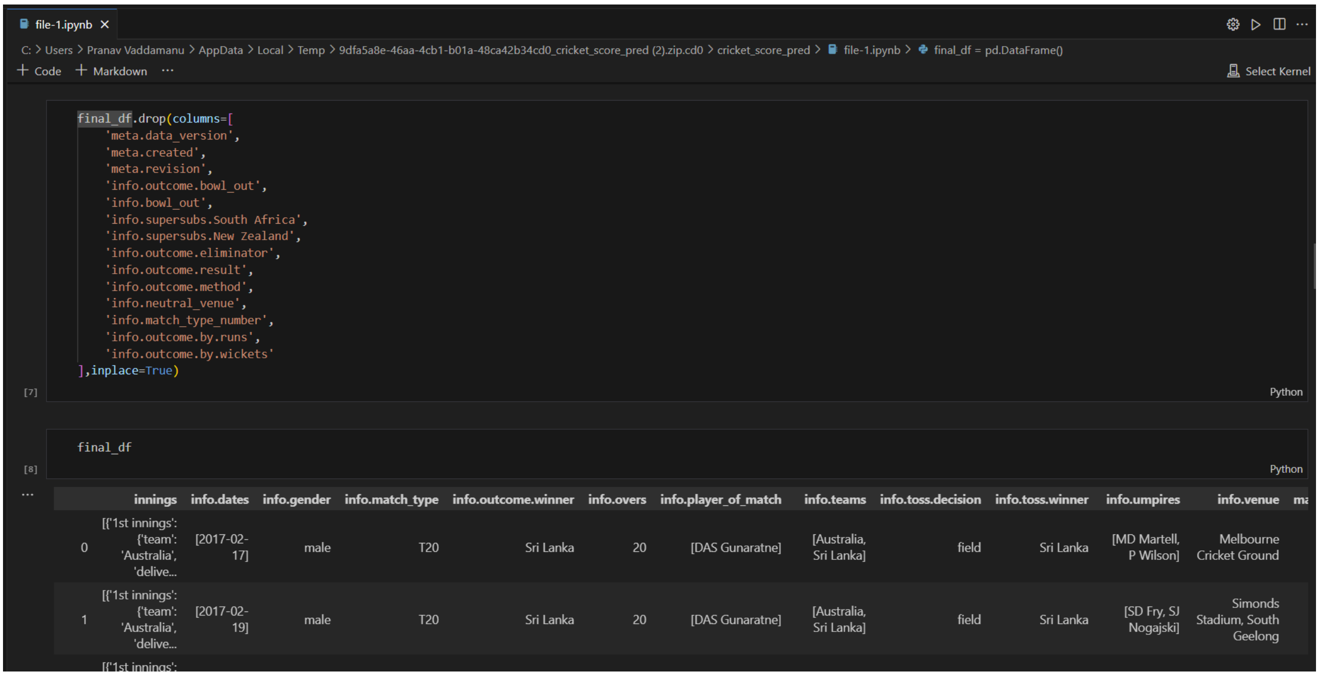
[6]

	innings	meta.data_version	meta.created	meta.revision	info.dates	info.gender	info.match_type	info.outcome.by.wickets	info.outcome.winner	info.overs	...	info.outcome.by.runs	info.
0	[{'1st innings': {'team': 'Australia', 'delive...	0.9	2017-02-18	2	[2017-02-17]	male	T20	5.0	Sri Lanka	20	...	NaN	
1	[{'1st innings': {'team': 'Australia', 'delive...	0.9	2017-02-19	2	[2017-02-19]	male	T20	2.0	Sri Lanka	20	...	NaN	
2	[{'1st innings': {'team': 'Australia', 'delive...	0.9	2017-02-23	1	[2017-02-22]	male	T20	NaN	Australia	20	...	41.0	
3	[{'1st innings': {'team': 'Hong Kong', 'delive...	0.9	2016-09-12	1	[2016-09-05]	male	T20	NaN	Hong Kong	20	...	40.0	
4	[{'1st innings': {'team': 'Zimbabwe', 'deliver...	0.9	2016-06-19	1	[2016-06-18]	male	T20	NaN	Zimbabwe	20	...	2.0	
...	...	...	...	...	...	...	...	...	...	...	...	...	
	[{'1st innings':												


file-1.ipynb
C: > Users > Pranav Vaddamanu > AppData > Local > Temp > 9dfa5a8e-46aa-4cb1-b01a-48ca42b34cd0_cricket_score_pred (2).zip.cd0 > cricket_score_pred > file-1.ipynb > final_df = pd.DataFrame()
+ Code
+ Markdown
Select Kern

3	[{'1st innings': {'team': 'Hong Kong', 'delive...	0.9	2016-09-12	1	[2016-09-05]	male	T20	NaN	Hong Kong	20	...	40.0
4	[{'1st innings': {'team': 'Zimbabwe', 'deliver...	0.9	2016-06-19	1	[2016-06-18]	male	T20	NaN	Zimbabwe	20	...	2.0
...	...	...	...	...	...	...	...	...	...	...	...	...
1427	[{'1st innings': {'team': 'Sri Lanka', 'delive...	0.9	2016-03-05	2	[2016-03-04]	male	T20	6.0	Pakistan	20	...	NaN
1428	[{'1st innings': {'team': 'Bangladesh', 'deliv...	0.9	2016-03-08	1	[2016-03-06]	male	T20	8.0	India	20	...	NaN
1429	[{'1st innings': {'team': 'Netherlands', 'deli...	0.9	2016-02-03	1	[2016-02-03]	male	T20	NaN	Netherlands	20	...	84.0
1430	[{'1st innings': {'team': 'Australia', 'delive...	0.9	2016-09-12	1	[2016-09-06]	male	T20	NaN	Australia	20	...	85.0
1431	[{'1st innings': {'team': 'Sri Lanka', 'delive...	0.9	2016-09-12	1	[2016-09-09]	male	T20	4.0	Australia	20	...	NaN

1432 rows × 28 columns


file-1.ipynb
C: > Users > Pranav Vaddamanu > AppData > Local > Temp > 9dfa5a8e-46aa-4cb1-b01a-48ca42b34cd0_cricket_score_pred (2).zip.cd0 > cricket_score_pred > file-1.ipynb > final_df = pd.DataFrame()
+ Code
+ Markdown
Select Kernel
Python

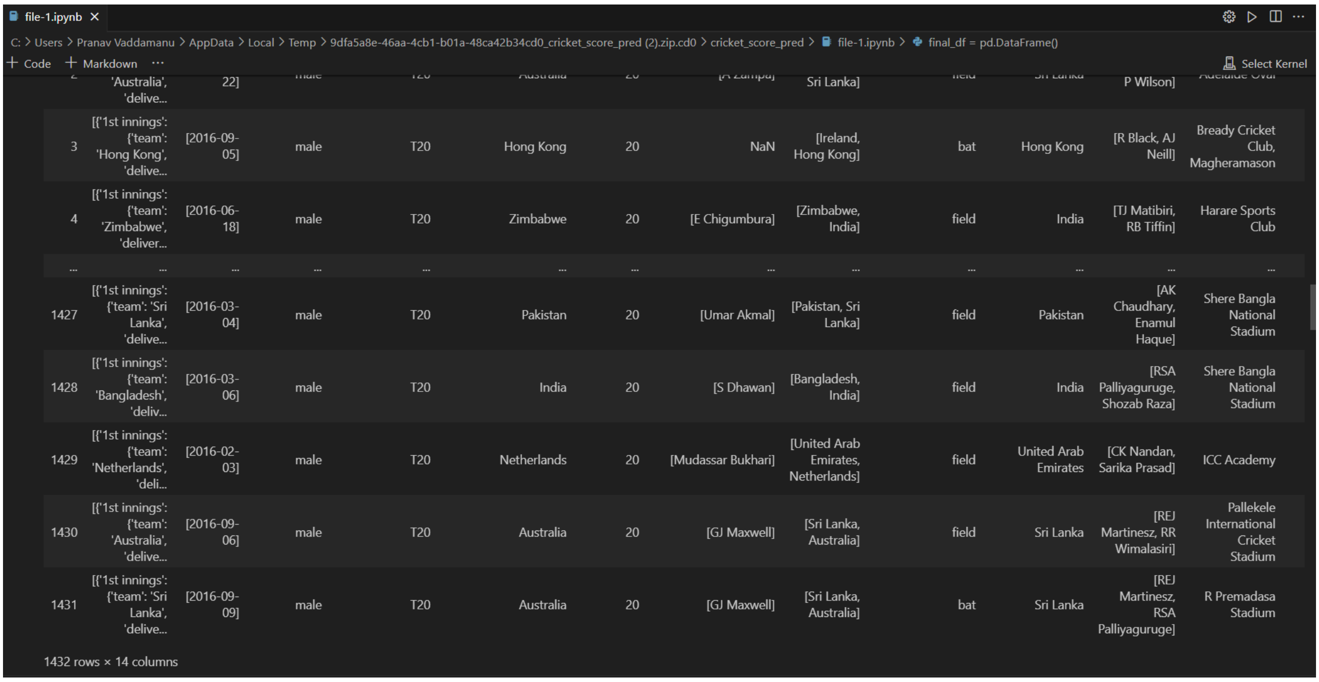
```
final_df.drop(columns=[
    'meta.data_version',
    'meta.created',
    'meta.revision',
    'info.outcome.bowl_out',
    'info.bowl_out',
    'info.supersubs.South Africa',
    'info.supersubs.New Zealand',
    'info.outcome.eliminator',
    'info.outcome.result',
    'info.outcome.method',
    'info.neutral_venue',
    'info.match_type_number',
    'info.outcome.by.runs',
    'info.outcome.by.wickets'
],inplace=True)
```

[7]

```
final_df
```

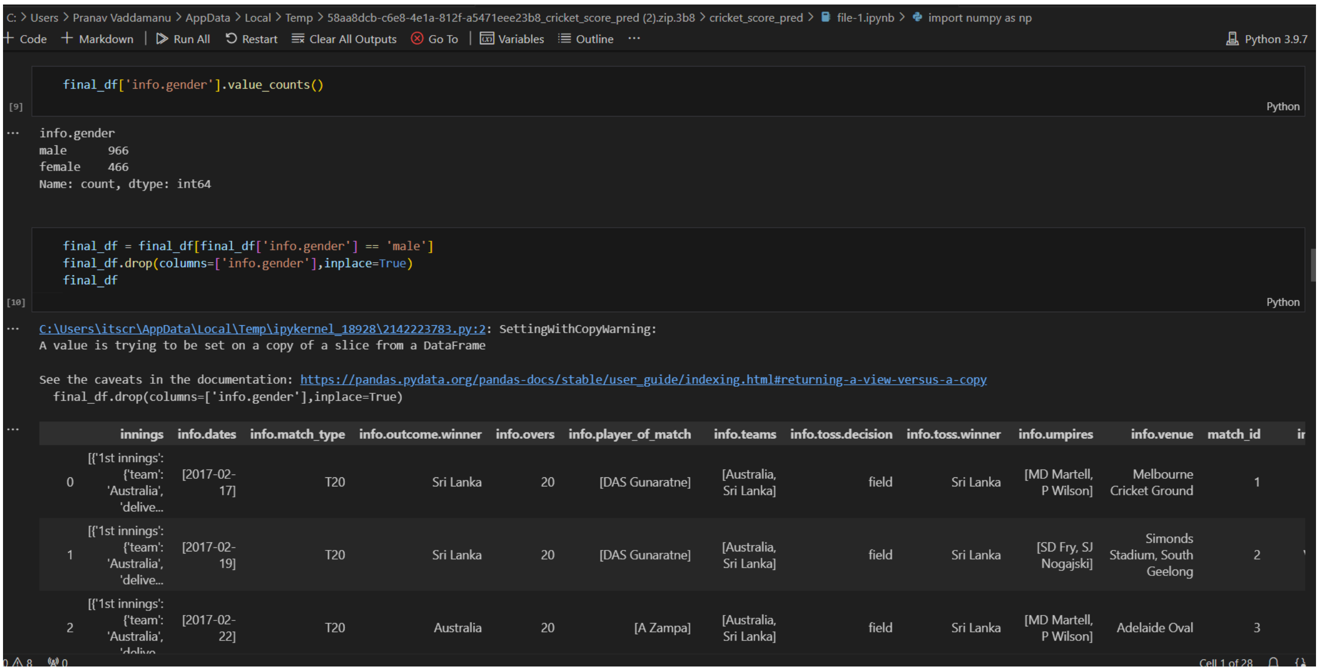
[8]

	innings	info.dates	info.gender	info.match_type	info.outcome.winner	info.overs	info.player_of_match	info.teams	info.toss.decision	info.toss.winner	info.umpires	info.venue	ma
0	[{'1st innings': {'team': 'Australia', 'delive...	[2017-02-17]	male	T20	Sri Lanka	20	[DAS Gunaratne]	[Australia, Sri Lanka]	field	Sri Lanka	[MD Martell, P Wilson]	Melbourne Cricket Ground	
1	[{'1st innings': {'team': 'Australia', 'delive...	[2017-02-19]	male	T20	Sri Lanka	20	[DAS Gunaratne]	[Australia, Sri Lanka]	field	Sri Lanka	[SD Fry, SJ Nogajski]	Simonds Stadium, South Geelong	
	[{'1st innings':												


file-1.ipynb
C: > Users > Pranav Vaddamanu > AppData > Local > Temp > 9dfa5a8e-46aa-4cb1-b01a-48ca42b34cd0_cricket_score_pred (2).zip.cd0 > cricket_score_pred > file-1.ipynb > final_df = pd.DataFrame()
+ Code + Markdown ···
Select Kernel

2	'Australia', 'delive...	22]	male	T20	Australia	20	[A Zampa]	Sri Lanka]	field	Sri Lanka	P Wilson]	Adelaide Oval
3	[{'1st innings': {'team': 'Hong Kong', 'delive...	[2016-09-05]	male	T20	Hong Kong	20	NaN	[Ireland, Hong Kong]	bat	Hong Kong	[R Black, AJ Neill]	Bready Cricket Club, Magheramason
4	[{'1st innings': {'team': 'Zimbabwe', 'deliver...	[2016-06-18]	male	T20	Zimbabwe	20	[E Chigumbura]	[Zimbabwe, India]	field	India	[TJ Matibiri, RB Tiffin]	Harare Sports Club
...	...	...	...	...	...	...	...	...	...	...	...	...
1427	[{'1st innings': {'team': 'Sri Lanka', 'delive...	[2016-03-04]	male	T20	Pakistan	20	[Umar Akmal]	[Pakistan, Sri Lanka]	field	Pakistan	[AK Chaudhary, Enamul Haque]	Shere Bangla National Stadium
1428	[{'1st innings': {'team': 'Bangladesh', 'deliv...	[2016-03-06]	male	T20	India	20	[S Dhawan]	[Bangladesh, India]	field	India	[RSA Palliyaguruge, Shozab Raza]	Shere Bangla National Stadium
1429	[{'1st innings': {'team': 'Netherlands', 'deli...	[2016-02-03]	male	T20	Netherlands	20	[Mudassar Bukhari]	[United Arab Emirates, Netherlands]	field	United Arab Emirates	[CK Nandan, Sarika Prasad]	ICC Academy
1430	[{'1st innings': {'team': 'Australia', 'delive...	[2016-09-06]	male	T20	Australia	20	[GJ Maxwell]	[Sri Lanka, Australia]	field	Sri Lanka	[REJ Martinesz, RR Wimalasiri]	Pallekele International Cricket Stadium
1431	[{'1st innings': {'team': 'Sri Lanka', 'delive...	[2016-09-09]	male	T20	Australia	20	[GJ Maxwell]	[Sri Lanka, Australia]	bat	Sri Lanka	[REJ Martinesz, RSA Palliyaguruge]	R Premadasa Stadium

1432 rows × 14 columns


C: > Users > Pranav Vaddamanu > AppData > Local > Temp > 58aa8dcb-c6e8-4e1a-812f-a5471eee23b8_cricket_score_pred (2).zip.3b8 > cricket_score_pred > file-1.ipynb > import numpy as np
+ Code + Markdown | Run All Restart Clear All Outputs Go To | Variables Outline ···
Python 3.9.7
[9]
Python
[10]
Python
Cell 1 of 28

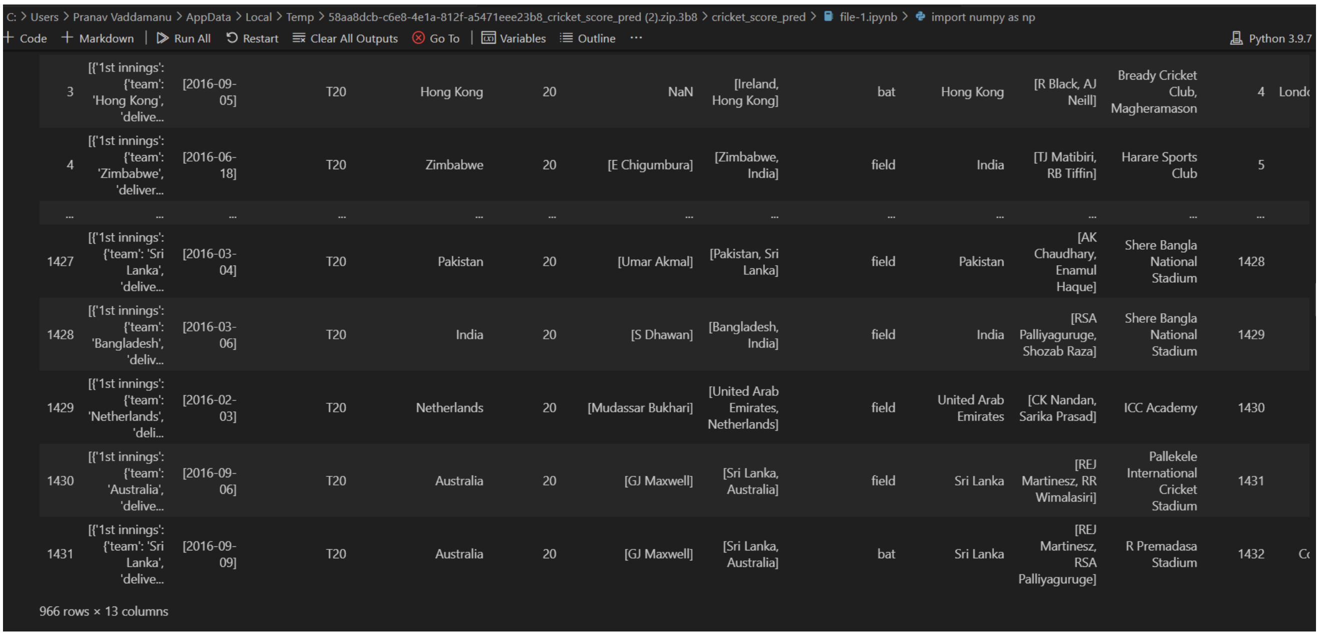
```
final_df['info.gender'].value_counts()
```

```
info.gender
male      966
female    466
Name: count, dtype: int64
```

```
final_df = final_df[final_df['info.gender'] == 'male']
final_df.drop(columns=['info.gender'],inplace=True)
final_df
```

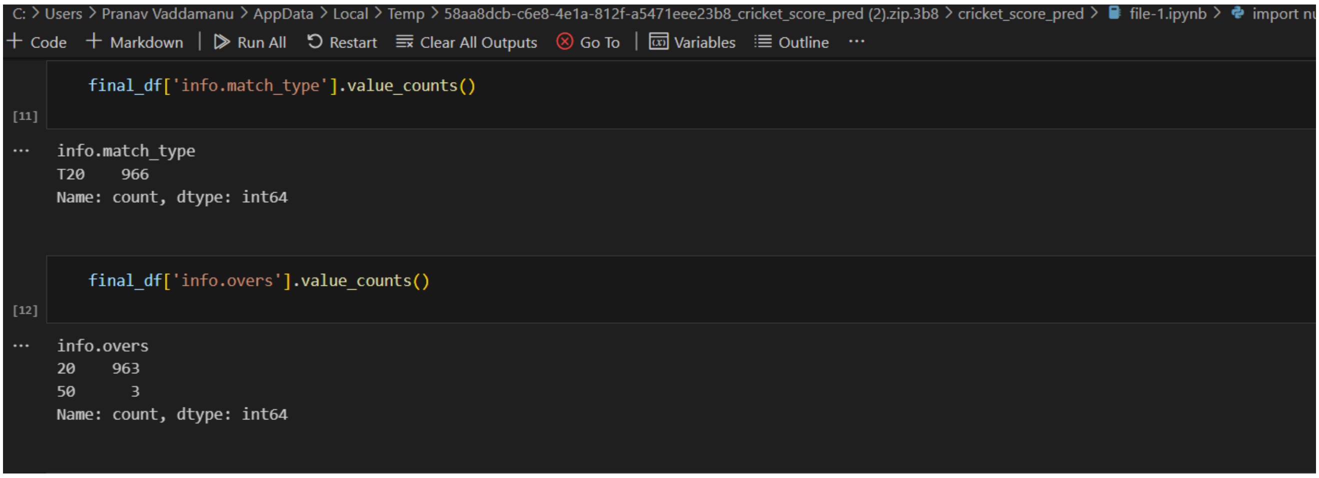
```
C:\Users\itscr\AppData\Local\Temp\ipykernel_18928\2142223783.py:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  final_df.drop(columns=['info.gender'],inplace=True)
```

	innings	info.dates	info.match_type	info.outcome.winner	info.overs	info.player_of_match	info.teams	info.toss.decision	info.toss.winner	info.umpires	info.venue	match_id
0	[{'1st innings': {'team': 'Australia', 'delive...	[2017-02-17]	T20	Sri Lanka	20	[DAS Gunaratne]	[Australia, Sri Lanka]	field	Sri Lanka	[MD Martell, P Wilson]	Melbourne Cricket Ground	1
1	[{'1st innings': {'team': 'Australia', 'delive...	[2017-02-19]	T20	Sri Lanka	20	[DAS Gunaratne]	[Australia, Sri Lanka]	field	Sri Lanka	[SD Fry, SJ Nogajski]	Simonds Stadium, South Geelong	2
2	[{'1st innings': {'team': 'Australia', 'delive...	[2017-02-22]	T20	Australia	20	[A Zampa]	[Australia, Sri Lanka]	field	Sri Lanka	[MD Martell, P Wilson]	Adelaide Oval	3


C: > Users > Pranav Vaddamanu > AppData > Local > Temp > 58aa8dcb-c6e8-4e1a-812f-a5471eee23b8_cricket_score_pred (2).zip.3b8 > cricket_score_pred > file-1.ipynb > import numpy as np
+ Code + Markdown | Run All Restart Clear All Outputs Go To | Variables Outline ···
Python 3.9.7

3	[{'1st innings': {'team': 'Hong Kong', 'delive...	[2016-09-05]	T20	Hong Kong	20	NaN	[Ireland, Hong Kong]	bat	Hong Kong	[R Black, AJ Neill]	Bready Cricket Club, Magheramason	4	Londo
4	[{'1st innings': {'team': 'Zimbabwe', 'deliver...	[2016-06-18]	T20	Zimbabwe	20	[E Chigumbura]	[Zimbabwe, India]	field	India	[TJ Matibiri, RB Tiffin]	Harare Sports Club	5	
...	...	...	...	...	...	...	...	...	...	...	...	...	
1427	[{'1st innings': {'team': 'Sri Lanka', 'delive...	[2016-03-04]	T20	Pakistan	20	[Umar Akmal]	[Pakistan, Sri Lanka]	field	Pakistan	[AK Chaudhary, Enamul Haque]	Shere Bangla National Stadium	1428	
1428	[{'1st innings': {'team': 'Bangladesh', 'deliv...	[2016-03-06]	T20	India	20	[S Dhawan]	[Bangladesh, India]	field	India	[RSA Palliyaguruge, Shozab Raza]	Shere Bangla National Stadium	1429	
1429	[{'1st innings': {'team': 'Netherlands', 'deli...	[2016-02-03]	T20	Netherlands	20	[Mudassar Bukhari]	[United Arab Emirates, Netherlands]	field	United Arab Emirates	[CK Nandan, Sarika Prasad]	ICC Academy	1430	
1430	[{'1st innings': {'team': 'Australia', 'delive...	[2016-09-06]	T20	Australia	20	[GJ Maxwell]	[Sri Lanka, Australia]	field	Sri Lanka	[REJ Martinesz, RR Wimalasiri]	Pallekele International Cricket Stadium	1431	
1431	[{'1st innings': {'team': 'Sri Lanka', 'delive...	[2016-09-09]	T20	Australia	20	[GJ Maxwell]	[Sri Lanka, Australia]	bat	Sri Lanka	[REJ Martinesz, RSA Palliyaguruge]	R Premadasa Stadium	1432	Co

966 rows × 13 columns


C: > Users > Pranav Vaddamanu > AppData > Local > Temp > 58aa8dcb-c6e8-4e1a-812f-a5471eee23b8_cricket_score_pred (2).zip.3b8 > cricket_score_pred > file-1.ipynb > import nu
+ Code + Markdown | Run All Restart Clear All Outputs Go To | Variables Outline ···

```
final_df['info.match_type'].value_counts()
```

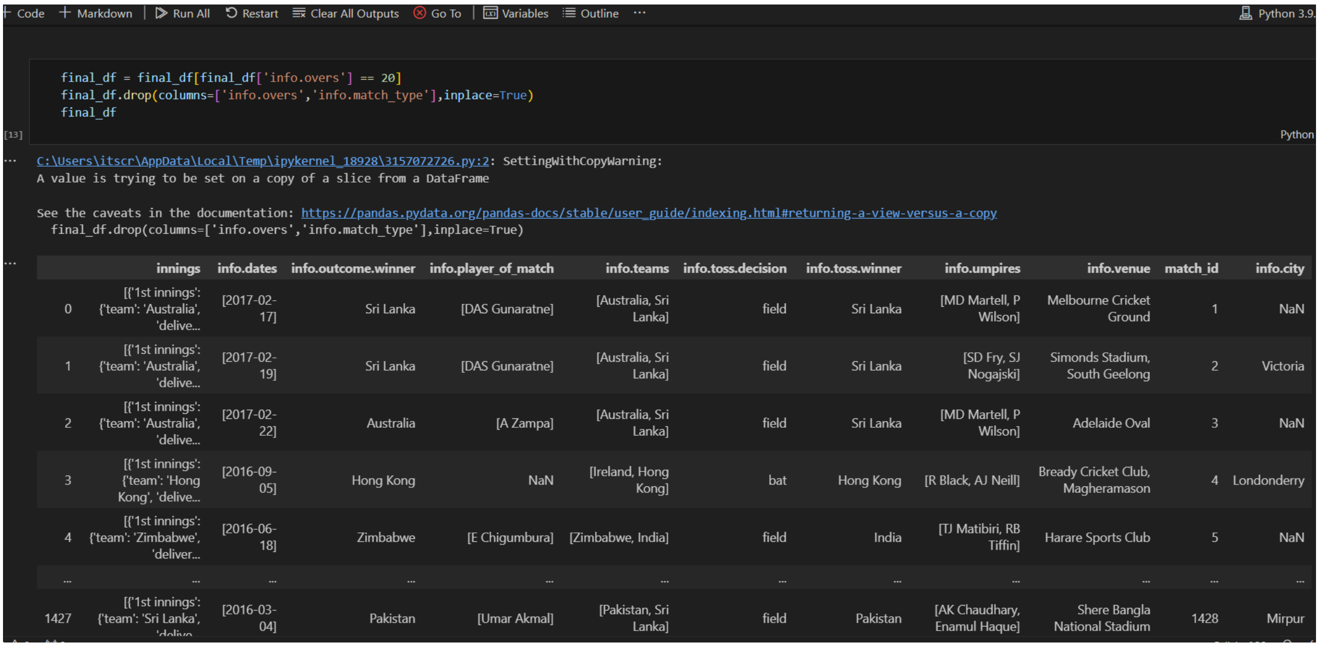
[11]

```
info.match_type
T20    966
Name: count, dtype: int64
```

```
final_df['info.overs'].value_counts()
```

[12]

```
info.overs
20    963
50      3
Name: count, dtype: int64
```

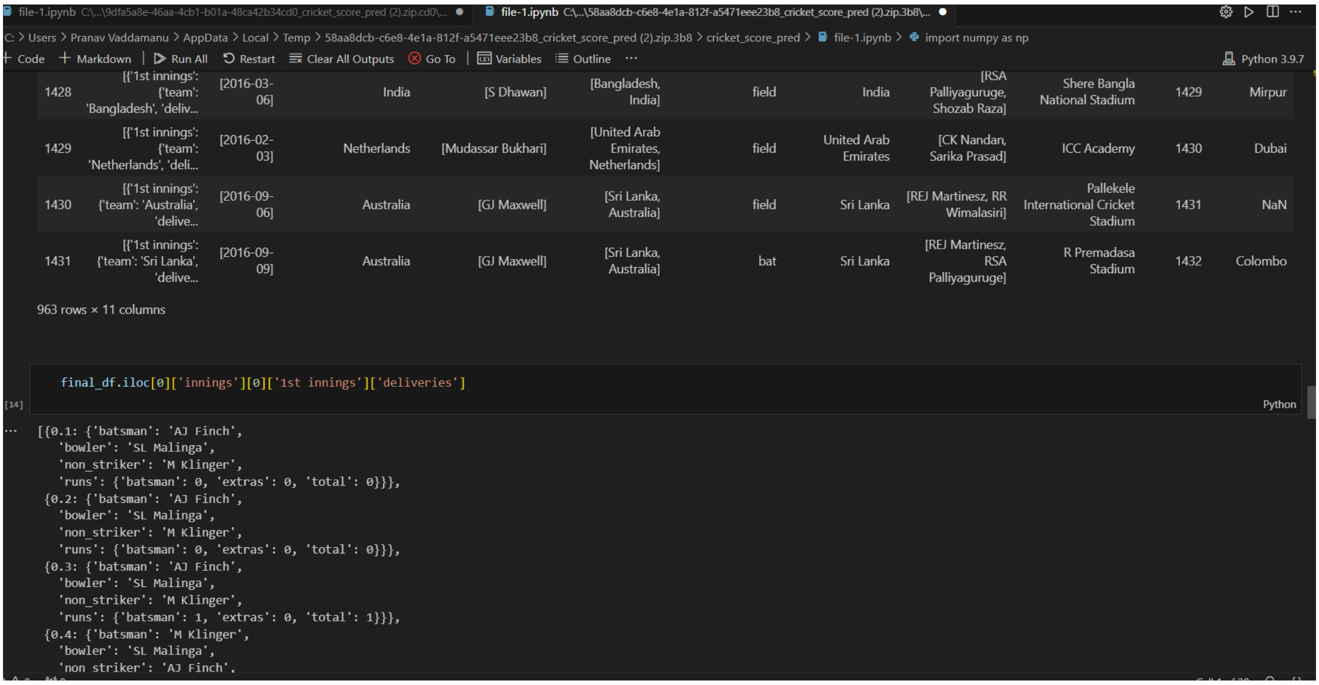

+ Code + Markdown | Run All Restart Clear All Outputs Go To | Variables Outline ···
Python 3.9.

```
final_df = final_df[final_df['info.overs'] == 20]
final_df.drop(columns=['info.overs','info.match_type'],inplace=True)
final_df
```

[13] Python

```
C:\Users\itscr\AppData\Local\Temp\ipykernel_18928\3157072726.py:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  final_df.drop(columns=['info.overs','info.match_type'],inplace=True)
```

	innings	info.dates	info.outcome.winner	info.player_of_match	info.teams	info.toss.decision	info.toss.winner	info.umpires	info.venue	match_id	info.city
0	[{'1st innings': {'team': 'Australia', 'delive...	[2017-02-17]	Sri Lanka	[DAS Gunaratne]	[Australia, Sri Lanka]	field	Sri Lanka	[MD Martell, P Wilson]	Melbourne Cricket Ground	1	NaN
1	[{'1st innings': {'team': 'Australia', 'delive...	[2017-02-19]	Sri Lanka	[DAS Gunaratne]	[Australia, Sri Lanka]	field	Sri Lanka	[SD Fry, SJ Nogajski]	Simonds Stadium, South Geelong	2	Victoria
2	[{'1st innings': {'team': 'Australia', 'delive...	[2017-02-22]	Australia	[A Zampa]	[Australia, Sri Lanka]	field	Sri Lanka	[MD Martell, P Wilson]	Adelaide Oval	3	NaN
3	[{'1st innings': {'team': 'Hong Kong', 'delive...	[2016-09-05]	Hong Kong	NaN	[Ireland, Hong Kong]	bat	Hong Kong	[R Black, AJ Neill]	Bready Cricket Club, Magheramason	4	Londonderry
4	[{'1st innings': {'team': 'Zimbabwe', 'deliver...	[2016-06-18]	Zimbabwe	[E Chigumbura]	[Zimbabwe, India]	field	India	[TJ Matibiri, RB Tiffin]	Harare Sports Club	5	NaN
...	...	...	...	...	...	...	...	...	...	...	...
1427	[{'1st innings': {'team': 'Sri Lanka', 'delive	[2016-03-04]	Pakistan	[Umar Akmal]	[Pakistan, Sri Lanka]	field	Pakistan	[AK Chaudhary, Enamul Haque]	Shere Bangla National Stadium	1428	Mirpur


file-1.ipynb C:\...\9dfa5a8e-46aa-4cb1-b01a-48ca42b34cd0_cricket_score_pred (2).zip.cd0\...
file-1.ipynb C:\...\58aa8dcb-c6e8-4e1a-812f-a5471eee23b8_cricket_score_pred (2).zip.3b8\...
C: > Users > Pranav Vaddamanu > AppData > Local > Temp > 58aa8dcb-c6e8-4e1a-812f-a5471eee23b8_cricket_score_pred (2).zip.3b8 > cricket_score_pred > file-1.ipynb > import numpy as np
+ Code + Markdown | Run All Restart Clear All Outputs Go To | Variables Outline ···
Python 3.9.7

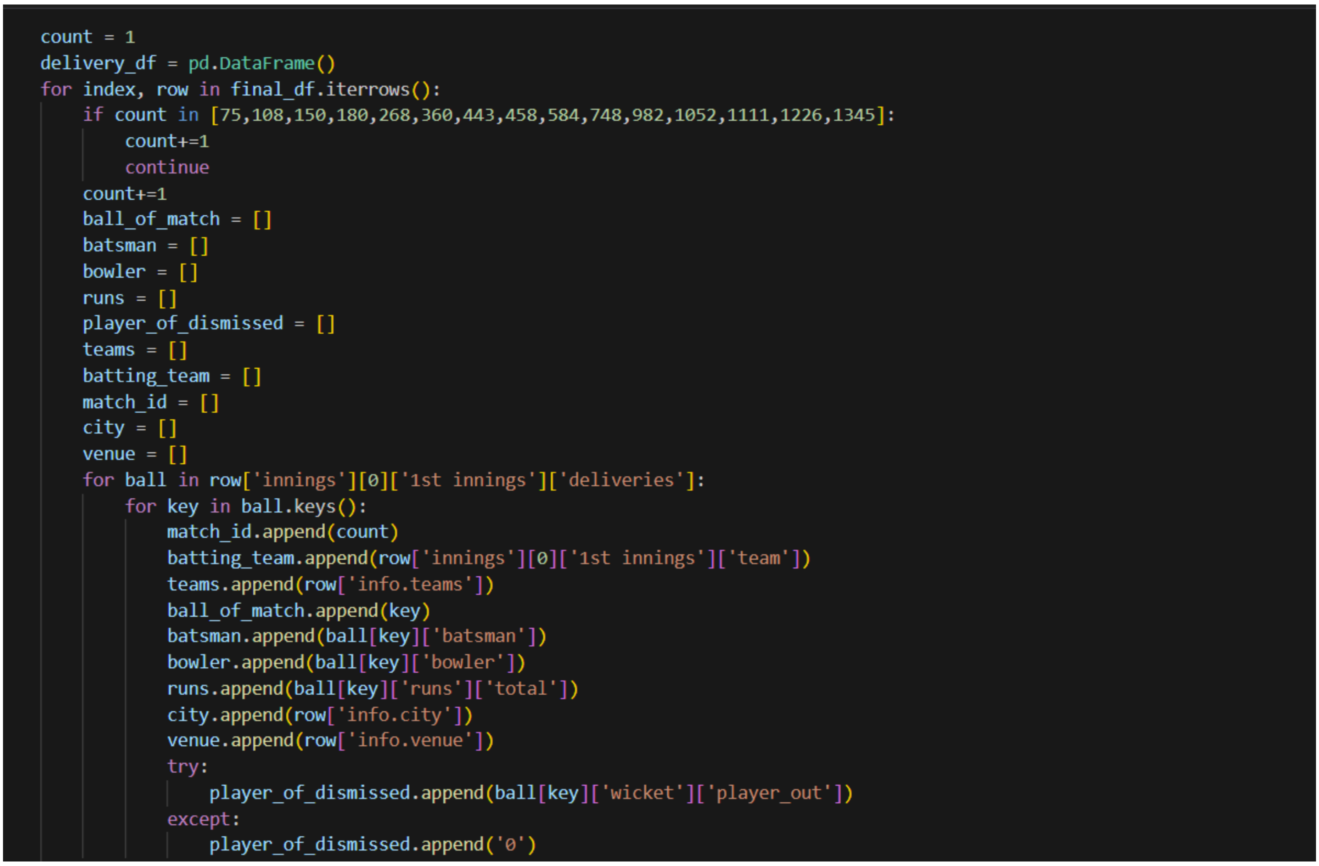
1428	[{'1st innings': {'team': 'Bangladesh', 'deliv...	[2016-03-06]	India	[S Dhawan]	[Bangladesh, India]	field	India	[RSA Palliyaguruge, Shozab Raza]	Shere Bangla National Stadium	1429	Mirpur
1429	[{'1st innings': {'team': 'Netherlands', 'deli...	[2016-02-03]	Netherlands	[Mudassar Bukhari]	[United Arab Emirates, Netherlands]	field	United Arab Emirates	[CK Nandan, Sarika Prasad]	ICC Academy	1430	Dubai
1430	[{'1st innings': {'team': 'Australia', 'delive...	[2016-09-06]	Australia	[GJ Maxwell]	[Sri Lanka, Australia]	field	Sri Lanka	[REJ Martinesz, RR Wimalasiri]	Pallekele International Cricket Stadium	1431	NaN
1431	[{'1st innings': {'team': 'Sri Lanka', 'delive...	[2016-09-09]	Australia	[GJ Maxwell]	[Sri Lanka, Australia]	bat	Sri Lanka	[REJ Martinesz, RSA Palliyaguruge]	R Premadasa Stadium	1432	Colombo

963 rows × 11 columns

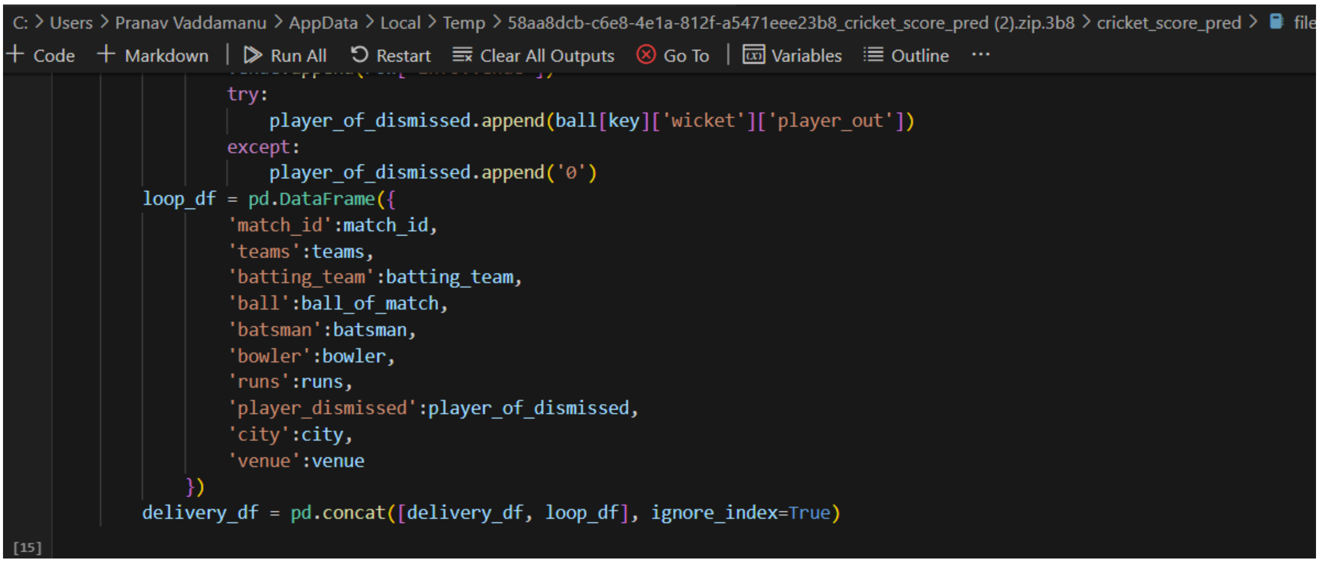
```
final_df.iloc[0]['innings'][0]['1st innings']['deliveries']
```

[14] Python

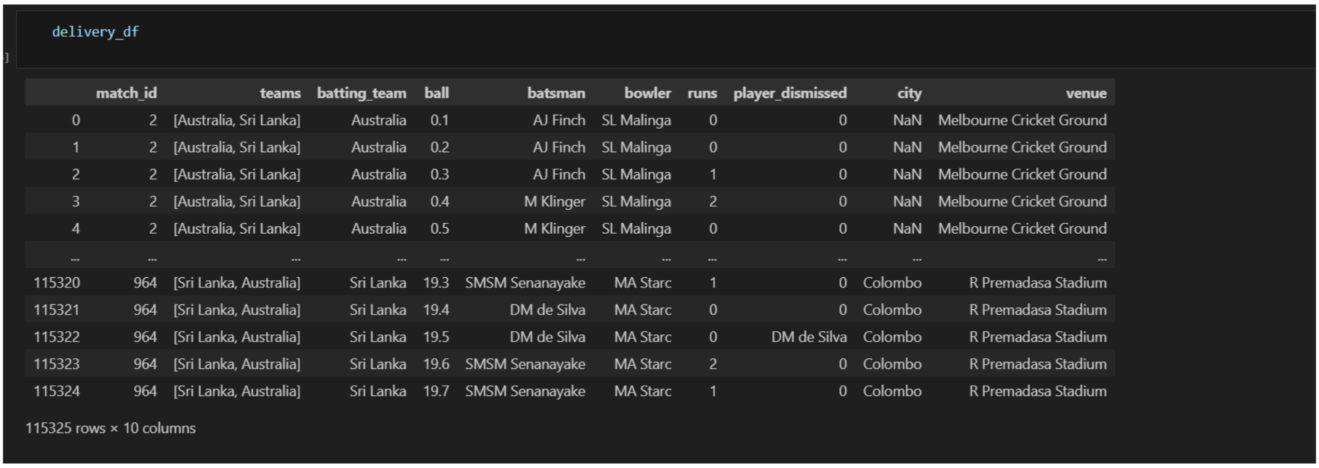
```
[{0.1: {'batsman': 'AJ Finch',
   'bowler': 'SL Malinga',
   'non_striker': 'M Klinger',
   'runs': {'batsman': 0, 'extras': 0, 'total': 0}}},
 {0.2: {'batsman': 'AJ Finch',
   'bowler': 'SL Malinga',
   'non_striker': 'M Klinger',
   'runs': {'batsman': 0, 'extras': 0, 'total': 0}}},
 {0.3: {'batsman': 'AJ Finch',
   'bowler': 'SL Malinga',
   'non_striker': 'M Klinger',
   'runs': {'batsman': 1, 'extras': 0, 'total': 1}}},
 {0.4: {'batsman': 'M Klinger',
   'bowler': 'SL Malinga',
   'non_striker': 'AJ Finch',
```



```python
count = 1
delivery_df = pd.DataFrame()
for index, row in final_df.iterrows():
    if count in [75,108,150,180,268,360,443,458,584,748,982,1052,1111,1226,1345]:
        count+=1
        continue
    count+=1
    ball_of_match = []
    batsman = []
    bowler = []
    runs = []
    player_of_dismissed = []
    teams = []
    batting_team = []
    match_id = []
    city = []
    venue = []
    for ball in row['innings'][0]['1st innings']['deliveries']:
        for key in ball.keys():
            match_id.append(count)
            batting_team.append(row['innings'][0]['1st innings']['team'])
            teams.append(row['info.teams'])
            ball_of_match.append(key)
            batsman.append(ball[key]['batsman'])
            bowler.append(ball[key]['bowler'])
            runs.append(ball[key]['runs']['total'])
            city.append(row['info.city'])
            venue.append(row['info.venue'])
            try:
                player_of_dismissed.append(ball[key]['wicket']['player_out'])
            except:
                player_of_dismissed.append('0')
```


C: > Users > Pranav Vaddamanu > AppData > Local > Temp > 58aa8dcb-c6e8-4e1a-812f-a5471eee23b8_cricket_score_pred (2).zip.3b8 > cricket_score_pred > file
+ Code
+ Markdown
Run All
Restart
Clear All Outputs
Go To
Variables
Outline
···
[15]

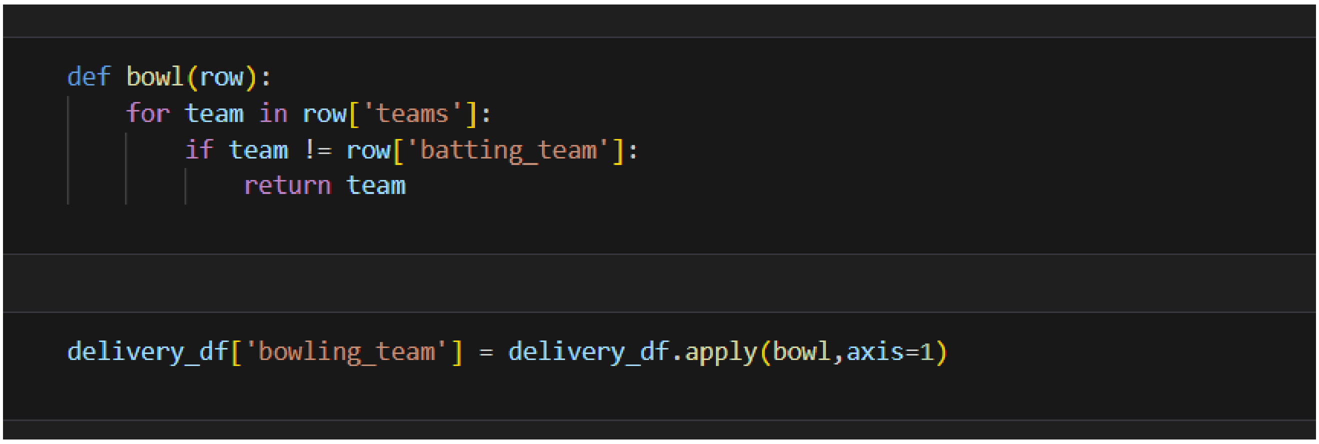
```python
            try:
                player_of_dismissed.append(ball[key]['wicket']['player_out'])
            except:
                player_of_dismissed.append('0')
    loop_df = pd.DataFrame({
            'match_id':match_id,
            'teams':teams,
            'batting_team':batting_team,
            'ball':ball_of_match,
            'batsman':batsman,
            'bowler':bowler,
            'runs':runs,
            'player_dismissed':player_of_dismissed,
            'city':city,
            'venue':venue
        })
    delivery_df = pd.concat([delivery_df, loop_df], ignore_index=True)
```



```python
delivery_df
```

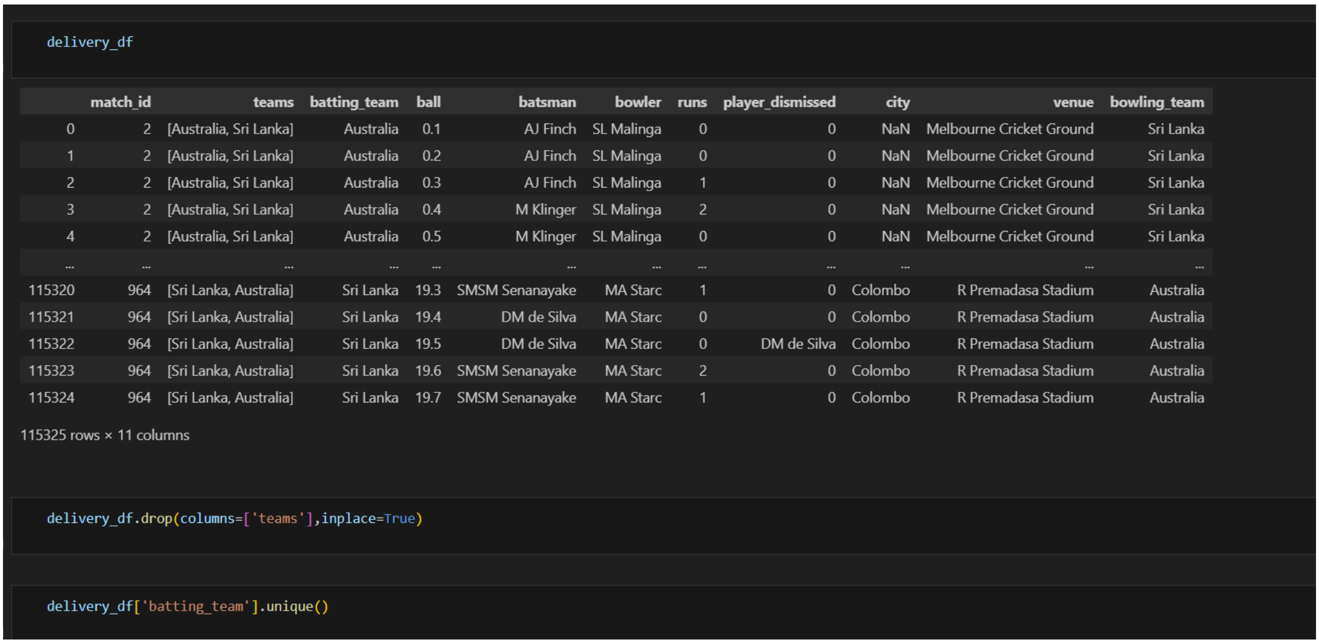
	match_id	teams	batting_team	ball	batsman	bowler	runs	player_dismissed	city	venue
0	2	[Australia, Sri Lanka]	Australia	0.1	AJ Finch	SL Malinga	0	0	NaN	Melbourne Cricket Ground
1	2	[Australia, Sri Lanka]	Australia	0.2	AJ Finch	SL Malinga	0	0	NaN	Melbourne Cricket Ground
2	2	[Australia, Sri Lanka]	Australia	0.3	AJ Finch	SL Malinga	1	0	NaN	Melbourne Cricket Ground
3	2	[Australia, Sri Lanka]	Australia	0.4	M Klinger	SL Malinga	2	0	NaN	Melbourne Cricket Ground
4	2	[Australia, Sri Lanka]	Australia	0.5	M Klinger	SL Malinga	0	0	NaN	Melbourne Cricket Ground
...	...	...	...	...	...	...	...	...	...	...
115320	964	[Sri Lanka, Australia]	Sri Lanka	19.3	SMSM Senanayake	MA Starc	1	0	Colombo	R Premadasa Stadium
115321	964	[Sri Lanka, Australia]	Sri Lanka	19.4	DM de Silva	MA Starc	0	0	Colombo	R Premadasa Stadium
115322	964	[Sri Lanka, Australia]	Sri Lanka	19.5	DM de Silva	MA Starc	0	DM de Silva	Colombo	R Premadasa Stadium
115323	964	[Sri Lanka, Australia]	Sri Lanka	19.6	SMSM Senanayake	MA Starc	2	0	Colombo	R Premadasa Stadium
115324	964	[Sri Lanka, Australia]	Sri Lanka	19.7	SMSM Senanayake	MA Starc	1	0	Colombo	R Premadasa Stadium

115325 rows × 10 columns



```
def bowl(row):
    for team in row['teams']:
        if team != row['batting_team']:
            return team
```

```
delivery_df['bowling_team'] = delivery_df.apply(bowl,axis=1)
```



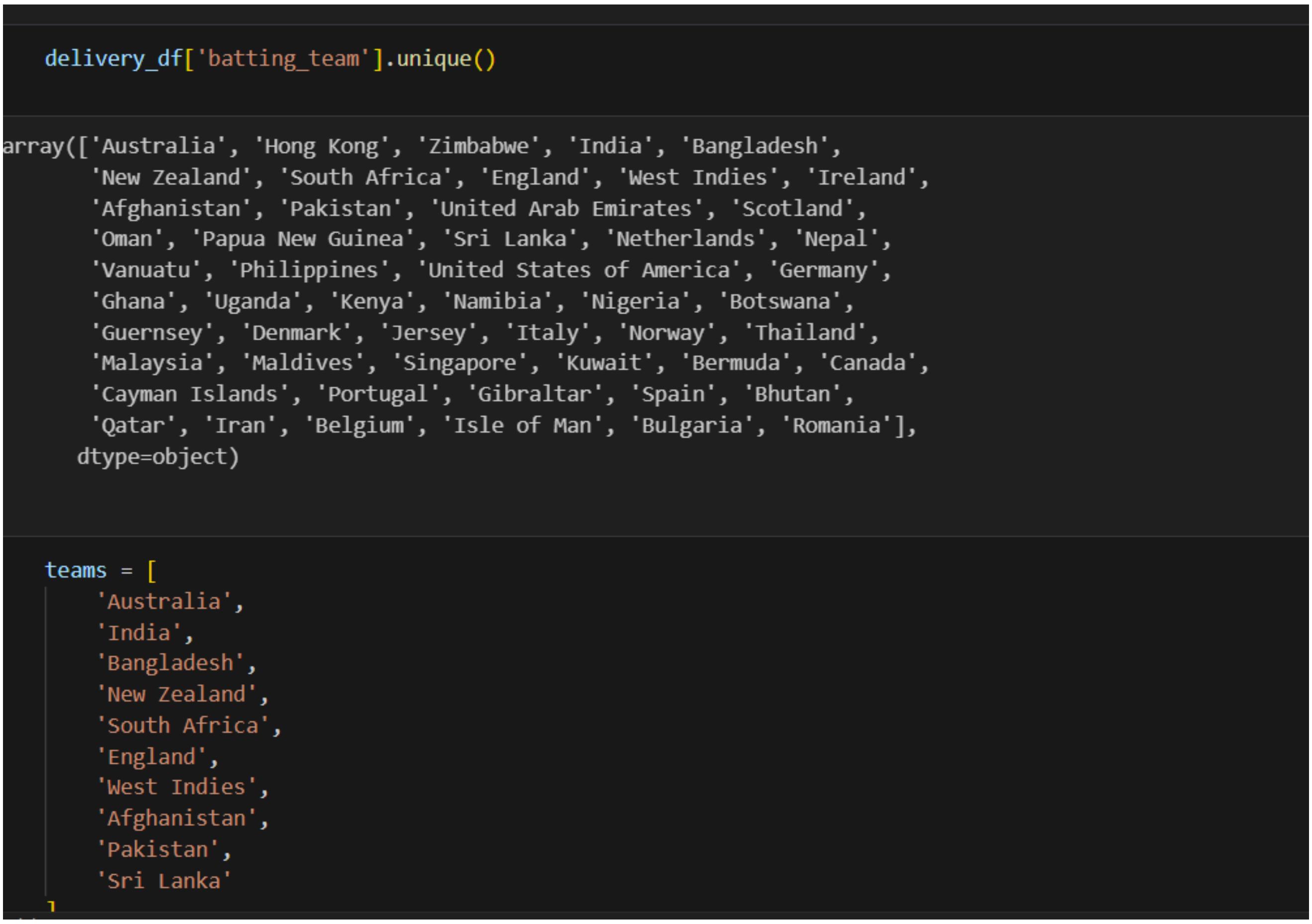
```
delivery_df
```

	match_id	teams	batting_team	ball	batsman	bowler	runs	player_dismissed	city	venue	bowling_team
0	2	[Australia, Sri Lanka]	Australia	0.1	AJ Finch	SL Malinga	0	0	NaN	Melbourne Cricket Ground	Sri Lanka
1	2	[Australia, Sri Lanka]	Australia	0.2	AJ Finch	SL Malinga	0	0	NaN	Melbourne Cricket Ground	Sri Lanka
2	2	[Australia, Sri Lanka]	Australia	0.3	AJ Finch	SL Malinga	1	0	NaN	Melbourne Cricket Ground	Sri Lanka
3	2	[Australia, Sri Lanka]	Australia	0.4	M Klinger	SL Malinga	2	0	NaN	Melbourne Cricket Ground	Sri Lanka
4	2	[Australia, Sri Lanka]	Australia	0.5	M Klinger	SL Malinga	0	0	NaN	Melbourne Cricket Ground	Sri Lanka
...	...	...	...	...	...	...	...	...	...	...	...
115320	964	[Sri Lanka, Australia]	Sri Lanka	19.3	SMSM Senanayake	MA Starc	1	0	Colombo	R Premadasa Stadium	Australia
115321	964	[Sri Lanka, Australia]	Sri Lanka	19.4	DM de Silva	MA Starc	0	0	Colombo	R Premadasa Stadium	Australia
115322	964	[Sri Lanka, Australia]	Sri Lanka	19.5	DM de Silva	MA Starc	0	DM de Silva	Colombo	R Premadasa Stadium	Australia
115323	964	[Sri Lanka, Australia]	Sri Lanka	19.6	SMSM Senanayake	MA Starc	2	0	Colombo	R Premadasa Stadium	Australia
115324	964	[Sri Lanka, Australia]	Sri Lanka	19.7	SMSM Senanayake	MA Starc	1	0	Colombo	R Premadasa Stadium	Australia

115325 rows × 11 columns

```
delivery_df.drop(columns=['teams'],inplace=True)
```

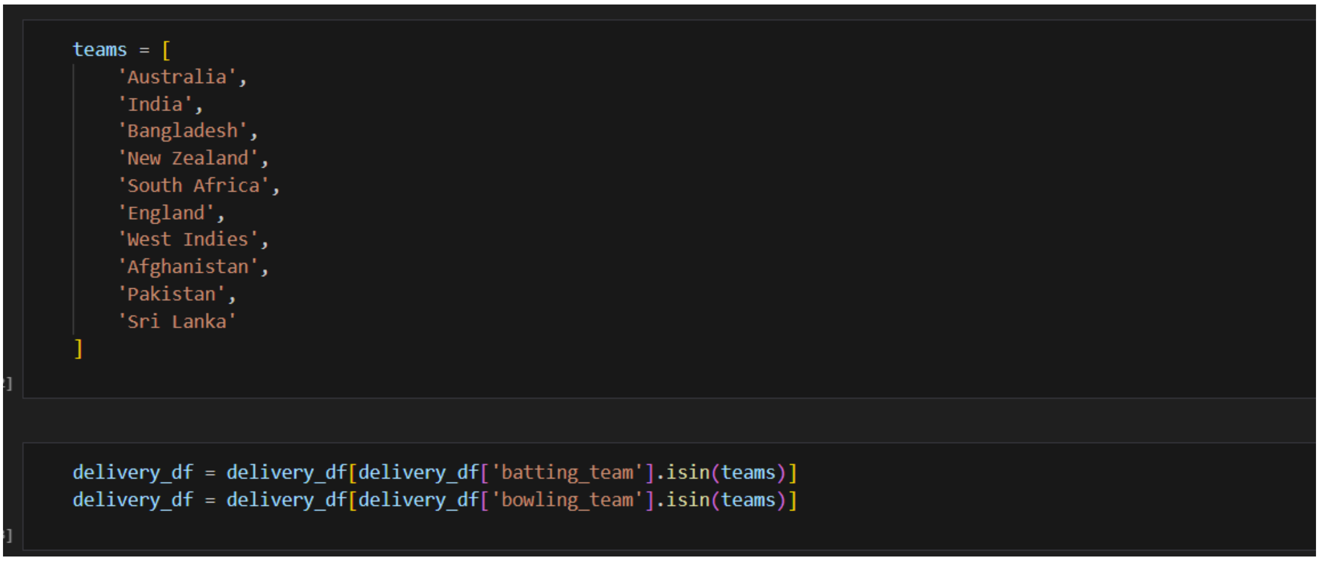
```
delivery_df['batting_team'].unique()
```



```
delivery_df['batting_team'].unique()
```

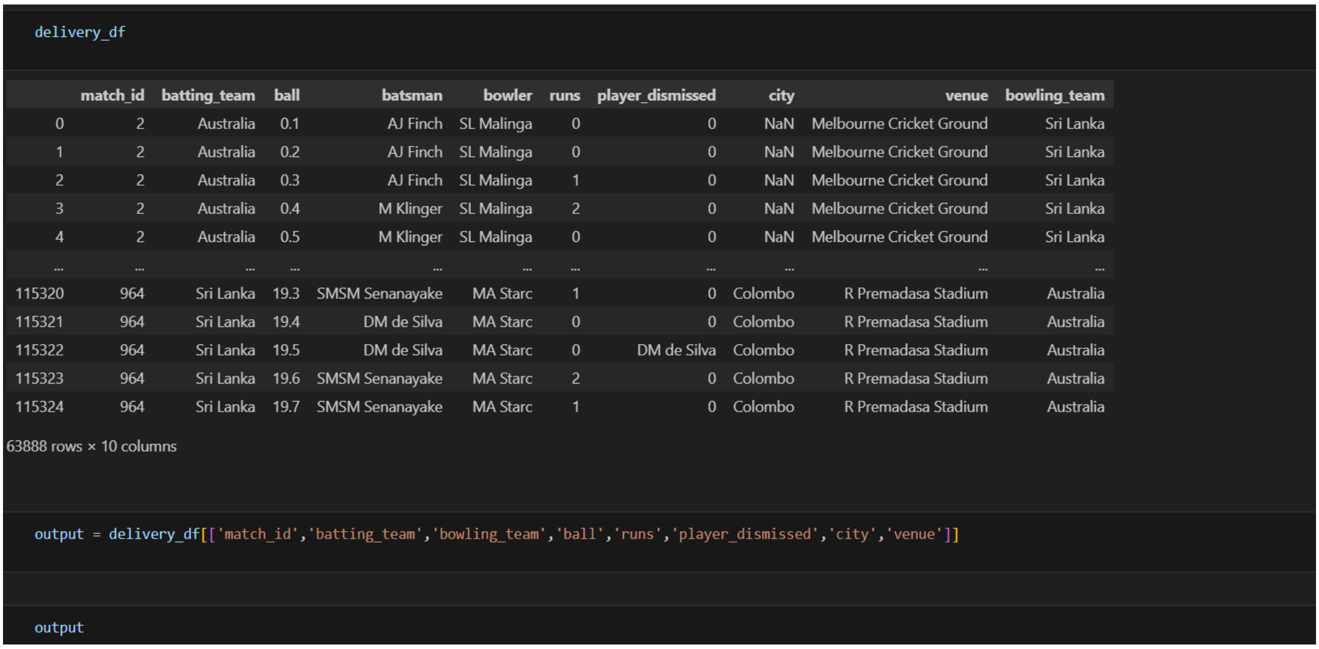
```
array(['Australia', 'Hong Kong', 'Zimbabwe', 'India', 'Bangladesh',
       'New Zealand', 'South Africa', 'England', 'West Indies', 'Ireland',
       'Afghanistan', 'Pakistan', 'United Arab Emirates', 'Scotland',
       'Oman', 'Papua New Guinea', 'Sri Lanka', 'Netherlands', 'Nepal',
       'Vanuatu', 'Philippines', 'United States of America', 'Germany',
       'Ghana', 'Uganda', 'Kenya', 'Namibia', 'Nigeria', 'Botswana',
       'Guernsey', 'Denmark', 'Jersey', 'Italy', 'Norway', 'Thailand',
       'Malaysia', 'Maldives', 'Singapore', 'Kuwait', 'Bermuda', 'Canada',
       'Cayman Islands', 'Portugal', 'Gibraltar', 'Spain', 'Bhutan',
       'Qatar', 'Iran', 'Belgium', 'Isle of Man', 'Bulgaria', 'Romania'],
      dtype=object)
```

```
teams = [
    'Australia',
    'India',
    'Bangladesh',
    'New Zealand',
    'South Africa',
    'England',
    'West Indies',
    'Afghanistan',
    'Pakistan',
    'Sri Lanka'
]
```



```python
teams = [
    'Australia',
    'India',
    'Bangladesh',
    'New Zealand',
    'South Africa',
    'England',
    'West Indies',
    'Afghanistan',
    'Pakistan',
    'Sri Lanka'
]
```

```python
delivery_df = delivery_df[delivery_df['batting_team'].isin(teams)]
delivery_df = delivery_df[delivery_df['bowling_team'].isin(teams)]
```



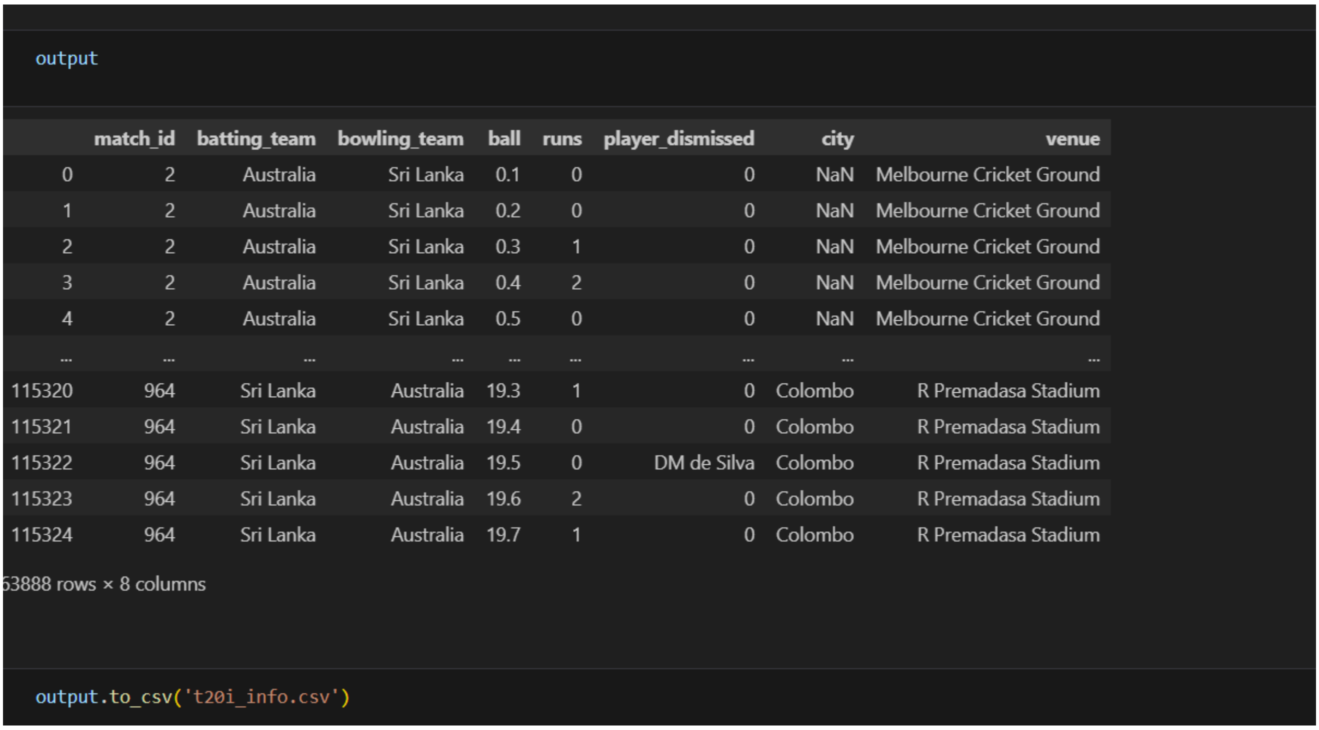
```python
delivery_df
```

	match_id	batting_team	ball	batsman	bowler	runs	player_dismissed	city	venue	bowling_team
0	2	Australia	0.1	AJ Finch	SL Malinga	0	0	NaN	Melbourne Cricket Ground	Sri Lanka
1	2	Australia	0.2	AJ Finch	SL Malinga	0	0	NaN	Melbourne Cricket Ground	Sri Lanka
2	2	Australia	0.3	AJ Finch	SL Malinga	1	0	NaN	Melbourne Cricket Ground	Sri Lanka
3	2	Australia	0.4	M Klinger	SL Malinga	2	0	NaN	Melbourne Cricket Ground	Sri Lanka
4	2	Australia	0.5	M Klinger	SL Malinga	0	0	NaN	Melbourne Cricket Ground	Sri Lanka
...	...	...	...	...	...	...	...	...	...	...
115320	964	Sri Lanka	19.3	SMSM Senanayake	MA Starc	1	0	Colombo	R Premadasa Stadium	Australia
115321	964	Sri Lanka	19.4	DM de Silva	MA Starc	0	0	Colombo	R Premadasa Stadium	Australia
115322	964	Sri Lanka	19.5	DM de Silva	MA Starc	0	DM de Silva	Colombo	R Premadasa Stadium	Australia
115323	964	Sri Lanka	19.6	SMSM Senanayake	MA Starc	2	0	Colombo	R Premadasa Stadium	Australia
115324	964	Sri Lanka	19.7	SMSM Senanayake	MA Starc	1	0	Colombo	R Premadasa Stadium	Australia

63888 rows × 10 columns

```python
output = delivery_df[['match_id','batting_team','bowling_team','ball','runs','player_dismissed','city','venue']]
```

```python
output
```



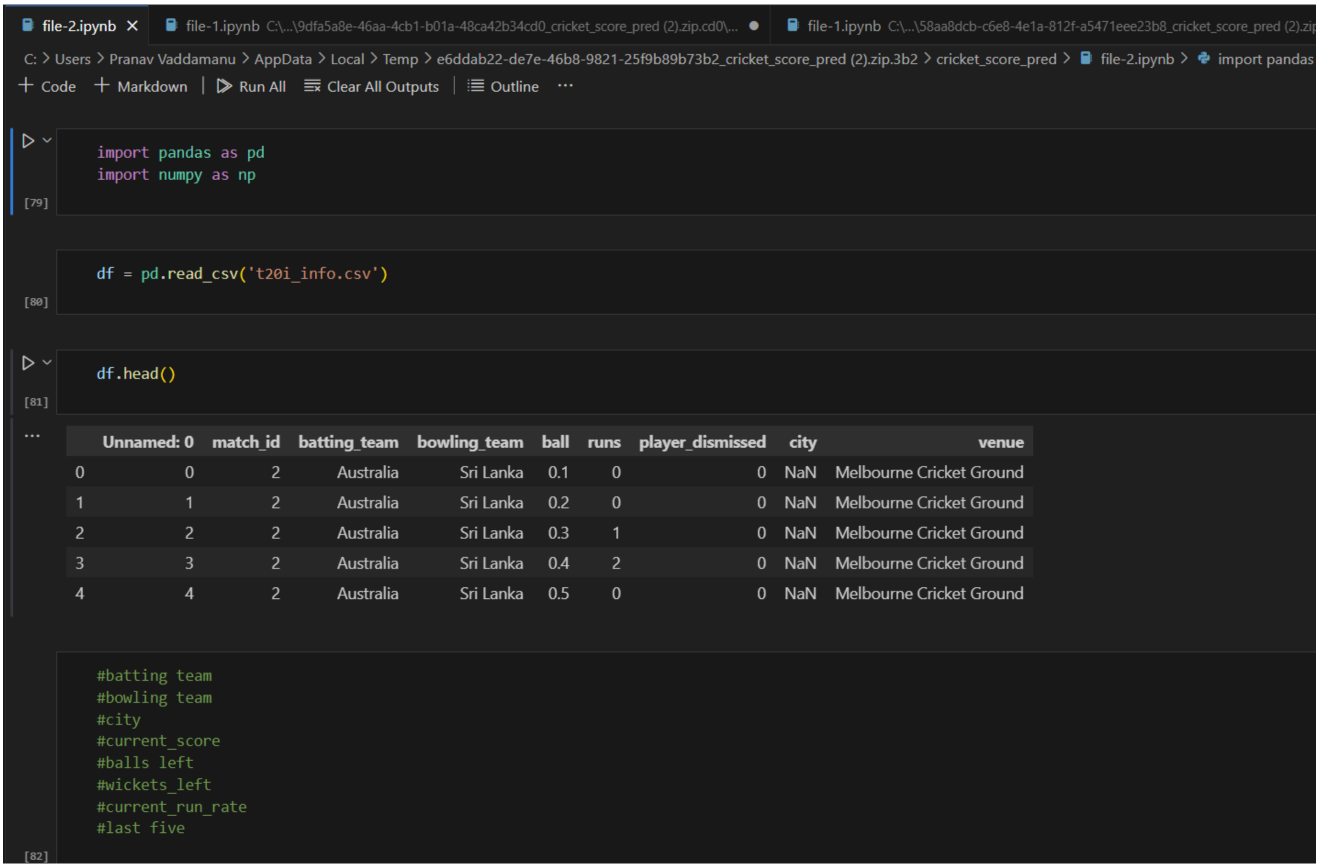
```python
output
```

	match_id	batting_team	bowling_team	ball	runs	player_dismissed	city	venue
0	2	Australia	Sri Lanka	0.1	0	0	NaN	Melbourne Cricket Ground
1	2	Australia	Sri Lanka	0.2	0	0	NaN	Melbourne Cricket Ground
2	2	Australia	Sri Lanka	0.3	1	0	NaN	Melbourne Cricket Ground
3	2	Australia	Sri Lanka	0.4	2	0	NaN	Melbourne Cricket Ground
4	2	Australia	Sri Lanka	0.5	0	0	NaN	Melbourne Cricket Ground
...	...	...	...	...	...	...	...	...
115320	964	Sri Lanka	Australia	19.3	1	0	Colombo	R Premadasa Stadium
115321	964	Sri Lanka	Australia	19.4	0	0	Colombo	R Premadasa Stadium
115322	964	Sri Lanka	Australia	19.5	0	DM de Silva	Colombo	R Premadasa Stadium
115323	964	Sri Lanka	Australia	19.6	2	0	Colombo	R Premadasa Stadium
115324	964	Sri Lanka	Australia	19.7	1	0	Colombo	R Premadasa Stadium

63888 rows × 8 columns

```python
output.to_csv('t20i_info.csv')
```

File-2: Model Training and Prediction python


file-2.ipynb
file-1.ipynb C:\...\9dfa5a8e-46aa-4cb1-b01a-48ca42b34cd0_cricket_score_pred (2).zip.cd0\...
file-1.ipynb C:\...\58aa8dcb-c6e8-4e1a-812f-a5471eee23b8_cricket_score_pred (2).zip
C: > Users > Pranav Vaddamanu > AppData > Local > Temp > e6ddab22-de7e-46b8-9821-25f9b89b73b2_cricket_score_pred (2).zip.3b2 > cricket_score_pred > file-2.ipynb > import pandas
+ Code + Markdown | Run All Clear All Outputs | Outline ...
[79]
[80]
[81]
[82]

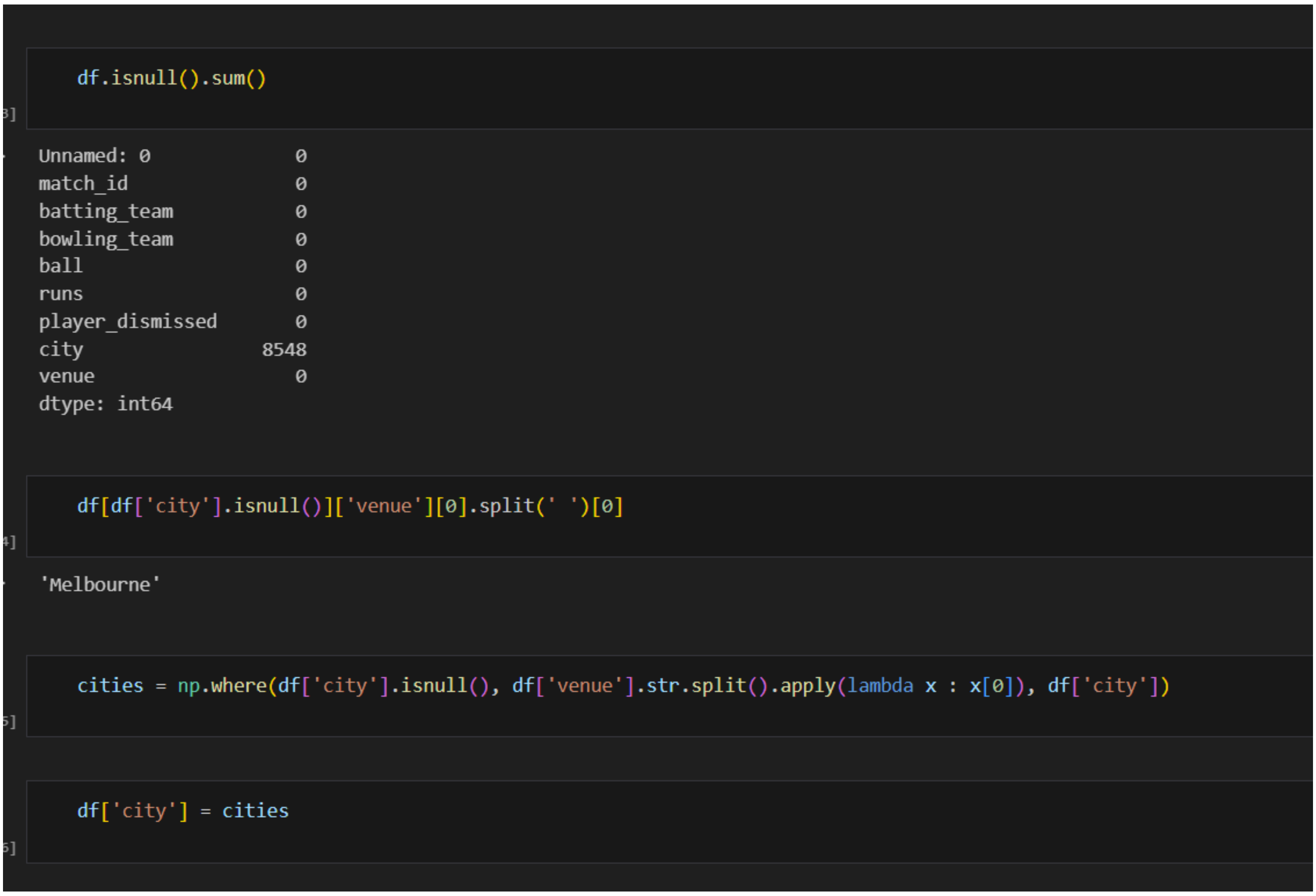
```
import pandas as pd
import numpy as np
```

```
df = pd.read_csv('t20i_info.csv')
```

```
df.head()
```

	Unnamed: 0	match_id	batting_team	bowling_team	ball	runs	player_dismissed	city	venue
0	0	2	Australia	Sri Lanka	0.1	0	0	NaN	Melbourne Cricket Ground
1	1	2	Australia	Sri Lanka	0.2	0	0	NaN	Melbourne Cricket Ground
2	2	2	Australia	Sri Lanka	0.3	1	0	NaN	Melbourne Cricket Ground
3	3	2	Australia	Sri Lanka	0.4	2	0	NaN	Melbourne Cricket Ground
4	4	2	Australia	Sri Lanka	0.5	0	0	NaN	Melbourne Cricket Ground

```
#batting team
#bowling team
#city
#current_score
#balls left
#wickets_left
#current_run_rate
#last five
```



```
df.isnull().sum()
```

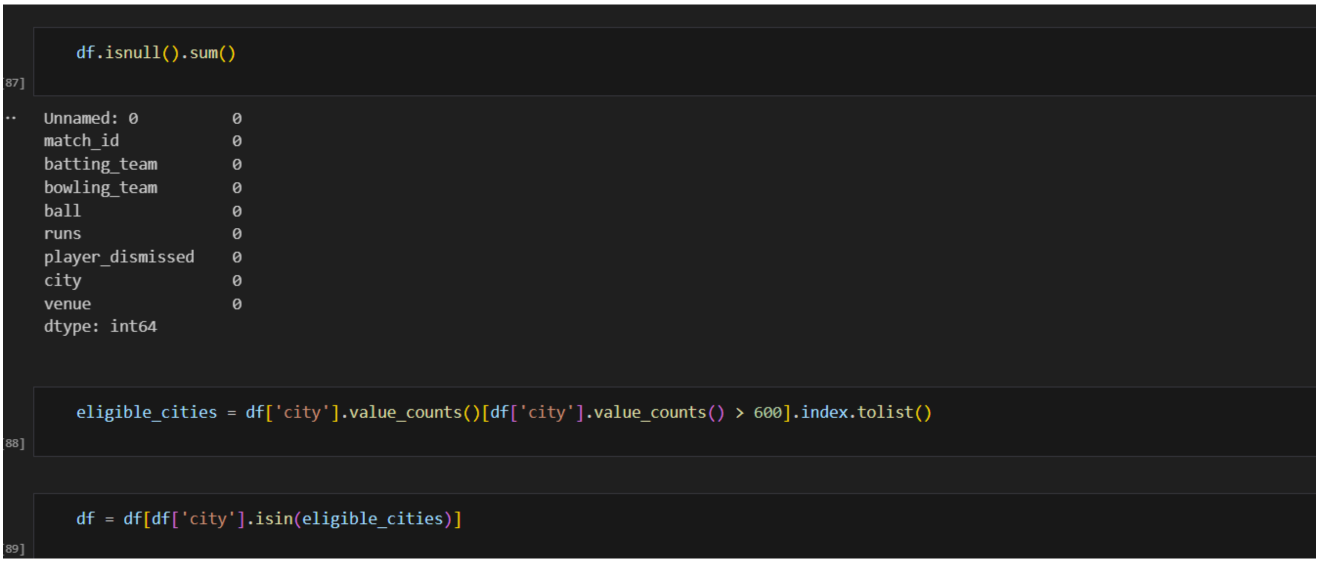
```
Unnamed: 0             0
match_id               0
batting_team           0
bowling_team           0
ball                   0
runs                   0
player_dismissed       0
city                8548
venue                  0
dtype: int64
```

```
df[df['city'].isnull()]['venue'][0].split(' ')[0]
```

```
'Melbourne'
```

```
cities = np.where(df['city'].isnull(), df['venue'].str.split().apply(lambda x : x[0]), df['city'])
```

```
df['city'] = cities
```

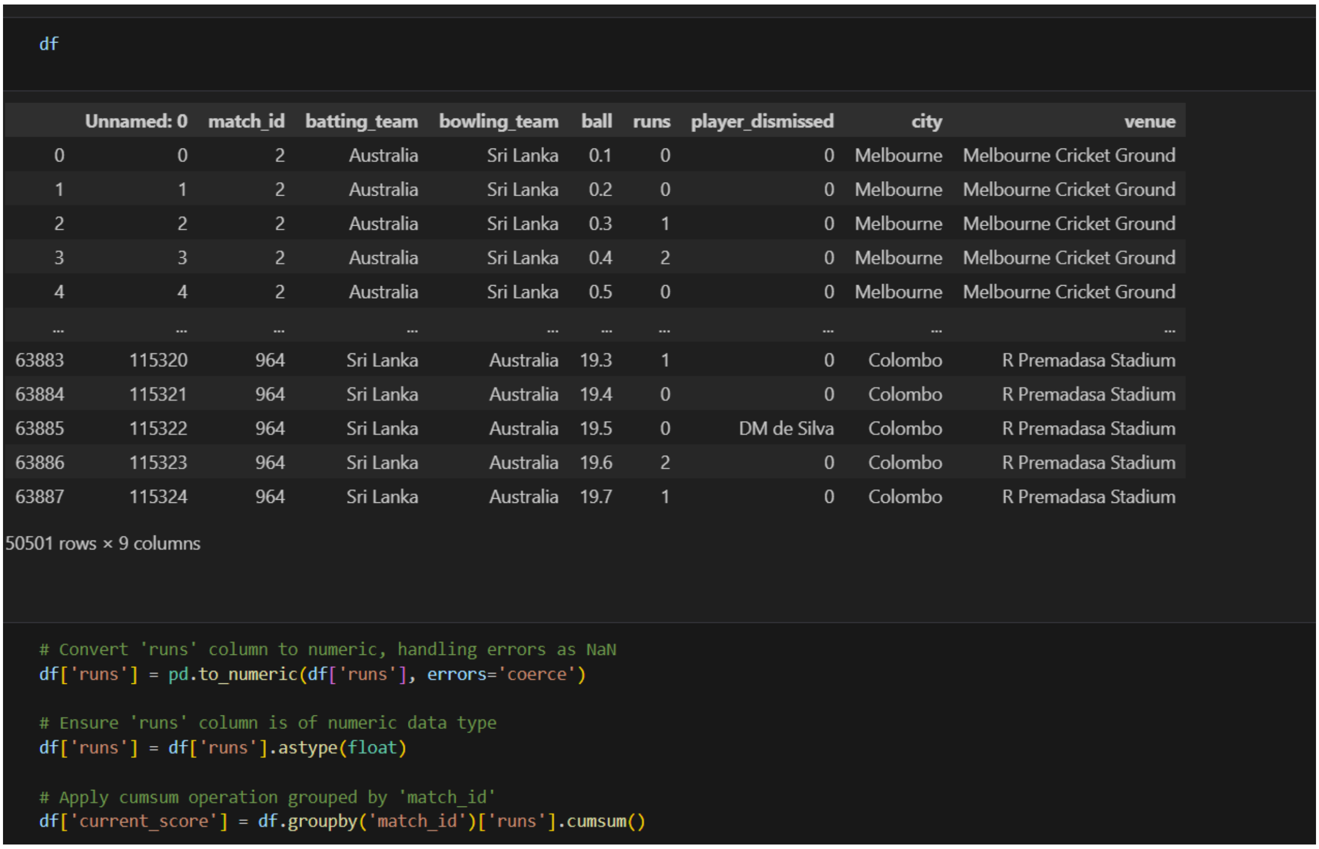


```python
df.isnull().sum()
```

```
Unnamed: 0          0
match_id            0
batting_team        0
bowling_team        0
ball                0
runs                0
player_dismissed    0
city                0
venue               0
dtype: int64
```

```python
eligible_cities = df['city'].value_counts()[df['city'].value_counts() > 600].index.tolist()
```

```python
df = df[df['city'].isin(eligible_cities)]
```

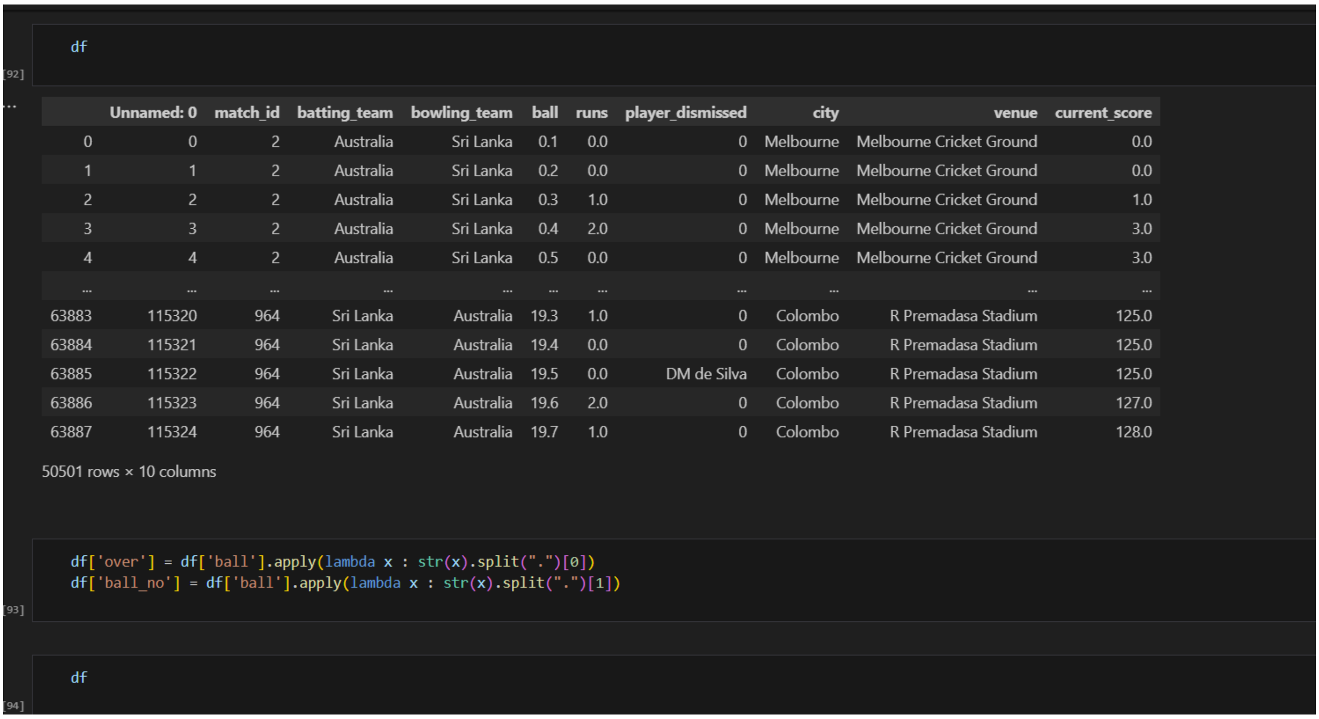


```python
df
```

	Unnamed: 0	match_id	batting_team	bowling_team	ball	runs	player_dismissed	city	venue
0	0	2	Australia	Sri Lanka	0.1	0	0	Melbourne	Melbourne Cricket Ground
1	1	2	Australia	Sri Lanka	0.2	0	0	Melbourne	Melbourne Cricket Ground
2	2	2	Australia	Sri Lanka	0.3	1	0	Melbourne	Melbourne Cricket Ground
3	3	2	Australia	Sri Lanka	0.4	2	0	Melbourne	Melbourne Cricket Ground
4	4	2	Australia	Sri Lanka	0.5	0	0	Melbourne	Melbourne Cricket Ground
...	...	...	...	...	...	...	...	...	...
63883	115320	964	Sri Lanka	Australia	19.3	1	0	Colombo	R Premadasa Stadium
63884	115321	964	Sri Lanka	Australia	19.4	0	0	Colombo	R Premadasa Stadium
63885	115322	964	Sri Lanka	Australia	19.5	0	DM de Silva	Colombo	R Premadasa Stadium
63886	115323	964	Sri Lanka	Australia	19.6	2	0	Colombo	R Premadasa Stadium
63887	115324	964	Sri Lanka	Australia	19.7	1	0	Colombo	R Premadasa Stadium

50501 rows × 9 columns

```python
# Convert 'runs' column to numeric, handling errors as NaN
df['runs'] = pd.to_numeric(df['runs'], errors='coerce')

# Ensure 'runs' column is of numeric data type
df['runs'] = df['runs'].astype(float)

# Apply cumsum operation grouped by 'match_id'
df['current_score'] = df.groupby('match_id')['runs'].cumsum()
```



```
df
```

[92]

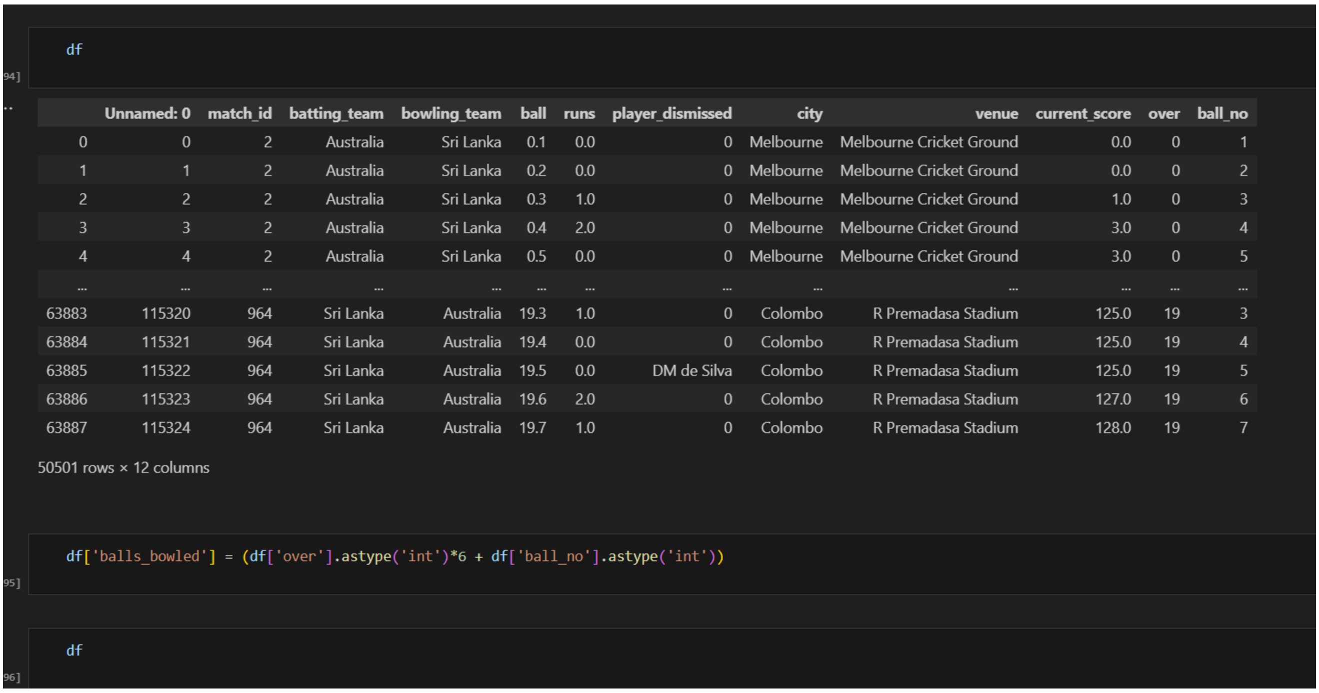
```
       Unnamed: 0  match_id  batting_team  bowling_team  ball  runs  player_dismissed       city                   venue  current_score
0               0         2     Australia     Sri Lanka   0.1   0.0                 0  Melbourne  Melbourne Cricket Ground            0.0
1               1         2     Australia     Sri Lanka   0.2   0.0                 0  Melbourne  Melbourne Cricket Ground            0.0
2               2         2     Australia     Sri Lanka   0.3   1.0                 0  Melbourne  Melbourne Cricket Ground            1.0
3               3         2     Australia     Sri Lanka   0.4   2.0                 0  Melbourne  Melbourne Cricket Ground            3.0
4               4         2     Australia     Sri Lanka   0.5   0.0                 0  Melbourne  Melbourne Cricket Ground            3.0
...           ...       ...           ...           ...   ...   ...               ...        ...                       ...            ...
63883      115320       964     Sri Lanka     Australia  19.3   1.0                 0    Colombo      R Premadasa Stadium          125.0
63884      115321       964     Sri Lanka     Australia  19.4   0.0                 0    Colombo      R Premadasa Stadium          125.0
63885      115322       964     Sri Lanka     Australia  19.5   0.0       DM de Silva    Colombo      R Premadasa Stadium          125.0
63886      115323       964     Sri Lanka     Australia  19.6   2.0                 0    Colombo      R Premadasa Stadium          127.0
63887      115324       964     Sri Lanka     Australia  19.7   1.0                 0    Colombo      R Premadasa Stadium          128.0

50501 rows × 10 columns
```

```
df['over'] = df['ball'].apply(lambda x : str(x).split(".")[0])
df['ball_no'] = df['ball'].apply(lambda x : str(x).split(".")[1])
```

[93]

```
df
```

[94]



```
df
```

94]

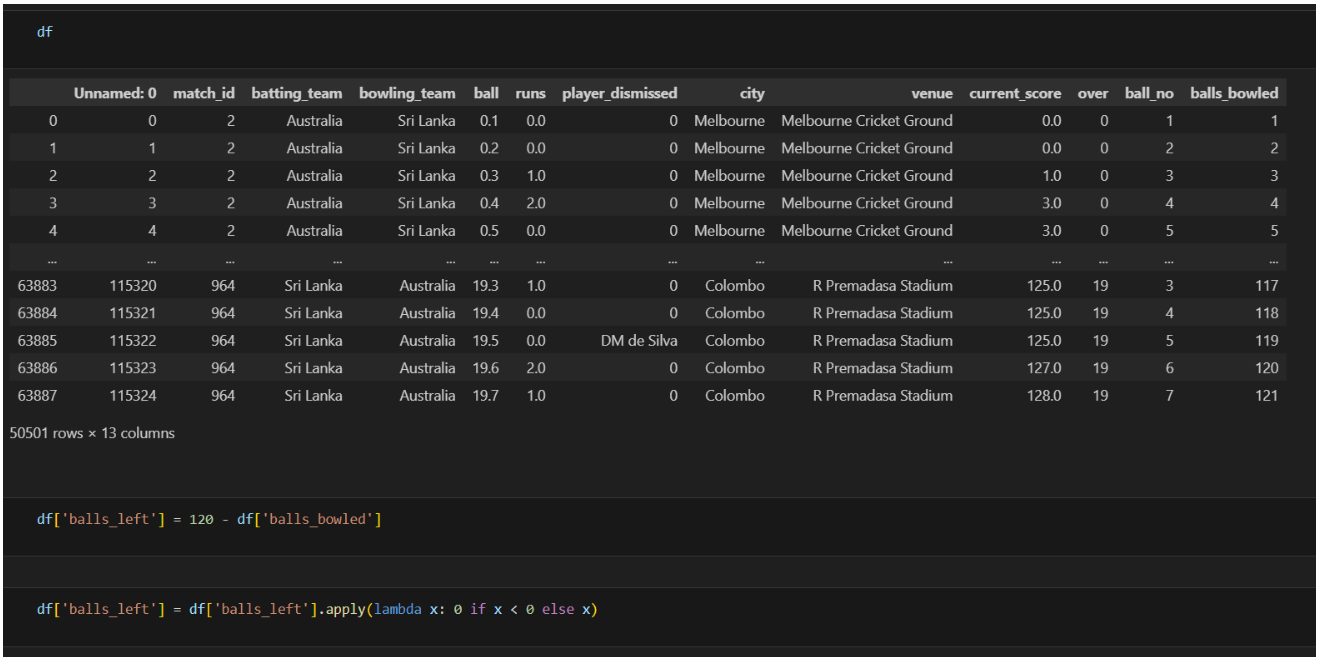
```
       Unnamed: 0  match_id  batting_team  bowling_team  ball  runs  player_dismissed       city                   venue  current_score  over  ball_no
0               0         2     Australia     Sri Lanka   0.1   0.0                 0  Melbourne  Melbourne Cricket Ground            0.0     0        1
1               1         2     Australia     Sri Lanka   0.2   0.0                 0  Melbourne  Melbourne Cricket Ground            0.0     0        2
2               2         2     Australia     Sri Lanka   0.3   1.0                 0  Melbourne  Melbourne Cricket Ground            1.0     0        3
3               3         2     Australia     Sri Lanka   0.4   2.0                 0  Melbourne  Melbourne Cricket Ground            3.0     0        4
4               4         2     Australia     Sri Lanka   0.5   0.0                 0  Melbourne  Melbourne Cricket Ground            3.0     0        5
...           ...       ...           ...           ...   ...   ...               ...        ...                       ...            ...   ...      ...
63883      115320       964     Sri Lanka     Australia  19.3   1.0                 0    Colombo      R Premadasa Stadium          125.0    19        3
63884      115321       964     Sri Lanka     Australia  19.4   0.0                 0    Colombo      R Premadasa Stadium          125.0    19        4
63885      115322       964     Sri Lanka     Australia  19.5   0.0       DM de Silva    Colombo      R Premadasa Stadium          125.0    19        5
63886      115323       964     Sri Lanka     Australia  19.6   2.0                 0    Colombo      R Premadasa Stadium          127.0    19        6
63887      115324       964     Sri Lanka     Australia  19.7   1.0                 0    Colombo      R Premadasa Stadium          128.0    19        7

50501 rows × 12 columns
```

```
df['balls_bowled'] = (df['over'].astype('int')*6 + df['ball_no'].astype('int'))
```

95]

```
df
```

96]



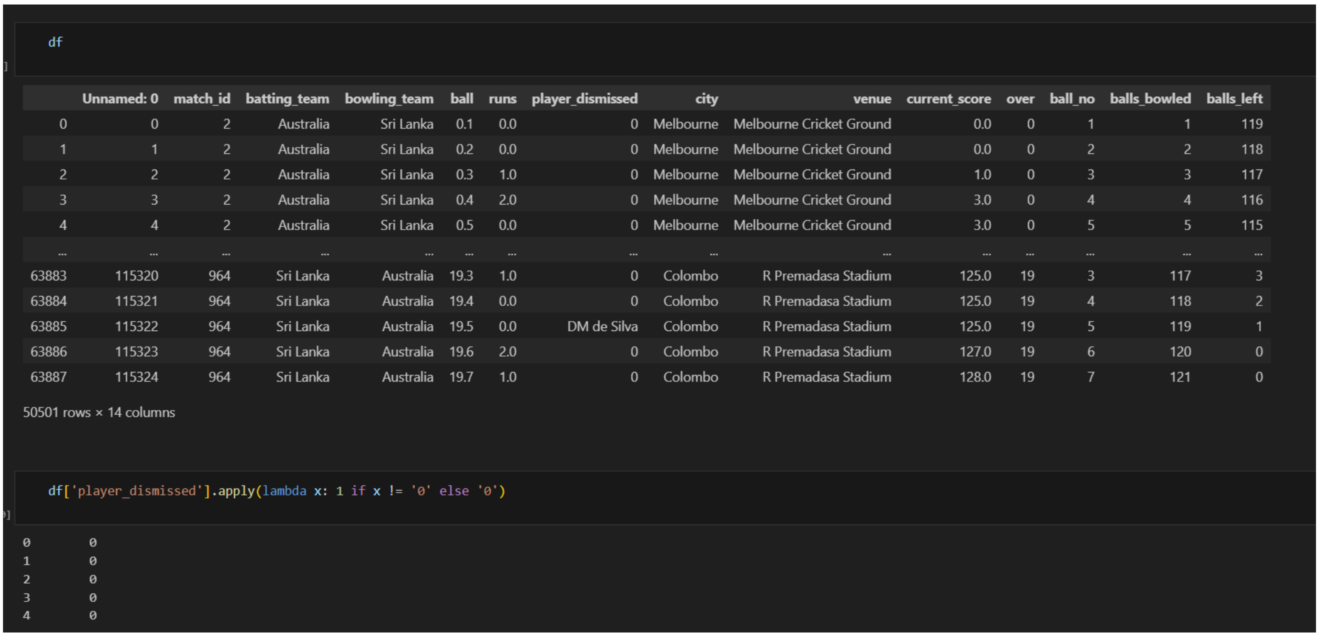
```
df
```

	Unnamed: 0	match_id	batting_team	bowling_team	ball	runs	player_dismissed	city	venue	current_score	over	ball_no	balls_bowled
0	0	2	Australia	Sri Lanka	0.1	0.0	0	Melbourne	Melbourne Cricket Ground	0.0	0	1	1
1	1	2	Australia	Sri Lanka	0.2	0.0	0	Melbourne	Melbourne Cricket Ground	0.0	0	2	2
2	2	2	Australia	Sri Lanka	0.3	1.0	0	Melbourne	Melbourne Cricket Ground	1.0	0	3	3
3	3	2	Australia	Sri Lanka	0.4	2.0	0	Melbourne	Melbourne Cricket Ground	3.0	0	4	4
4	4	2	Australia	Sri Lanka	0.5	0.0	0	Melbourne	Melbourne Cricket Ground	3.0	0	5	5
...	...	...	...	...	...	...	...	...	...	...	...	...	...
63883	115320	964	Sri Lanka	Australia	19.3	1.0	0	Colombo	R Premadasa Stadium	125.0	19	3	117
63884	115321	964	Sri Lanka	Australia	19.4	0.0	0	Colombo	R Premadasa Stadium	125.0	19	4	118
63885	115322	964	Sri Lanka	Australia	19.5	0.0	DM de Silva	Colombo	R Premadasa Stadium	125.0	19	5	119
63886	115323	964	Sri Lanka	Australia	19.6	2.0	0	Colombo	R Premadasa Stadium	127.0	19	6	120
63887	115324	964	Sri Lanka	Australia	19.7	1.0	0	Colombo	R Premadasa Stadium	128.0	19	7	121

50501 rows × 13 columns

```
df['balls_left'] = 120 - df['balls_bowled']
```

```
df['balls_left'] = df['balls_left'].apply(lambda x: 0 if x < 0 else x)
```



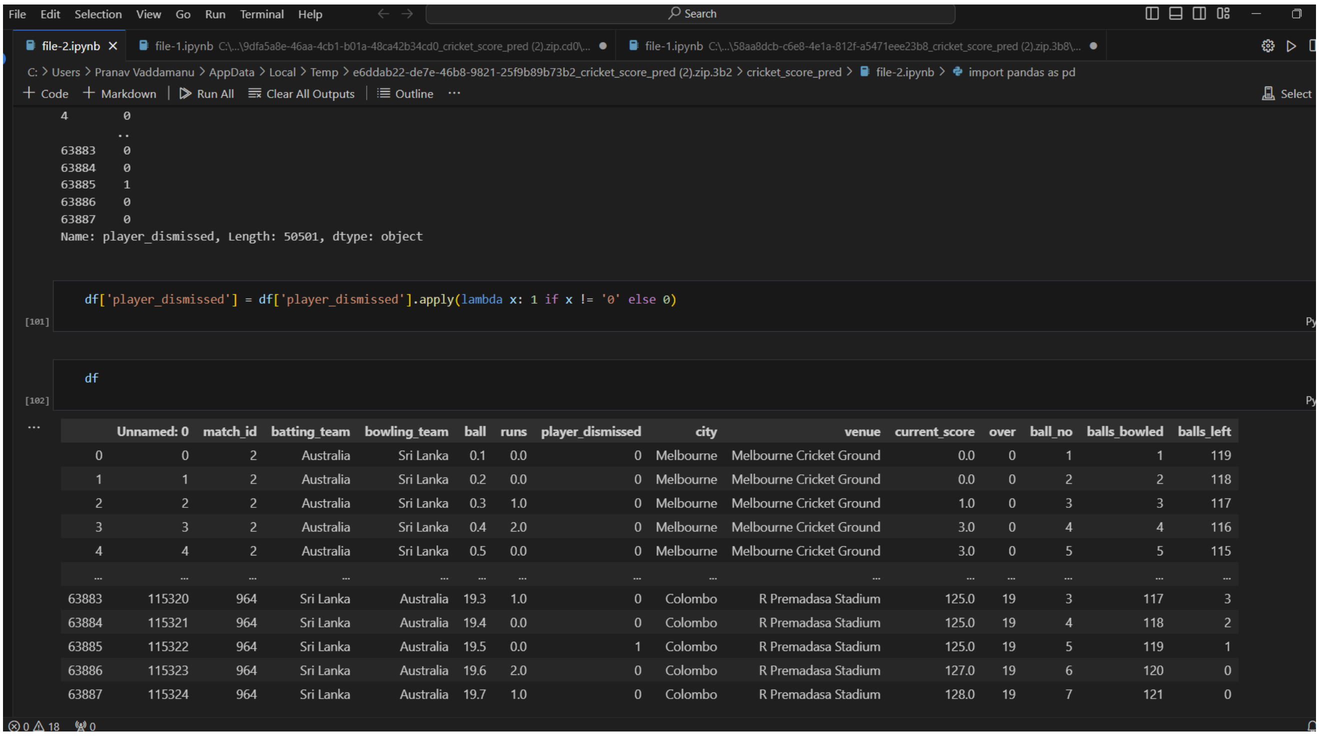
```
df
```

	Unnamed: 0	match_id	batting_team	bowling_team	ball	runs	player_dismissed	city	venue	current_score	over	ball_no	balls_bowled	balls_left
0	0	2	Australia	Sri Lanka	0.1	0.0	0	Melbourne	Melbourne Cricket Ground	0.0	0	1	1	119
1	1	2	Australia	Sri Lanka	0.2	0.0	0	Melbourne	Melbourne Cricket Ground	0.0	0	2	2	118
2	2	2	Australia	Sri Lanka	0.3	1.0	0	Melbourne	Melbourne Cricket Ground	1.0	0	3	3	117
3	3	2	Australia	Sri Lanka	0.4	2.0	0	Melbourne	Melbourne Cricket Ground	3.0	0	4	4	116
4	4	2	Australia	Sri Lanka	0.5	0.0	0	Melbourne	Melbourne Cricket Ground	3.0	0	5	5	115
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
63883	115320	964	Sri Lanka	Australia	19.3	1.0	0	Colombo	R Premadasa Stadium	125.0	19	3	117	3
63884	115321	964	Sri Lanka	Australia	19.4	0.0	0	Colombo	R Premadasa Stadium	125.0	19	4	118	2
63885	115322	964	Sri Lanka	Australia	19.5	0.0	DM de Silva	Colombo	R Premadasa Stadium	125.0	19	5	119	1
63886	115323	964	Sri Lanka	Australia	19.6	2.0	0	Colombo	R Premadasa Stadium	127.0	19	6	120	0
63887	115324	964	Sri Lanka	Australia	19.7	1.0	0	Colombo	R Premadasa Stadium	128.0	19	7	121	0

50501 rows × 14 columns

```
df['player_dismissed'].apply(lambda x: 1 if x != '0' else '0')
```

```
0        0
1        0
2        0
3        0
4        0
```



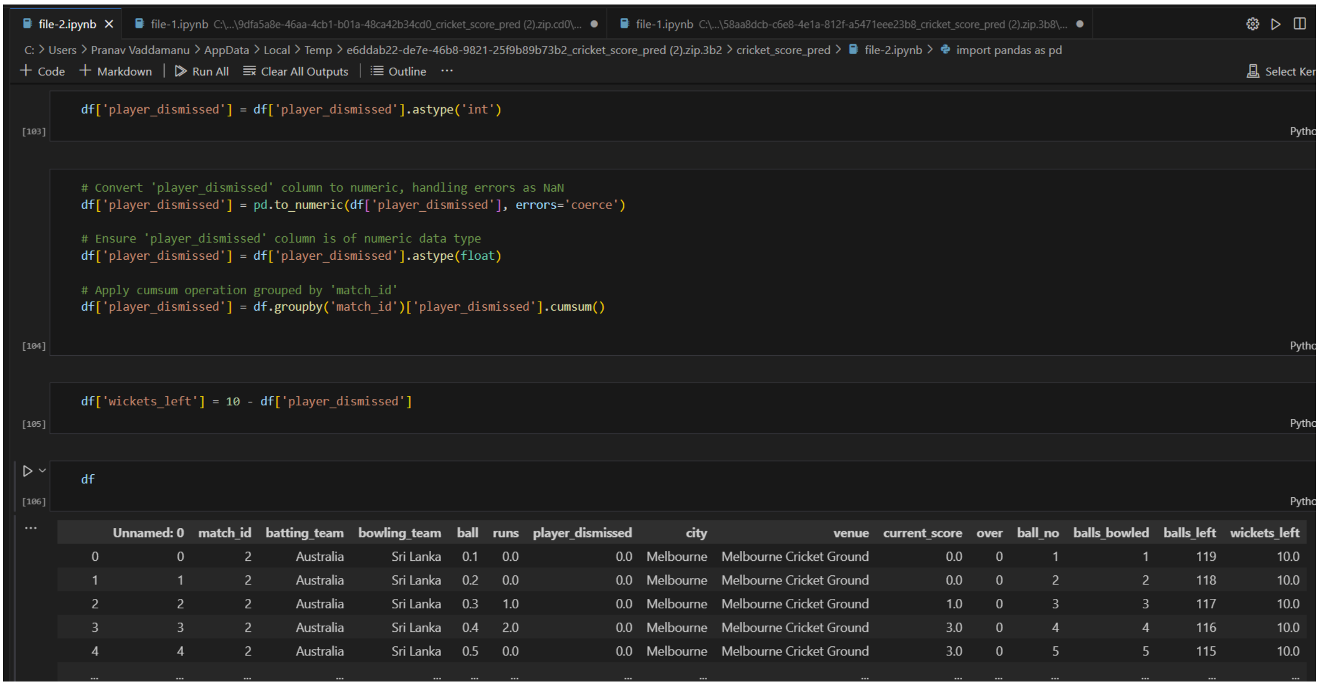
File Edit Selection View Go Run Terminal Help
Search
file-2.ipynb
file-1.ipynb C:\...\9dfa5a8e-46aa-4cb1-b01a-48ca42b34cd0_cricket_score_pred (2).zip.cd0\...
file-1.ipynb C:\...\58aa8dcb-c6e8-4e1a-812f-a5471eee23b8_cricket_score_pred (2).zip.3b8\...
C: > Users > Pranav Vaddamanu > AppData > Local > Temp > e6ddab22-de7e-46b8-9821-25f9b89b73b2_cricket_score_pred (2).zip.3b2 > cricket_score_pred > file-2.ipynb > import pandas as pd
+ Code + Markdown | Run All Clear All Outputs | Outline ...
Select

```
4        0
        ..
63883    0
63884    0
63885    1
63886    0
63887    0
Name: player_dismissed, Length: 50501, dtype: object
```

```
[101]
df['player_dismissed'] = df['player_dismissed'].apply(lambda x: 1 if x != '0' else 0)
```

```
[102]
df
```

	Unnamed: 0	match_id	batting_team	bowling_team	ball	runs	player_dismissed	city	venue	current_score	over	ball_no	balls_bowled	balls_left
0	0	2	Australia	Sri Lanka	0.1	0.0	0	Melbourne	Melbourne Cricket Ground	0.0	0	1	1	119
1	1	2	Australia	Sri Lanka	0.2	0.0	0	Melbourne	Melbourne Cricket Ground	0.0	0	2	2	118
2	2	2	Australia	Sri Lanka	0.3	1.0	0	Melbourne	Melbourne Cricket Ground	1.0	0	3	3	117
3	3	2	Australia	Sri Lanka	0.4	2.0	0	Melbourne	Melbourne Cricket Ground	3.0	0	4	4	116
4	4	2	Australia	Sri Lanka	0.5	0.0	0	Melbourne	Melbourne Cricket Ground	3.0	0	5	5	115
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
63883	115320	964	Sri Lanka	Australia	19.3	1.0	0	Colombo	R Premadasa Stadium	125.0	19	3	117	3
63884	115321	964	Sri Lanka	Australia	19.4	0.0	0	Colombo	R Premadasa Stadium	125.0	19	4	118	2
63885	115322	964	Sri Lanka	Australia	19.5	0.0	1	Colombo	R Premadasa Stadium	125.0	19	5	119	1
63886	115323	964	Sri Lanka	Australia	19.6	2.0	0	Colombo	R Premadasa Stadium	127.0	19	6	120	0
63887	115324	964	Sri Lanka	Australia	19.7	1.0	0	Colombo	R Premadasa Stadium	128.0	19	7	121	0


file-2.ipynb ✕
file-1.ipynb C:\...\9dfa5a8e-46aa-4cb1-b01a-48ca42b34cd0_cricket_score_pred (2).zip.cd0\... ●
file-1.ipynb C:\...\58aa8dcb-c6e8-4e1a-812f-a5471eee23b8_cricket_score_pred (2).zip.3b8\... ●
C: > Users > Pranav Vaddamanu > AppData > Local > Temp > e6ddab22-de7e-46b8-9821-25f9b89b73b2_cricket_score_pred (2).zip.3b2 > cricket_score_pred > file-2.ipynb > import pandas as pd
+ Code + Markdown | Run All Clear All Outputs | Outline ···
Select Ker

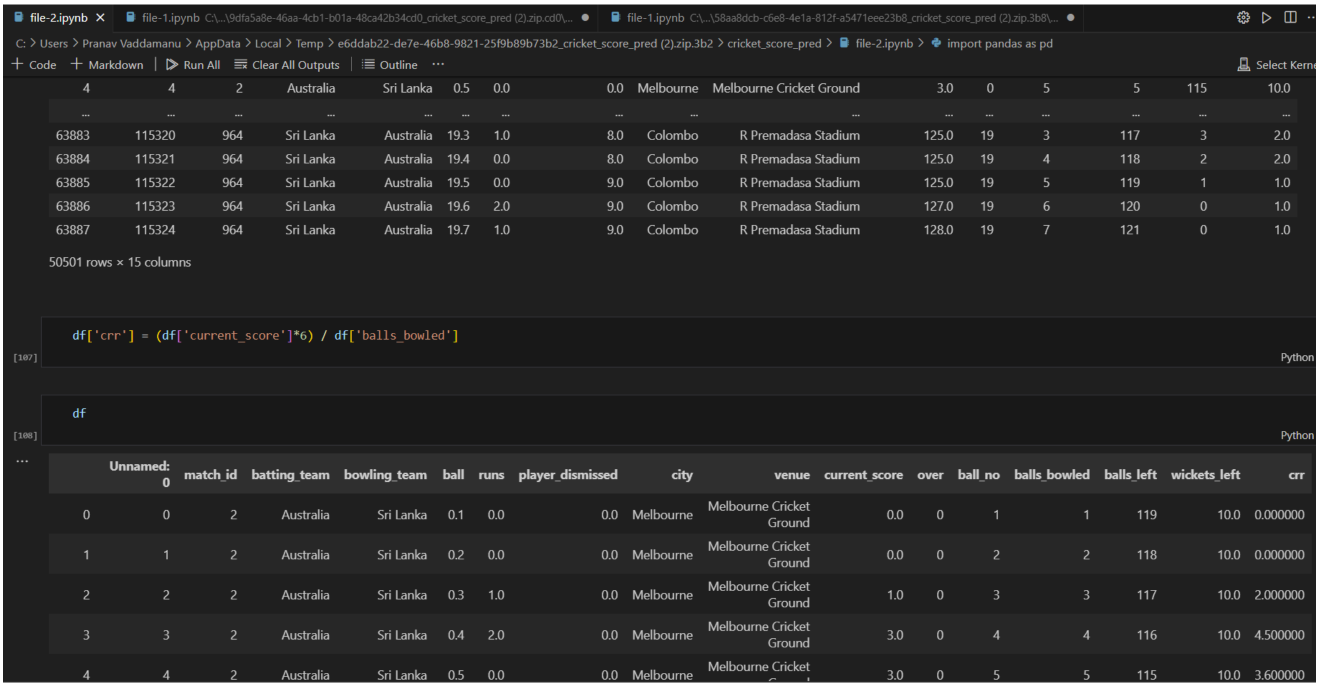
```
df['player_dismissed'] = df['player_dismissed'].astype('int')
```

```
# Convert 'player_dismissed' column to numeric, handling errors as NaN
df['player_dismissed'] = pd.to_numeric(df['player_dismissed'], errors='coerce')

# Ensure 'player_dismissed' column is of numeric data type
df['player_dismissed'] = df['player_dismissed'].astype(float)

# Apply cumsum operation grouped by 'match_id'
df['player_dismissed'] = df.groupby('match_id')['player_dismissed'].cumsum()
```

```
df['wickets_left'] = 10 - df['player_dismissed']
```

```
df
```

	Unnamed: 0	match_id	batting_team	bowling_team	ball	runs	player_dismissed	city	venue	current_score	over	ball_no	balls_bowled	balls_left	wickets_left
0	0	2	Australia	Sri Lanka	0.1	0.0	0.0	Melbourne	Melbourne Cricket Ground	0.0	0	1	1	119	10.0
1	1	2	Australia	Sri Lanka	0.2	0.0	0.0	Melbourne	Melbourne Cricket Ground	0.0	0	2	2	118	10.0
2	2	2	Australia	Sri Lanka	0.3	1.0	0.0	Melbourne	Melbourne Cricket Ground	1.0	0	3	3	117	10.0
3	3	2	Australia	Sri Lanka	0.4	2.0	0.0	Melbourne	Melbourne Cricket Ground	3.0	0	4	4	116	10.0
4	4	2	Australia	Sri Lanka	0.5	0.0	0.0	Melbourne	Melbourne Cricket Ground	3.0	0	5	5	115	10.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...


file-2.ipynb ✕
file-1.ipynb C:\...\9dfa5a8e-46aa-4cb1-b01a-48ca42b34cd0_cricket_score_pred (2).zip.cd0\... ●
file-1.ipynb C:\...\58aa8dcb-c6e8-4e1a-812f-a5471eee23b8_cricket_score_pred (2).zip.3b8\... ●
C: > Users > Pranav Vaddamanu > AppData > Local > Temp > e6ddab22-de7e-46b8-9821-25f9b89b73b2_cricket_score_pred (2).zip.3b2 > cricket_score_pred > file-2.ipynb > import pandas as pd
+ Code + Markdown | Run All Clear All Outputs | Outline ···
Select Kerne

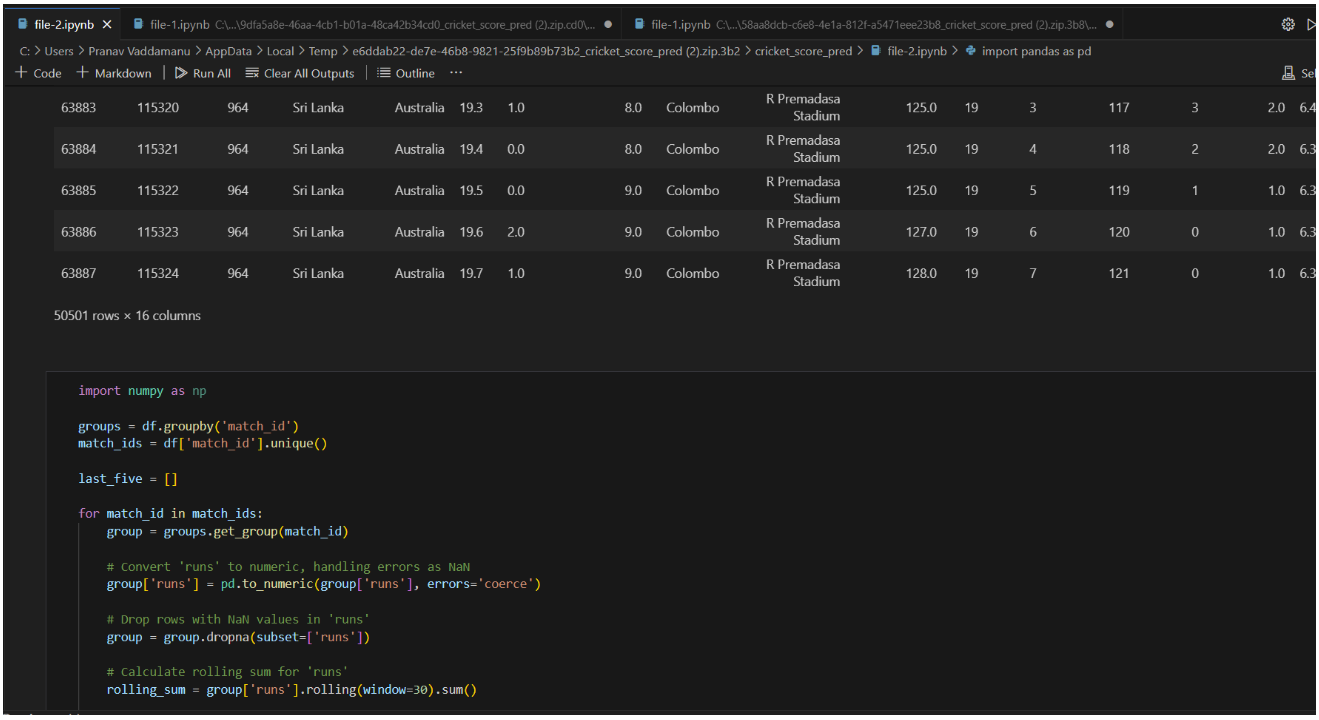
4	4	2	Australia	Sri Lanka	0.5	0.0	0.0	Melbourne	Melbourne Cricket Ground	3.0	0	5	5	115	10.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
63883	115320	964	Sri Lanka	Australia	19.3	1.0	8.0	Colombo	R Premadasa Stadium	125.0	19	3	117	3	2.0
63884	115321	964	Sri Lanka	Australia	19.4	0.0	8.0	Colombo	R Premadasa Stadium	125.0	19	4	118	2	2.0
63885	115322	964	Sri Lanka	Australia	19.5	0.0	9.0	Colombo	R Premadasa Stadium	125.0	19	5	119	1	1.0
63886	115323	964	Sri Lanka	Australia	19.6	2.0	9.0	Colombo	R Premadasa Stadium	127.0	19	6	120	0	1.0
63887	115324	964	Sri Lanka	Australia	19.7	1.0	9.0	Colombo	R Premadasa Stadium	128.0	19	7	121	0	1.0

50501 rows × 15 columns

```
df['crr'] = (df['current_score']*6) / df['balls_bowled']
```

```
df
```

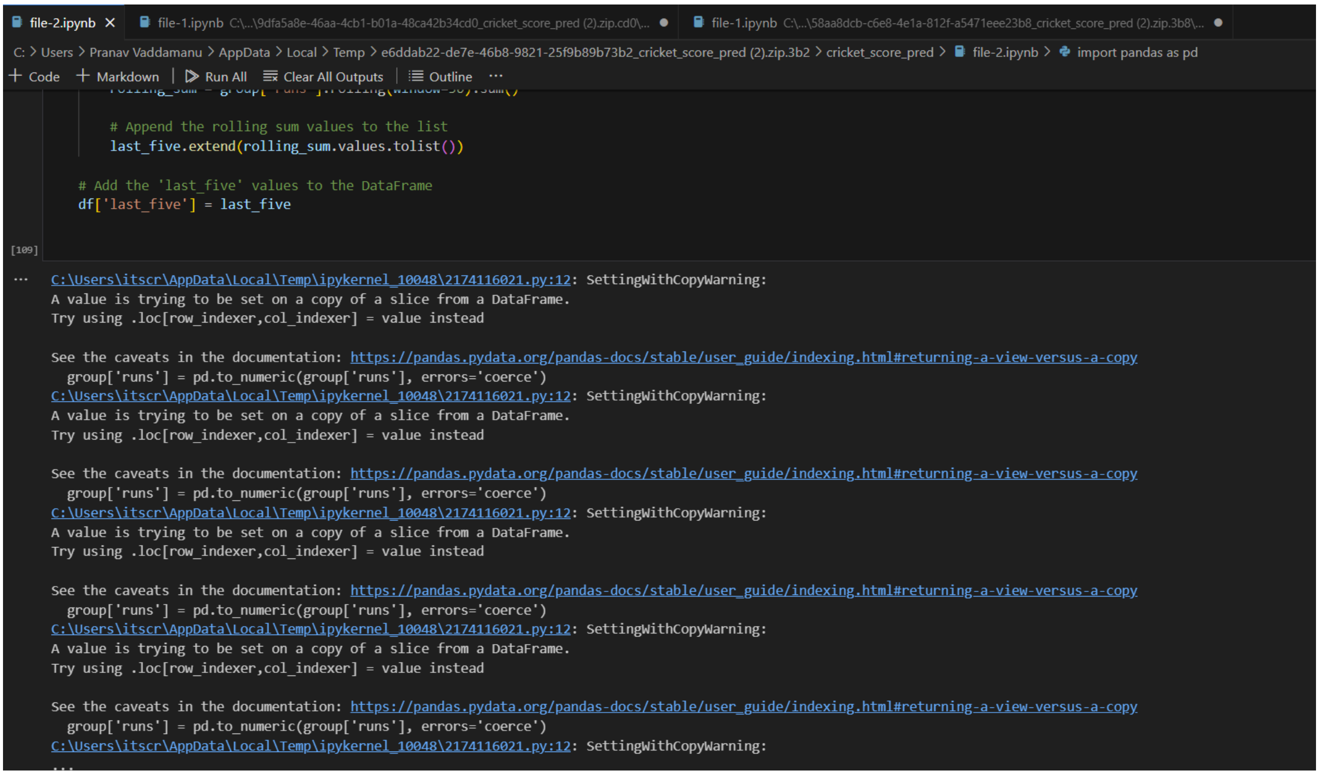
	Unnamed: 0	match_id	batting_team	bowling_team	ball	runs	player_dismissed	city	venue	current_score	over	ball_no	balls_bowled	balls_left	wickets_left	crr
0	0	2	Australia	Sri Lanka	0.1	0.0	0.0	Melbourne	Melbourne Cricket Ground	0.0	0	1	1	119	10.0	0.000000
1	1	2	Australia	Sri Lanka	0.2	0.0	0.0	Melbourne	Melbourne Cricket Ground	0.0	0	2	2	118	10.0	0.000000
2	2	2	Australia	Sri Lanka	0.3	1.0	0.0	Melbourne	Melbourne Cricket Ground	1.0	0	3	3	117	10.0	2.000000
3	3	2	Australia	Sri Lanka	0.4	2.0	0.0	Melbourne	Melbourne Cricket Ground	3.0	0	4	4	116	10.0	4.500000
4	4	2	Australia	Sri Lanka	0.5	0.0	0.0	Melbourne	Melbourne Cricket	3.0	0	5	5	115	10.0	3.600000


file-2.ipynb ×
file-1.ipynb C:\...\9dfa5a8e-46aa-4cb1-b01a-48ca42b34cd0_cricket_score_pred (2).zip.cd0\...
file-1.ipynb C:\...\58aa8dcb-c6e8-4e1a-812f-a5471eee23b8_cricket_score_pred (2).zip.3b8\...
C: > Users > Pranav Vaddamanu > AppData > Local > Temp > e6ddab22-de7e-46b8-9821-25f9b89b73b2_cricket_score_pred (2).zip.3b2 > cricket_score_pred > file-2.ipynb > import pandas as pd
+ Code + Markdown | Run All Clear All Outputs | Outline ···

63883	115320	964	Sri Lanka	Australia	19.3	1.0	8.0	Colombo	R Premadasa Stadium	125.0	19	3	117	3	2.0	6.4
63884	115321	964	Sri Lanka	Australia	19.4	0.0	8.0	Colombo	R Premadasa Stadium	125.0	19	4	118	2	2.0	6.3
63885	115322	964	Sri Lanka	Australia	19.5	0.0	9.0	Colombo	R Premadasa Stadium	125.0	19	5	119	1	1.0	6.3
63886	115323	964	Sri Lanka	Australia	19.6	2.0	9.0	Colombo	R Premadasa Stadium	127.0	19	6	120	0	1.0	6.3
63887	115324	964	Sri Lanka	Australia	19.7	1.0	9.0	Colombo	R Premadasa Stadium	128.0	19	7	121	0	1.0	6.3

50501 rows × 16 columns

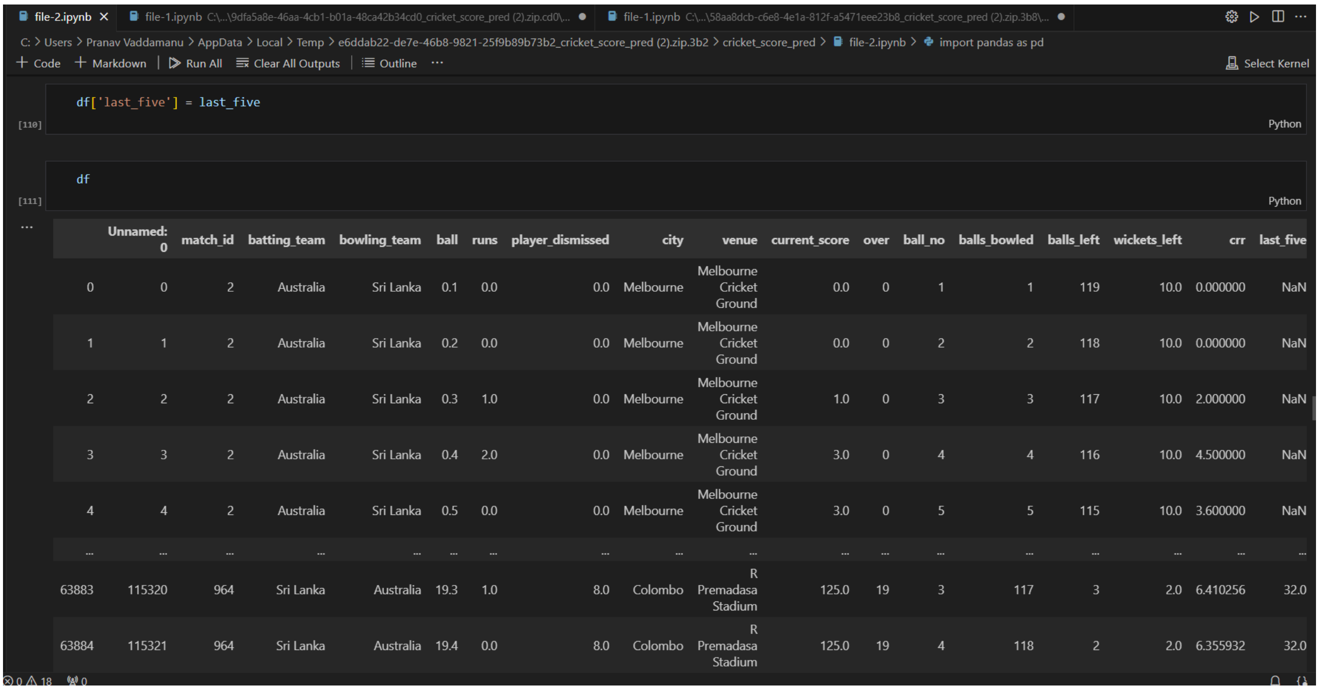
```
import numpy as np

groups = df.groupby('match_id')
match_ids = df['match_id'].unique()

last_five = []

for match_id in match_ids:
    group = groups.get_group(match_id)

    # Convert 'runs' to numeric, handling errors as NaN
    group['runs'] = pd.to_numeric(group['runs'], errors='coerce')

    # Drop rows with NaN values in 'runs'
    group = group.dropna(subset=['runs'])

    # Calculate rolling sum for 'runs'
    rolling_sum = group['runs'].rolling(window=30).sum()
```


file-2.ipynb ×
file-1.ipynb C:\...\9dfa5a8e-46aa-4cb1-b01a-48ca42b34cd0_cricket_score_pred (2).zip.cd0\...
file-1.ipynb C:\...\58aa8dcb-c6e8-4e1a-812f-a5471eee23b8_cricket_score_pred (2).zip.3b8\...
C: > Users > Pranav Vaddamanu > AppData > Local > Temp > e6ddab22-de7e-46b8-9821-25f9b89b73b2_cricket_score_pred (2).zip.3b2 > cricket_score_pred > file-2.ipynb > import pandas as pd
+ Code + Markdown | Run All Clear All Outputs | Outline ···

```
    rolling_sum = group['runs'].rolling(window=30).sum()

    # Append the rolling sum values to the list
    last_five.extend(rolling_sum.values.tolist())

# Add the 'last_five' values to the DataFrame
df['last_five'] = last_five
```

[109]

```
C:\Users\itscr\AppData\Local\Temp\ipykernel_10048\2174116021.py:12: SettingWithCopyWarning: 
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  group['runs'] = pd.to_numeric(group['runs'], errors='coerce')
C:\Users\itscr\AppData\Local\Temp\ipykernel_10048\2174116021.py:12: SettingWithCopyWarning: 
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  group['runs'] = pd.to_numeric(group['runs'], errors='coerce')
C:\Users\itscr\AppData\Local\Temp\ipykernel_10048\2174116021.py:12: SettingWithCopyWarning: 
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  group['runs'] = pd.to_numeric(group['runs'], errors='coerce')
C:\Users\itscr\AppData\Local\Temp\ipykernel_10048\2174116021.py:12: SettingWithCopyWarning: 
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  group['runs'] = pd.to_numeric(group['runs'], errors='coerce')
C:\Users\itscr\AppData\Local\Temp\ipykernel_10048\2174116021.py:12: SettingWithCopyWarning: 
...
```


file-2.ipynb ×
file-1.ipynb C:\...\9dfa5a8e-46aa-4cb1-b01a-48ca42b34cd0_cricket_score_pred (2).zip.cd0\... ●
file-1.ipynb C:\...\58aa8dcb-c6e8-4e1a-812f-a5471eee23b8_cricket_score_pred (2).zip.3b8\... ●
C: > Users > Pranav Vaddamanu > AppData > Local > Temp > e6ddab22-de7e-46b8-9821-25f9b89b73b2_cricket_score_pred (2).zip.3b2 > cricket_score_pred > file-2.ipynb > import pandas as pd
+ Code + Markdown | Run All Clear All Outputs | Outline ···
Select Kernel

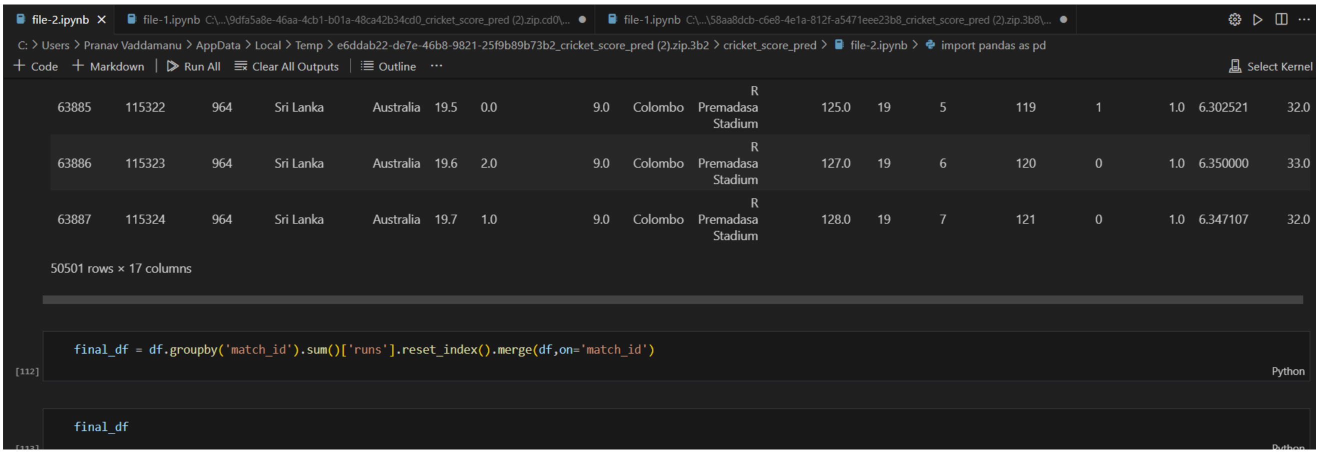
```
df['last_five'] = last_five
```

[110] Python

```
df
```

[111] Python

	Unnamed: 0	match_id	batting_team	bowling_team	ball	runs	player_dismissed	city	venue	current_score	over	ball_no	balls_bowled	balls_left	wickets_left	crr	last_five
0	0	2	Australia	Sri Lanka	0.1	0.0	0.0	Melbourne	Melbourne Cricket Ground	0.0	0	1	1	119	10.0	0.000000	NaN
1	1	2	Australia	Sri Lanka	0.2	0.0	0.0	Melbourne	Melbourne Cricket Ground	0.0	0	2	2	118	10.0	0.000000	NaN
2	2	2	Australia	Sri Lanka	0.3	1.0	0.0	Melbourne	Melbourne Cricket Ground	1.0	0	3	3	117	10.0	2.000000	NaN
3	3	2	Australia	Sri Lanka	0.4	2.0	0.0	Melbourne	Melbourne Cricket Ground	3.0	0	4	4	116	10.0	4.500000	NaN
4	4	2	Australia	Sri Lanka	0.5	0.0	0.0	Melbourne	Melbourne Cricket Ground	3.0	0	5	5	115	10.0	3.600000	NaN
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
63883	115320	964	Sri Lanka	Australia	19.3	1.0	8.0	Colombo	R Premadasa Stadium	125.0	19	3	117	3	2.0	6.410256	32.0
63884	115321	964	Sri Lanka	Australia	19.4	0.0	8.0	Colombo	R Premadasa Stadium	125.0	19	4	118	2	2.0	6.355932	32.0
63885	115322	964	Sri Lanka	Australia	19.5	0.0	9.0	Colombo	R Premadasa Stadium	125.0	19	5	119	1	1.0	6.302521	32.0
63886	115323	964	Sri Lanka	Australia	19.6	2.0	9.0	Colombo	R Premadasa Stadium	127.0	19	6	120	0	1.0	6.350000	33.0
63887	115324	964	Sri Lanka	Australia	19.7	1.0	9.0	Colombo	R Premadasa Stadium	128.0	19	7	121	0	1.0	6.347107	32.0


file-2.ipynb ×
file-1.ipynb C:\...\9dfa5a8e-46aa-4cb1-b01a-48ca42b34cd0_cricket_score_pred (2).zip.cd0\... ●
file-1.ipynb C:\...\58aa8dcb-c6e8-4e1a-812f-a5471eee23b8_cricket_score_pred (2).zip.3b8\... ●
C: > Users > Pranav Vaddamanu > AppData > Local > Temp > e6ddab22-de7e-46b8-9821-25f9b89b73b2_cricket_score_pred (2).zip.3b2 > cricket_score_pred > file-2.ipynb > import pandas as pd
+ Code + Markdown | Run All Clear All Outputs | Outline ···
Select Kernel

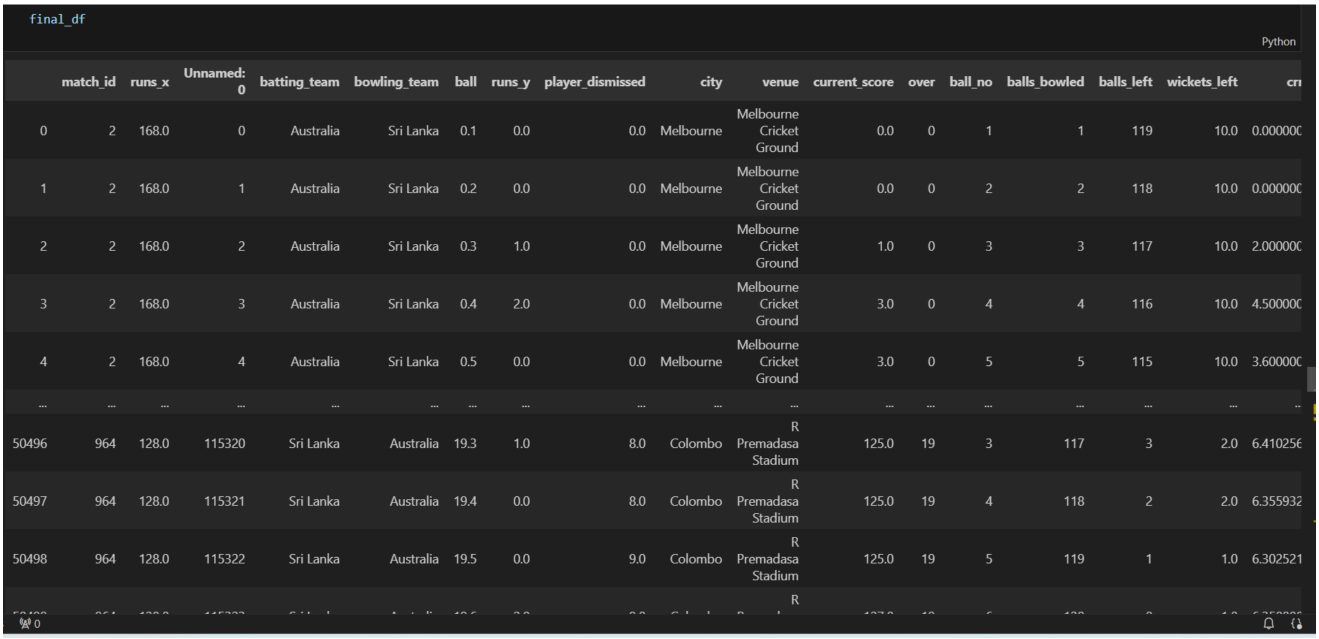
50501 rows × 17 columns

```
final_df = df.groupby('match_id').sum()['runs'].reset_index().merge(df,on='match_id')
```

[112] Python

```
final_df
```

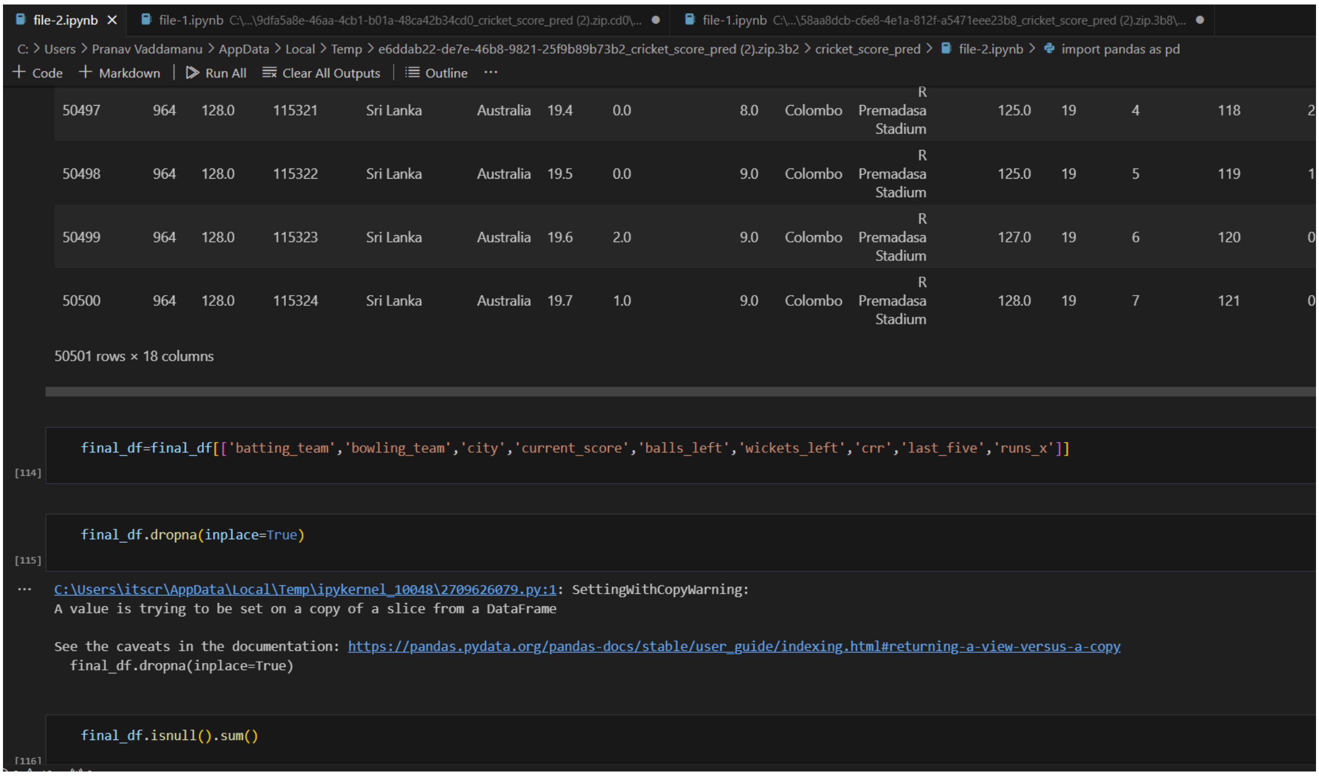
[113] Python



```
final_df
```

Python

	match_id	runs_x	Unnamed: 0	batting_team	bowling_team	ball	runs_y	player_dismissed	city	venue	current_score	over	ball_no	balls_bowled	balls_left	wickets_left	cr
0	2	168.0	0	Australia	Sri Lanka	0.1	0.0	0.0	Melbourne	Melbourne Cricket Ground	0.0	0	1	1	119	10.0	0.000000
1	2	168.0	1	Australia	Sri Lanka	0.2	0.0	0.0	Melbourne	Melbourne Cricket Ground	0.0	0	2	2	118	10.0	0.000000
2	2	168.0	2	Australia	Sri Lanka	0.3	1.0	0.0	Melbourne	Melbourne Cricket Ground	1.0	0	3	3	117	10.0	2.000000
3	2	168.0	3	Australia	Sri Lanka	0.4	2.0	0.0	Melbourne	Melbourne Cricket Ground	3.0	0	4	4	116	10.0	4.500000
4	2	168.0	4	Australia	Sri Lanka	0.5	0.0	0.0	Melbourne	Melbourne Cricket Ground	3.0	0	5	5	115	10.0	3.600000
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
50496	964	128.0	115320	Sri Lanka	Australia	19.3	1.0	8.0	Colombo	R Premadasa Stadium	125.0	19	3	117	3	2.0	6.410256
50497	964	128.0	115321	Sri Lanka	Australia	19.4	0.0	8.0	Colombo	R Premadasa Stadium	125.0	19	4	118	2	2.0	6.355932
50498	964	128.0	115322	Sri Lanka	Australia	19.5	0.0	9.0	Colombo	R Premadasa Stadium	125.0	19	5	119	1	1.0	6.302521


file-2.ipynb ✕
file-1.ipynb C:\...\9dfa5a8e-46aa-4cb1-b01a-48ca42b34cd0_cricket_score_pred (2).zip.cd0\... ●
file-1.ipynb C:\...\58aa8dcb-c6e8-4e1a-812f-a5471eee23b8_cricket_score_pred (2).zip.3b8\... ●
C: > Users > Pranav Vaddamanu > AppData > Local > Temp > e6ddab22-de7e-46b8-9821-25f9b89b73b2_cricket_score_pred (2).zip.3b2 > cricket_score_pred > file-2.ipynb > import pandas as pd
+ Code + Markdown | Run All Clear All Outputs | Outline ···

50497	964	128.0	115321	Sri Lanka	Australia	19.4	0.0	8.0	Colombo	R Premadasa Stadium	125.0	19	4	118	2
50498	964	128.0	115322	Sri Lanka	Australia	19.5	0.0	9.0	Colombo	R Premadasa Stadium	125.0	19	5	119	1
50499	964	128.0	115323	Sri Lanka	Australia	19.6	2.0	9.0	Colombo	R Premadasa Stadium	127.0	19	6	120	0
50500	964	128.0	115324	Sri Lanka	Australia	19.7	1.0	9.0	Colombo	R Premadasa Stadium	128.0	19	7	121	0

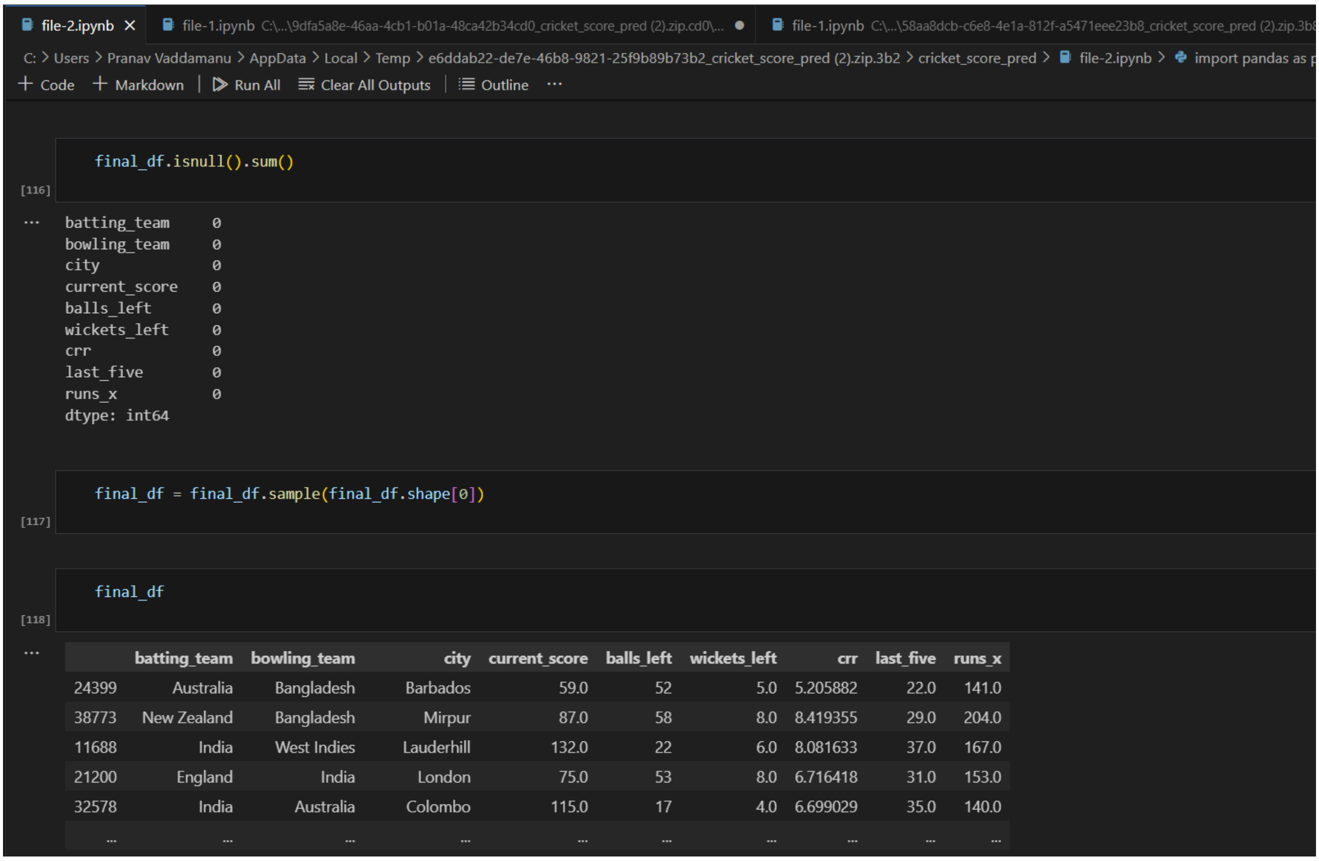
50501 rows × 18 columns

[114]
```
final_df=final_df[['batting_team','bowling_team','city','current_score','balls_left','wickets_left','crr','last_five','runs_x']]
```

[115]
```
final_df.dropna(inplace=True)
```

```
C:\Users\itscr\AppData\Local\Temp\ipykernel_10048\2709626079.py:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  final_df.dropna(inplace=True)
```

[116]
```
final_df.isnull().sum()
```


file-2.ipynb ✕
file-1.ipynb C:\...\9dfa5a8e-46aa-4cb1-b01a-48ca42b34cd0_cricket_score_pred (2).zip.cd0\... ●
file-1.ipynb C:\...\58aa8dcb-c6e8-4e1a-812f-a5471eee23b8_cricket_score_pred (2).zip.3b8
C: > Users > Pranav Vaddamanu > AppData > Local > Temp > e6ddab22-de7e-46b8-9821-25f9b89b73b2_cricket_score_pred (2).zip.3b2 > cricket_score_pred > file-2.ipynb > import pandas as p
+ Code + Markdown | Run All Clear All Outputs | Outline ···

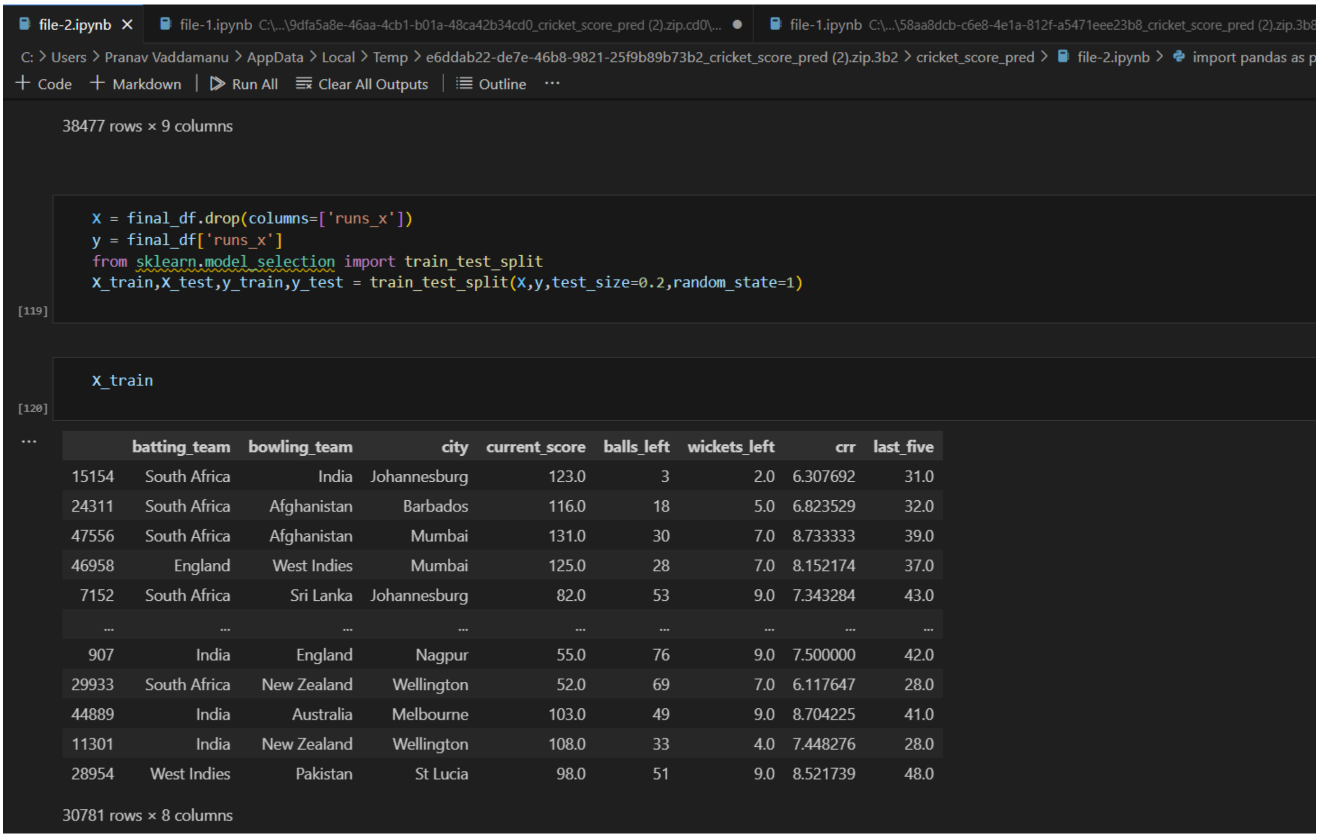
[116]
```
final_df.isnull().sum()
```

```
batting_team     0
bowling_team     0
city             0
current_score    0
balls_left       0
wickets_left     0
crr              0
last_five        0
runs_x           0
dtype: int64
```

[117]
```
final_df = final_df.sample(final_df.shape[0])
```

[118]
```
final_df
```

	batting_team	bowling_team	city	current_score	balls_left	wickets_left	crr	last_five	runs_x
24399	Australia	Bangladesh	Barbados	59.0	52	5.0	5.205882	22.0	141.0
38773	New Zealand	Bangladesh	Mirpur	87.0	58	8.0	8.419355	29.0	204.0
11688	India	West Indies	Lauderhill	132.0	22	6.0	8.081633	37.0	167.0
21200	England	India	London	75.0	53	8.0	6.716418	31.0	153.0
32578	India	Australia	Colombo	115.0	17	4.0	6.699029	35.0	140.0
...	...	...	...	...	...	...	...	...	...


file-2.ipynb ✕
file-1.ipynb C:\...\9dfa5a8e-46aa-4cb1-b01a-48ca42b34cd0_cricket_score_pred (2).zip.cd0\... ●
file-1.ipynb C:\...\58aa8dcb-c6e8-4e1a-812f-a5471eee23b8_cricket_score_pred (2).zip.3b8
C: > Users > Pranav Vaddamanu > AppData > Local > Temp > e6ddab22-de7e-46b8-9821-25f9b89b73b2_cricket_score_pred (2).zip.3b2 > cricket_score_pred > file-2.ipynb > import pandas as p
+ Code + Markdown | Run All Clear All Outputs | Outline ···
[119]
[120]
···

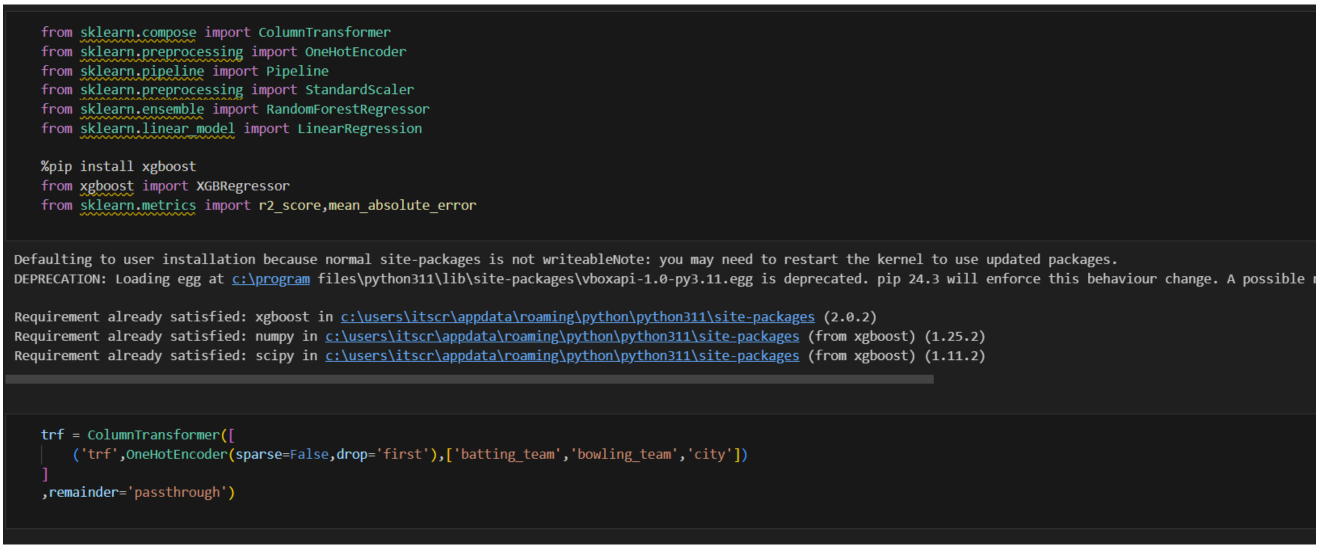
```
38477 rows × 9 columns
```

```python
X = final_df.drop(columns=['runs_x'])
y = final_df['runs_x']
from sklearn.model_selection import train_test_split
X_train,X_test,y_train,y_test = train_test_split(X,y,test_size=0.2,random_state=1)
```

```python
X_train
```

	batting_team	bowling_team	city	current_score	balls_left	wickets_left	crr	last_five
15154	South Africa	India	Johannesburg	123.0	3	2.0	6.307692	31.0
24311	South Africa	Afghanistan	Barbados	116.0	18	5.0	6.823529	32.0
47556	South Africa	Afghanistan	Mumbai	131.0	30	7.0	8.733333	39.0
46958	England	West Indies	Mumbai	125.0	28	7.0	8.152174	37.0
7152	South Africa	Sri Lanka	Johannesburg	82.0	53	9.0	7.343284	43.0
...	...	...	...	...	...	...	...	...
907	India	England	Nagpur	55.0	76	9.0	7.500000	42.0
29933	South Africa	New Zealand	Wellington	52.0	69	7.0	6.117647	28.0
44889	India	Australia	Melbourne	103.0	49	9.0	8.704225	41.0
11301	India	New Zealand	Wellington	108.0	33	4.0	7.448276	28.0
28954	West Indies	Pakistan	St Lucia	98.0	51	9.0	8.521739	48.0

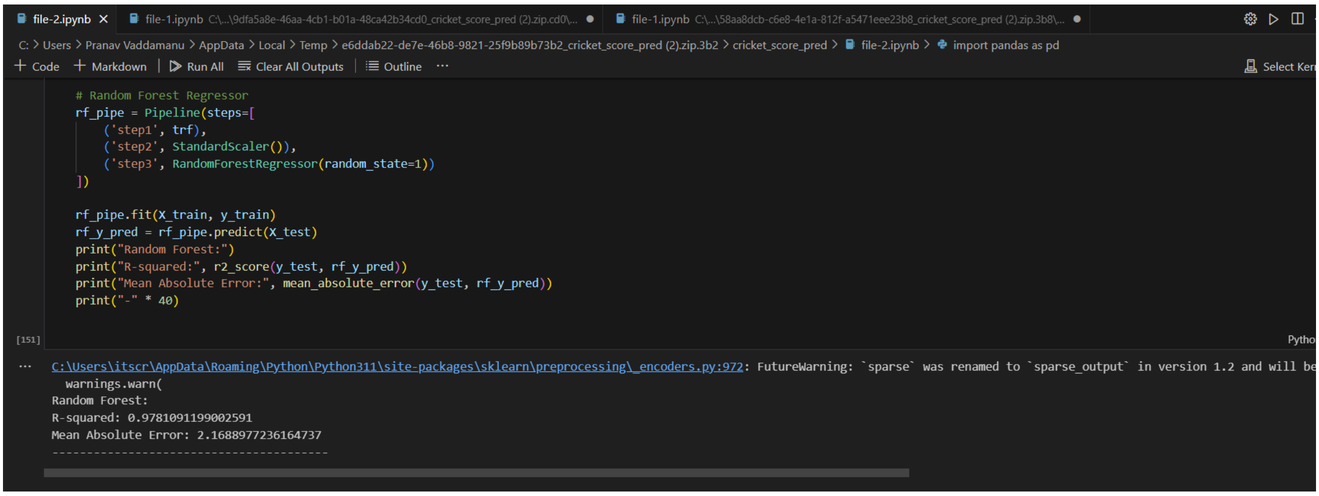
```
30781 rows × 8 columns
```



```python
from sklearn.compose import ColumnTransformer
from sklearn.preprocessing import OneHotEncoder
from sklearn.pipeline import Pipeline
from sklearn.preprocessing import StandardScaler
from sklearn.ensemble import RandomForestRegressor
from sklearn.linear_model import LinearRegression

%pip install xgboost
from xgboost import XGBRegressor
from sklearn.metrics import r2_score,mean_absolute_error
```

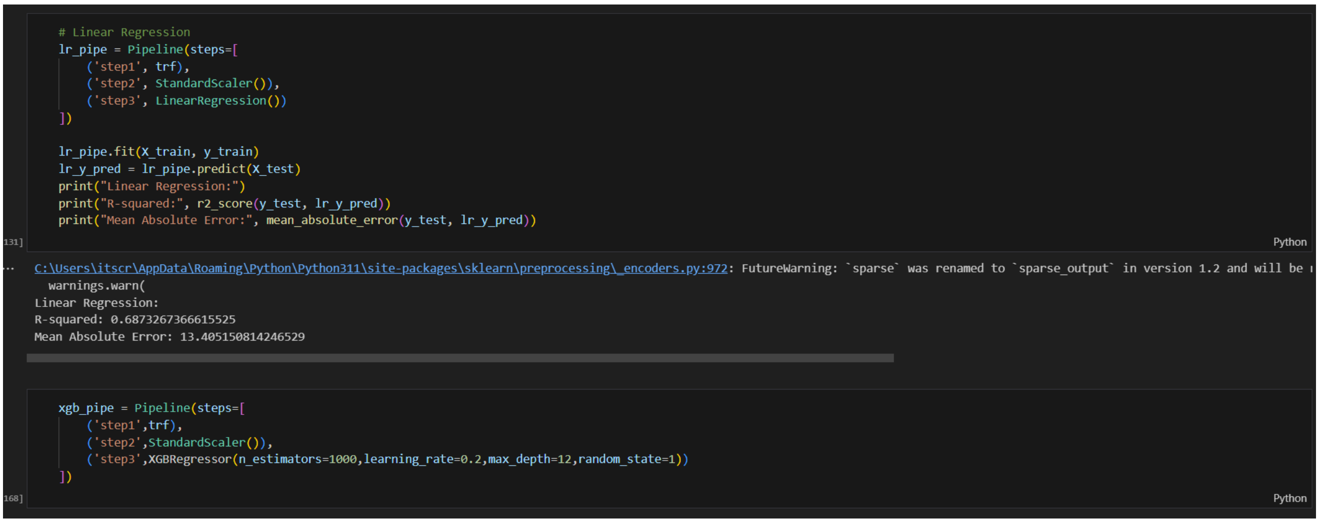
```
Defaulting to user installation because normal site-packages is not writeableNote: you may need to restart the kernel to use updated packages.
DEPRECATION: Loading egg at c:\program files\python311\lib\site-packages\vboxapi-1.0-py3.11.egg is deprecated. pip 24.3 will enforce this behaviour change. A possible 

Requirement already satisfied: xgboost in c:\users\itscr\appdata\roaming\python\python311\site-packages (2.0.2)
Requirement already satisfied: numpy in c:\users\itscr\appdata\roaming\python\python311\site-packages (from xgboost) (1.25.2)
Requirement already satisfied: scipy in c:\users\itscr\appdata\roaming\python\python311\site-packages (from xgboost) (1.11.2)
```

```python
trf = ColumnTransformer([
    ('trf',OneHotEncoder(sparse=False,drop='first'),['batting_team','bowling_team','city'])
]
,remainder='passthrough')
```


file-2.ipynb ✕
file-1.ipynb C:\...\9dfa5a8e-46aa-4cb1-b01a-48ca42b34cd0_cricket_score_pred (2).zip.cd0\... ●
file-1.ipynb C:\...\58aa8dcb-c6e8-4e1a-812f-a5471eee23b8_cricket_score_pred (2).zip.3b8\... ●
C: > Users > Pranav Vaddamanu > AppData > Local > Temp > e6ddab22-de7e-46b8-9821-25f9b89b73b2_cricket_score_pred (2).zip.3b2 > cricket_score_pred > file-2.ipynb > import pandas as pd
+ Code + Markdown | Run All Clear All Outputs | Outline ···
Select Ker
[151]
Pytho
···

```python
# Random Forest Regressor
rf_pipe = Pipeline(steps=[
    ('step1', trf),
    ('step2', StandardScaler()),
    ('step3', RandomForestRegressor(random_state=1))
])

rf_pipe.fit(X_train, y_train)
rf_y_pred = rf_pipe.predict(X_test)
print("Random Forest:")
print("R-squared:", r2_score(y_test, rf_y_pred))
print("Mean Absolute Error:", mean_absolute_error(y_test, rf_y_pred))
print("-" * 40)
```

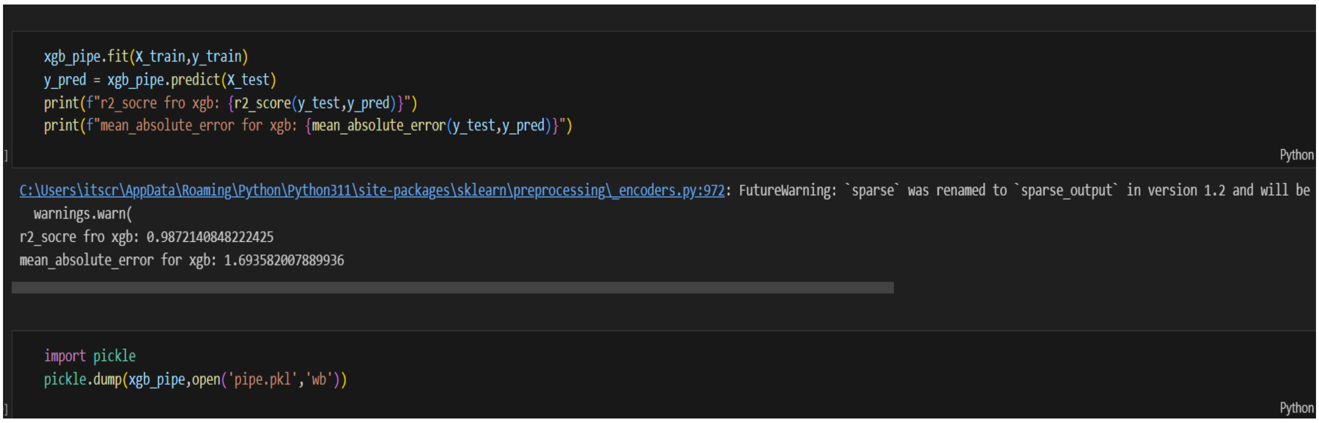
```
C:\Users\itscr\AppData\Roaming\Python\Python311\site-packages\sklearn\preprocessing\_encoders.py:972: FutureWarning: `sparse` was renamed to `sparse_output` in version 1.2 and will be
  warnings.warn(
Random Forest:
R-squared: 0.9781091199002591
Mean Absolute Error: 2.1688977236164737
----------------------------------------
```



```python
# Linear Regression
lr_pipe = Pipeline(steps=[
    ('step1', trf),
    ('step2', StandardScaler()),
    ('step3', LinearRegression())
])

lr_pipe.fit(X_train, y_train)
lr_y_pred = lr_pipe.predict(X_test)
print("Linear Regression:")
print("R-squared:", r2_score(y_test, lr_y_pred))
print("Mean Absolute Error:", mean_absolute_error(y_test, lr_y_pred))
```

```
C:\Users\itscr\AppData\Roaming\Python\Python311\site-packages\sklearn\preprocessing\_encoders.py:972: FutureWarning: `sparse` was renamed to `sparse_output` in version 1.2 and will be
  warnings.warn(
Linear Regression:
R-squared: 0.6873267366615525
Mean Absolute Error: 13.405150814246529
```

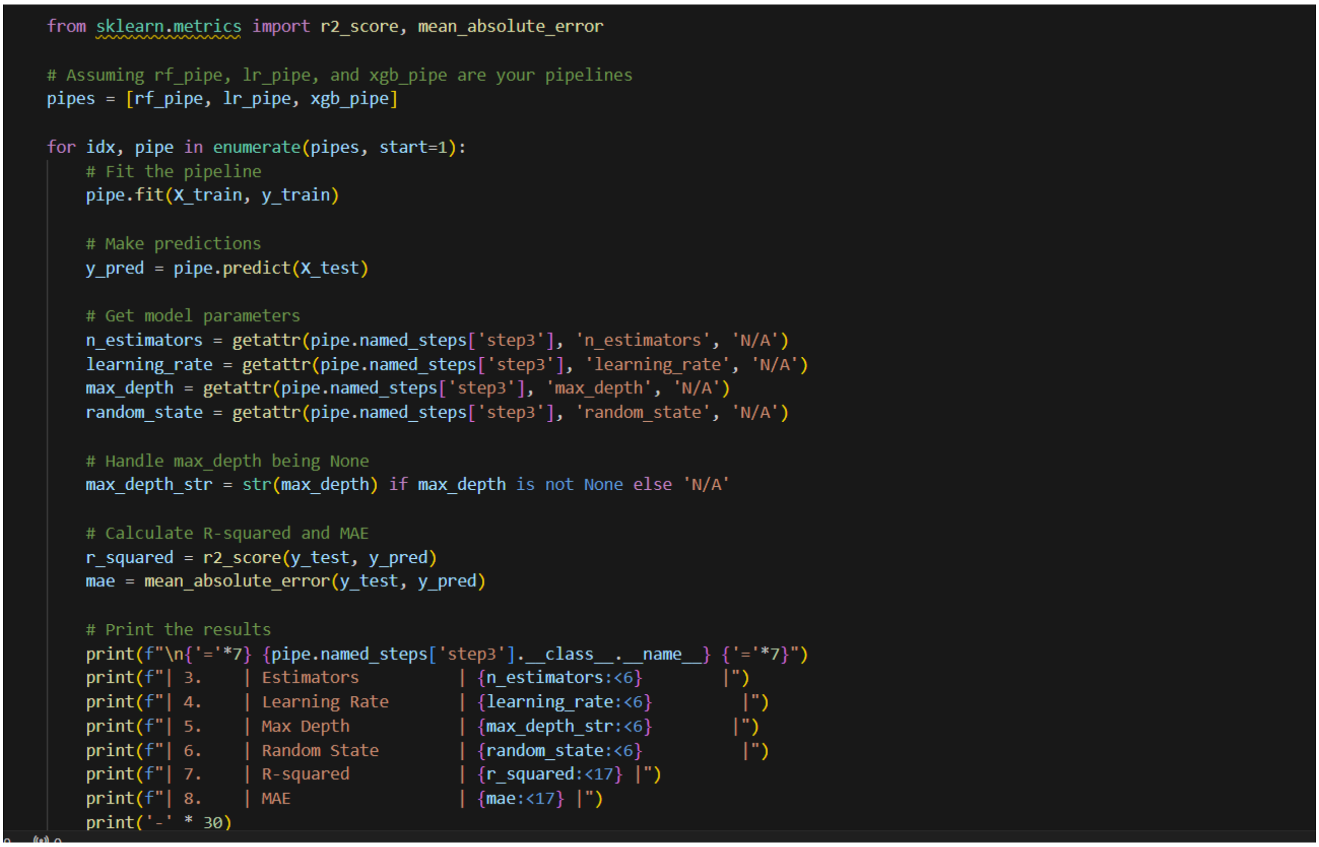
```python
xgb_pipe = Pipeline(steps=[
    ('step1',trf),
    ('step2',StandardScaler()),
    ('step3',XGBRegressor(n_estimators=1000,learning_rate=0.2,max_depth=12,random_state=1))
])
```



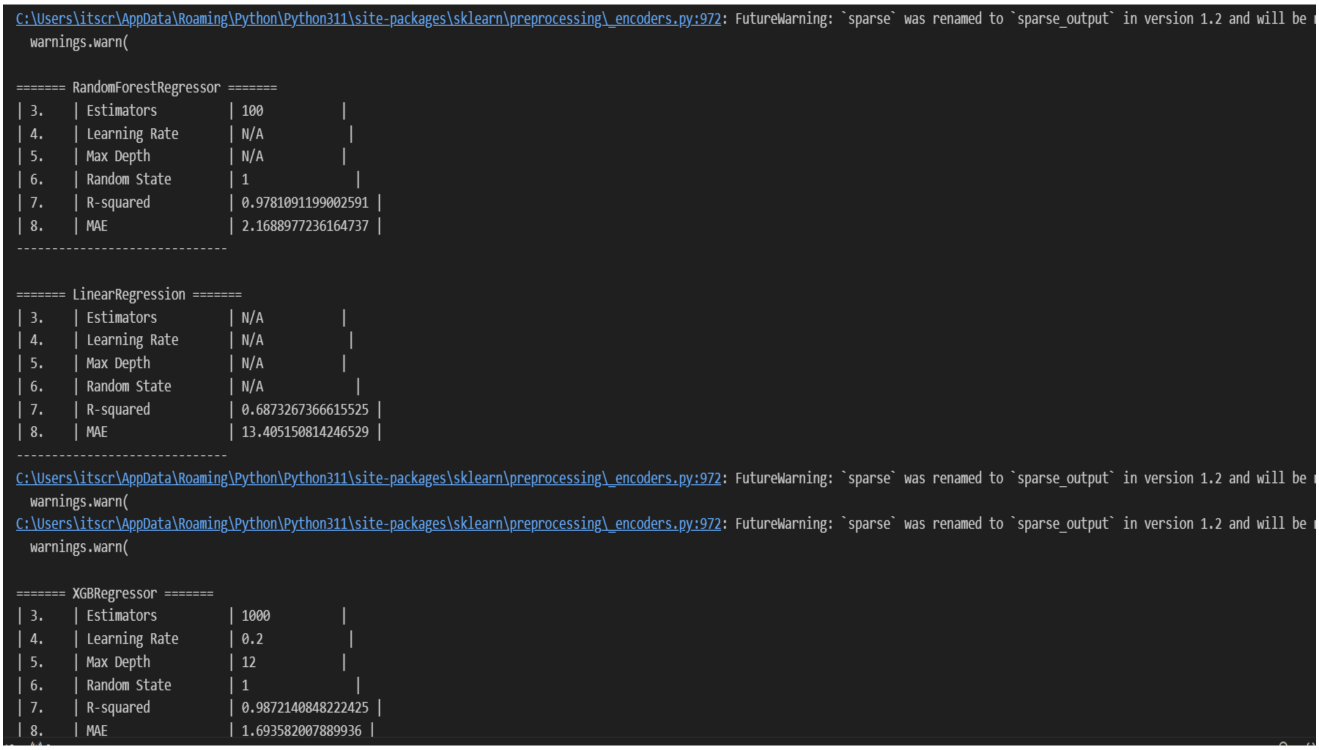
```python
xgb_pipe.fit(X_train,y_train)
y_pred = xgb_pipe.predict(X_test)
print(f"r2_socre fro xgb: {r2_score(y_test,y_pred)}")
print(f"mean_absolute_error for xgb: {mean_absolute_error(y_test,y_pred)}")
```

```
C:\Users\itscr\AppData\Roaming\Python\Python311\site-packages\sklearn\preprocessing\_encoders.py:972: FutureWarning: `sparse` was renamed to `sparse_output` in version 1.2 and will be
  warnings.warn(
r2_socre fro xgb: 0.9872140848222425
mean_absolute_error for xgb: 1.693582007889936
```

```python
import pickle
pickle.dump(xgb_pipe,open('pipe.pkl','wb'))
```

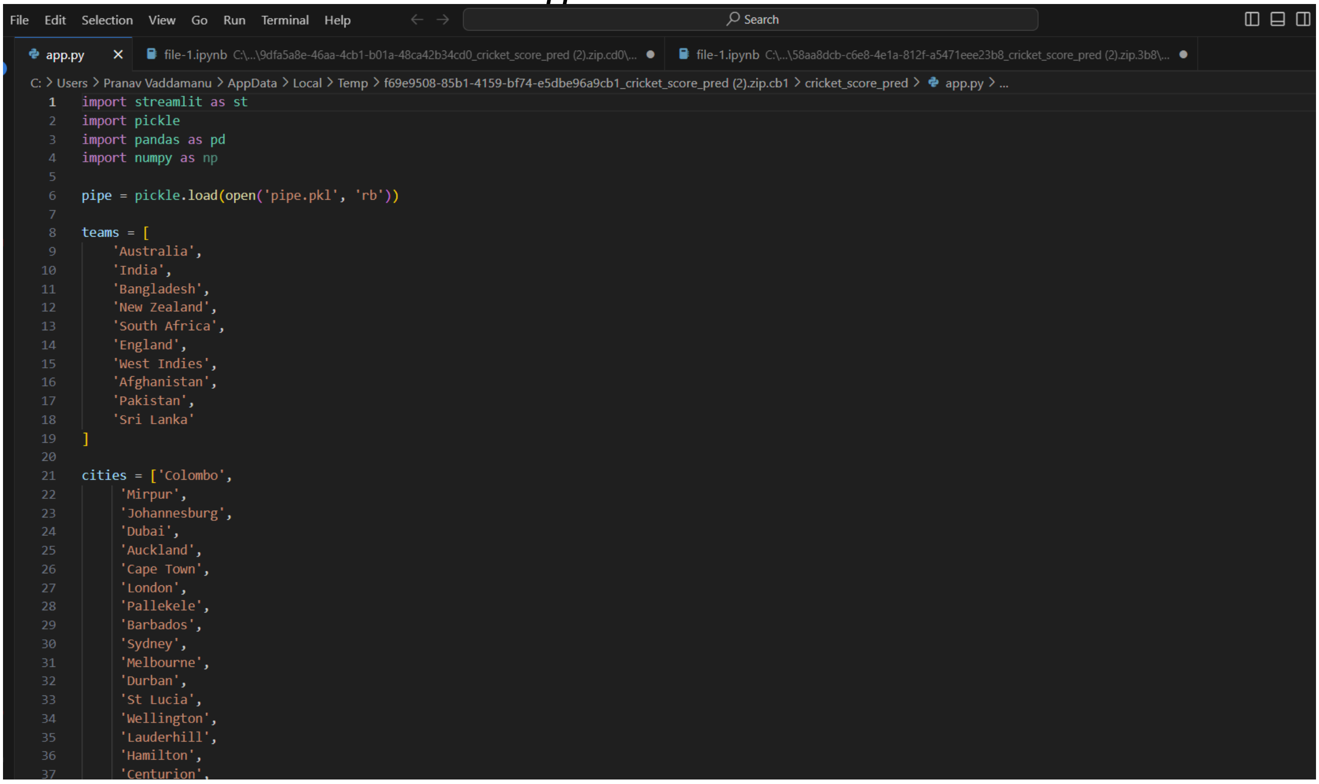


```python
from sklearn.metrics import r2_score, mean_absolute_error

# Assuming rf_pipe, lr_pipe, and xgb_pipe are your pipelines
pipes = [rf_pipe, lr_pipe, xgb_pipe]

for idx, pipe in enumerate(pipes, start=1):
    # Fit the pipeline
    pipe.fit(X_train, y_train)

    # Make predictions
    y_pred = pipe.predict(X_test)

    # Get model parameters
    n_estimators = getattr(pipe.named_steps['step3'], 'n_estimators', 'N/A')
    learning_rate = getattr(pipe.named_steps['step3'], 'learning_rate', 'N/A')
    max_depth = getattr(pipe.named_steps['step3'], 'max_depth', 'N/A')
    random_state = getattr(pipe.named_steps['step3'], 'random_state', 'N/A')

    # Handle max_depth being None
    max_depth_str = str(max_depth) if max_depth is not None else 'N/A'

    # Calculate R-squared and MAE
    r_squared = r2_score(y_test, y_pred)
    mae = mean_absolute_error(y_test, y_pred)

    # Print the results
    print(f"\n{'='*7} {pipe.named_steps['step3'].__class__.__name__} {'='*7}")
    print(f"| 3.    | Estimators          | {n_estimators:<6}        |")
    print(f"| 4.    | Learning Rate       | {learning_rate:<6}         |")
    print(f"| 5.    | Max Depth           | {max_depth_str:<6}        |")
    print(f"| 6.    | Random State        | {random_state:<6}          |")
    print(f"| 7.    | R-squared           | {r_squared:<17} |")
    print(f"| 8.    | MAE                 | {mae:<17} |")
    print('-' * 30)
```

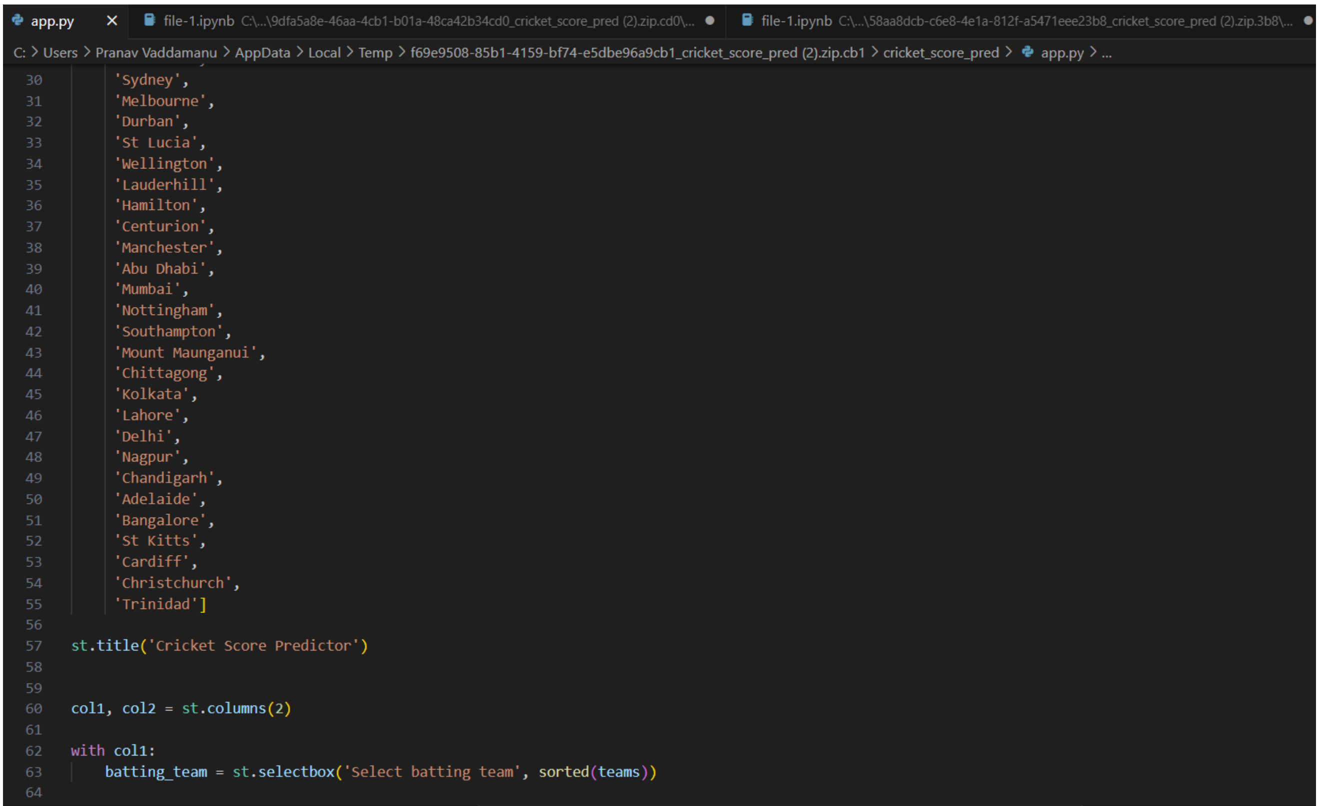


```
C:\Users\itscr\AppData\Roaming\Python\Python311\site-packages\sklearn\preprocessing\_encoders.py:972: FutureWarning: `sparse` was renamed to `sparse_output` in version 1.2 and will be
  warnings.warn(

======= RandomForestRegressor =======
| 3.    | Estimators          | 100          |
| 4.    | Learning Rate       | N/A           |
| 5.    | Max Depth           | N/A          |
| 6.    | Random State        | 1              |
| 7.    | R-squared           | 0.9781091199002591 |
| 8.    | MAE                 | 2.1688977236164737 |
----------------------------

====== LinearRegression ======
| 3.    | Estimators          | N/A          |
| 4.    | Learning Rate       | N/A           |
| 5.    | Max Depth           | N/A          |
| 6.    | Random State        | N/A            |
| 7.    | R-squared           | 0.6873267366615525 |
| 8.    | MAE                 | 13.405150814246529 |
----------------------------
C:\Users\itscr\AppData\Roaming\Python\Python311\site-packages\sklearn\preprocessing\_encoders.py:972: FutureWarning: `sparse` was renamed to `sparse_output` in version 1.2 and will be
  warnings.warn(
C:\Users\itscr\AppData\Roaming\Python\Python311\site-packages\sklearn\preprocessing\_encoders.py:972: FutureWarning: `sparse` was renamed to `sparse_output` in version 1.2 and will be
  warnings.warn(

====== XGBRegressor ======
| 3.    | Estimators          | 1000         |
| 4.    | Learning Rate       | 0.2           |
| 5.    | Max Depth           | 12           |
| 6.    | Random State        | 1              |
| 7.    | R-squared           | 0.9872140848222425 |
| 8.    | MAE                 | 1.693582007889936 |
```

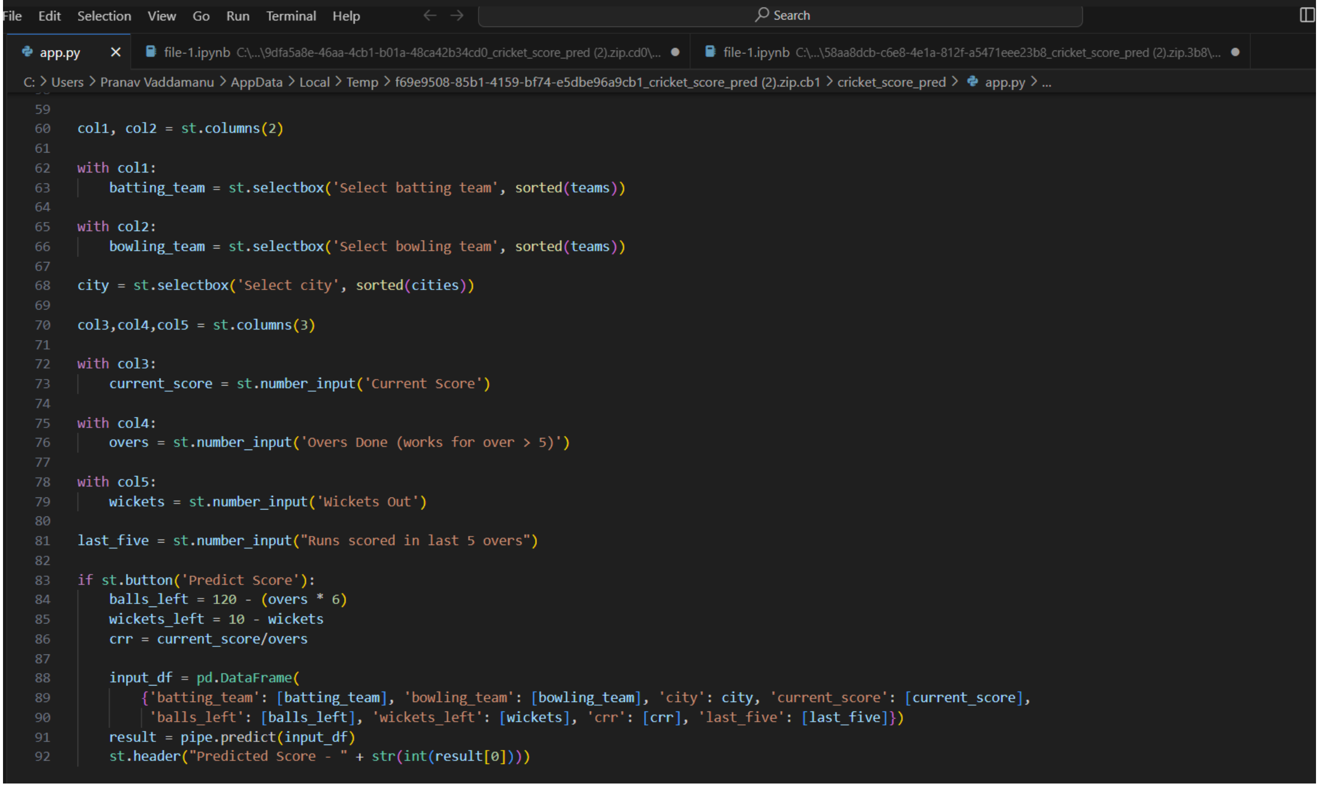
Web Application Source Code:


File
Edit
Selection
View
Go
Run
Terminal
Help
Search
app.py
file-1.ipynb C:\...\9dfa5a8e-46aa-4cb1-b01a-48ca42b34cd0_cricket_score_pred (2).zip.cd0\...
file-1.ipynb C:\...\58aa8dcb-c6e8-4e1a-812f-a5471eee23b8_cricket_score_pred (2).zip.3b8\...
C: > Users > Pranav Vaddamanu > AppData > Local > Temp > f69e9508-85b1-4159-bf74-e5dbe96a9cb1_cricket_score_pred (2).zip.cb1 > cricket_score_pred > app.py > ...

```python
import streamlit as st
import pickle
import pandas as pd
import numpy as np

pipe = pickle.load(open('pipe.pkl', 'rb'))

teams = [
    'Australia',
    'India',
    'Bangladesh',
    'New Zealand',
    'South Africa',
    'England',
    'West Indies',
    'Afghanistan',
    'Pakistan',
    'Sri Lanka'
]

cities = ['Colombo',
    'Mirpur',
    'Johannesburg',
    'Dubai',
    'Auckland',
    'Cape Town',
    'London',
    'Pallekele',
    'Barbados',
    'Sydney',
    'Melbourne',
    'Durban',
    'St Lucia',
    'Wellington',
    'Lauderhill',
    'Hamilton',
    'Centurion',
```

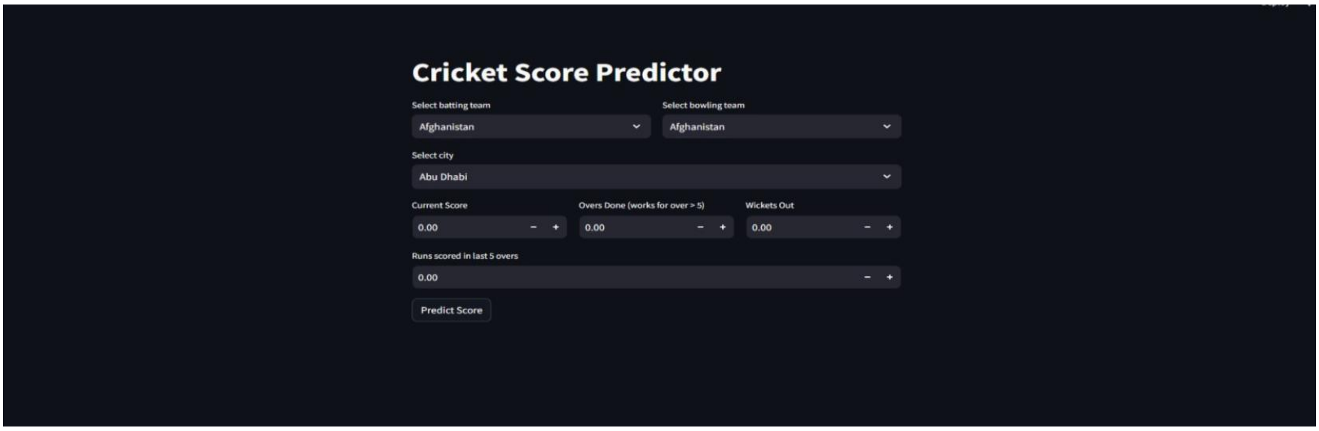

app.py
file-1.ipynb C:\...\9dfa5a8e-46aa-4cb1-b01a-48ca42b34cd0_cricket_score_pred (2).zip.cd0\...
file-1.ipynb C:\...\58aa8dcb-c6e8-4e1a-812f-a5471eee23b8_cricket_score_pred (2).zip.3b8\...
C: > Users > Pranav Vaddamanu > AppData > Local > Temp > f69e9508-85b1-4159-bf74-e5dbe96a9cb1_cricket_score_pred (2).zip.cb1 > cricket_score_pred > app.py > ...

```python
    'Sydney',
    'Melbourne',
    'Durban',
    'St Lucia',
    'Wellington',
    'Lauderhill',
    'Hamilton',
    'Centurion',
    'Manchester',
    'Abu Dhabi',
    'Mumbai',
    'Nottingham',
    'Southampton',
    'Mount Maunganui',
    'Chittagong',
    'Kolkata',
    'Lahore',
    'Delhi',
    'Nagpur',
    'Chandigarh',
    'Adelaide',
    'Bangalore',
    'St Kitts',
    'Cardiff',
    'Christchurch',
    'Trinidad']

st.title('Cricket Score Predictor')


col1, col2 = st.columns(2)

with col1:
    batting_team = st.selectbox('Select batting team', sorted(teams))

```


File Edit Selection View Go Run Terminal Help
Search
app.py
file-1.ipynb C:\...\9dfa5a8e-46aa-4cb1-b01a-48ca42b34cd0_cricket_score_pred (2).zip.cd0\...
file-1.ipynb C:\...\58aa8dcb-c6e8-4e1a-812f-a5471eee23b8_cricket_score_pred (2).zip.3b8\...
C: > Users > Pranav Vaddamanu > AppData > Local > Temp > f69e9508-85b1-4159-bf74-e5dbe96a9cb1_cricket_score_pred (2).zip.cb1 > cricket_score_pred > app.py > ...

```python

col1, col2 = st.columns(2)

with col1:
    batting_team = st.selectbox('Select batting team', sorted(teams))

with col2:
    bowling_team = st.selectbox('Select bowling team', sorted(teams))

city = st.selectbox('Select city', sorted(cities))

col3,col4,col5 = st.columns(3)

with col3:
    current_score = st.number_input('Current Score')

with col4:
    overs = st.number_input('Overs Done (works for over > 5)')

with col5:
    wickets = st.number_input('Wickets Out')

last_five = st.number_input("Runs scored in last 5 overs")

if st.button('Predict Score'):
    balls_left = 120 - (overs * 6)
    wickets_left = 10 - wickets
    crr = current_score/overs

    input_df = pd.DataFrame(
        {'batting_team': [batting_team], 'bowling_team': [bowling_team], 'city': city, 'current_score': [current_score],
         'balls_left': [balls_left], 'wickets_left': [wickets], 'crr': [crr], 'last_five': [last_five]})
    result = pipe.predict(input_df)
    st.header("Predicted Score - " + str(int(result[0])))
```

OUTPUT



Cricket Score Predictor

Select batting team: Afghanistan

Select bowling team: Afghanistan

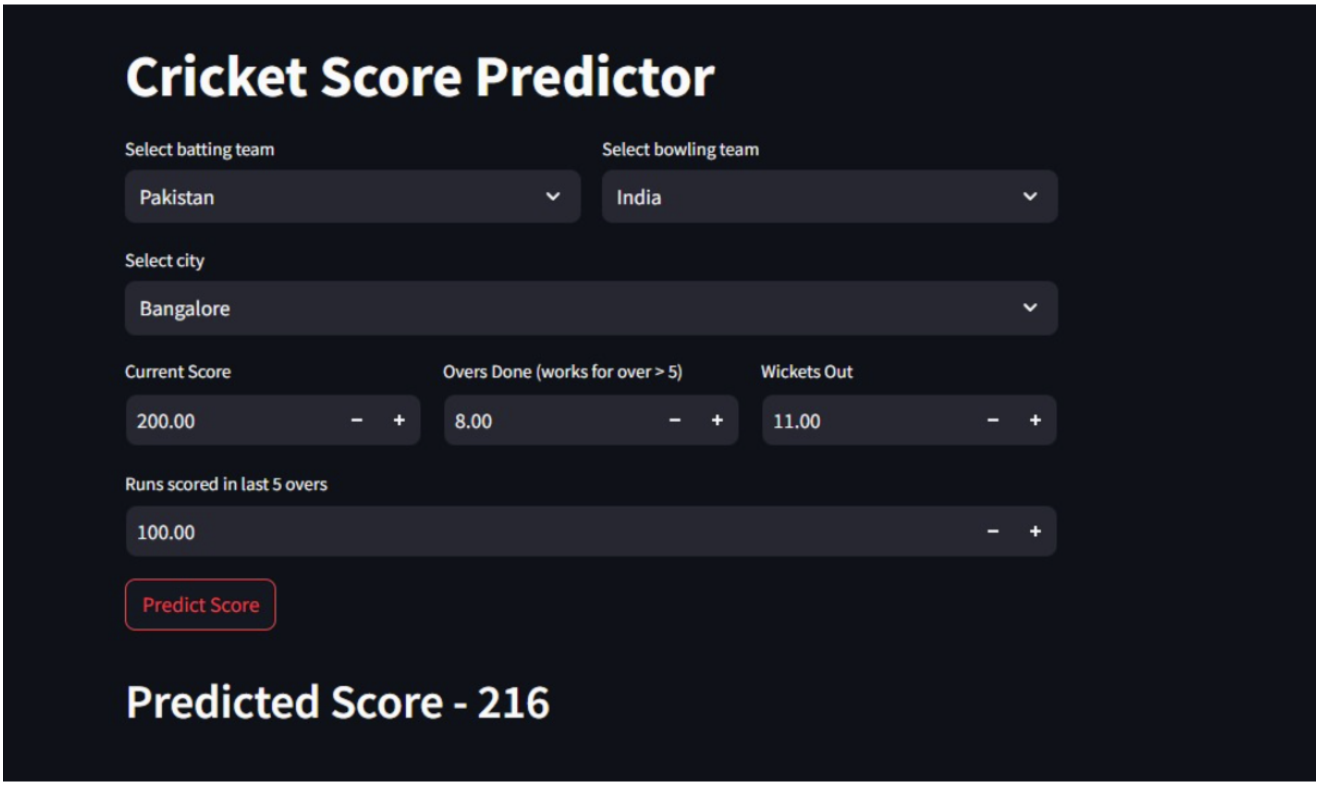
Select city: Abu Dhabi

Current Score: 0.00 − +

Overs Done (works for over > 5): 0.00 − +

Wickets Out: 0.00 − +

Runs scored in last 5 overs: 0.00 − +



Cricket Score Predictor

Select batting team: Pakistan

Select bowling team: India

Select city: Bangalore

Current Score: 200.00

Overs Done (works for over > 5): 8.00

Wickets Out: 11.00

Runs scored in last 5 overs: 100.00

Predicted Score - 216

13.2 GitHub Repository

The complete project, including source code, datasets, and documentation, is hosted on GitHub. The repository can be accessed at the following link: https://github.com/smartinternz02/SI-GuidedProject-612871-1700582187

13.3 Project Demo

For a live demonstration of the T20 Cricket Score Prediction application, please visit the following link:

https://drive.google.com/file/d/1Q0dJhlmadLRnru8jX2hzWil6WiO1x8qt/view?usp=sharing

13.4 Data Dictionary

The data dictionary provides detailed information about the columns in the processed dataset, including their meanings and data types.

Column Name	Description	Data Type
batting_team	Team currently batting	String
bowling_team	Team currently bowling	String
city	City where the match is played	String
current_score	Current total runs scored by the batting team	Integer
balls_left	Number of balls left in the match	Integer
wickets_left	Number of wickets left for the batting team	Integer
crr	Current run rate	Float
last_five	Runs scored in the last five overs	Integer
runs_x	Target runs for the batting team	Integer

13.5 Project Documentation

The project documentation, including detailed information on project overview, literature survey, requirements analysis, and more, can be accessed in the provided PDF document: T20 Cricket Score Prediction Documentation (link-to-documentation.pdf)

13.6 Contact Information

For inquiries or additional information, please contact:

- Project Team:
  Pranav Vaddamanu: pranav.vaddamanu2021@vitstudent.ac.in
  Sriram S: sriram.s2021@vitstudent.ac.in
  Lanka Vasanth: lanka.vasanth2021@vitstudent.ac.in

Note: Ensure that the GitHub repository is regularly updated with the latest code changes and documentation revisions.