A

Project Report

on

DATA ANALYSIS AND PREDICTION OF FOOTBALL GOALS

Submitted for partial fulfilment of the requirements for the award of the degree of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

By

M.HAREEN (2451-19-733-041)
NAGASREEDIVYA KINDI (2451-19-733-049)
KOTHAPALLI CHANDRA SHEKAR (2451-19-733-301)

Under the guidance of

MEDURI ANUPAMA
Associate Professor
Department of CSE

Maturi Venkata Subba Rao (MVSR) ENGINEERING COLLEGE
Department of Computer Science and Engineering
(Affiliated to Osmania University & Recognized by AICTE)
Nadergul, Saroor Nagar Mandal, Hyderabad – 501 510

Academic Year: 2022-23

CERTIFICATE

This is to certify that the project work entitled "Data Analysis and Prediction of Football Goals" is a bonafide work carried out by ***Mr. M HAREEN (2451-19-733-041), Ms. NAGASREEDIVYA KINDI (2451-19-733-049), Mr. KOTHAPALLI CHANDRA SHEKAR (2451-19-733-301)*** *in partial fulfilment of the requirements for the award of degree of* ***Bachelor of Engineering*** *in* ***Computer Science and Engineering*** *from* ***Maturi Venkata Subba Rao (MVSR) Engineering College,*** *affiliated to OSMANIA UNIVERSITY, Hyderabad, during the Academic Year 2022-23 under our guidance and supervision.*

The results embodied in this report have not been submitted to any other university or institute for the award of any degree or diploma to the best of our knowledge and belief.

Internal Guide
Anupama Meduri
Associate Professor
Department of CSE
MVSREC.

Head of the Department
Prof. J. Prasanna Kumar
Professor
Department of CSE
MVSREC.

DECLARATION

This is to certify that the work reported in the present project entitled "Data Analysis and Prediction of Football Goals" is a record of bonafide work done by us in the Department of Computer Science and Engineering, Maturi Venkata Subba Rao (MVSR) Engineering College, Osmania University during the Academic Year 2022-23. The reports are based on the project work done entirely by us and not copied from any other source. The results embodied in this project report have not been submitted to any other University or Institute for the award of any degree or diploma.

M. Hareen	Nagasreedivya Kindi	K. Chandra Shekar
(2451-19-733-041)	(2451-19-733-049)	(2451-19-733-301)

ACKNOWLEDGEMENT

We would like to express our sincere gratitude and indebtedness to our project coordinator / guide **M. Anupama**, Associate Professor, Department of CSE for her valuable suggestions and interest throughout the course of this project.

We are also thankful to our principal **Dr. G. Kanaka Durga** and **Mr. J. Prasanna Kumar,** Head, Department of Computer Science and Engineering, Maturi Venkata Subba Rao (MVSR) Engineering College, Hyderabad for providing supporting infrastructure, environment and ambiance for completing this project successfully as a part of our B.E.(CSE) Degree.

We convey our heartfelt thanks to the lab staff for allowing us to use the required equipment whenever needed.

Finally, we would like to take this opportunity to thank our families for their support through the work. We sincerely acknowledge and thank all those who gave directly or indirectly their support in completion of this work.

M. HAREEN (2451-19-733-041)
NAGASREEDIVYA KINDI (2451-19-733-049)
K. CHANDRA SHEKAR (2451-19-733-301)

VISION

- To impart technical education of the highest standards, producing competent and confident engineers with an ability to use computer science knowledge to solve societal problems.

MISSION

- To make learning process exciting, stimulating and interesting.
- To impart adequate fundamental knowledge and soft skills to students.
- To expose students to advanced computer technologies in order to excel in engineering practices by bringing out the creativity in students.
- To develop economically feasible and socially acceptable software.

PEOs:

PEO-1: Achieve recognition through demonstration of technical competence for successful execution of software projects to meet customer business objectives..

PEO-2: Practice life-long learning by pursuing professional certifications, higher education or research in the emerging areas of information processing and intelligent systems at a global level.

PEO-3: Contribute to society by understanding the impact of computing using a multidisciplinary and ethical approach.

PROGRAM OUTCOMES (POs)

At the end of the program the students (Engineering Graduates) will be able to:

1. **Engineering knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialisation for the solution of complex engineering problems.
2. **Problem analysis:** Identify, formulate, research literature, and analyse complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

3. **Design/development of solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for public health and safety, and cultural, societal, and environmental considerations.
4. **Conduct investigations of complex problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
5. **Modern tool usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modelling to complex engineering activities with an understanding of the limitations.
6. **The engineer and society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal, and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
7. **Environment and sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and the need for sustainable development.
8. **Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
9. **Individual and team work:** Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
10. **Communication:** Communicate effectively on complex engineering activities with the engineering community and with the society at large, such as being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions**.**
11. **Project management and finance:** Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's work,

as a member and leader in a team, to manage projects and in multidisciplinary environments.

12. **Lifelong learning:** Recognise the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

PROGRAM SPECIFIC OUTCOMES (PSOs)

1. (PSO-1) Demonstrate competence to build effective solutions for computational real-world problems using software and hardware across multi-disciplinary domains.
2. (PSO-2) Adapt to current computing trends for meeting the industrial and societal needs through a holistic professional development leading to pioneering careers or entrepreneurship

COURSE OBJECTIVES AND OUTCOMES

Course Objectives:

➢To enhance practical and professional skills.

➢ To familiarize tools and techniques of systematic literature survey and documentation

➢ To expose the students to industry practices and teamwork.

➢ To encourage students to work with innovative and entrepreneurial ideas

Course Outcomes:

Course code PW 861 CS	**Statement** Student will be able to
CO1	Demonstrate the ability to synthesize and apply the knowledge and skills acquired in the academic program to real-world problems.
CO2	Evaluate different solutions based on economic and technical feasibility
CO3	Effectively plan a project and confidently perform all aspects of project management
CO4	Demonstrate effective written and oral communication skills
CO5	Understand tools and techniques, apply for problem statements and prepare reports

ABSTRACT

This project aims to demonstrate how advanced analytics and predictive modeling can provide deeper insights into the sports world compared to traditional methods of analysis. Despite the availability of extensive statistical information, the implementation of analytics in sports has been inconsistent. Therefore, a dataset will be created by converting sample videos into data using video tracking with OpenCV that can be further used for analysis and prediction in Python. The focus is on showcasing the effectiveness of cutting-edge analytical methods in providing a more comprehensive understanding of sports. By doing so, the project adds to the growing body of evidence that Artificial Intelligence and Machine Learning models are transforming the operations of various businesses and can significantly enhance sports analysis.

M. Hareen (2451-19-733-041)
Nagasreedivya Kindi (2451-19-733-049)
Kothapalli Chandra Shekar (2451-19-733-301)

TABLE OF CONTENTS

CONTENTS	PAGE NO.
Certificate	i
Declaration	ii
Acknowledgement	iii
Vision & Mission, PEOs, POs and PSOs	iv
Course Objectives & Outcomes	vii
Abstract	viii
Table of Contents	ix
List of Figures	xi

Chapters:

1. INTRODUCTION

1.1 Problem Statement	1
1.2 Objective	1
1.3 Motivation	2
1.4 Scope	2
1.5 Hardware & Software Requirements	2

2. BACKGROUND/LITERATURE SURVEY

2.1 Survey of Major Areas	3
2.2 Techniques and Algorithms Relevant to Project	4
2.3 Applications	5

3. SYSTEM DESIGN

4.1 System Architecture	6
4.2 Each Module Description	6
4.3 Project Plan	8

4. SYSTEM DESIGN

4.1 Environmental Setup 9
4.2 Implementation of Each Module 13

5. Testing & Results
5.1 Software Training and Testing 18
5.2 Code Snippets & IDE Screenshots 21
5.3 Results 26

6. Conclusion & Future Enhancements

6.1 Conclusion 47
6.2 Future Enhancements 47

References **48**

Appendix

A: source/pseudo code 49

B: Rubrics of the Project 81

LIST OF FIGURES

Figure No	Figure Name	Page No
3.1.1	Flowchart of the project	5
3.3.1	Project Plan	7
5.1.1	Sample Results of the Tracker	18
5.1.2	Mean Average Precision for Player Detection	19
5.1.3	Confusion Matrix of Player Detection (match date:16-10-22)	19
5.1.4	Training Metrics of Player Detection	20
5.1.5	Confusion Matrix of Player Detection (match date:20-3-23)	20
5.2.1	Snippet Shot_Map code	21
5.2.2	Snippet of PassMap_Barca code	21
5.2.3	Snippet of PassMap_Rmd code	22

5.2.4	Snippet of PassHeatMap_Barca code	22
5.2.5	Snippet of PassHeatMap_Rmd code	23
5.2.6	Snippet of HeatMap_Barca code	23
5.2.7	Snippet of HeatMap_Rmd code	24
5.2.8	Snippet of Possession Chain	24
5.2.9	Snippet of xG_to_LoadData code	25
5.2.10	Snippet of xG_to_FitModel code	25
5.3.1	Shot map of Real Madrid and Barcelona (match date: 16-10-2022)	26
5.3.2	Pass map of Barcelona (match date:16-10-2022)	26
5.3.3	Pass map of Real Madrid (match date: 16-10-2022)	27
5.3.4	Dangerous Pass map by Barcelona (match date: 16-10-2022)	27

5.3.5	Dangerous Pass map by Real Madrid (match date: 16-10-2022)	28
5.3.6	Heat map of Barcelona (match date: 16-10-2022)	28
5.3.7	Heat map of Real Madrid (match date:16-10-2022)	29
5.3.8	Possession chains map of Real Madrid & Barcelona (match date: 16-10-2022)	29
5.3.9	Shot map of Real Madrid and Barcelona (match date: 20-03-2023)	30
5.3.10	Pass map of Barcelona (match date: 20-03-2023)	30
5.3.11	Pass map of Real Madrid (match date: 20-03-2023)	31
5.3.12	Dangerous Pass map by Barcelona (match date: 20-03-2023)	31
5.3.13	Dangerous Pass map by Real Madrid (match date: 20-03-2023)	32
5.3.14	Heat map of Barcelona (match date: 20-03-2023)	32
5.3.15	Heat map of Real Madrid (match date:20-03-2023)	33

5.3.16	Possession chains map of Real Madrid & Barcelona (match date: 20-03-2023)	33
5.3.17	Shot map of Real Madrid and Barcelona (match date: 03-03-2023)	34
5.3.18	Pass map of Barcelona (match date: 03-03-2023)	34
5.3.19	Pass map of Real Madrid (match date: 03-03-2023)	35
5.3.20	Dangerous Pass map by Barcelona (match date: 03-03-2023)	35
5.3.21	Dangerous Pass map by Real Madrid (match date: 03-03-2023)	36
5.3.22	Heat map of Barcelona (match date: 03-03-2023)	36
5.3.23	Heat map of Real Madrid (match date:03-03-2023)	37
5.3.24	Possession chains map of Real Madrid & Barcelona (match date: 03-03-2023)	37
5.3.25	Shot map of Real Madrid and Barcelona (match date: 16-01-2023)	38
5.3.26	Pass map of Barcelona (match date: 16-01-2023)	38

5.3.27	Pass map of Real Madrid (match date: 16-01-2023)	39
5.3.28	Dangerous Pass map by Barcelona (match date: 16-01-2023)	39
5.3.29	Dangerous Pass map by Real Madrid (match date: 16-01-2023)	40
5.3.30	Heat map of Barcelona (match date: 16-01-2023)	40
5.3.31	Heat map of Real Madrid (match date:16-01-2023)	41
5.3.32	Possession chains map of Real Madrid & Barcelona (match date: 16-01-2023)	41
5.3.33	Number of Shots	42
5.3.34	Number of Goals	42
5.3.35	Probability of Scoring	43
5.3.36	Shot goal Angle	43
5.3.37	Goal Scored Shot Angle	44

5.3.38	Probability of Scoring Distance	44
5.3.39	Probability of Scoring Distance Squared	45
5.3.40	Probability of Goal	45
5.3.41	ROC Curve	46

1. INTRODUCTION

Sports enthusiasts exhibit a profound interest in even the minutest aspects of a game, and their access to a wealth of statistical information renders them more knowledgeable. This discourse delves into how advanced analytics and predictive modeling surpass traditional analysis methods in the realm of sports. The application of Artificial Intelligence and state-of-the-art Machine Learning models have already demonstrated superior efficacy compared to archaic analyst techniques, transforming various industries. Despite the profusion of available data, the integration of analytics in sports has been erratic. In this project, a dataset is created by converting sample videos into analyzable data. The conversion process entails video tracking utilizing OpenCV, while the project's programming language shall be Python. The aim is to showcase how cutting-edge analytical approaches can offer insights beyond traditional analysis and foster a more comprehensive comprehension of the sports world.

1.1 PROBLEM STATEMENT

The present undertaking concerns the examination of football games, wherein each match is accompanied by an mp4 file that undergoes conversion to a JSON file, facilitating match analysis and goal prediction.

1.2 OBJECTIVE

This project aims to demonstrate the benefits of advanced analysis and predictive modeling in the sports sector compared to traditional methods. The efficacy of AI and state-of-the-art machine learning models over conventional analyst techniques is exemplified through their profound impact on various businesses. Despite the abundance of data in the sports industry, the uptake of analytics has been sluggish and irregular. This discourse endeavors to elucidate this issue.

1.3 MOTIVATION

The prime motivator for this initiative is to furnish football team managers with recommendations pertaining to optimal player selection based on data-driven insights, alongside forecasting the likelihood of a goal being scored. The outcomes of this analysis will equip team managers to make judicious decisions, resulting in superior team performance and outcomes.

1.4 SCOPE

This mode of analysis is implemented to be customized for other sports providing requisite data and algorithmic modifications. Given suitable datasets, an identical analytical methodology may be employed to scrutinize other sports. This adaptability confers a practical advantage to this approach across diverse sporting domains, transcending football.

1.5 SOFTWARE & HARDWARE REQUIREMENTS

1.5.1 Software Requirements

- Football videos
- Roboflow
- Anaconda
- Spyder
- Google Collab
- YoloV8

1.5.2 Hardware Requirements

- Intel(R) Core (TM) i7-1065G7 CPU @ 1.30GHz 1.50 GHz
- Intel IRIS 655
- 12.0 GB RAM

2. BACKGROUND/LITERATURE SURVEY

2.1 SURVEY OF MAJOR AREAS

2.1.1 Data Science

Data science is the study of data to extract meaningful insights for business. It is a multidisciplinary approach that combines principles and practices from the fields of mathematics, statistics, artificial intelligence, and computer engineering to analyse large amounts of data. This analysis helps data scientists to ask and answer questions like what happened, why it happened, what will happen, and what can be done with the results.

2.1.2 Computer Vision

Computer vision is a field of artificial intelligence (AI) that enables computers and systems to derive meaningful information from digital images, videos, and, other visual inputs — and take actions or make recommendations based on that information.

2.1.3 Machine Learning

Machine Learning is the field of study that gives computers the capability to learn without being explicitly programmed. Machine learning is a branch of artificial intelligence (AI) and computer science that uses data and algorithms to imitate how humans learn and work by gradually improving its accuracy.

2.1.4 Artificial Intelligence

AI, short for artificial intelligence, refers to the simulation of human intelligence in machines. It involves the development of algorithms and computer programs that can perform tasks that typically require human intelligence, such as visual perception, speech recognition, decision-making, and language translation. AI is a rapidly growing field with the potential to revolutionize many industries and change the way we live and work. However, it also raises ethical and societal concerns, such as job displacement and privacy issues.

2.1.5 Deep Learning

Deep learning is a subset of machine learning that involves the use of artificial neural networks with multiple layers to process and learn from large amounts of data. It is inspired by the structure and function of the human brain and has been successful in a wide range of applications, including image and speech recognition, natural language processing, and autonomous driving. Deep learning models are capable of learning complex patterns and representations from raw data and can achieve state-of-the-art performance in many tasks.

2.2 TECHNIQUES AND ALGORITHMS RELEVANT W.R.T PROJECT

2.2.1 Supervised Learning

Supervised learning, also known as supervised machine learning, is a subcategory of machine learning and artificial intelligence. It is defined by its use of labeled datasets to train algorithms that classify data or predict outcomes accurately.

2.2.2 Logistic Regression

Logistic regression is a statistical method used for modeling the relationship between a binary dependent variable (such as yes/no, true/false, or 0/1) and one or more independent variables (also called predictors or features). The goal of logistic regression is to predict the probability that the dependent variable takes a particular value, given the values of the independent variables.

2.2.3 Roc Curve

The ROC curve (Receiver Operating Characteristic curve) is a graphical representation of the performance of a binary classification model. It is a plot of the true positive rate (TPR) against the false positive rate (FPR) at different classification thresholds. The ROC curve is created by plotting the TPR (also called sensitivity or recall) on the y-axis and the FPR on the x-axis for different threshold values.

2.3 APPLICATIONS

- Simulating match results.
- Analyzing the performance of a team.
- Team tactic optimizations.

3. SYSTEM DESIGN

3.1 SYSTEM ARCHITECTURE

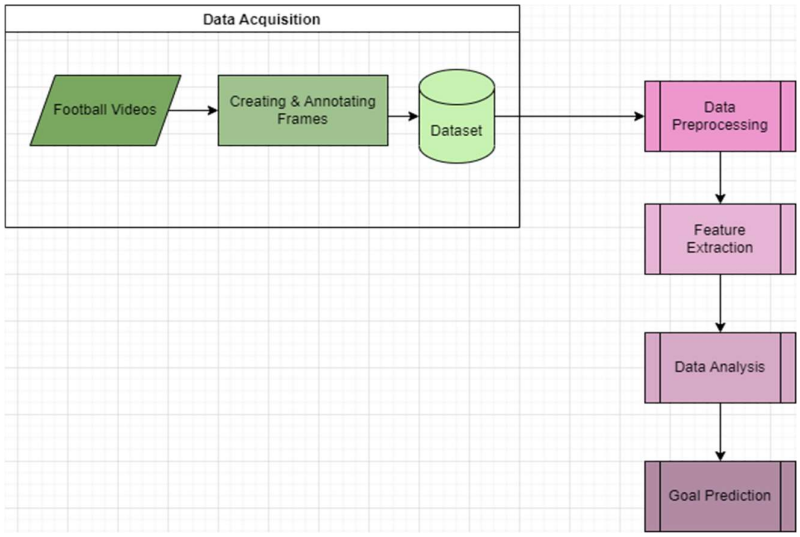


Fig. 3.1.1. Flowchart of the Project

Following a left-to-right approach, the initial step involves obtaining the dataset through object detection techniques applied to diverse football videos. The dataset thus obtained from the previous step is of JSON file format, rich in positional data of the player and the pass and shot metrics of the player. The dataset is then trained and tested using YOLOv8 model with a training-validation-testing split of 80-10-10. Subsequently, the dataset undergoes preprocessing procedures aimed at extracting the essential features for further analysis. Afterward, a comprehensive individual analysis of the datasets is conducted, along with predicting the probability of scoring a goal.

3.2 EACH MODULE DESCRIPTION

3.2.1 Data Acquisition

This study compiled a selection of recent matches between FC Barcelona and Real Madrid CF in Spanish football. Subsequently, utilizing the software 'Roboflow', the frames of the matches were annotated at a frequency of 5 frames per second, and categorized into five distinct classes, namely "ball", "referee", "fcb_player", "fcb_goalie", "rmd_player", and "rmd_goalie". This resulted in a total of approximately

8500 frames used for the project. The YOLOv8 model in conjunction with PyTorch was utilized for the dataset's custom training, validation, and testing. The output dataset was stored and saved as .json files for future use.

3.2.2 Data Pre-Processing

The .json files undergo a conversion process into .csv format to facilitate examination and comprehension of their distinctive attributes and nuances, which in turn enables their utilization for subsequent analysis and prediction.

3.2.3 Feature Extraction

The .json files are subjected to a meticulous selection and extraction process to identify the essential attributes that are crucial for subsequent analysis and prediction. This procedure involves a comprehensive evaluation of the available data to isolate. The selected attributes are then carefully scrutinized to ensure their accuracy and validity, and any extraneous or irrelevant information is eliminated to enhance the precision and quality of the data. The resultant dataset is thus optimized for effective analysis and prediction, ensuring that the outcomes are robust and dependable.

3.2.4 Data Analysis

The attributes that were extracted in the preceding step serve as the foundational elements for generating a diverse range of data analysis and visualization techniques. By utilizing the selected features, various analytical methods are employed to uncover patterns and insights within the dataset. The findings are then presented using a variety of data visualization tools, such as charts, graphs, or interactive dashboards, to facilitate a clear and concise representation of the insights obtained from the data.

3.2.5 Data Prediction

The predictive task undertaken in this project entails employing logistic regression to forecast the likelihood of goals occurring. The logistic regression model is utilized as a statistical tool to determine the probability of a specific outcome based

on the relationship between the predictor variables and the response variable. By fitting the relevant features to the model, the probability of goals taking place can be estimated, based on the underlying characteristics of the dataset.

3.3 PROJECT PLAN

Fig. 3.3.1 Project Plan

4. IMPLEMENTATION

4.1 ENVIRONMENTAL SETUP

4.1.1 Roboflow

Roboflow is a cutting-edge computer vision platform that offers a suite of advanced tools and techniques to expedite and optimize the process of building computer vision models. By improving data collection, preprocessing, and model training, Roboflow enables users to achieve higher levels of accuracy and efficiency in their computer vision projects. The platform provides a comprehensive set of resources and functionalities that facilitate streamlined model development and deployment, empowering users to achieve their computer vision objectives with greater speed and precision.

4.1.2 Anaconda

The Anaconda distribution incorporates Anaconda Navigator, which provides users with a graphical interface for launching applications, and managing conda packages, environments, and channels, thereby obviating the need for command-line commands. Navigator enables users to easily search for packages, install them within an environment, execute the packages, and update them, streamlining the package management process and enhancing the ease of use.

4.1.3 Spyder

Spyder is a free and open-source integrated development environment (IDE) that enables scientific programming in Python, compatible with multiple operating systems. As a robust and versatile tool, Spyder is designed to support a broad range of scientific programming applications, offering a comprehensive suite of features and functionalities to support developers and researchers in their work. With its user-friendly interface and extensive

capabilities, Spyder has become a go-to choice for many users seeking a powerful and efficient environment for Python-based scientific programming.

4.1.4 Google Colab

Google Collaboratory, also known as "Colab", is a Google Research tool that provides a user-friendly platform for executing arbitrary Python code through a browser. It is particularly well-suited for tasks such as machine learning, data analysis, and educational purposes. Essentially, Colab is a Jupyter notebook service that is hosted on a remote server, requiring no setup and allowing users to access computing resources, including GPUs, free of charge. By providing a hassle-free interface for Python coding and a wealth of computing resources, Colab has become a preferred option for many researchers, students, and developers seeking to accomplish complex tasks in a streamlined and efficient manner.

4.1.5 YoloV8

The field of computer vision has taken a significant stride forward with the unveiling of YOLOv8, a groundbreaking model that sets a new benchmark for object detection and instance segmentation. In addition to its enhanced model architecture, YOLOv8 provides developers with a user-friendly interface via a PIP package, simplifying the utilization of the YOLO model. This cutting-edge tool is poised to revolutionize the world of computer vision, offering researchers and developers an innovative and efficient means of achieving their objectives in this rapidly evolving field.

4.1.6 Python

Python can be used for developing machine learning and deep learning projects because of its simplicity, fast development, massive online support, flexible integrations, fast code tests, performance, and visualization tools. The Python language comes with many libraries and frameworks that make coding deep learning models easy. Python is dynamically-typed and garbage-

collected. It supports multiple programming paradigms, including structured (particularly, procedural), object-oriented, and functional programming. It is often described as a "batteries included" language due to its comprehensive standard library.

4.1.7 Libraries Used

4.1.7.1 Pandas

Pandas is a widely used Python library that facilitates working with data sets, offering a suite of functions for data analysis, cleaning, exploration, and manipulation. Created by Wes McKinney in 2008, the name "Pandas" is a portmanteau of "Panel Data" and "Python Data Analysis", reflecting its focus on providing tools and resources for working with complex data structures. With its rich set of capabilities and intuitive interface, Pandas has become a staple tool for many data scientists, researchers, and developers seeking to analyze and process data with ease and efficiency.

4.1.7.2 Matplotlib

Pyplot is a powerful API (Application Programming Interface) that extends the functionality of Python's matplotlib library, providing a robust open-source alternative to MATLAB. Matplotlib is a data visualization library that enables the creation of a variety of informative plots, graphs, and charts. By leveraging the Pyplot API, users can more easily and efficiently create complex visualizations and analytical outputs in Python, streamlining the process of data analysis and enabling researchers and developers to achieve their objectives with greater speed and precision.

4.1.7.3 Numpy

NumPy is a highly regarded Python library that is widely used for working with arrays, offering a rich set of functions for linear algebra, Fourier transform, and matrix operations. Created by Travis Oliphant in 2005, NumPy has evolved into a powerful open-source

project, providing a comprehensive suite of mathematical functions and tools for manipulating arrays. With its intuitive interface and extensive capabilities, NumPy has become a go-to tool for researchers and developers seeking to work with numerical data and perform complex calculations with ease and efficiency. In short, NumPy - an acronym for Numerical Python - has transformed the landscape of scientific computing, enabling users to tackle complex problems in a streamlined and effective manner.

4.1.7.4 Open CV

OpenCV, or Open-Source Computer Vision Library, is a popular open-source software library that provides a powerful set of tools for computer vision and machine learning applications. Developed to establish a standardized framework for computer vision research, OpenCV has since become a go-to resource for researchers, developers, and commercial organizations seeking to leverage the power of machine perception in a wide range of products and services. With its robust suite of features and comprehensive functionality, OpenCV has helped to drive advances in computer vision and machine learning, enabling users to tackle complex problems with greater ease and precision than ever before.

4.1.7.5 PyTorch

PyTorch is a comprehensive framework that enables the creation of deep learning models, a form of machine learning that has gained widespread use in image recognition and natural language processing applications. Developed using the Python programming language, PyTorch offers a powerful yet accessible toolset that can be easily learned and utilized by machine learning developers of all levels. By providing a user-friendly interface and a rich suite of features for model building, training, and deployment, PyTorch has helped to make deep learning more accessible and democratized the field of artificial intelligence.

4.1.7.6 Mplsoccer

Mplsoccer is a specialized Python library that allows for the creation of football charts within the Matplotlib data visualization framework. With its focus on football-related data visualization, mplsoccer offers a range of tools and functionality that are

tailored specifically to the unique needs of football analysts, researchers, and enthusiasts. By providing a seamless interface for working with football data and creating high-quality visualizations, mplsoccer has become a popular choice among those looking to explore and analyze football-related information using Python.

4.1.7.7 Json

JSON, short for JavaScript Object Notation, is a widely-used open data interchange format that can be read and understood by both humans and machines. Although it bears the name of the JavaScript programming language, JSON is language-independent and can be utilized in a broad range of applications and contexts. As a common output format for many APIs, JSON has emerged as a popular choice for data exchange due to its simplicity, flexibility, and broad compatibility with a wide range of programming languages and platforms.

4.1.7.8 FCPython

FC Python strives to provide readily available learning materials for individuals with an interest in sports who are seeking to acquire fundamental programming and data skills using the Python language.

4.1.7.9 Random

The random module is an inherent module that produces pseudo-random variables. It enables various random actions, including generating a random number, selecting random elements from a list, and shuffling elements randomly, among others.

4.2 IMPLEMENTATION OF EACH MODULE

Latest sports matches were selected for object detection analysis. Manual annotation of 8,500 frames identified players' positions, enabling insights into team strategies. The dataset was pre-processed and trained with YOLOv8 on Google Colab. JSON and CSV formats provided much needed flexibility. Analyses of shots, goals and possession chain was performed and by logistic regression goal likelihood was predicted to enhance team performance.

4.2.1 Selection of Videos

In order to carry out a project that involved object detection in sports videos, it was necessary to use footage of the latest matches. Therefore, the selection process

focused on matches that were played on specific dates, namely 16-10-2022, 20-3-2023, 3-3-2023, and 16-1-2023. These matches were chosen based on their recency, as they were the most up-to-date matches available at the time of the project.

After the selection process was completed, the chosen videos underwent object detection analysis. By using the latest matches for the project and employing object detection techniques, the project team was able to obtain accurate and relevant data that could be used to improve the performance of sports teams or enhance the viewing experience of fans.

4.2.2 Player Annotation using Roboflow

A meticulous annotation process on approximately 8500 frames was conducted to distinguish each player of both teams. This was a manual process done using YOLOv8, which required great attention to detail and significant effort. The annotation process allowed the team to track the positions of each player and generate various types of data, including passes, shots, and attempts on goal. By identifying each player and their movements in the frames, analysis on their interactions during the game is made, providing valuable insights into the strategies and tactics employed by both teams.

Overall, the manual annotation process and tracking of players in the sports videos allowed the team to extract valuable insights from the data, which could have significant implications for the sports industry and contribute to ongoing research in the field of computer vision and machine learning.

4.2.2.1 Labels Created

- The “ball” label tracks the position of the ball.
- “Fcb_player” tracks the position of the Real Madrid CF outfield players.
- “Rmd_player” tracks the position of the Real Madrid CF outfield players.
- “Rmd_goalie” tracks the position of the Real Madrid CF goalkeeper.
- “Fcb_goalie” tracks the position of the FC Barcelona goalkeeper.
- “Referee” tracks the position of the referee to separate them from the rest of the players.

4.2.3 Collecting the Dataset from Roboflow

Once the manual annotation of the sports video frames was completed, a preprocessing stage was carried out to prepare the dataset for analysis and model training. This involved resizing and flipping the images, a technique commonly used in computer vision to ensure that the images are of a consistent size and orientation.

Following pre-processing, the dataset approximately of size 0.8 GB is generated which has the positional data, shot data and pass data of the players tracked in the match and trained using the YOLOv8 model, a popular object detection algorithm, on Google Colab, a cloud-based development platform. This involved using the annotated images as input into the YOLOv8 model, which then learned to detect and track objects within the images.

The generated dataset was extracted in two different file formats, JSON and CSV. The JSON format was selected to enable the utilization of the dataset in the analysis and prediction stages of the project, while the CSV format was chosen to provide a structured and organized view of the features of the dataset.

Overall, the annotation and preprocessing stages were critical for generating a high-quality dataset, which was then used to train the YOLOv8 model. The ability to extract the dataset in different file formats provides flexibility in how the data can be analyzed and utilized in future stages of the project.

4.2.4 Performing Analysis on the Dataset

A multitude of analyses were performed on the datasets of individual matches.

- For each football match analyzed, a shot map is generated, which is a visual representation of the football pitch that displays the locations of all the shots taken during the game. This shot map is color-coordinated to distinguish between the two teams, providing an easy-to-read summary of the match's shooting patterns. The shot map provides valuable insights into the tactics and strategies employed by both teams, as well as highlighting the strengths and weaknesses of individual players.
- A pass map is a graphical representation of all the passes that occur on the football field during a game. The pass map provides a visual representation of the

movement of the ball between players and the overall flow of the game. This visualization is a crucial tool in the analysis of football games as it enables coaches, analysts, and players to identify patterns and trends in passing, which can be used to develop and fine-tune strategies and tactics. Pass maps can also be used to compare the performance of different teams and players, identifying areas where one team may have an advantage over another. By analyzing the pass maps of opposing teams, coaches and analysts can develop game plans that exploit weaknesses in their opponents' play while leveraging their own strengths.

- As part of the analysis of the four football matches, a record of all the dangerous passes made by both teams was compiled. A dangerous pass is defined as one that leads to a goal or, at the very least, an attempt at the goal. These passes are critical in determining the attacking prowess and effectiveness of a team. By compiling and analyzing the dangerous passes made during the four matches, the team can identify the most effective strategies for attacking the opponent's goal. The data generated through this analysis provides valuable insights into the types of passes that lead to the most successful outcomes and can be used to develop targeted training programs for players. Moreover, the analysis of dangerous passes can also be used to assess the performance of individual players, identifying those who are particularly adept at creating goal-scoring opportunities.
- In addition to the other analyses conducted on the four football matches, heat maps for both teams were generated for individual matches. A heat map is a graphical representation of a football pitch that indicates the level of a player's activity and effectiveness in different parts of the field. The heat map generates colored areas that indicate where a player had more composure with the ball. Typically, red is the most common color used to indicate these heated areas. Different types of heat maps are represented in the analysis to segregate the football pitch in various ways. These maps provide a more nuanced and detailed view of player activity, allowing coaches and analysts to identify specific areas of strength and weakness. For example, a heat map that segregates the pitch based on the location of the ball during set pieces such as corner kicks, free kicks, or throw-ins can provide valuable insights into the positioning and effectiveness of players in those situations.

- For all four matches, a possession chain for both teams is generated, which provides an insight into the passing sequences without losing the ball to the opposing team. This possession chain is presented in the form of an HTML file, specifically designed to utilize the full functionality of the 'add_trace' function of the "plotly" library. By creating a possession chain using the 'plotly' library and presenting it in an HTML format, a more interactive and dynamic visualization of the passing sequences is generated. The 'plotly' library allows us to create rich, interactive charts that can be easily explored and analyzed. This makes it easier to identify patterns and trends in a team's play and to develop strategies for improving their possession play.

4.2.5 Goal Prediction

Using supervised learning and logistic regression, the likelihood of a goal being scored is predicted. Supervised learning is a machine learning technique where an algorithm is trained on a labeled dataset to predict outcomes or identify patterns. Logistic regression is a statistical method used to model the probability of a binary outcome, such as whether a goal will be scored or not. To make the prediction, the model is trained using various factors that influence the likelihood of a goal, such as the position of the ball, the number of players in the attacking and defending teams, and the position of the players on the field. The model then used this information to calculate the probability of a goal being scored. By using logistic regression, a model is developed that could accurately predict the likelihood of a goal being scored based on various factors. This information can be used by coaches and analysts to develop strategies for improving their team's offensive play and to identify weaknesses in their defensive play. Ultimately, this type of analysis can help teams to improve their performance on the field and achieve better results.

5. TESTING AND RESULTS

5.1 SOFTWARE TRAINING AND TESTING

The dataset was created from different football matches of Barcelona and Real Madrid teams in the most recent times. 6 classes of labels were created for the object detection namely Ball, Referee, Fcb_player, Fcb_goalie, Rmd_player and Rmd_goalie. Roboflow software along with OpenCV and YOLOv8 model was used to train and test the datasets. The split was done such that 80% of the data is used for training, 10% data is used for validation and 10% of the data is used for testing.

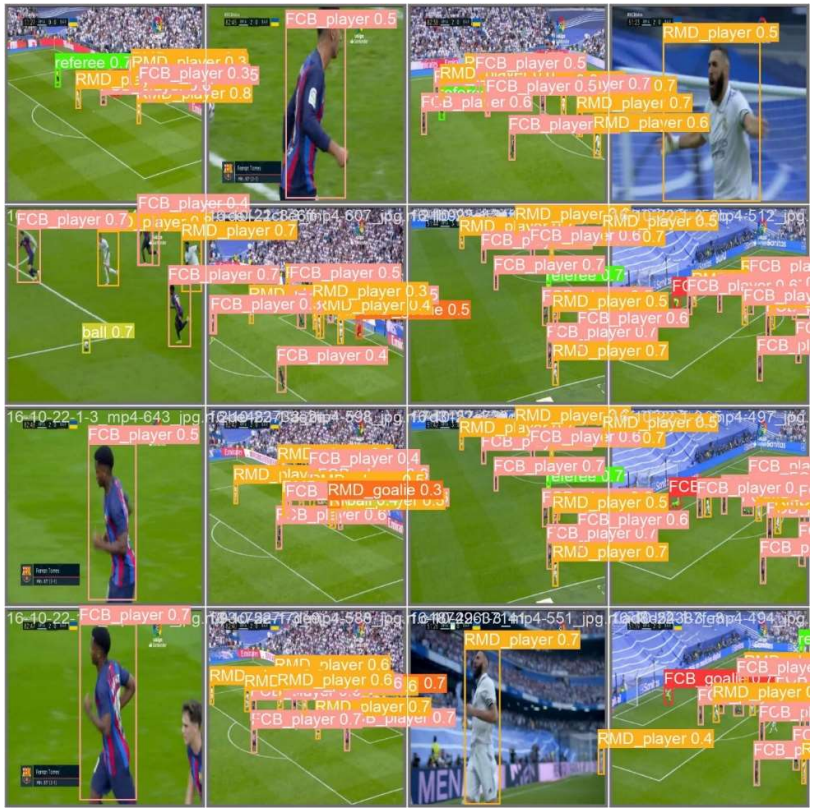


Fig. 5.1.1 Sample Results of the Tracker (match date: 16-10-2022)

The above figure shows how the players, the referee and the ball are being tracked in a football match using OpenCV and YOLOv8.

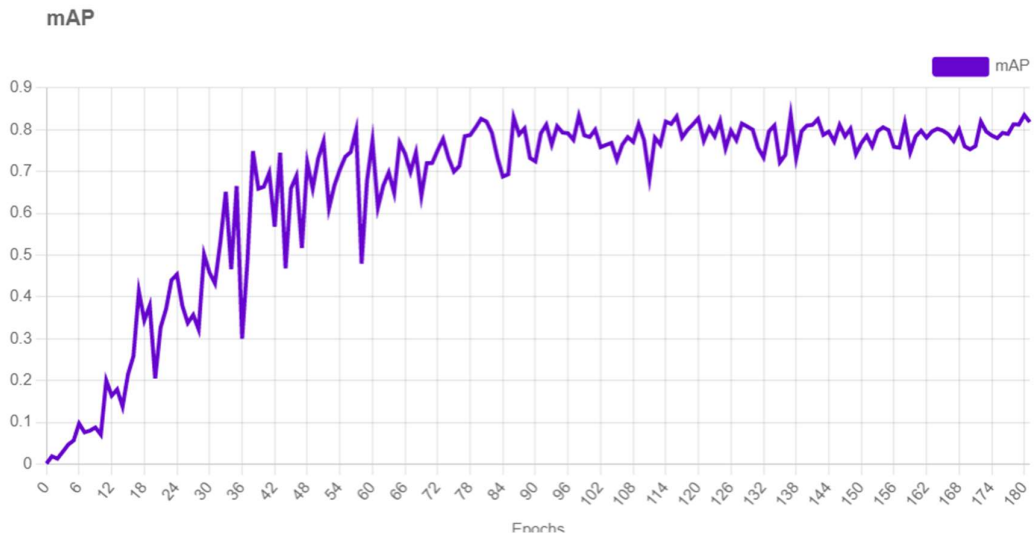


Fig. 5.1.2 Mean Average Precision for Player Detection (match date: 16-10-2022)

Fig. 5.1.2 represents the mAP of the player detection which compares the ground-truth bounding box to the detected bounding box and returns a score for the match dated 16-10-22.

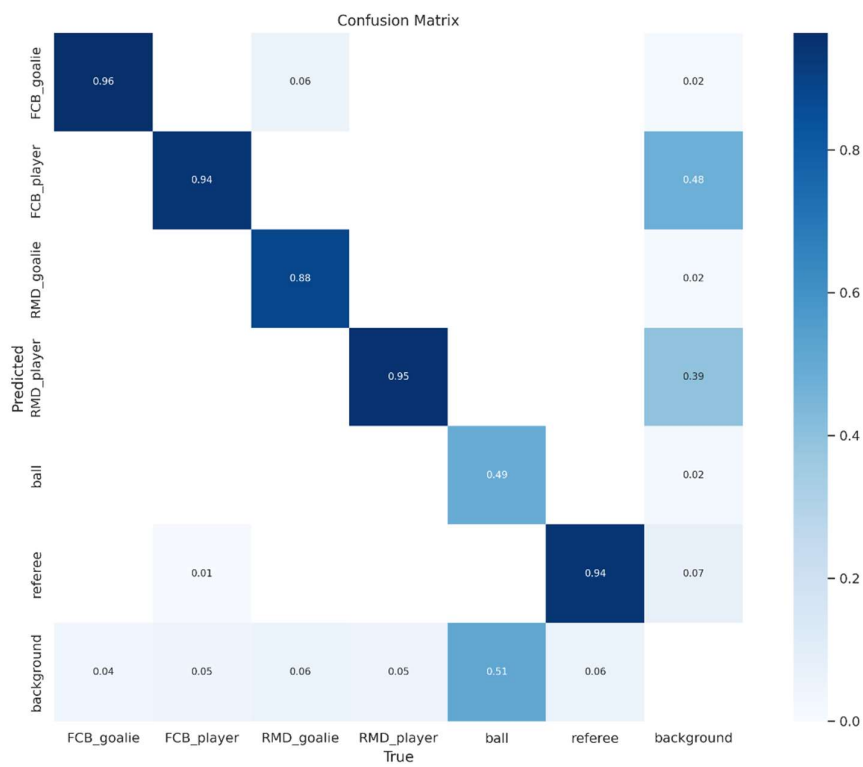


Fig. 5.1.3 Confusion Matrix of Player Detection (match date: 16-10-2022)

The above figure represents confusion matrix showing the performance of the classifier of the player detection for the match dated 16-10-22.

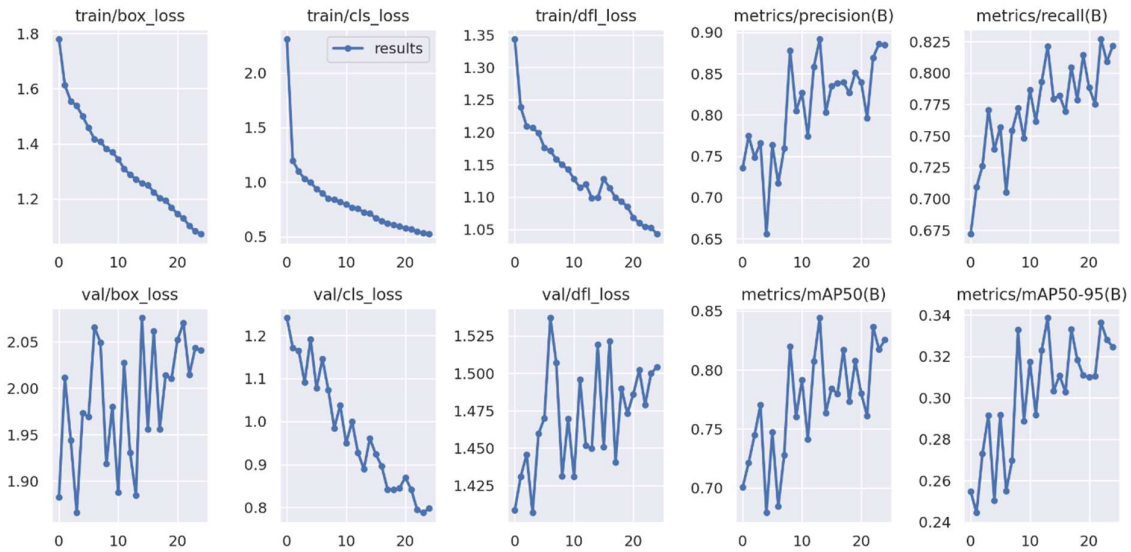


Fig. 5.1.4 Training Metrics of Player Detection (match date: 16-10-2022)

The plots in the above figure show the different metrics used to evaluate the Detection model.

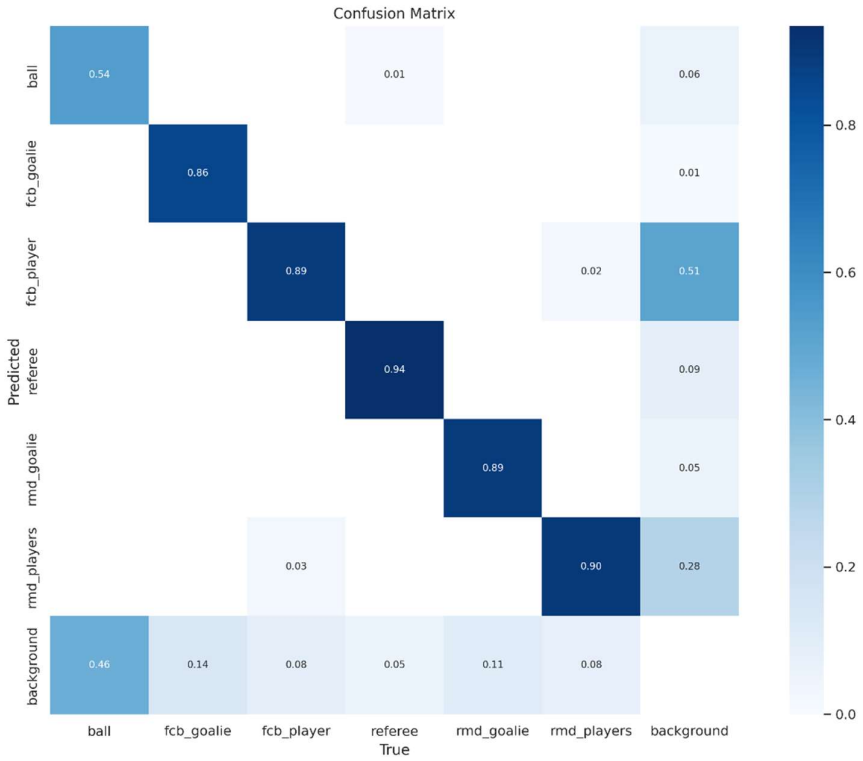
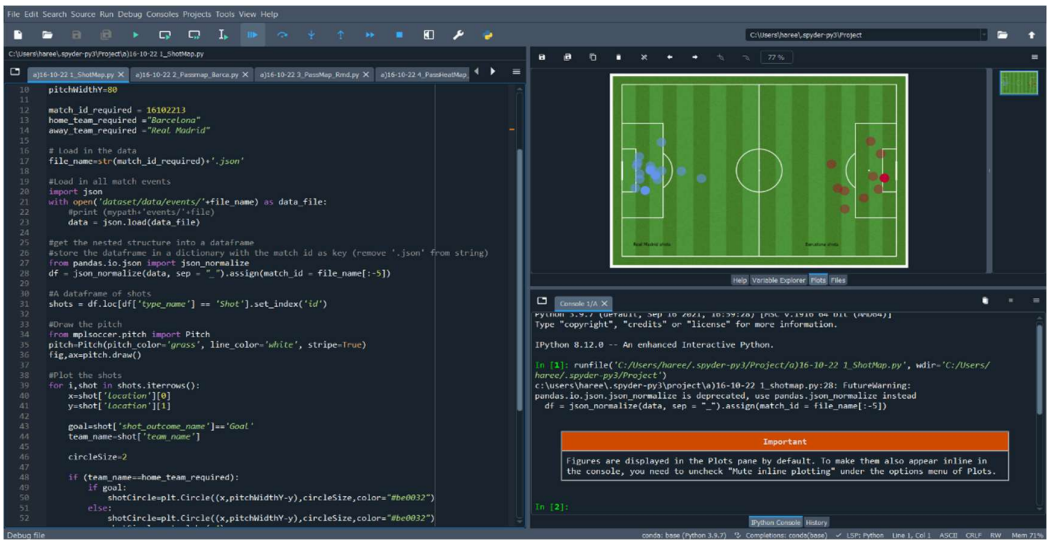


Fig. 5.1.3 Confusion Matrix of Player Detection (match date: 20-03-2023)

The above figure represents confusion matrix showing the performance of the classifier of the player detection for the match dated 16-10-22.

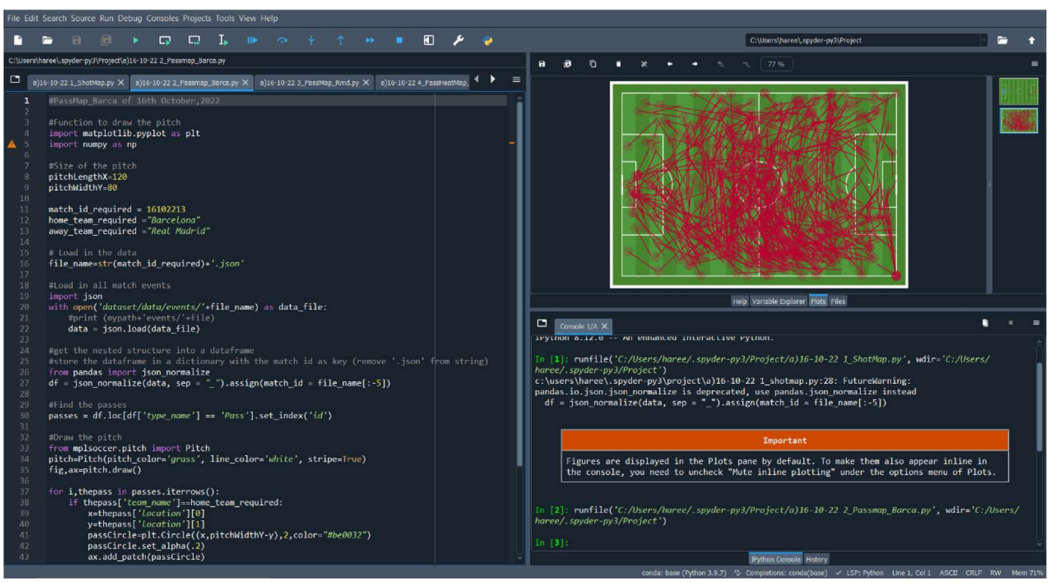
5.2 Code Snippets & IDE Screenshots



```
pitchWidthY=80

match_id_required = 16102213
home_team_required ="Barcelona"
away_team_required ="Real Madrid"

# Load in the data
file_name=str(match_id_required)+'.json'

#Load in all match events
import json
with open('dataset/data/events/'+file_name) as data_file:
    #print (mypath+'events/'+file)
    data = json.load(data_file)

#get the nested structure into a dataframe
#store the dataframe in a dictionary with the match id as key (remove '.json' from string)
from pandas.io.json import json_normalize
df = json_normalize(data, sep = "_").assign(match_id = file_name[:-5])

#A dataframe of shots
shots = df.loc[df['type_name'] == 'Shot'].set_index('id')

#Draw the pitch
from mplsoccer.pitch import Pitch
pitch=Pitch(pitch_color='grass', line_color='white', stripe=True)
fig,ax=pitch.draw()

#Plot the shots
for i,shot in shots.iterrows():
    x=shot['location'][0]
    y=shot['location'][1]

    goal=shot['shot_outcome_name']=='Goal'
    team_name=shot['team_name']

    circleSize=2

    if (team_name==home_team_required):
        if goal:
            shotCircle=plt.Circle((x,pitchWidthY-y),circleSize,color="#be0032")
        else:
            shotCircle=plt.Circle((x,pitchWidthY-y),circleSize,color="#be0032")
```

Fig. 5.2.1. Snippet Shot_Map code

The above code snippet shows the code written to generate the shot map of a match.



```
#PassMap_Barca of 16th October,2022

#Function to draw the pitch
import matplotlib.pyplot as plt
import numpy as np

#Size of the pitch
pitchLengthX=120
pitchWidthY=80

match_id_required = 16102213
home_team_required ="Barcelona"
away_team_required ="Real Madrid"

# Load in the data
file_name=str(match_id_required)+'.json'

#Load in all match events
import json
with open('dataset/data/events/'+file_name) as data_file:
    #print (mypath+'events/'+file)
    data = json.load(data_file)

#get the nested structure into a dataframe
#store the dataframe in a dictionary with the match id as key (remove '.json' from string)
from pandas import json_normalize
df = json_normalize(data, sep = "_").assign(match_id = file_name[:-5])

#Find the passes
passes = df.loc[df['type_name'] == 'Pass'].set_index('id')

#Draw the pitch
from mplsoccer.pitch import Pitch
pitch=Pitch(pitch_color='grass', line_color='white', stripe=True)
fig,ax=pitch.draw()

for i,thepass in passes.iterrows():
    if thepass['team_name']==home_team_required:
        x=thepass['location'][0]
        y=thepass['location'][1]
        passCircle=plt.Circle((x,pitchWidthY-y),2,color="#be0032")
        passCircle.set_alpha(.2)
        ax.add_patch(passCircle)
```

Fig. 5.2.2. Snippet of PassMap_Barca code

The snippet shown in Fig. 5.2.2 shows the pass map code of Barcelona team in a single match.

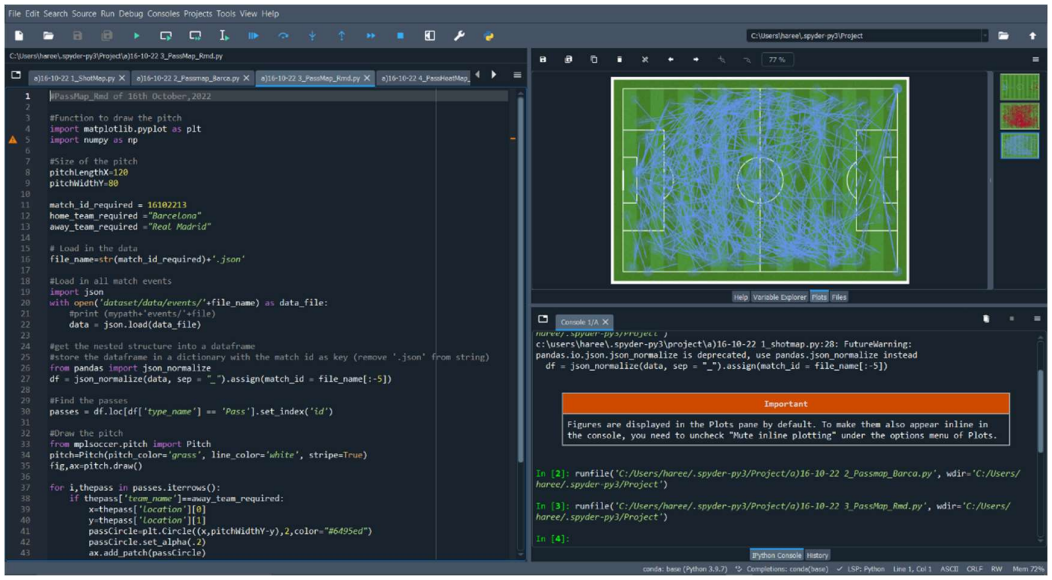


Fig. 5.2.3 Snippet of PassMap_Rmd code

The code snippet above in Fig. 5.2.3 shows the pass map code of Madrid team in a single match.

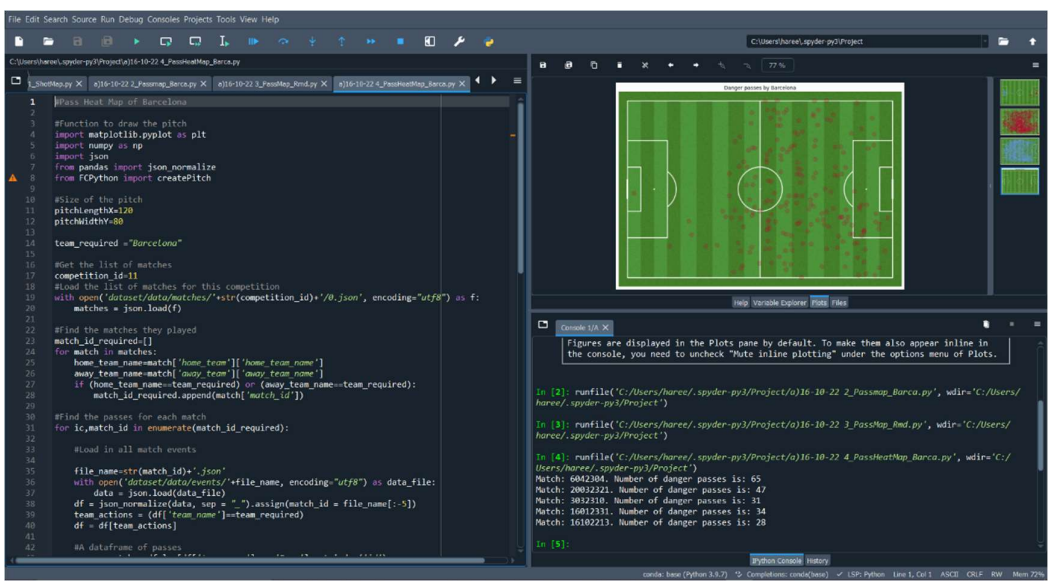


Fig. 5.2.4 Snippet of PassHeatMap_Barca code

The code snippet above shows the code written to generate the pass heat map of Barcelona team across different matches.

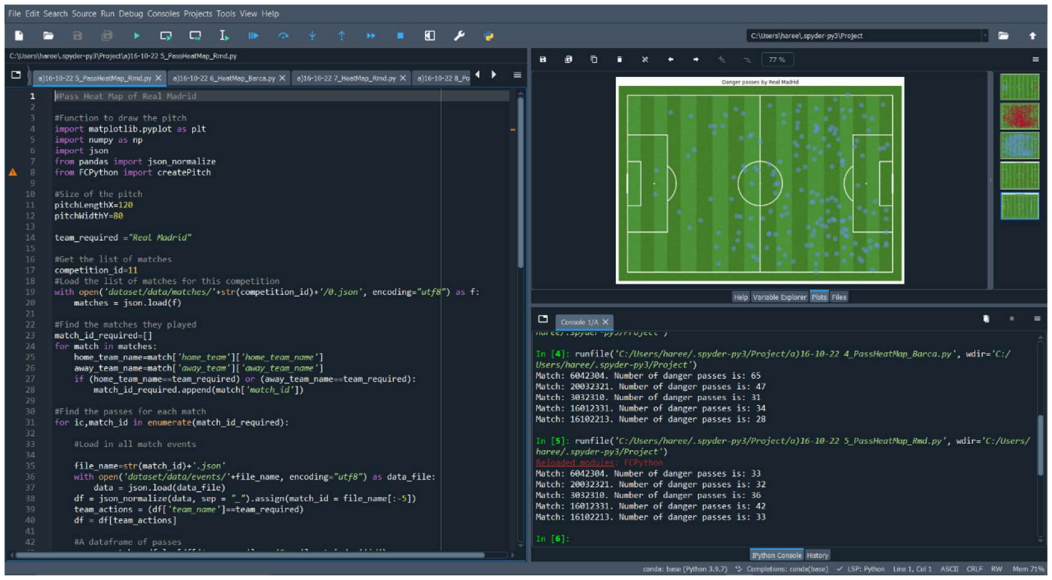


Fig. 5.2.5 Snippet of PassHeatMap_Rmd code

The code snipped given in Fig. 5.2.5 shows the code written to generate the pass heat map of Madrid team across different matches.

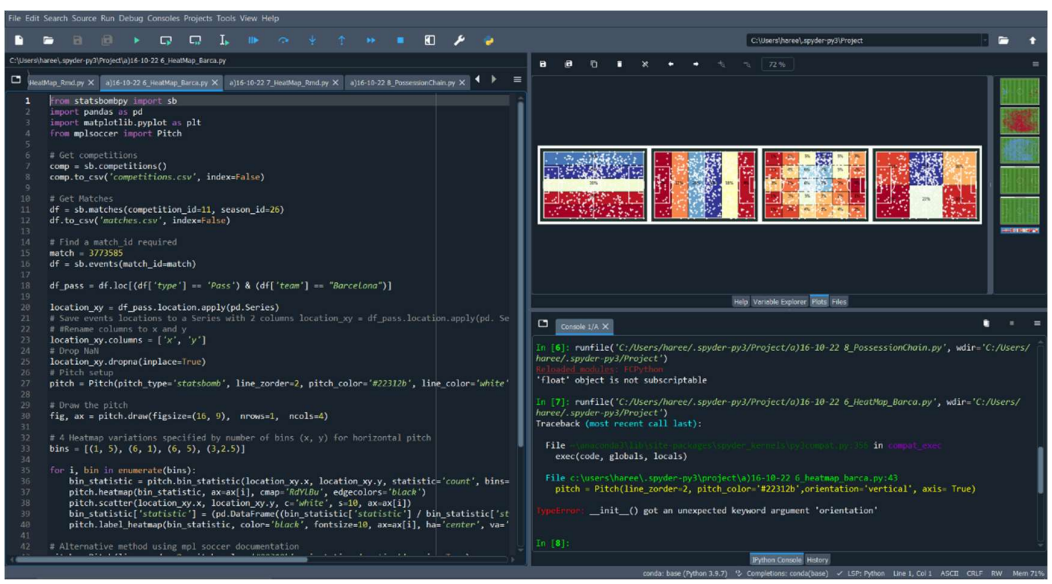


Fig. 5.2.6. Snippet of HeatMap_Barca code

The code snippet above in Fig. 5.2.6 shows code written to generate a heat map of Barcelona team in a single match.

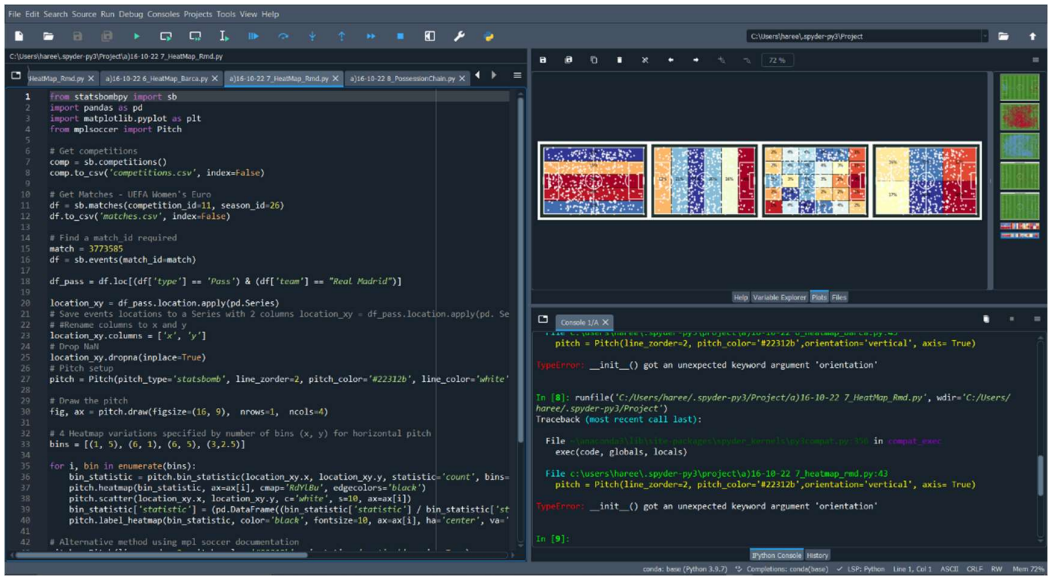


Fig. 5.2.7 Snippet of HeatMap_Rmd code

The code snippet shown above in Fig. 5.2.7 is the code written to generate the heat map of Madrid team in a single match.

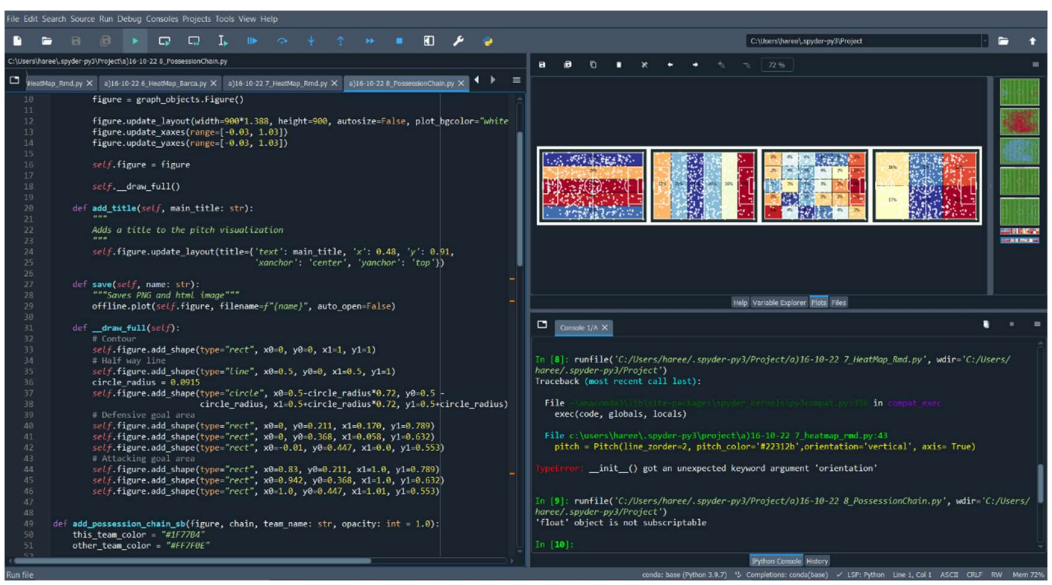


Fig. 5.2.8 Snippet of PossessionChain code

The code snippet shown above in Fig. 5.2.8 is the code written to generate a possession chain of both the teams in a single match.

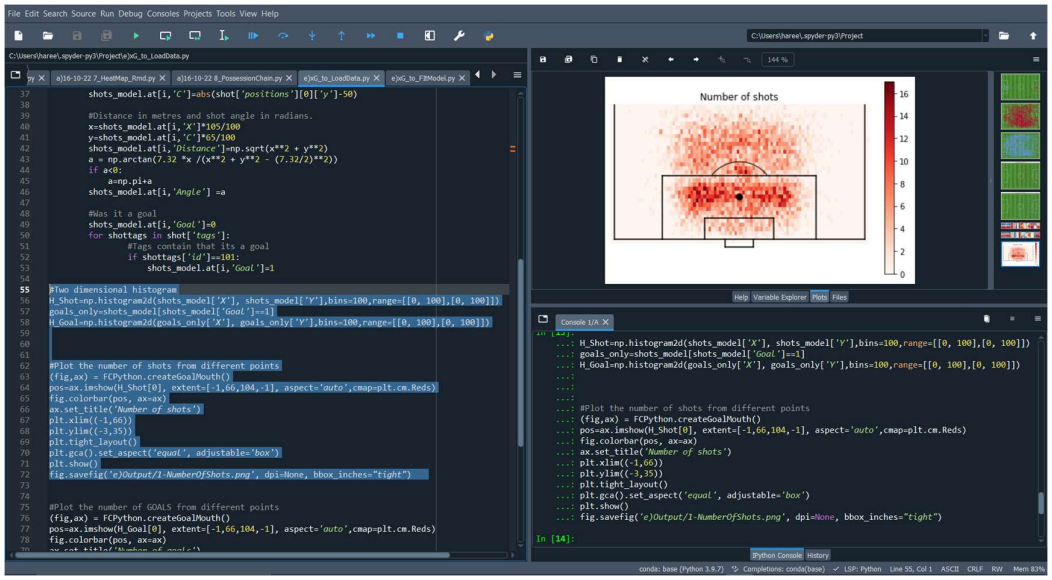


Fig. 5.2.9 Snippet of xG_to_LoadData code

The code snippet shown above in Fig.5.2.9 is the code written to load the goal data used in the prediction.

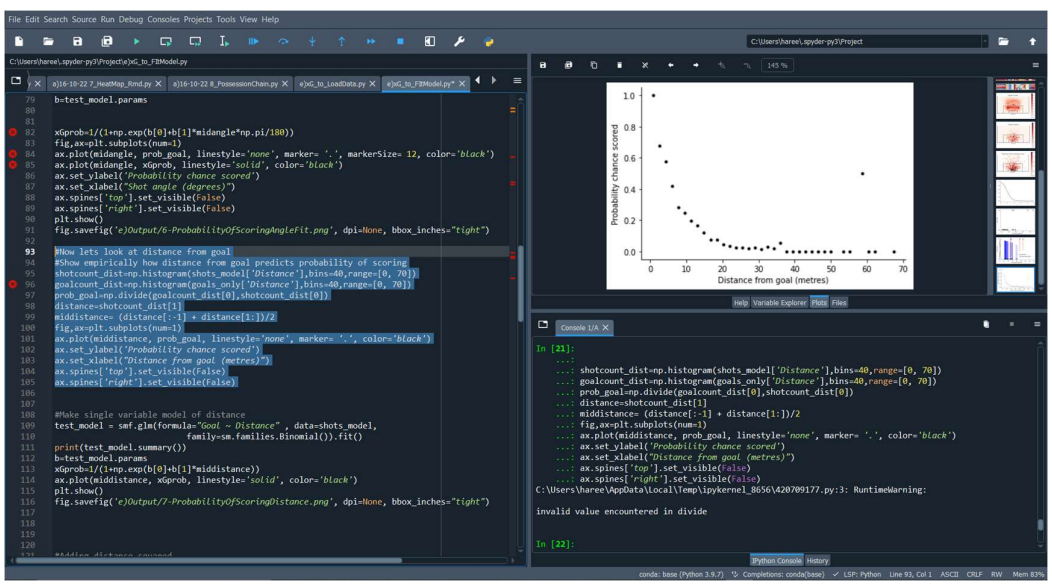


Fig. 5.2.10 Snippet of xG_to_FitModel code

The code snippet shown above in Fig. 5.2.10 is the code written to predict the goals scored by the teams.

5.3 RESULTS

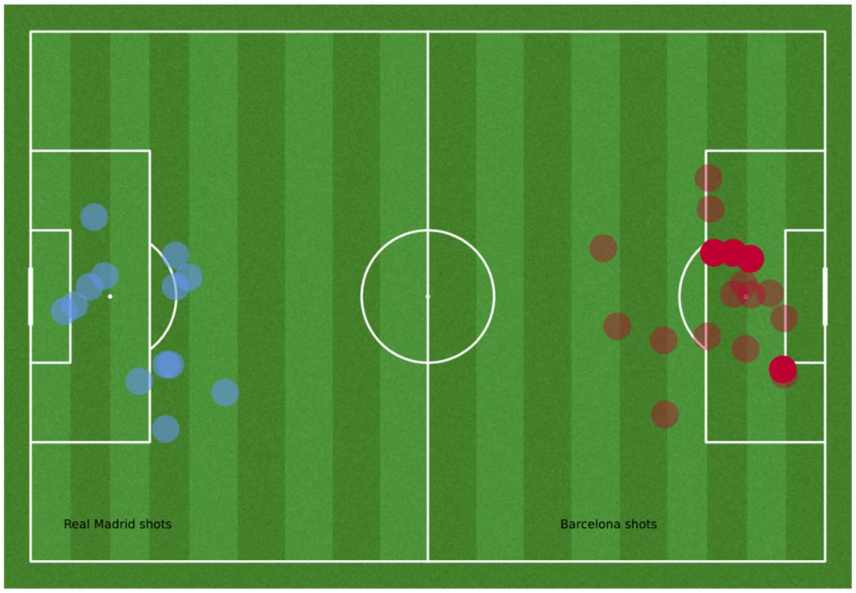


Fig. 5.3.1 Shot map of Real Madrid & Barcelona (match date: 16-10-2022)

The plot shown above in Fig.5.3.1 represents the shot map which is a representation of all the locations a shot was taken on the pitch for the match dated 16-10-22.

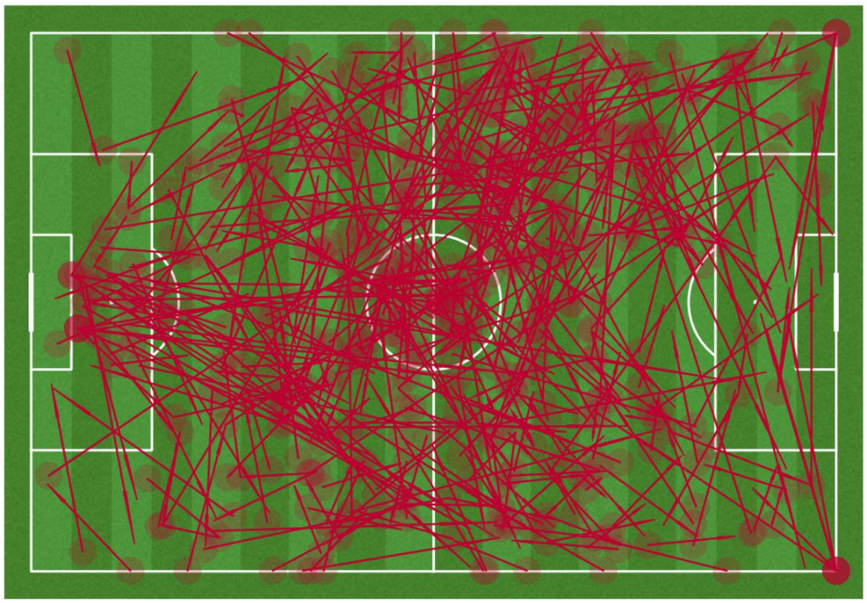


Fig 5.3.2 Pass map of Barcelona (match date: 16-10-2022)

The plot shown above in Fig.5.3.2 represents the pass map which is a network of all the successful passes made by Barcelona team for the match dated 16-10-22.

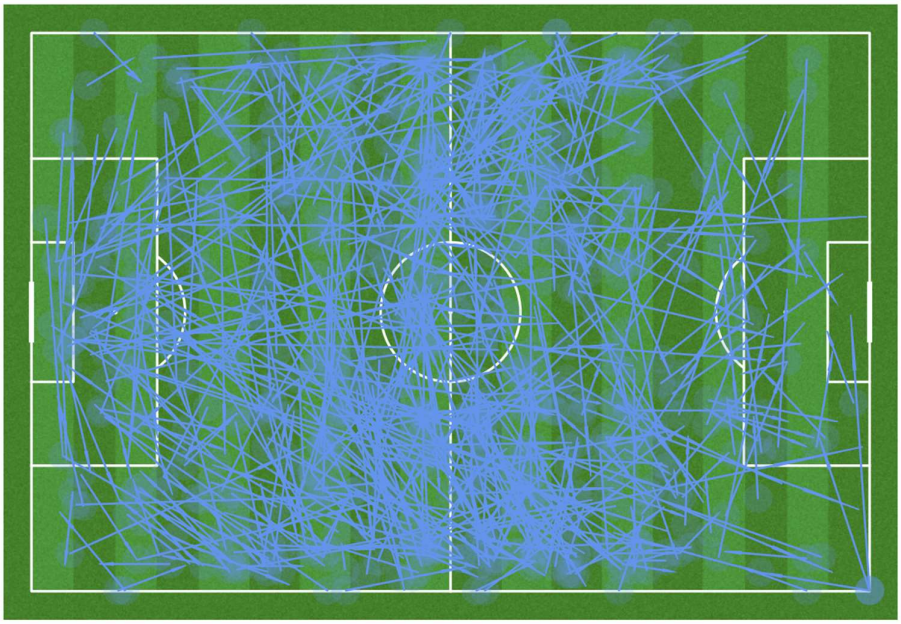


Fig 5.3.3 Pass map of Real Madrid (match date: 16-10-2022)

The plot shown above in Fig.5.3.2 represents the pass map which is a network of all the successful passes made by Madrid team for the match dated 16-10-22.

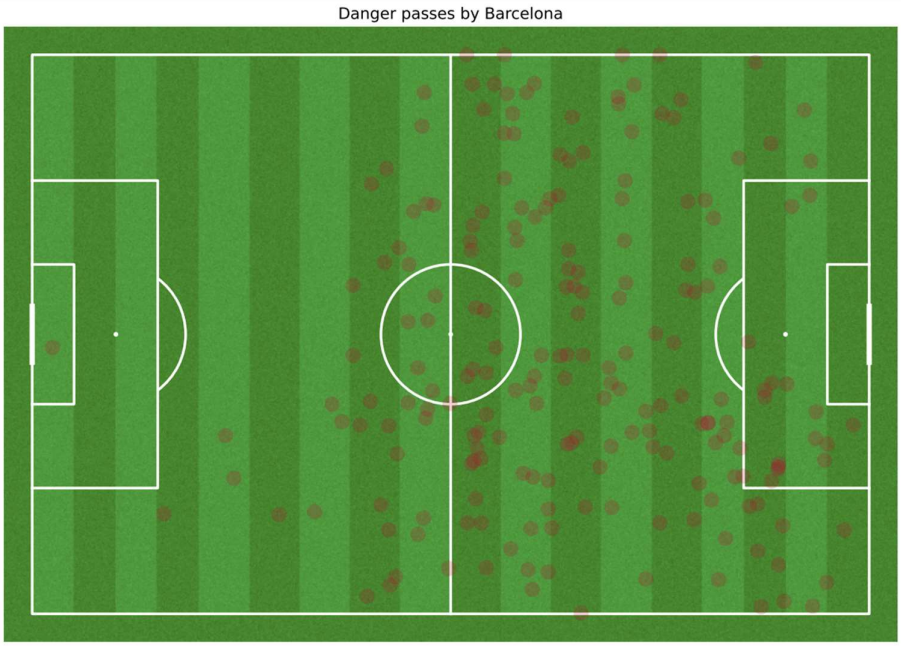


Fig 5.3.4 Dangerous Pass map by Barcelona (match date: 16-10-2022)

The plot above in Fig.5.2.4 shows the map of all the dangerous passes made by Barcelona team.

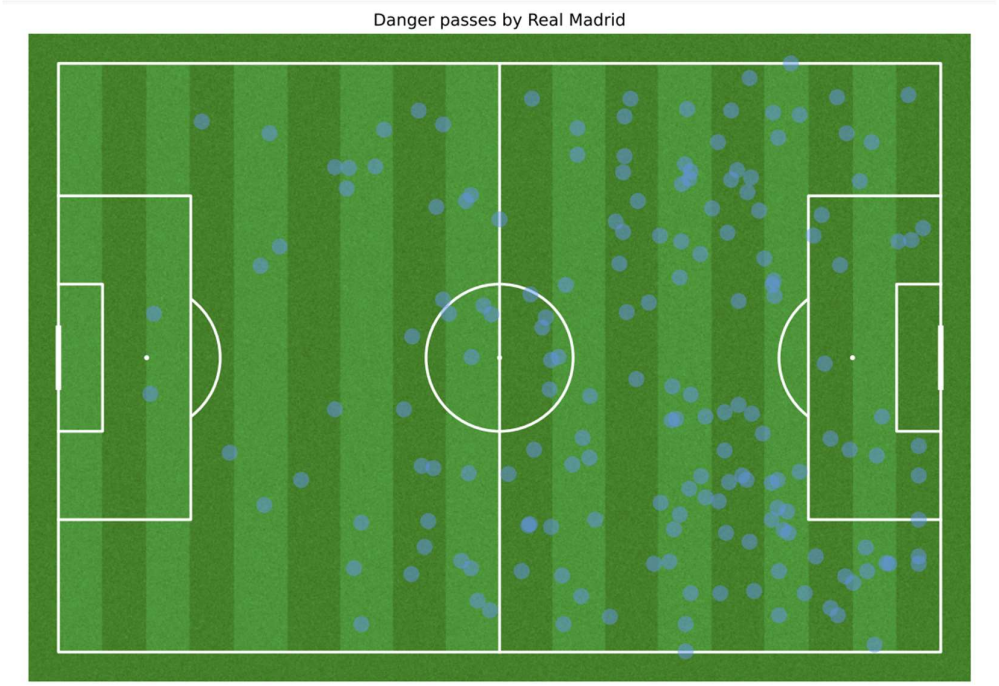


Fig 5.3.5 Dangerous Pass map by Real Madrid (match date: 16-10-2022)

The plot shown above in Fig.5.3.5 shows the map of all the dangerous passes made by Madrid team.

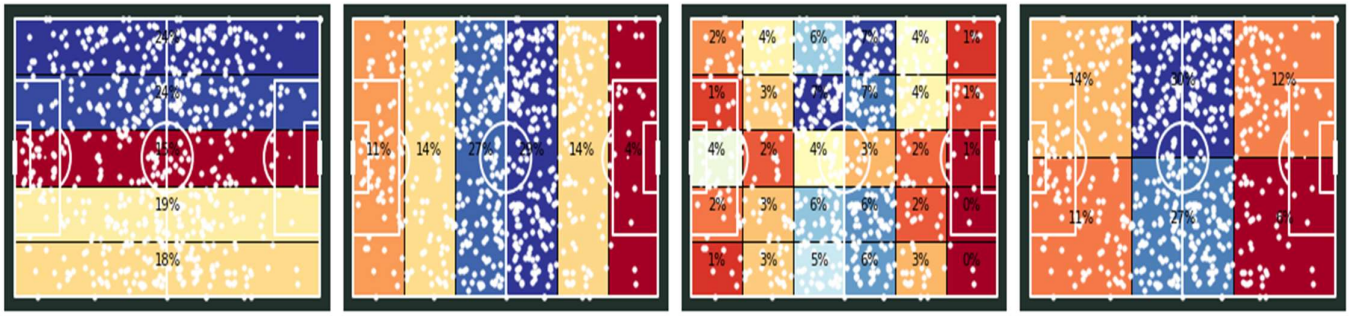


Fig 5.3.6 Heat map of Barcelona (match date: 16-10-2022)

The plot shown above in Fig.5.3.6 shows different heat maps generated for Barcelona team in the match dated 16-10-22.

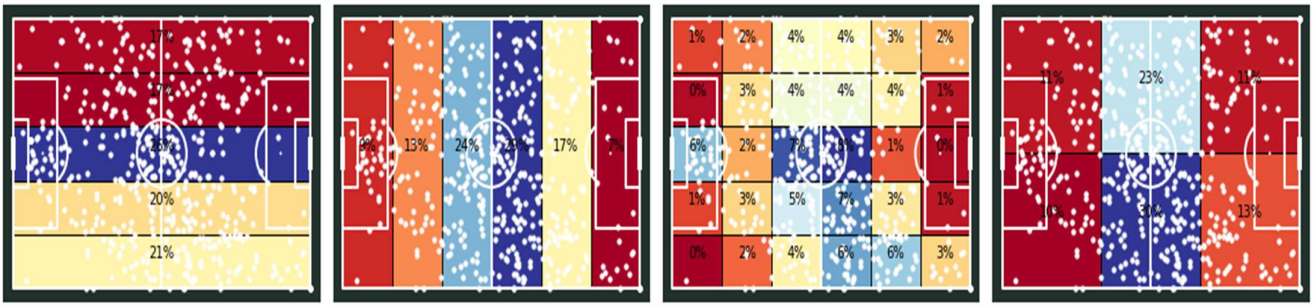


Fig 5.3.7 Heat map of Real Madrid (match date: 16-10-2022)

The plot shown above in Fig.5.3.7 shows different heat maps generated for Barcelona team in the match dated 16-10-22.

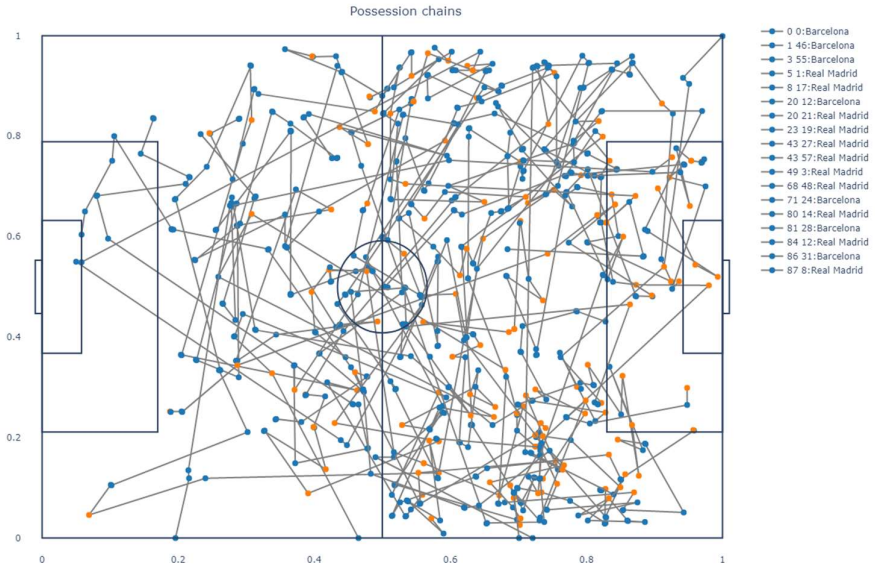


Fig 5.3.8 Possession chains map of Real Madrid & Barcelona (match date: 16-10-2022)

The plot shown above in Fig.5.3.8 shows the possession chain of both the teams in the match dated 16-10-22.

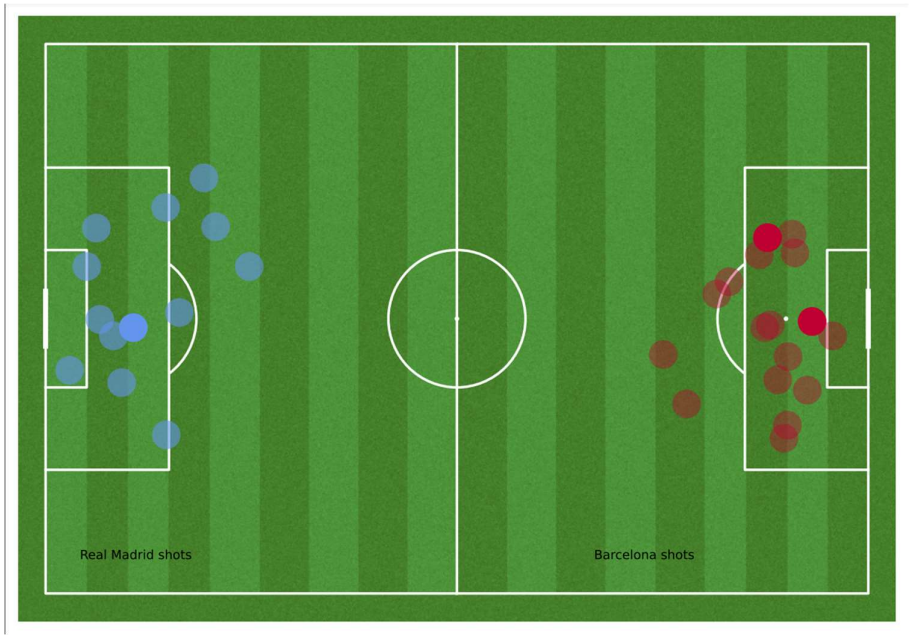


Fig. 5.3.9 Short map of Real Madrid & Barcelona (match date: 20-03-2023)

The plot shown above in Fig.5.3.9 represents the shot map which is a representation of all the locations a shot was taken on the pitch for the match dated 20-03-22.

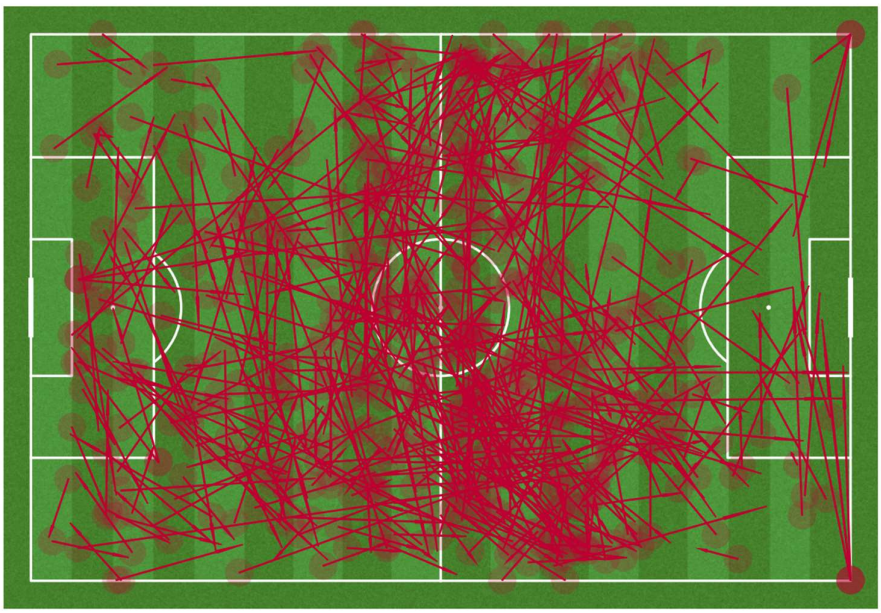


Fig 5.3.10 Pass map of Barcelona (match date: 20-03-2023)

The plot shown above in Fig.5.3.10 represents the pass map which is a network of all the successful passes made by Barcelona team for the match dated 20-03-22.

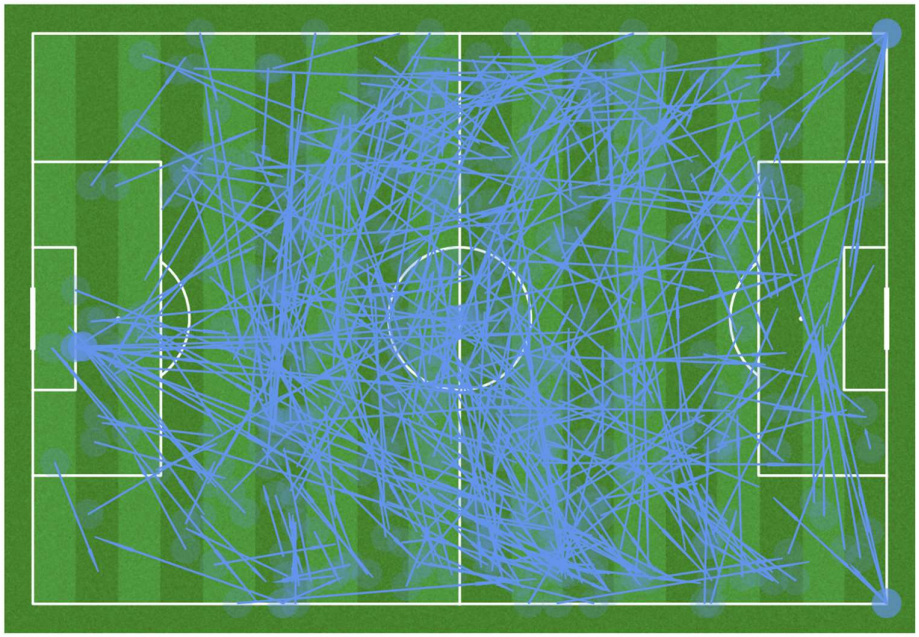


Fig 5.3.11 Pass map of Real Madrid (match date: 20-03-2023)

The plot shown above in Fig.5.3.11 represents the pass map which is a network of all the successful passes made by Barcelona team for the match dated 20-03-22.

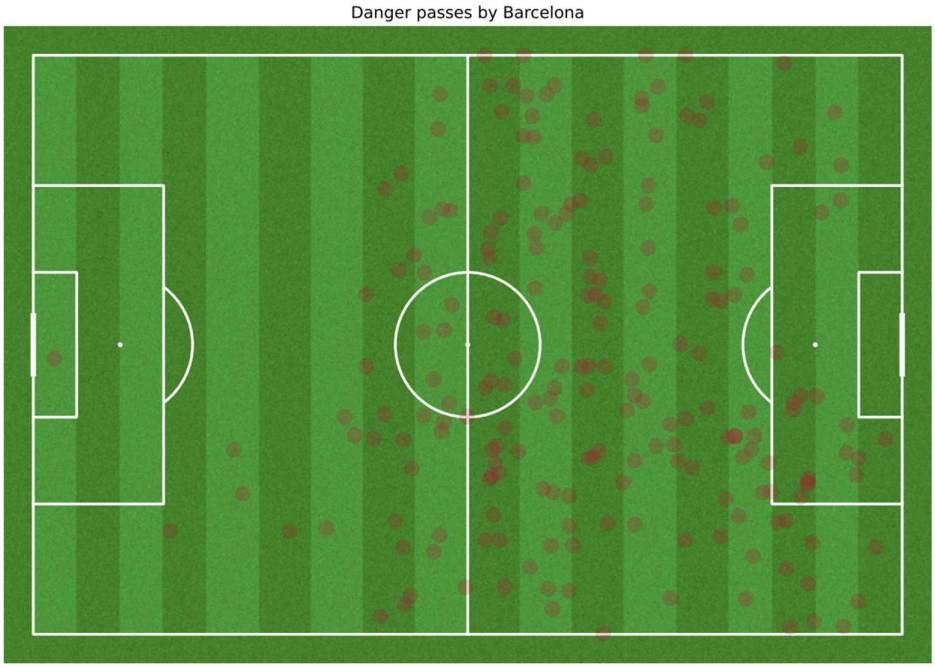


Fig 5.3.12 Dangerous Pass map by Barcelona (match date: 20-03-2023)

The plot above in Fig.5.3.12 shows the map of all the dangerous passes made by Barcelona team.

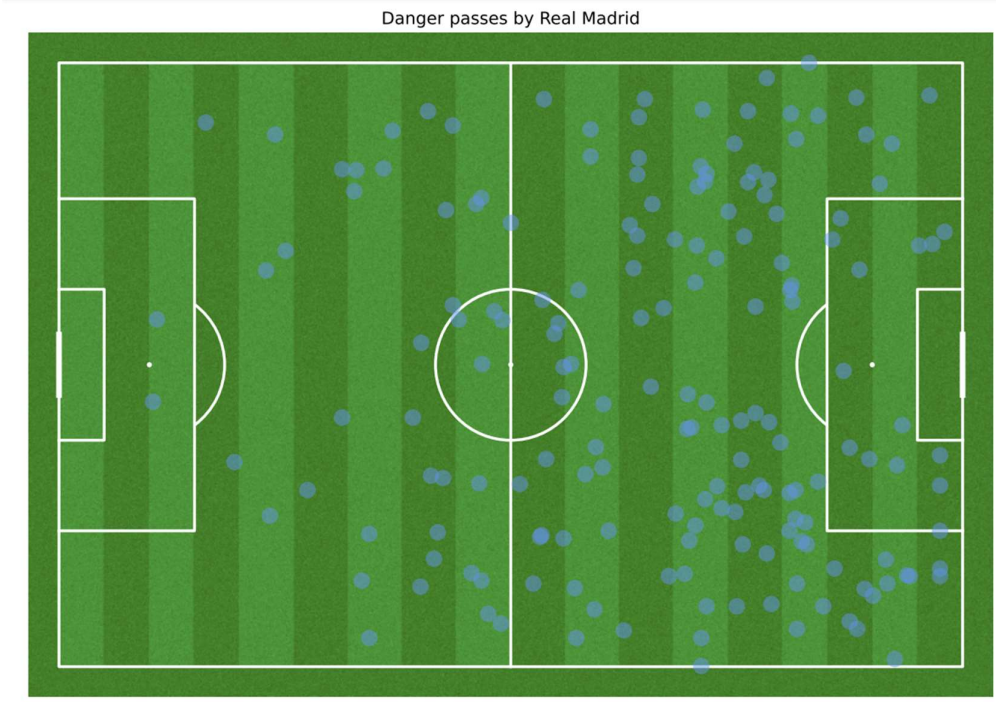


Fig 5.3.13 Dangerous Pass map by Real Madrid (match date: 20-03-2023)

The plot shown above in Fig.5.3.5 shows the map of all the dangerous passes made by Madrid team.

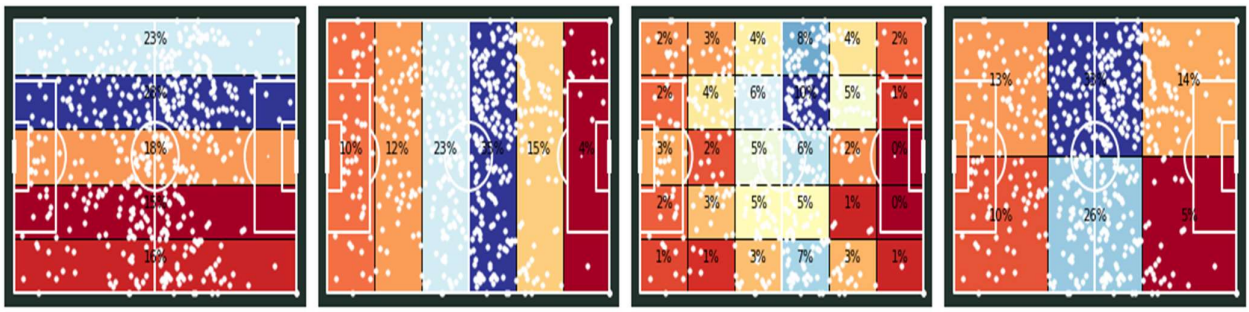


Fig 5.3.14 Heat map of Barcelona (match date: 20-03-2023)

The plot shown above in Fig.5.3.14 shows different heat maps generated for Barcelona team in the match dated 20-03-22.

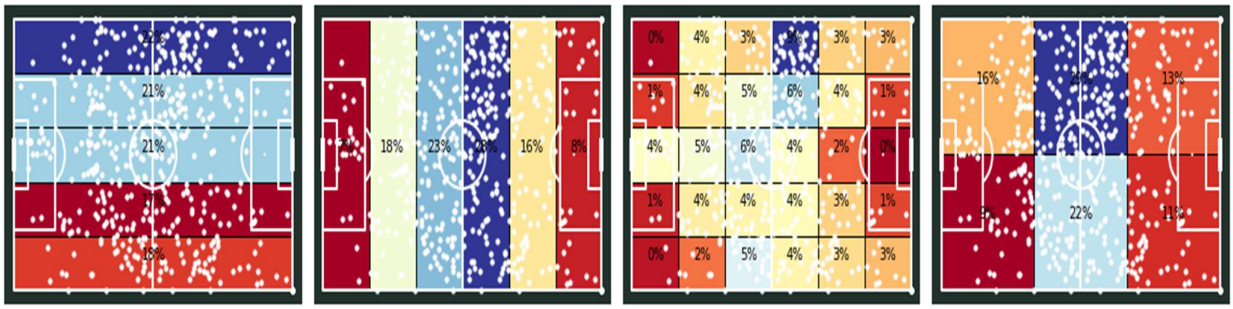


Fig 5.3.15 Heat map of Real Madrid (match date: 20-03-2023)

The plot shown above in Fig.5.3.15 shows different heat maps generated for Barcelona team in the match dated 20-03-22.

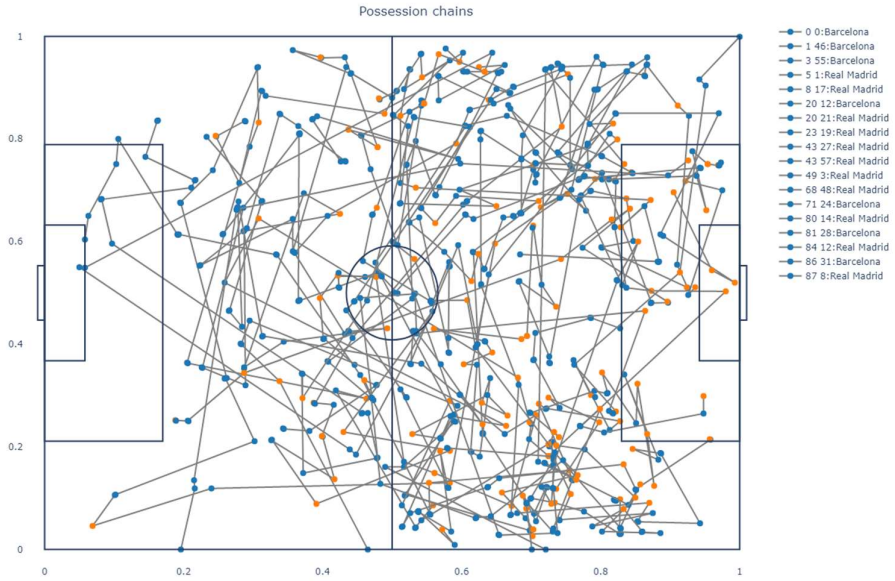


Fig 5.3.16 Possession chains map of Real Madrid & Barcelona (match date: 20-03-2023)

The plot shown above in Fig.5.3.16 shows the possession chain of both the teams in the match dated 20-03-22.

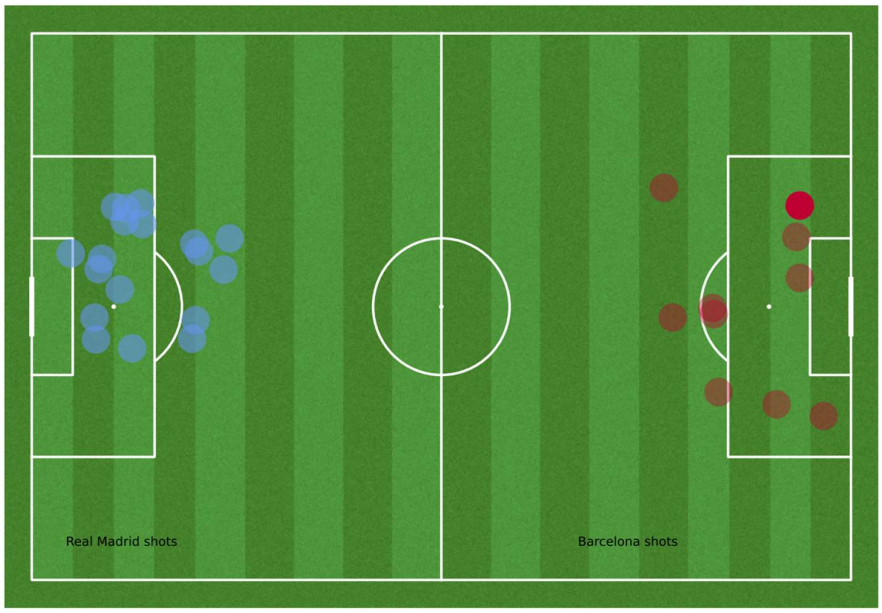


Fig. 5.3.17 Short map of Real Madrid & Barcelona (match date: 03-03-2023)

The plot shown above in Fig.5.3.17 represents the shot map which is a representation of all the locations a shot was taken on the pitch for the match dated 03-03-22.

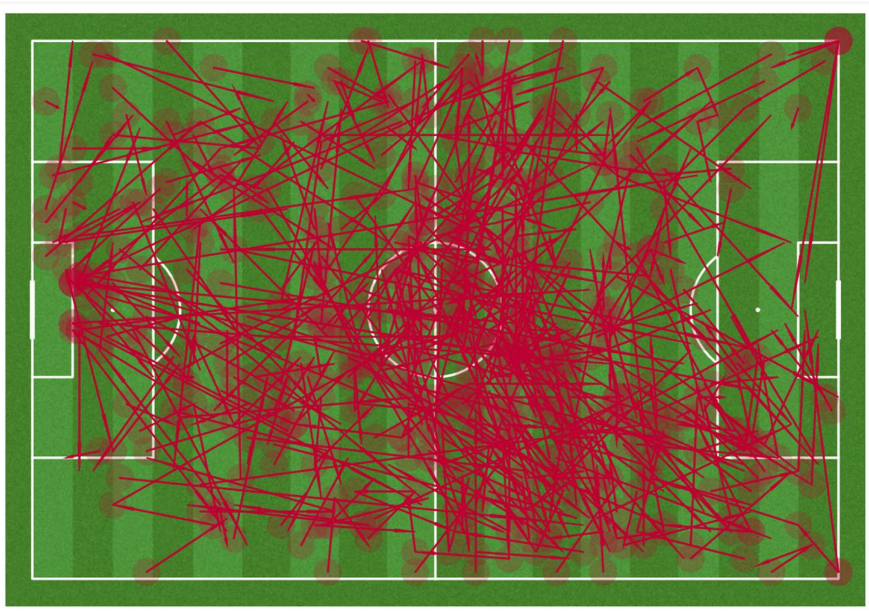


Fig 5.3.18 Pass map of Barcelona (match date: 03-03-2023)

The plot shown above in Fig.5.3.18 represents the pass map which is a network of all the successful passes made by Barcelona team for the match dated 03-03-22.

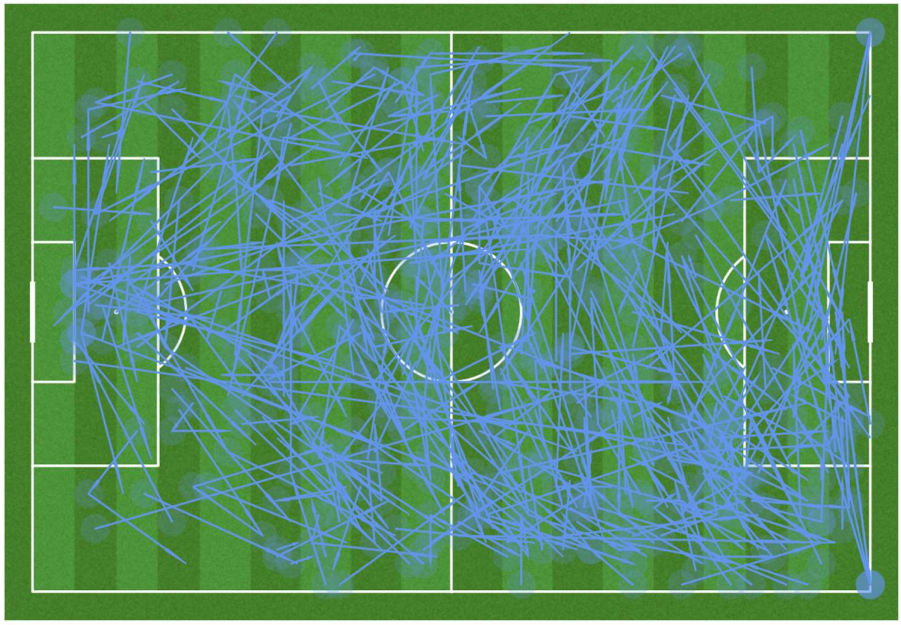


Fig 5.3.19 Pass map of Real Madrid (match date: 03-03-2023)

The plot shown above in Fig.5.3.19 represents the pass map which is a network of all the successful passes made by Barcelona team for the match dated 03-03-22.

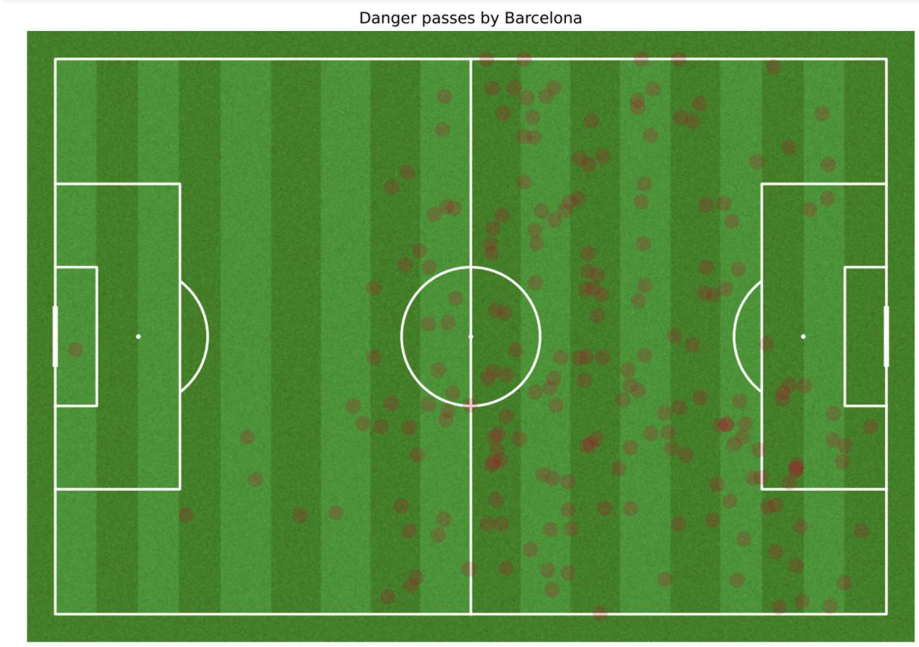


Fig 5.3.20 Dangerous Pass map by Barcelona (match date: 03-03-2023)

The plot above in Fig.5.3.20 shows the map of all the dangerous passes made by Barcelona team.

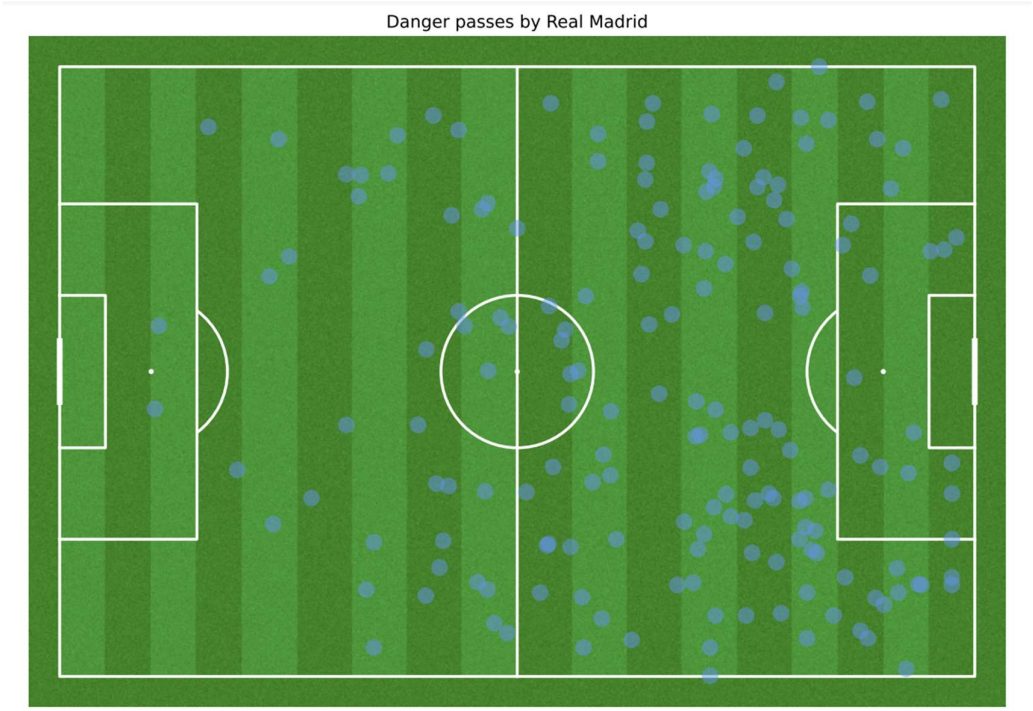


Fig 5.3.21 Dangerous Pass map by Real Madrid (match date: 03-03-2023)

The plot shown above in Fig.5.3.21 shows the map of all the dangerous passes made by Madrid team.

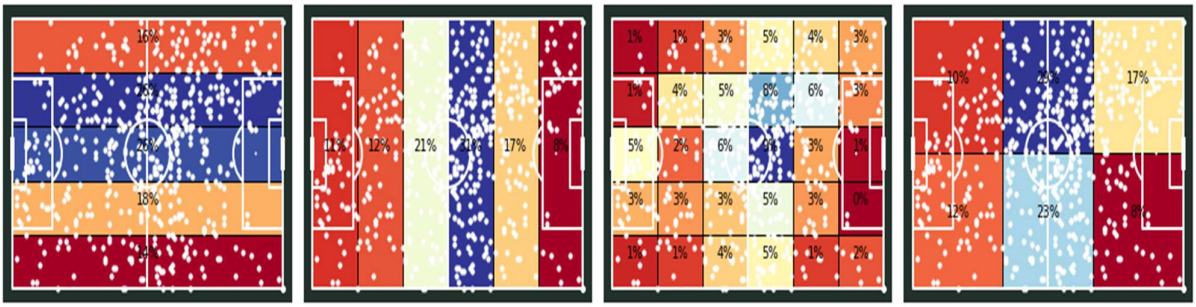


Fig 5.3.22 Heat map of Barcelona (match date: 03-03-2023)

The plot shown above in Fig.5.3.22 shows different heat maps generated for Barcelona team in the match dated 03-03-22.

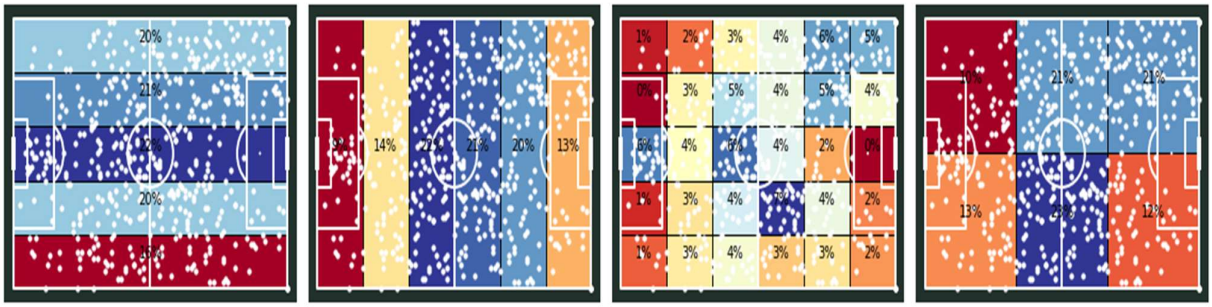


Fig 5.3.23 Heat map of Real Madrid (match date: 03-03-2023)

The plot shown above in Fig. 5.3.23 shows the different heat maps generated for Madrid team in the match dated 03-03-22.

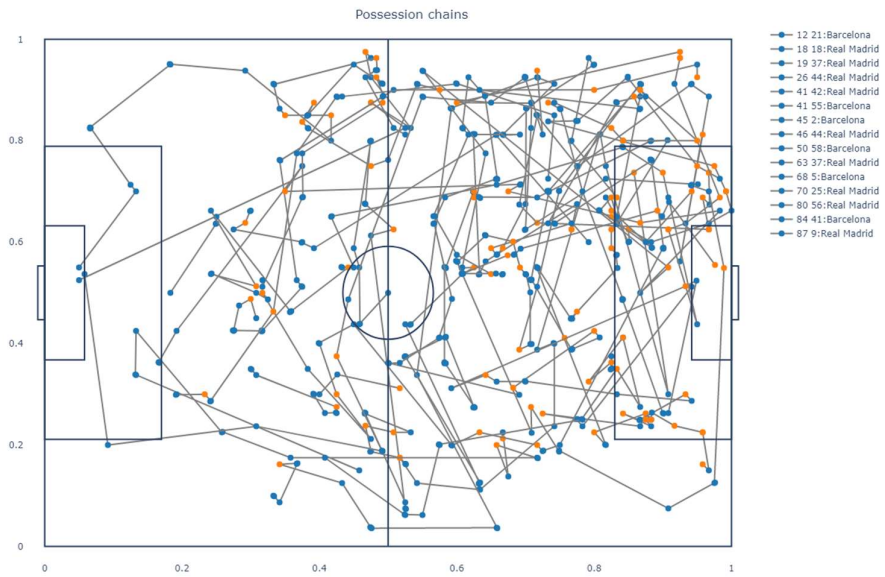


Fig 5.3.24 Possession chains map of Real Madrid & Barcelona (match date: 03-03-2023)

The plot above in Fig. 5.3.25 shows the possession chain of both the teams for the match dated 03-03-22.

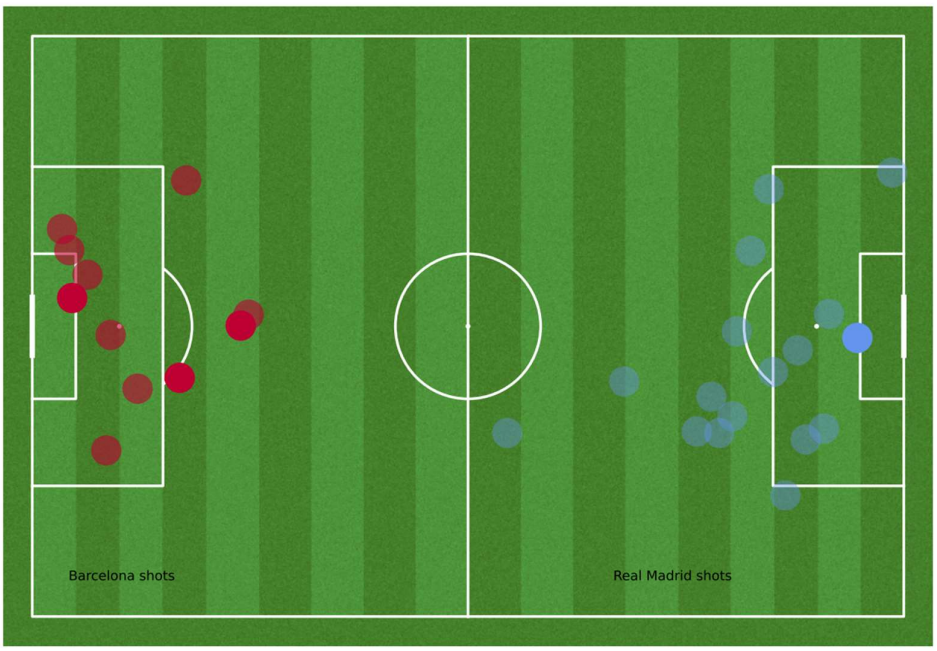


Fig. 5.3.25 Short map of Real Madrid & Barcelona (match date: 16-01-2023)

The plot shown above in Fig.5.3.25 represents the shot map which is a representation of all the locations a shot was taken on the pitch for the match dated 16-01-23.

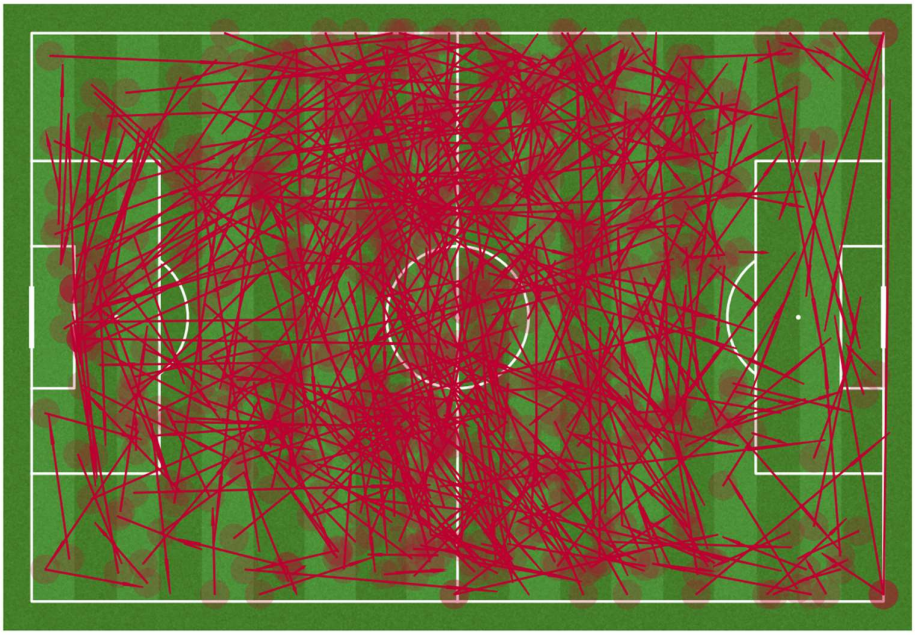


Fig 5.3.26 Pass map of Barcelona (match date: 16-01-2023)

The plot above in Fig. 5.3.26 represents the pass map which is a network of all the successful passes made by Barcelona team for the match 16-01-23.

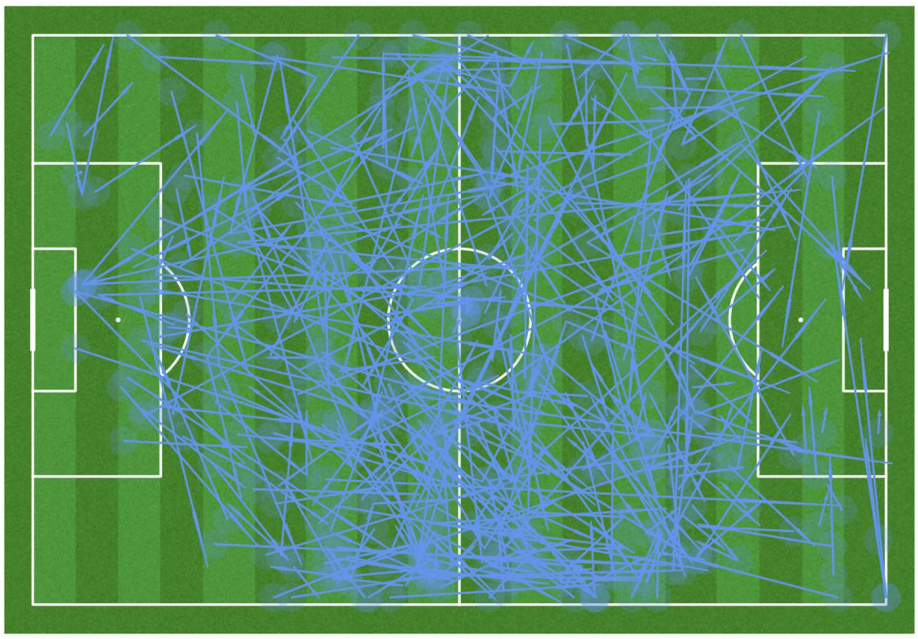


Fig 5.3.27 Pass map of Real Madrid (match date: 16-01-2023)

The plot above in Fig. 5.3.27 represents a pass map which is a network of all the successful passes made by Madrid team for the match 16-01-23.

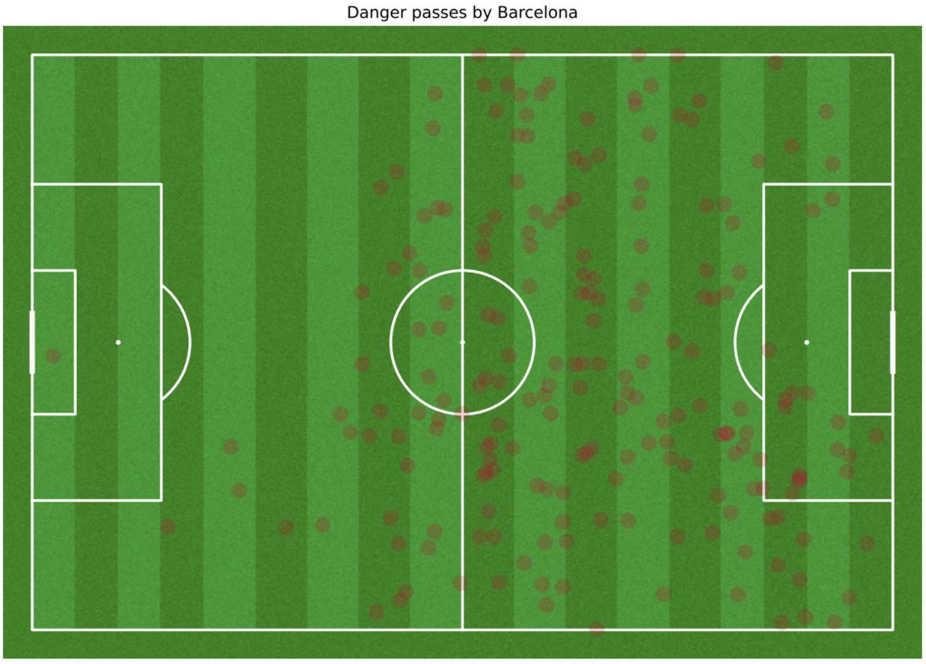


Fig 5.3.28 Dangerous Pass map by Barcelona (match date: 16-01-2023)

The plot shown above in Fig. 5.3.29 show all the dangerous passes made bby Barcelona team.

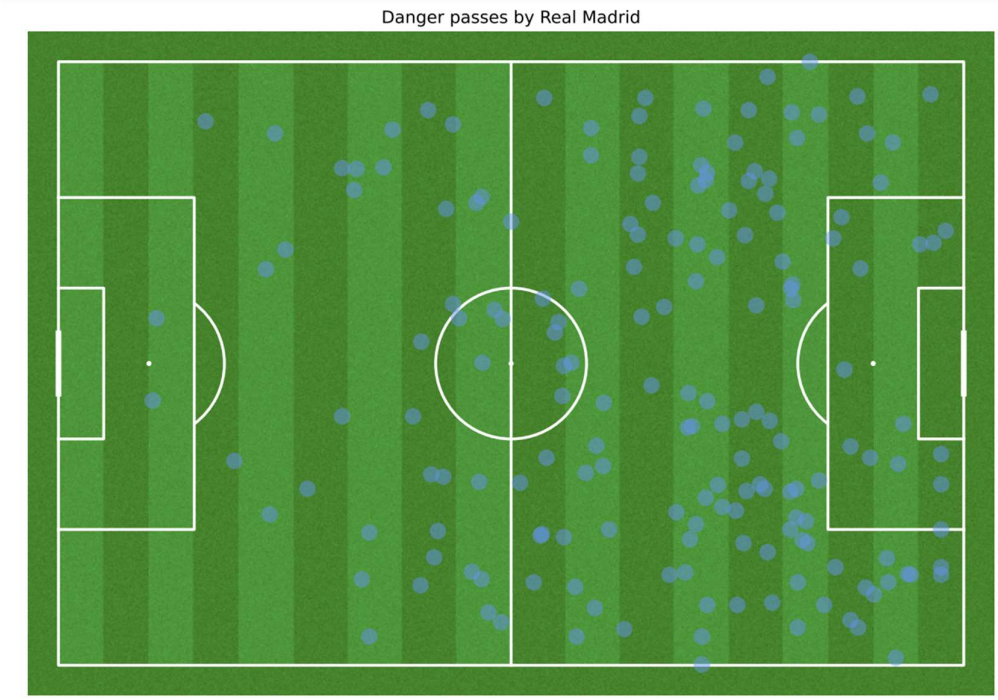


Fig 5.3.29 Dangerous Pass map by Real Madrid (match date: 16-01-2023)

The plot shown above in Fig. 5.3.29 shows all the dangerous passes made by the Madrid team.

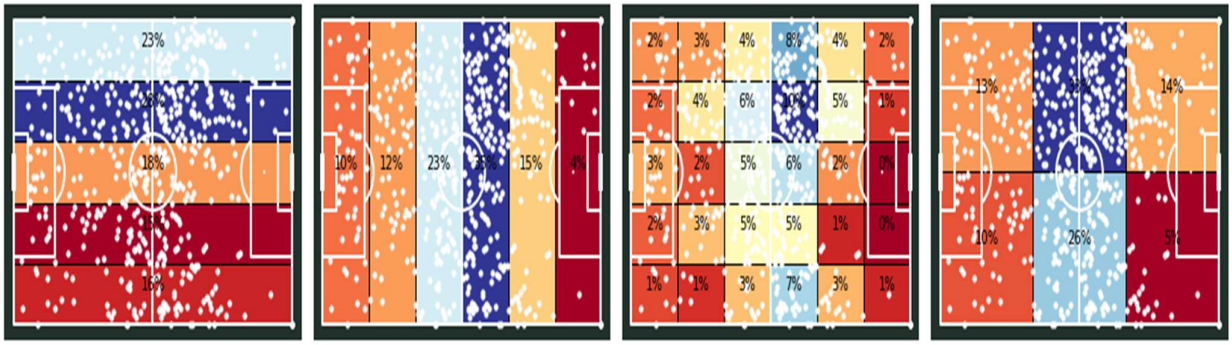


Fig 5.3.30 Heat map of Barcelona (match date: 16-01-2023)

The plot shown above in Fig. 5.3.30 shows the different heat maps generated by Barcelona for the match dated 16-01-23.

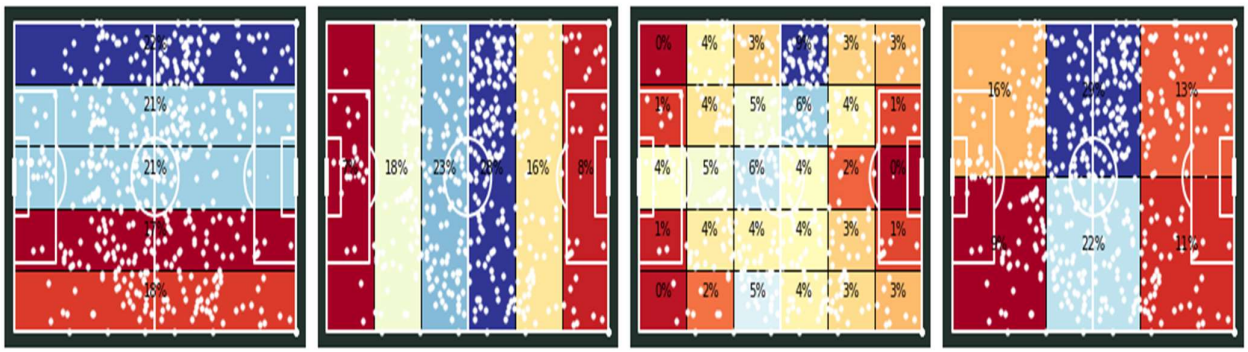


Fig 5.3.31 Heat map of Real Madrid (match date: 16-01-2023)

The plot shown above in Fig. 5.3.31 show the different heat maps generated by Madrid for the match dated 16-01-23.

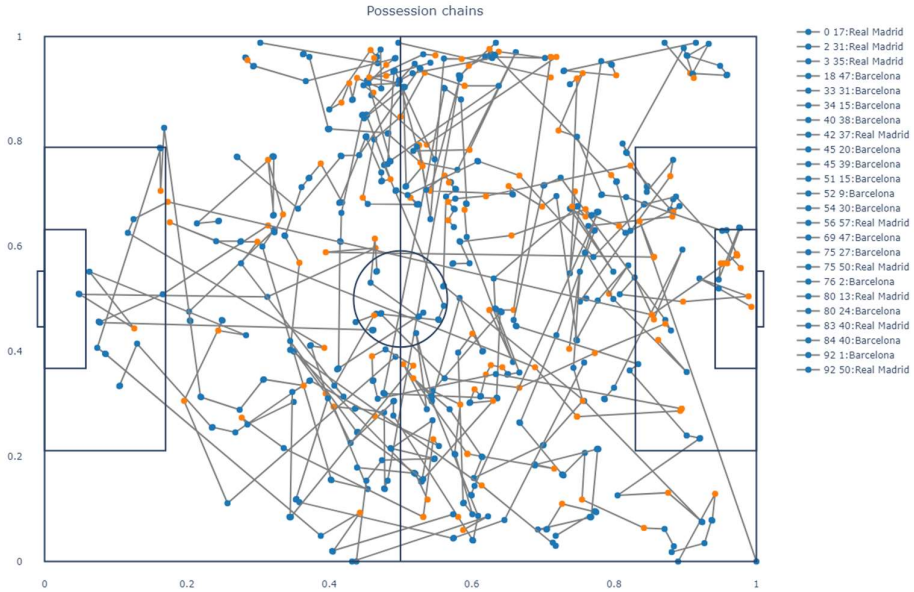


Fig 5.3.32 Possession chains map of Real Madrid & Barcelona (match date: 16-01-2023)

The plot above in Fig. 5.3.32 shows the possession chains of both the teams for the match dated 16-01-23.

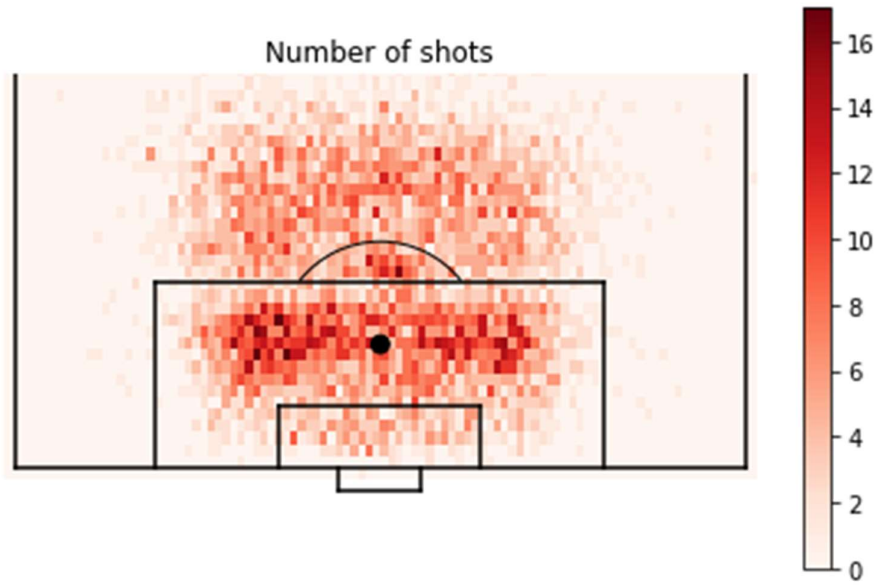


Fig. 5.3.33 Number of Shots

The plot shown above in Fig. 5.3.33 represents the locations from which various shots on goal were attempted. It should also be noted that these are just attempts made to score a goal and not the actual goal.

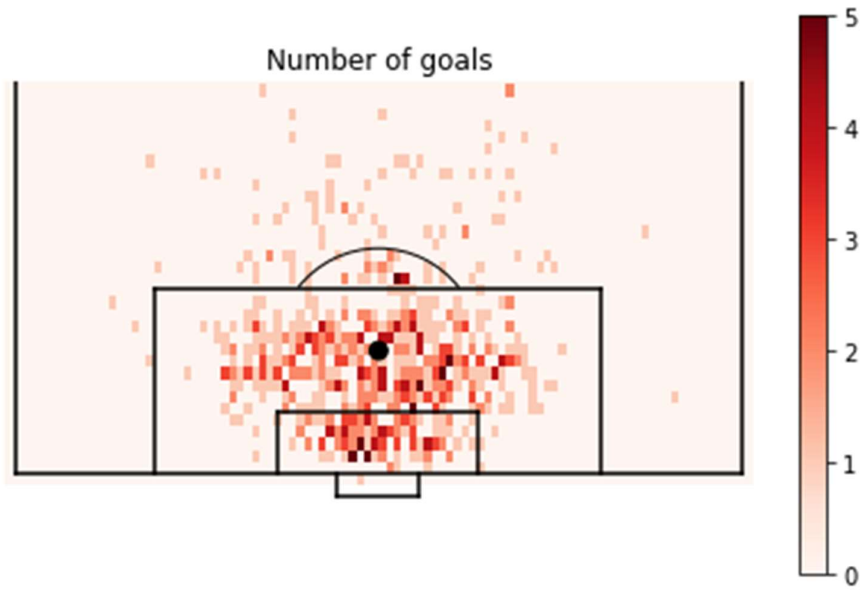


Fig. 5.2.34 Number of Goals

The plot above in Fig. 5.3.34 represents the locations from which various goals have occurred during open play.

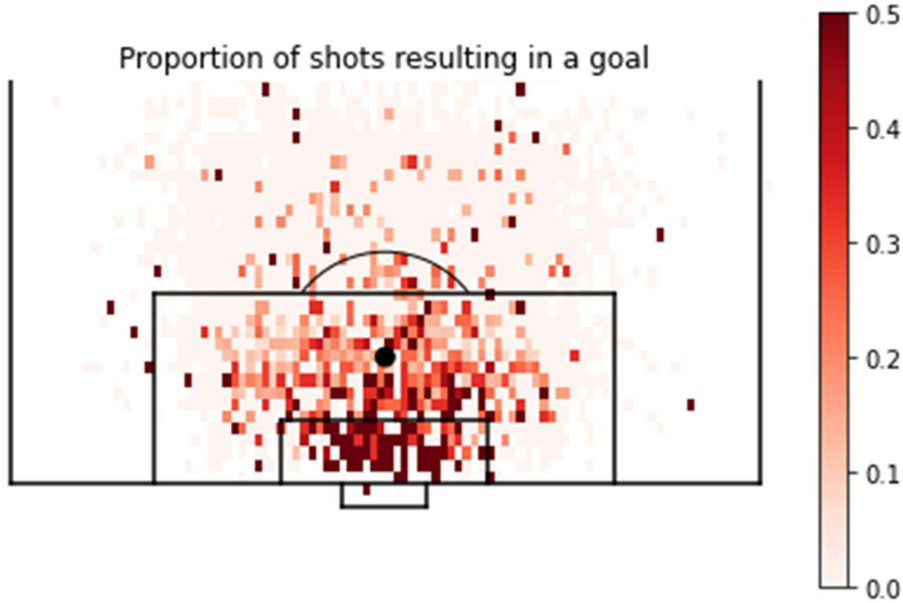


Fig 5.3.35 Probability of Scoring

The plot shown in Fig. 5.3.35 represents the probability of scoring from a specific location. It is created by taking the number of goals scored from a location and dividing it with the number of shots attempted from that position.

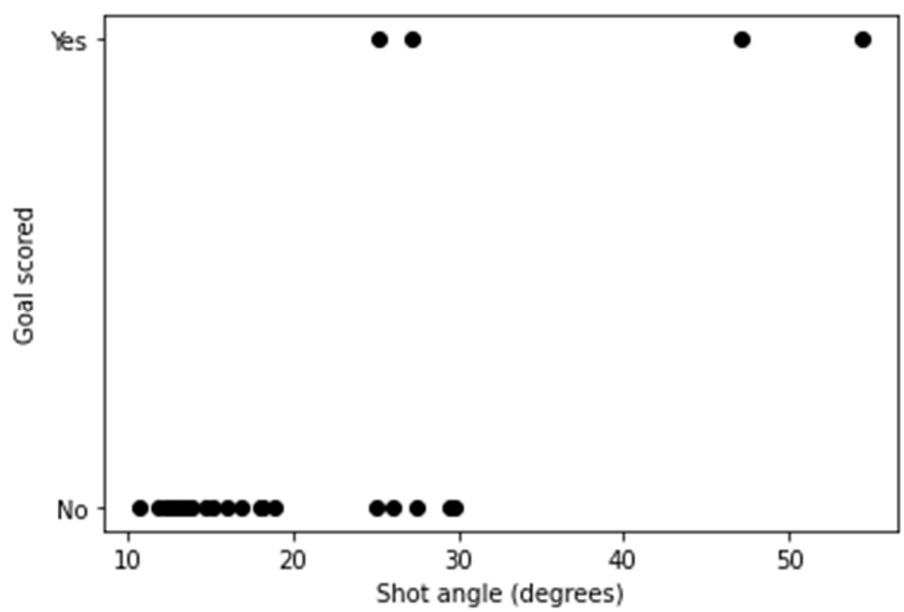


Fig 5.3.36 Shot Goal Angle

The graph shown above in Fig. 5.3.36 represents whether a goal was scored from a specific shot angle. The formula used to calculate the goal angle is:

$$\tan\theta = \frac{(7.32)x}{x^2 + y^2 - (\frac{7.32}{2})^2}$$

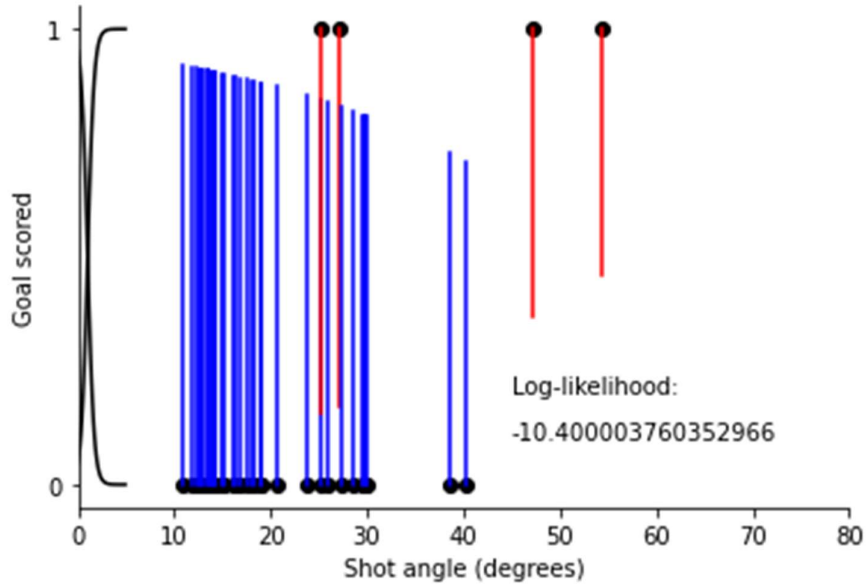


Fig. 5.3.37 Goal Scored Shot Angle

The graph shown in Fig. 5.3.37 represents the log likelihood of a goal being scored from a given shot angle.

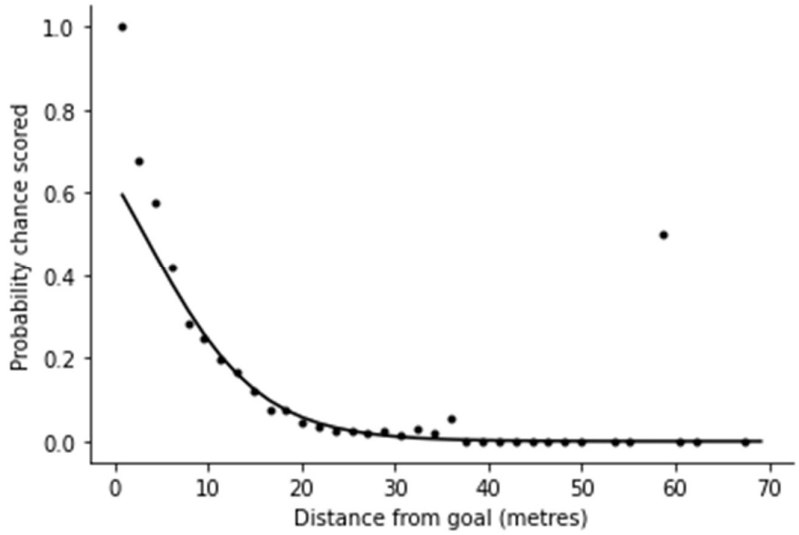


Fig. 5.3.38 Probability of Scoring Distance

The graph shown in Fig. 5.3.38 represents the probability of scoring from a certain distance from the goal. This can also be interpreted that the probability of scoring a goal decrease with increase in the distance from goal.

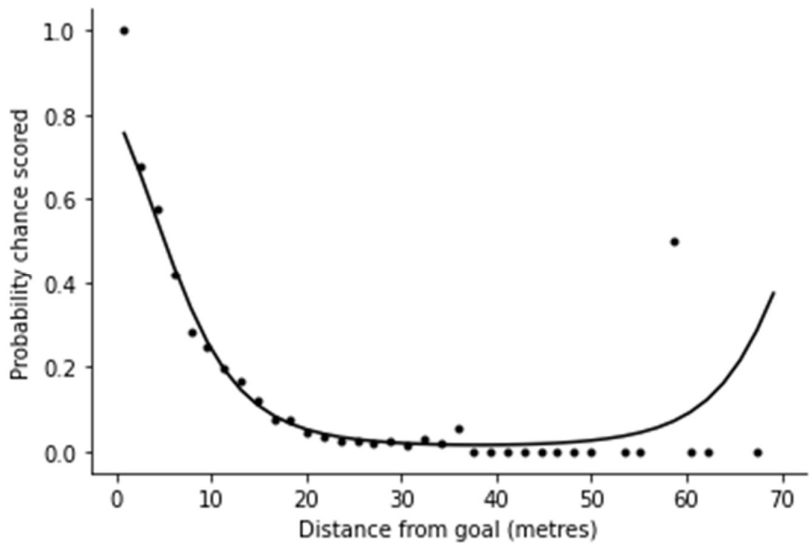


Fig. 5.3.39 Probability of Scoring Distance Squared

The graph shown in Fig. 5.3.39 represents the probability of the scoring distance squared. This can be done using logistic regression and the derivation acquired is used in predicting the goals.

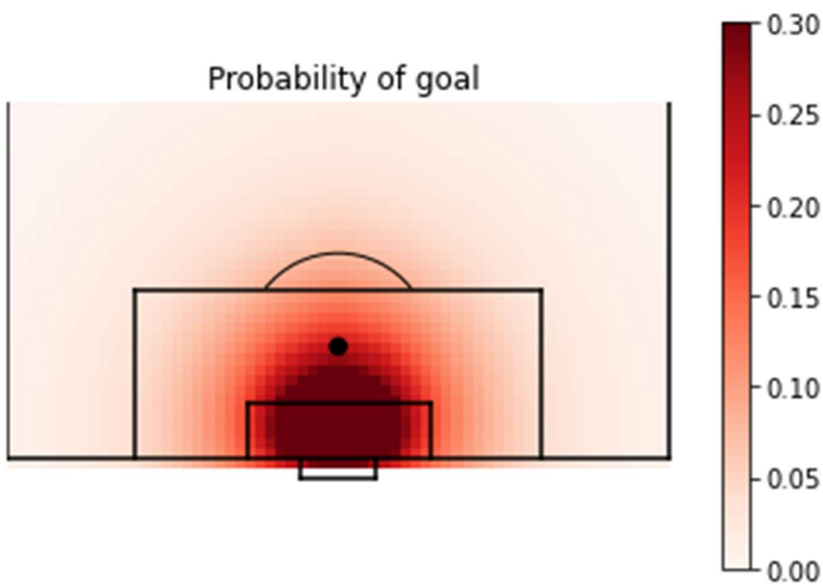


Fig. 5.3.40 Probability of Goal

The plot shown in Fig. 5.3.40 represents the probability of scoring a goal from a specific location using a heat map and fined tuned using logistic regression.

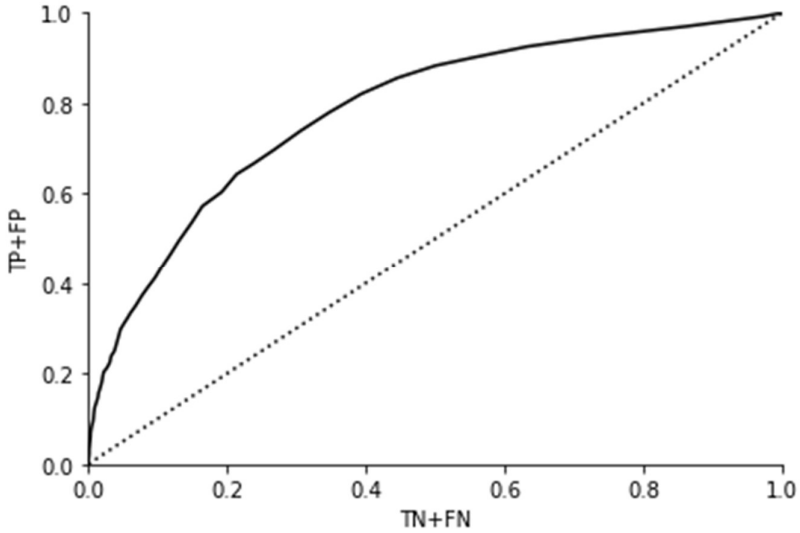


Fig. 5.3.41 ROC Curve

The graph shown above in Fig. 5.3.41 represents the ROC curve also curved as a Receiver Operating Characteristic curve which shows the performance of the prediction model at all the different thresholds.

6. CONCLUSION & FUTURE ENHANCEMENTS

6.1. CONCLUSION

In the sports industry, prediction and analysis are used hand-in-hand by both sports analysts and sports scientists. The primary objective of this project is to predict the probability of the goals scored by both the teams and analyse the different aspects of the football match. In the first phase computer vision was used to annotate the frames with the help of object detection and YOLOv8. Using the Roboflow software the frames were converted into dataset (JSON file). A total of four datasets were created for four different matches. In the second phase the datasets were studied and the necessary attributes and parameters like the position of the player, the position of the ball etc. were taken to perform further analysis.

In the third phase individual analysis of each dataset for specific match was performed during which the specific team's performance was analysed and shot map's, pass map's, dangerous pass map's, heat maps and possession chains of both teams were generated. In the fourth phase using the analysis acquired the possibility of goals in football match is predicted using logistic regression. The resultant ROC curve shows the performance of the prediction model.

6.2. FUTURE ENHANCEMENTS

The following points focus on the future scope of this project:

- Identifying and analysing individual player's movement and judging their technical abilities.
- Including parameters such as pitch conditions and weather conditions for prediction of goals.
- Predicting the winner of a game by analysing complete football matches and thus increase the reliability of the model.

REFERENCES

1. https://www.researchgate.net/publication/273164409_An_Improved_Prediction_System_for_Football_a_Match_Result
2. https://www.researchgate.net/publication/336785507_Data_Analytics_in_Football_Positional_Data_Collection_Modeling_and_Analysis
3. https://www.ibm.com/topics/artificial-intelligence
4. https://github.com/ultralytics/ultralytics.git
5. https://www.youtube.com/watch?v=oXlwWbU8l2o
6. https://www.ibm.com/topics/computer-vision
7. https://ischoolonline.berkeley.edu/data-science/what-is-data-science/
8. https://www.ibm.com/topics/machine-learning
9. https://www.ibm.com/topics/deep-learning
10. https://developers.google.com/machine-learning/crash-course/classification/roc-and-auc

APPENDIX

A: source/pseudo code

16_10_22_cv.py

```
import os
HOME = os.getcwd()
print(HOME)
# Pip install method (recommended)
!pip install ultralytics==8.0.20
from IPython import display
display.clear_output()
import ultralytics
ultralytics.checks()
from ultralytics import YOLO
from PIL import Image
!pip install roboflow
from roboflow import Roboflow
rf = Roboflow(api_key="o7rqZ6ii3o9wMfN5lhKn")
project = rf.workspace("programmers").project("16-10-22-1-3")
dataset = project.version(1).download("yolov5")
!yolo task=detect mode=train model=yolov8s.pt data=/content/datasets/16-10-22-1-3-1/data.yaml epochs=25 imgsz=800 plots=True
!ls /content/runs/detect/train2
im = Image.open('/content/runs/detect/train2/confusion_matrix.png')
im.show()
im = Image.open('/content/runs/detect/train2/results.png')
im.show()
im = Image.open('/content/runs/detect/train2/val_batch0_pred.jpg')
im.show()
!yolo task=detect mode=val model=/content/runs/detect/train2/weights/best.pt data=/content/datasets/16-10-22-1-3-1/data.yaml
!yolo task=detect mode=predict model=/content/runs/detect/train2/weights/best.pt conf=0.25 source=/content/datasets/16-10-22-1-3-1/test/images save=True
import glob
from IPython.display import Image, display
for image_path in glob.glob(f'/content/runs/detect/predict/*.jpg')[:3]:
    display(Image(filename=image_path, width=600))
    print("\n")
!yolo task=detect mode=predict model=/content/runs/detect/train2/weights/best.pt conf=0.25 source=/content/runs/16thMarch2022.mp4 save=True
```

20_3_23_cv.py

```
!nvidia-smi
import os
HOME = os.getcwd()
print(HOME)
!pip install ultralytics==8.0.20
from IPython import display
display.clear_output()
import ultralytics
ultralytics.checks()
from ultralytics import YOLO
from PIL import Image
!pip install roboflow
from roboflow import Roboflow
rf = Roboflow(api_key="o7rqZ6ii3o9wMfN5lhKn")
project = rf.workspace("programmers").project("20-3-23-2-1")
dataset = project.version(1).download("yolov5")
!yolo task=detect mode=train model=yolov8s.pt data=/content/datasets/20-3-23-2-1-1/data.yaml epochs=25 imgsz=800 plots=True
!ls /content/runs/detect/train
im = Image.open('/content/runs/detect/train/confusion_matrix.png')
im.show()
im = Image.open('/content/runs/detect/train/results.png')
im.show()
im = Image.open('/content/runs/detect/train/val_batch0_pred.jpg')
im.show()
!yolo task=detect mode=val model=/content/runs/detect/train/weights/best.pt data=/content/datasets/20-3-23-2-1-1/data.yaml
!yolo task=detect mode=predict model=/content/runs/detect/train/weights/best.pt conf=0.25 source=/content/datasets/20-3-23-2-1-1/test/images save=True
import glob
from IPython.display import Image, display
for image_path in glob.glob(f'/content/runs/detect/predict/*.jpg')[:3]:
    display(Image(filename=image_path, width=600))
    print("\n")
!yolo task=detect mode=predict model=/content/runs/detect/train/weights/best.pt conf=0.25 source=/content/runs/20thMarch2023.mp4 save=True
```

3_3_23_cv.py

```
!nvidia-smi
!pip install ultralytics==8.0.20
from IPython import display
display.clear_output()
import ultralytics
ultralytics.checks()
from ultralytics import YOLO
from PIL import Image
!pip install roboflow
from roboflow import Roboflow
rf = Roboflow(api_key="o7rqZ6ii3o9wMfN5lhKn")
project = rf.workspace("programmers").project("3-3-23-1-0")
dataset = project.version(1).download("yolov5")
!yolo task=detect mode=train model=yolov8s.pt data=/content/datasets/3-3-23-1-0-1/data.yaml epochs=25 imgsz=800 plots=True
!ls /content/runs/detect/train
im = Image.open('/content/runs/detect/train/confusion_matrix.png')
im.show()
im = Image.open('/content/runs/detect/train/results.png')
im.show()
im = Image.open('/content/runs/detect/train/val_batch0_pred.jpg')
im.show()
!yolo task=detect mode=val model=/content/runs/detect/train/weights/best.pt data=/content/datasets/3-3-23-1-0-1/data.yaml
!yolo task=detect mode=predict model=/content/runs/detect/train/weights/best.pt conf=0.25 source=/content/datasets/3-3-23-1-0-1/test/images save=True
import glob
from IPython.display import Image, display
for image_path in glob.glob(f'/content/runs/detect/predict/*.jpg')[:3]:
    display(Image(filename=image_path, width=600))
    print("\n")
!yolo task=detect mode=predict model=/content/runs/detect/train/weights/best.pt conf=0.25 source=/content/runs/3rdMarch2023.mp4 save=True
```

16_1_23_cv.py

```
!nvidia-smi
!pip install ultralytics==8.0.20
from IPython import display
display.clear_output()
import ultralytics
ultralytics.checks()
from ultralytics import YOLO
from PIL import Image
!pip install roboflow
from roboflow import Roboflow
rf = Roboflow(api_key="o7rqZ6ii3o9wMfN5lhKn")
project = rf.workspace("programmers").project("16-1-23-3-1")
dataset = project.version(1).download("yolov5")
!yolo task=detect mode=train model=yolov8s.pt data=/content/datasets/3-3-23-1-0-1/data.yaml epochs=25 imgsz=800 plots=True
!ls /content/runs/detect/train
im = Image.open('/content/runs/detect/train/confusion_matrix.png')
im.show()
im = Image.open('/content/runs/detect/train/results.png')
im.show()
im = Image.open('/content/runs/detect/train/val_batch0_pred.jpg')
im.show()
!yolo task=detect mode=val model=/content/runs/detect/train/weights/best.pt data=/content/datasets/3-3-23-1-0-1/data.yaml
!yolo task=detect mode=predict model=/content/runs/detect/train/weights/best.pt conf=0.25 source=/content/datasets/3-3-23-1-0-1/test/images save=True
import glob
from IPython.display import Image, display
for image_path in glob.glob(f'/content/runs/detect/predict/*.jpg')[:3]:
    display(Image(filename=image_path, width=600))
    print("\n")
!yolo task=detect mode=predict model=/content/runs/detect/train/weights/best.pt conf=0.25 source=/content/runs/3rdMarch2023.mp4 save=True
```

a)16-10-22 1_ShotMap.py

```
#ShotMap of 16th October,2022
#Make a shot map

#Function to draw the pitch
import matplotlib.pyplot as plt
import numpy as np

#Size of the pitch
pitchLengthX=120
pitchWidthY=80

match_id_required = 16102213
home_team_required ="Barcelona"
away_team_required ="Real Madrid"

# Load in the data
file_name=str(match_id_required)+'.json'

#Load in all match events
import json
with open('dataset/data/events/'+file_name) as data_file:
    #print (mypath+'events/'+file)
    data = json.load(data_file)

#get the nested structure into a dataframe
#store the dataframe in a dictionary with the match id as key (remove '.json' from string)
from pandas.io.json import json_normalize
df = json_normalize(data, sep = "_").assign(match_id = file_name[:-5])

#A dataframe of shots
shots = df.loc[df['type_name'] == 'Shot'].set_index('id')

#Draw the pitch
from mplsoccer.pitch import Pitch
pitch=Pitch(pitch_color='grass', line_color='white', stripe=True)
fig,ax=pitch.draw()

#Plot the shots
for i,shot in shots.iterrows():
```

```
    x=shot['location'][0]
    y=shot['location'][1]

    goal=shot['shot_outcome_name']=='Goal'
    team_name=shot['team_name']

    circleSize=2

    if (team_name==home_team_required):
        if goal:
            shotCircle=plt.Circle((x,pitchWidthY-y),circleSize,color="#be0032")
        else:
            shotCircle=plt.Circle((x,pitchWidthY-y),circleSize,color="#be0032")
            shotCircle.set_alpha(.4)
    elif (team_name==away_team_required):
        if goal:
            shotCircle=plt.Circle((pitchLengthX-x,y),circleSize,color="#6495ed")
        else:
            shotCircle=plt.Circle((pitchLengthX-x,y),circleSize,color="#6495ed")
            shotCircle.set_alpha(.6)
    ax.add_patch(shotCircle)


plt.text(5,75,away_team_required + ' shots')
plt.text(80,75,home_team_required + ' shots')

fig.set_size_inches(10, 7)
fig.savefig('a)Output/1-ShotMap.png', dpi=100)
plt.show()
```

a)16-10-22 2_PassMap_Barca.py

```
#PassMap_Barca of 16th October,2022

#Function to draw the pitch
import matplotlib.pyplot as plt
import numpy as np

#Size of the pitch
pitchLengthX=120
pitchWidthY=80
```

```
match_id_required = 16102213
home_team_required ="Barcelona"
away_team_required ="Real Madrid"

# Load in the data
file_name=str(match_id_required)+'.json'

#Load in all match events
import json
with open('dataset/data/events/'+file_name) as data_file:
    #print (mypath+'events/'+file)
    data = json.load(data_file)

#get the nested structure into a dataframe
#store the dataframe in a dictionary with the match id as key (remove '.json' from string)
from pandas import json_normalize
df = json_normalize(data, sep = "_").assign(match_id = file_name[:-5])

#Find the passes
passes = df.loc[df['type_name'] == 'Pass'].set_index('id')

#Draw the pitch
from mplsoccer.pitch import Pitch
pitch=Pitch(pitch_color='grass', line_color='white', stripe=True)
fig,ax=pitch.draw()

for i,thepass in passes.iterrows():
    if thepass['team_name']==home_team_required:
        x=thepass['location'][0]
        y=thepass['location'][1]
        passCircle=plt.Circle((x,pitchWidthY-y),2,color="#be0032")
        passCircle.set_alpha(.2)
        ax.add_patch(passCircle)
        dx=thepass['pass_end_location'][0]-x
        dy=thepass['pass_end_location'][1]-y

        passArrow=plt.Arrow(x,pitchWidthY-y,dx,-dy,width=0.6,color="#be0032")
        ax.add_patch(passArrow)

fig.set_size_inches(10, 7)
```

```
fig.savefig('a)Output/2-PassMap_Barca.png', dpi=100)
plt.show()
```

a)16-10-22 3_PassMap_Rmd.py

```
#PassMap_Rmd of 16th October,2022

#Function to draw the pitch
import matplotlib.pyplot as plt
import numpy as np

#Size of the pitch
pitchLengthX=120
pitchWidthY=80

match_id_required = 16102213
home_team_required ="Barcelona"
away_team_required ="Real Madrid"

# Load in the data
file_name=str(match_id_required)+'.json'

#Load in all match events
import json
with open('dataset/data/events/'+file_name) as data_file:
    #print (mypath+'events/'+file)
    data = json.load(data_file)

#get the nested structure into a dataframe
#store the dataframe in a dictionary with the match id as key (remove '.json' from string)
from pandas import json_normalize
df = json_normalize(data, sep = "_").assign(match_id = file_name[:-5])

#Find the passes
passes = df.loc[df['type_name'] == 'Pass'].set_index('id')

#Draw the pitch
from mplsoccer.pitch import Pitch
pitch=Pitch(pitch_color='grass', line_color='white', stripe=True)
fig,ax=pitch.draw()
```

```
for i,thepass in passes.iterrows():
    if thepass['team_name']==away_team_required:
        x=thepass['location'][0]
        y=thepass['location'][1]
        passCircle=plt.Circle((x,pitchWidthY-y),2,color="#6495ed")
        passCircle.set_alpha(.2)
        ax.add_patch(passCircle)
        dx=thepass['pass_end_location'][0]-x
        dy=thepass['pass_end_location'][1]-y

        passArrow=plt.Arrow(x,pitchWidthY-y,dx,-dy,width=0.6,color="#6495ed")
        ax.add_patch(passArrow)

fig.set_size_inches(10, 7)
fig.savefig('a)Output/3-PassMap_Rmd.png', dpi=100)
plt.show()
```

a)16-10-22 4_PassHeatMap_Barca.py

```
#Pass Heat Map of Barcelona

#Function to draw the pitch
import matplotlib.pyplot as plt
import numpy as np
import json
from pandas import json_normalize
from FCPython import createPitch

#Size of the pitch
pitchLengthX=120
pitchWidthY=80

team_required ="Barcelona"

#Get the list of matches
competition_id=11
#Load the list of matches for this competition
with open('dataset/data/matches/'+str(competition_id)+'/0.json', encoding="utf8") as f:
    matches = json.load(f)
```

```
#Find the matches they played
match_id_required=[]
for match in matches:
    home_team_name=match['home_team']['home_team_name']
    away_team_name=match['away_team']['away_team_name']
    if (home_team_name==team_required) or (away_team_name==team_required):
        match_id_required.append(match['match_id'])

#Find the passes for each match
for ic,match_id in enumerate(match_id_required):

    #Load in all match events

    file_name=str(match_id)+'.json'
    with open('dataset/data/events/'+file_name, encoding="utf8") as data_file:
        data = json.load(data_file)
    df = json_normalize(data, sep = "_").assign(match_id = file_name[:-5])
    team_actions = (df['team_name']==team_required)
    df = df[team_actions]

    #A dataframe of passes
    passes_match = df.loc[df['type_name'] == 'Pass'].set_index('id')
    #A dataframe of shots
    shots_match = df.loc[df['type_name'] == 'Shot'].set_index('id')

    #Find shot times in seconds
    #This should be adjusted to account for overlapping halves of the match.
    shot_times = shots_match['minute']*60+shots_match['second']
    shot_window = 15
    shot_start = shot_times - shot_window
    pass_times = passes_match['minute']*60+passes_match['second']

    #Check with passes are whitin [shot_window] seconds of a shot
    def in_range(pass_time,start,finish):
        return (True in ((start < pass_time) & (pass_time < finish)).unique())

    pass_to_shot = pass_times.apply(lambda x: in_range(x,shot_start,shot_times))
```

```
    #Exclude corners
    iscorner = passes_match['pass_type_name']=='Corner'

    danger_passes=passes_match[np.logical_and(pass_to_shot,np.logical_not(iscorner))]

    if ic==0:
        passes =  danger_passes
    else:
        passes = passes.append(danger_passes)



    print('Match: ' + str(match_id) + '. Number of danger passes is: ' + str(len(danger_passes)))


#Set number of matches
number_of_matches=ic+1

#Size of the pitch in yards (!!!)
pitchLengthX=120
pitchWidthY=80

#Draw the pitch
from mplsoccer.pitch import Pitch
pitch=Pitch(pitch_color='grass', line_color='white', stripe=True)
fig,ax=pitch.draw()

#Plot the passes
for i,thepass in passes.iterrows():
    x=thepass['location'][0]
    y=pitchWidthY-thepass['location'][1]
    passCircle=plt.Circle((x,y),1,color="#be0032")
    passCircle.set_alpha(.2)
    ax.add_patch(passCircle)

ax.set_title('Danger passes by ' + team_required)
fig.set_size_inches(10, 7)
fig.savefig('a)Output/4-PassHeatMap_Barca.png', dpi=100)
plt.show()
```

```
#Make x,y positions
x=[]
y=[]
for i,apass in passes.iterrows():
    x.append(apass['location'][0])
    y.append(pitchWidthY-apass['location'][1])

#Make a histogram of passes
H_Pass=np.histogram2d(y, x,bins=5,range=[[0, pitchWidthY],[0, pitchLengthX]])
```

a)16-10-22 5_PassHeatMap_Rmd.py

```
#Pass Heat Map of Real Madrid

#Function to draw the pitch
import matplotlib.pyplot as plt
import numpy as np
import json
from pandas import json_normalize
from FCPython import createPitch

#Size of the pitch
pitchLengthX=120
pitchWidthY=80

team_required ="Real Madrid"

#Get the list of matches
competition_id=11
#Load the list of matches for this competition
with open('dataset/data/matches/'+str(competition_id)+'/0.json', encoding="utf8") as f:
    matches = json.load(f)
```

```
#Find the matches they played
match_id_required=[]
for match in matches:
    home_team_name=match['home_team']['home_team_name']
    away_team_name=match['away_team']['away_team_name']
    if (home_team_name==team_required) or (away_team_name==team_required):
        match_id_required.append(match['match_id'])

#Find the passes for each match
for ic,match_id in enumerate(match_id_required):

    #Load in all match events

    file_name=str(match_id)+'.json'
    with open('dataset/data/events/'+file_name, encoding="utf8") as data_file:
        data = json.load(data_file)
    df = json_normalize(data, sep = "_").assign(match_id = file_name[:-5])
    team_actions = (df['team_name']==team_required)
    df = df[team_actions]

    #A dataframe of passes
    passes_match = df.loc[df['type_name'] == 'Pass'].set_index('id')
    #A dataframe of shots
    shots_match = df.loc[df['type_name'] == 'Shot'].set_index('id')

    #Find shot times in seconds
    #This should be adjusted to account for overlapping halves of the match.
    shot_times = shots_match['minute']*60+shots_match['second']
    shot_window = 15
    shot_start = shot_times - shot_window
    pass_times = passes_match['minute']*60+passes_match['second']
```

```
    #Check with passes are whitin [shot_window] seconds of a shot
    def in_range(pass_time,start,finish):
        return (True in ((start < pass_time) & (pass_time < finish)).unique())

    pass_to_shot = pass_times.apply(lambda x: in_range(x,shot_start,shot_times))

    #Exclude corners
    iscorner = passes_match['pass_type_name']=='Corner'

    danger_passes=passes_match[np.logical_and(pass_to_shot,np.logical_not(iscorner))]

    if ic==0:
        passes =  danger_passes
    else:
        passes = passes.append(danger_passes)



    print('Match: ' + str(match_id) + '. Number of danger passes is: ' + str(len(danger_passes)))


#Set number of matches
number_of_matches=ic+1

#Size of the pitch in yards (!!!)
pitchLengthX=120
pitchWidthY=80

#Draw the pitch
from mplsoccer.pitch import Pitch
pitch=Pitch(pitch_color='grass', line_color='white', stripe=True)
fig,ax=pitch.draw()
```

```
#Plot the passes
for i,thepass in passes.iterrows():
    x=thepass['location'][0]
    y=pitchWidthY-thepass['location'][1]
    passCircle=plt.Circle((x,y),1,color="#6495ed")
    passCircle.set_alpha(.5)
    ax.add_patch(passCircle)

ax.set_title('Danger passes by ' + team_required)
fig.set_size_inches(10, 7)
fig.savefig('a)Output/5-PassHeatMap_Rmd.png', dpi=100)
plt.show()

#Make x,y positions
x=[]
y=[]
for i,apass in passes.iterrows():
    x.append(apass['location'][0])
    y.append(pitchWidthY-apass['location'][1])

#Make a histogram of passes
H_Pass=np.histogram2d(y, x,bins=5,range=[[0, pitchWidthY],[0, pitchLengthX]])
```

a)16-10-22 6_HeatMap_Barca

```
from statsbombpy import sb
import pandas as pd
import matplotlib.pyplot as plt
from mplsoccer import Pitch

# Get competitions
comp = sb.competitions()
comp.to_csv('competitions.csv', index=False)
```

```
# Get Matches
df = sb.matches(competition_id=11, season_id=26)
df.to_csv('matches.csv', index=False)

# Find a match_id required
match = 3773585
df = sb.events(match_id=match)

df_pass = df.loc[(df['type'] == 'Pass') & (df['team'] == "Barcelona")]

location_xy = df_pass.location.apply(pd.Series)
# Save events locations to a Series with 2 columns location_xy = df_pass.location.apply(pd. Series)
# #Rename columns to x and y
location_xy.columns = ['x', 'y']
# Drop NaN
location_xy.dropna(inplace=True)
# Pitch setup
pitch = Pitch(pitch_type='statsbomb', line_zorder=2, pitch_color='#22312b', line_color='white')

# Draw the pitch
fig, ax = pitch.draw(figsize=(16, 9),  nrows=1,  ncols=4)

# 4 Heatmap variations specified by number of bins (x, y) for horizontal pitch
bins = [(1, 5), (6, 1), (6, 5), (3,2.5)]

for i, bin in enumerate(bins):
    bin_statistic = pitch.bin_statistic(location_xy.x, location_xy.y, statistic='count', bins=bin)
    pitch.heatmap(bin_statistic, ax=ax[i], cmap='RdYlBu', edgecolors='black')
    pitch.scatter(location_xy.x, location_xy.y, c='white', s=10, ax=ax[i])
    bin_statistic['statistic'] = (pd.DataFrame((bin_statistic['statistic'] /
bin_statistic['statistic'].sum())).applymap(lambda x: '{:.0%}'.format(x)).values)
    pitch.label_heatmap(bin_statistic, color='black', fontsize=10, ax=ax[i], ha='center', va='bottom')
```

```
# Alternative method using mpl soccer documentation
pitch = Pitch(line_zorder=2, pitch_color='#22312b',orientation='vertical', axis= True)
fig, ax = pitch.draw()
x = location_xy.x
y = location_xy.y
stats = pitch.bin_statistic_positional(x, y)
pitch.heatmap_positional(stats, edgecolors='black', cmap='RdYlBu', ax=ax)
pitch.scatter(x, y, c='white', s=10, ax=ax)
text = pitch.label_heatmap(stats, color='black', ax=ax, fontsize=10, ha='center')

plt.show()
plt.set_size_inches(10, 7)
plt.savefig('a)Output/6-HeatMap_Barca.png', dpi=100)
```

a)16-10-22 7_HeatMap_Rmd.py

```
from statsbombpy import sb
import pandas as pd
import matplotlib.pyplot as plt
from mplsoccer import Pitch

# Get competitions
comp = sb.competitions()
comp.to_csv('competitions.csv', index=False)

# Get Matches - UEFA Women's Euro
df = sb.matches(competition_id=11, season_id=26)
df.to_csv('matches.csv', index=False)

# Find a match_id required
match = 3773585
df = sb.events(match_id=match)
```

```
df_pass = df.loc[(df['type'] == 'Pass') & (df['team'] == "Real Madrid")]

location_xy = df_pass.location.apply(pd.Series)
# Save events locations to a Series with 2 columns location_xy = df_pass.location.apply(pd. Series)
# #Rename columns to x and y
location_xy.columns = ['x', 'y']
# Drop NaN
location_xy.dropna(inplace=True)
# Pitch setup
pitch = Pitch(pitch_type='statsbomb', line_zorder=2, pitch_color='#22312b', line_color='white')

# Draw the pitch
fig, ax = pitch.draw(figsize=(16, 9),  nrows=1,  ncols=4)

# 4 Heatmap variations specified by number of bins (x, y) for horizontal pitch
bins = [(1, 5), (6, 1), (6, 5), (3,2.5)]

for i, bin in enumerate(bins):
    bin_statistic = pitch.bin_statistic(location_xy.x, location_xy.y, statistic='count', bins=bin)
    pitch.heatmap(bin_statistic, ax=ax[i], cmap='RdYlBu', edgecolors='black')
    pitch.scatter(location_xy.x, location_xy.y, c='white', s=10, ax=ax[i])
    bin_statistic['statistic'] = (pd.DataFrame((bin_statistic['statistic'] /
bin_statistic['statistic'].sum())).applymap(lambda x: '{:.0%}'.format(x)).values)
    pitch.label_heatmap(bin_statistic, color='black', fontsize=10, ax=ax[i], ha='center', va='bottom')
# Alternative method using mpl soccer documentation
pitch = Pitch(line_zorder=2, pitch_color='#22312b',orientation='vertical', axis= True)
fig, ax = pitch.draw()
x = location_xy.x
y = location_xy.y
stats = pitch.bin_statistic_positional(x, y)
pitch.heatmap_positional(stats, edgecolors='black', cmap='RdYlBu', ax=ax)
pitch.scatter(x, y, c='white', s=10, ax=ax)
text = pitch.label_heatmap(stats, color='black', ax=ax, fontsize=10, ha='center')
```

```
lt.show()
plt.set_size_inches(10, 7)
plt.savefig('a)Output/6-HeatMap_Barca.png', dpi=100)
```

a)16-10-22 8_PossessionChain.py

```
import json
from pandas import json_normalize
import random
from plotly import graph_objects, offline


class Field ():

    def __init__(self):
        figure = graph_objects.Figure()

        figure.update_layout(width=900*1.388, height=900, autosize=False, plot_bgcolor="white")
        figure.update_xaxes(range=[-0.03, 1.03])
        figure.update_yaxes(range=[-0.03, 1.03])

        self.figure = figure

        self.__draw_full()

    def add_title(self, main_title: str):
        """
        Adds a title to the pitch visualization
        """
        self.figure.update_layout(title={'text': main_title, 'x': 0.48, 'y': 0.91,
                                         'xanchor': 'center', 'yanchor': 'top'})

    def save(self, name: str):
        """Saves PNG and html image"""
        offline.plot(self.figure, filename=f"{name}", auto_open=False)
```

```
    def __draw_full(self):
        # Contour
        self.figure.add_shape(type="rect", x0=0, y0=0, x1=1, y1=1)
        # Half way line
        self.figure.add_shape(type="line", x0=0.5, y0=0, x1=0.5, y1=1)
        circle_radius = 0.0915
        self.figure.add_shape(type="circle", x0=0.5-circle_radius*0.72, y0=0.5 -
                              circle_radius, x1=0.5+circle_radius*0.72, y1=0.5+circle_radius)
        # Defensive goal area
        self.figure.add_shape(type="rect", x0=0, y0=0.211, x1=0.170, y1=0.789)
        self.figure.add_shape(type="rect", x0=0, y0=0.368, x1=0.058, y1=0.632)
        self.figure.add_shape(type="rect", x0=-0.01, y0=0.447, x1=0.0, y1=0.553)
        # Attacking goal area
        self.figure.add_shape(type="rect", x0=0.83, y0=0.211, x1=1.0, y1=0.789)
        self.figure.add_shape(type="rect", x0=0.942, y0=0.368, x1=1.0, y1=0.632)
        self.figure.add_shape(type="rect", x0=1.0, y0=0.447, x1=1.01, y1=0.553)


def add_possession_chain_sb(figure, chain, team_name: str, opacity: int = 1.0):
    this_team_color = "#1F77B4"
    other_team_color = "#FF7F0E"

    chain_x = []
    chain_y = []
    text = []
    colors = []
    outcomes = []
    times = []
    counter = 0
    try:
        for idx, row in chain.iterrows():
            event_name = row['type_name']
            event_loc = row['location']
```

```
        # Transform  coordinates
        x = round((event_loc[0] * (100 / 120)) / 100, 3)
        y = round(((80 - event_loc[1]) * (100 / 80)) / 100, 3)

        color = this_team_color

        # slightly shift events that has the same coodinates
        if x in chain_x and y in chain_y:
            x = x + random.choice([-0.001, 0.001])
            y = y + random.choice([-0.001, 0.001])

        if row['possession_team_id'] != row['team_id']:
            color = other_team_color
            x = round(1.000 - x, 3)
            y = round(1.000 - y, 3)

        chain_x.append(x)
        chain_y.append(y)
        colors.append(color)
        times.append(f"{row['minute']}{row['second']}")
        outcomes.append('circle')
        counter += 10

except Exception as err:
    print(err)

line_color = "#7F7F7F"

figure.add_trace(
    {'mode': 'markers+lines',
     'textposition': 'top center',
     'type': 'scatter',
     'x': chain_x,
```

```
        'y': chain_y,
        'opacity': opacity,
        'marker': {'color': colors, 'size': 8, 'symbol': outcomes},
        'line': {'color': line_color, 'dash': 'solid'},
        'name': "{} {}:{} ".format(chain.iloc[0]['minute'], chain.iloc[0]['second'],team_name)
        }
    )

# MAIN METHOD

match_id_required = 16102213
home_team_required ="Real Madrid"
away_team_required ="Barcelona"

# Load in the data
file_name=str(match_id_required)+'.json'

#Load in all match events
import json
with open('dataset/data/events/'+file_name) as data_file:
    data = json.load(data_file)
match_events = json_normalize(data, sep="_") # Tranform data to DataFrmae
relevand_possessions = match_events[(match_events['type_id'] == 10)]['possession'].unique()  # all
uniques possessions with shot
 # Create Pitch
ground = Field()
ground.add_title('Possession chains')
for possession in relevand_possessions:
    df = match_events[match_events['possession'] == possession] # get all events of the possession
    # add each event
    add_possession_chain_sb(ground.figure, df, team_name= df.iloc[0]['team_name'])
#Look in the file below for the visulasiation of chains.
ground.save('a)Output/8-possessions.html')
```

e)xG_to_LoadData

```
#The basics
import pandas as pd
import numpy as np
import json

#Plotting
import matplotlib.pyplot as plt
import FCPython

#Statistical fitting of models
import statsmodels.api as sm
import statsmodels.formula.api as smf

#Decide which league to load
with open('Wyscout/events/events_England.json') as f:
    data = json.load(f)

#Create a data set of shots.
train = pd.DataFrame(data)
pd.unique(train['subEventName'])
shots=train[train['subEventName']=='Shot']
shots_model=pd.DataFrame(columns=['Goal','X','Y'])

#Go through the dataframe and calculate X, Y co-ordinates.
#Distance from a line in the centre
#Shot angle.
for i,shot in shots.iterrows():

    header=0
    for shottags in shot['tags']:
        if shottags['id']==403:
            header=1
```

```
    #Only include non-headers
    if not(header):
        shots_model.at[i,'X']=100-shot['positions'][0]['x']
        shots_model.at[i,'Y']=shot['positions'][0]['y']
        shots_model.at[i,'C']=abs(shot['positions'][0]['y']-50)
        #Distance in metres and shot angle in radians.
        x=shots_model.at[i,'X']*105/100
        y=shots_model.at[i,'C']*65/100
        shots_model.at[i,'Distance']=np.sqrt(x**2 + y**2)
        a = np.arctan(7.32 *x /(x**2 + y**2 - (7.32/2)**2))
        if a<0:
            a=np.pi+a
        shots_model.at[i,'Angle'] =a
        #Was it a goal
        shots_model.at[i,'Goal']=0
        for shottags in shot['tags']:
                #Tags contain that its a goal
                if shottags['id']==101:
                    shots_model.at[i,'Goal']=1
#Two dimensional histogram
H_Shot=np.histogram2d(shots_model['X'], shots_model['Y'],bins=100,range=[[0, 100],[0, 100]])
goals_only=shots_model[shots_model['Goal']==1]
H_Goal=np.histogram2d(goals_only['X'], goals_only['Y'],bins=100,range=[[0, 100],[0, 100]])
#Plot the number of shots from different points
(fig,ax) = FCPython.createGoalMouth()
pos=ax.imshow(H_Shot[0], extent=[-1,66,104,-1], aspect='auto',cmap=plt.cm.Reds)
fig.colorbar(pos, ax=ax)
ax.set_title('Number of shots')
plt.xlim((-1,66))
plt.ylim((-3,35))
plt.tight_layout()
plt.gca().set_aspect('equal', adjustable='box')
plt.show()
fig.savefig('e)Output/1-NumberOfShots.png', dpi=None, bbox_inches="tight")
```

```
#Plot the number of GOALS from different points
(fig,ax) = FCPython.createGoalMouth()
pos=ax.imshow(H_Goal[0], extent=[-1,66,104,-1], aspect='auto',cmap=plt.cm.Reds)
fig.colorbar(pos, ax=ax)
ax.set_title('Number of goals')
plt.xlim((-1,66))
plt.ylim((-3,35))
plt.tight_layout()
plt.gca().set_aspect('equal', adjustable='box')
plt.show()
fig.savefig('e)Output/2-NumberOfGoals.png', dpi=None, bbox_inches="tight")

#Plot the probability of scoring from different points
#Goals/Shots=Frequency of Scoring
(fig,ax) = FCPython.createGoalMouth()
pos=ax.imshow(H_Goal[0]/H_Shot[0], extent=[-1,66,104,-1],
aspect='auto',cmap=plt.cm.Reds,vmin=0, vmax=0.5)
fig.colorbar(pos, ax=ax)
ax.set_title('Proportion of shots resulting in a goal')
plt.xlim((-1,66))
plt.ylim((-3,35))
plt.tight_layout()
plt.gca().set_aspect('equal', adjustable='box')
plt.show()
fig.savefig('e)Output/3-ProbabilityOfScoring.png', dpi=None, bbox_inches="tight")
```

e)xG_to_FitModel

```
import pandas as pd
import numpy as np
import json
#Plotting
import matplotlib.pyplot as plt
import FCPython
```

```
#Statistical fitting of models
import statsmodels.api as sm
import statsmodels.formula.api as smf


#Plot a logistic curve
b=[3, -3]
x=np.arange(5,step=0.1)
y=1/(1+np.exp(-b[0]-b[1]*x))
fig,ax=plt.subplots(num=1)
plt.ylim((-0.05,1.05))
plt.xlim((0,5))
ax.set_ylabel('y')
ax.set_xlabel("x")
ax.plot(x, y, linestyle='solid', color='black')
ax.spines['top'].set_visible(False)
ax.spines['right'].set_visible(False)
plt.show()

shots_200=shots_model.iloc[:25]

#Plot first 25 shots goal angle
fig,ax=plt.subplots(num=1)
ax.plot(shots_200['Angle']*180/np.pi, shots_200['Goal'], linestyle='none', marker= '.', markerSize= 12,
color='black')
ax.set_ylabel('Goal scored')
ax.set_xlabel("Shot angle (degrees)")
plt.ylim((-0.05,1.05))
ax.set_yticks([0,1])
ax.set_yticklabels(['No','Yes'])
fig.savefig('e)Output/4-ShotGoalAngle.png', dpi=None, bbox_inches="tight")
plt.show()
```

```
#Now lets look at the likelihood of model given data
xG=1/(1+np.exp(b[0]+b[1]*shots_model['Angle']))
shots_model = shots_model.assign(xG=xG)
shots_40=shots_model.iloc[:40]
fig,ax=plt.subplots(num=1)
ax.plot(shots_40['Angle']*180/np.pi, shots_40['Goal'], linestyle='none', marker= '.', markerSize= 12,
color='black')
ax.plot(x, y, linestyle='solid', color='black')
ax.plot(x, 1-y, linestyle='solid', color='black')
loglikelihood=0
for item,shot in shots_40.iterrows():
    ang=shot['Angle']*180/np.pi
    if shot['Goal']==1:
        loglikelihood=loglikelihood+np.log(shot['xG'])
        ax.plot([ang,ang],[shot['Goal'],shot['xG']], color='red')
    else:
        loglikelihood=loglikelihood+np.log(1 - shot['xG'])
        ax.plot([ang,ang],[shot['Goal'],1-shot['xG']], color='blue')

ax.set_ylabel('Goal scored')
ax.set_xlabel("Shot angle (degrees)")
plt.ylim((-0.05,1.05))
plt.xlim((0,80))
plt.text(45,0.2,'Log-likelihood:')
plt.text(45,0.1,str(loglikelihood))
ax.set_yticks([0,1])
ax.spines['top'].set_visible(False)
ax.spines['right'].set_visible(False)
fig.savefig('e)Output/5-GoalScoredShotAngle.png', dpi=None, bbox_inches="tight")
plt.show()
```

```
#Make single variable model of angle
#Using logistic regression we find the optimal values of b
#This process minimizes the loglikelihood
test_model = smf.glm(formula="Goal ~ Angle" , data=shots_model,
                     family=sm.families.Binomial()).fit()
print(test_model.summary())
b=test_model.params


xGprob=1/(1+np.exp(b[0]+b[1]*midangle*np.pi/180))
fig,ax=plt.subplots(num=1)
ax.plot(midangle, prob_goal, linestyle='none', marker= '.', markerSize= 12, color='black')
ax.plot(midangle, xGprob, linestyle='solid', color='black')
ax.set_ylabel('Probability chance scored')
ax.set_xlabel("Shot angle (degrees)")
ax.spines['top'].set_visible(False)
ax.spines['right'].set_visible(False)
plt.show()
fig.savefig('e)Output/6-ProbabilityOfScoringAngleFit.png', dpi=None, bbox_inches="tight")

#Now lets look at distance from goal
#Show empirically how distance from goal predicts probability of scoring
shotcount_dist=np.histogram(shots_model['Distance'],bins=40,range=[0, 70])
goalcount_dist=np.histogram(goals_only['Distance'],bins=40,range=[0, 70])
prob_goal=np.divide(goalcount_dist[0],shotcount_dist[0])
distance=shotcount_dist[1]
middistance= (distance[:-1] + distance[1:])/2
fig,ax=plt.subplots(num=1)
ax.plot(middistance, prob_goal, linestyle='none', marker= '.', color='black')
ax.set_ylabel('Probability chance scored')
ax.set_xlabel("Distance from goal (metres)")
ax.spines['top'].set_visible(False)
ax.spines['right'].set_visible(False)
```

```
#Make single variable model of distance
test_model = smf.glm(formula="Goal ~ Distance" , data=shots_model,
                     family=sm.families.Binomial()).fit()
print(test_model.summary())
b=test_model.params
xGprob=1/(1+np.exp(b[0]+b[1]*middistance))
ax.plot(middistance, xGprob, linestyle='solid', color='black')
plt.show()
fig.savefig('e)Output/7-ProbabilityOfScoringDistance.png', dpi=None, bbox_inches="tight")

#Adding distance squared
squaredD = shots_model['Distance']**2
shots_model = shots_model.assign(D2=squaredD)
test_model = smf.glm(formula="Goal ~ Distance + D2" , data=shots_model,
                     family=sm.families.Binomial()).fit()
print(test_model.summary())
b=test_model.params
xGprob=1/(1+np.exp(b[0]+b[1]*middistance+b[2]*pow(middistance,2)))
fig,ax=plt.subplots(num=1)
ax.plot(middistance, prob_goal, linestyle='none', marker= '.', color='black')
ax.set_ylabel('Probability chance scored')
ax.set_xlabel("Distance from goal (metres)")
ax.spines['top'].set_visible(False)
ax.spines['right'].set_visible(False)
ax.plot(middistance, xGprob, linestyle='solid', color='black')
plt.show()
fig.savefig('e)Output/8-ProbabilityOfScoringDistanceSquared.png', dpi=None, bbox_inches="tight")
#Adding even more variables to the model.
squaredX = shots_model['X']**2
shots_model = shots_model.assign(X2=squaredX)
squaredC = shots_model['C']**2
shots_model = shots_model.assign(C2=squaredC)
AX = shots_model['Angle']*shots_model['X']
shots_model = shots_model.assign(AX=AX)
```

```
# A general model for fitting goal probability
# List the model variables you want here
model_variables = ['Angle','Distance','C']
model=''
for v in model_variables[:-1]:
    model = model  + v + ' + '
model = model + model_variables[-1]


#Fit the model
test_model = smf.glm(formula="Goal ~ " + model, data=shots_model,
                     family=sm.families.Binomial()).fit()
print(test_model.summary())
b=test_model.params


#Return xG value for more general model
def calculate_xG(sh):
   bsum=b[0]
   for i,v in enumerate(model_variables):
      bsum=bsum+b[i+1]*sh[v]
   xG = 1/(1+np.exp(bsum))
   return xG

#Add an xG to my dataframe
xG=shots_model.apply(calculate_xG, axis=1)
shots_model = shots_model.assign(xG=xG)
#Create a 2D map of xG
pgoal_2d=np.zeros((65,65))
for x in range(65):
    for y in range(65):
        sh=dict()
        a = np.arctan(7.32 *x /(x**2 + abs(y-65/2)**2 - (7.32/2)**2))
```

```
        if a<0:
            a = np.pi + a
        sh['Angle'] = a
        sh['Distance'] = np.sqrt(x**2 + abs(y-65/2)**2)
        sh['D2'] = x**2 + abs(y-65/2)**2
        sh['X'] = x
        sh['AX'] = x*a
        sh['X2'] = x**2
        sh['C'] = abs(y-65/2)
        sh['C2'] = (y-65/2)**2

        pgoal_2d[x,y] =  calculate_xG(sh)

(fig,ax) = FCPython.createGoalMouth()
pos=ax.imshow(pgoal_2d, extent=[-1,65,65,-1], aspect='auto',cmap=plt.cm.Reds,vmin=0, vmax=0.3)
fig.colorbar(pos, ax=ax)
ax.set_title('Probability of goal')
plt.xlim((0,66))
plt.ylim((-3,35))
plt.gca().set_aspect('equal', adjustable='box')
fig.savefig('e)Output/9-ProbabilityOfGoal_xG.png', dpi=None, bbox_inches="tight")
plt.show()

#Mcfaddens Rsquared for Logistic regression
null_model = smf.glm(formula="Goal ~ 1 ", data=shots_model,
                     family=sm.families.Binomial()).fit()
1-test_model.llf/null_model.llf

#ROC curve
numobs=100
TP=np.zeros(numobs)
FP=np.zeros(numobs)
TN=np.zeros(numobs)
FN=np.zeros(numobs)
```

```
for i,threshold in enumerate(np.arange(0,1,1/numobs)):
    for j,shot in shots_model.iterrows():
        if (shot['Goal']==1):
            if(shot['xG']>threshold):
                TP[i] = TP[i] + 1
            else:
                FN[i] = FN[i] + 1
        if (shot['Goal']==0):
            if(shot['xG']>threshold):
                FP[i] = FP[i] + 1
            else:
                TN[i] = TN[i] + 1


fig,ax=plt.subplots(num=1)
ax.plot(FP/(FP+TN), TP/(TP+FN), color='black')
ax.plot([0,1], [0,1], linestyle='dotted', color='black')
ax.set_ylabel("TP+FP")
ax.set_xlabel("TN+FN")
plt.ylim((0.00,1.00))
plt.xlim((0.00,1.00))
ax.spines['top'].set_visible(False)
ax.spines['right'].set_visible(False)
fig.savefig('e)Output/10-ROC_' + model  + '.png', dpi=None, bbox_inches="tight"
```

B. Rubrics for Project

Focus Areas	CNo	Criterion [c]	Exemplary 4	Satisfactory 3	Developing 2	Unsatisfactory 1
Problem Formulation (PO1,PO2, PO6, PO7,PSO1)	I	**Identify/Define Problem** Ability to identify a suitable problem and define the project objectives.	Demonstrates a skillful ability to identify / articulate a problem and the objectives are well defined and prioritized.	Demonstrates ability to Identify / articulate a problem and All major objectives are identified.	Demonstrates some ability to identify / articulate a problem that is partially connected to the issues and most major objectives are identified but one or two minor ones are missing or priorities are not established.	Demonstrates minimal or no ability to identify / articulate a problem and many major objectives are not identified.
	II	**Collection of Background Information:** Ability to gather background Information (existing knowledge, research, and/or indications of the problem)	Collects sufficient relevant back ground information from appropriate sources, and is able to identify pertinant/critical information;	Collects sufficient relevant background information from appropriate sources;	Collects some relevant background information from appropriate Sources.	Minimal or no ability to collect relevant background information

II I	**Define scope of the problem** Ability to identify problem scope suitable to the degree considering the impact on society and environment	Demonstrates a skillful ability to define the scope of problem accurately mentioning the relevant fields of engineering precisely. Considers, explains and evaluates the impact of engineering interventions on society and environment.	Demonstrates ability to define problem scope mentioning the relevant fields of engineering broadly. Considers and explains the impact of engineering interventions on society and environment	Demonstrates some ability to define problem scope mentioning some of the relevant fields . Some consideration of the impact of engineering interventions on society and environment.	Demonstrates minimal or no ability to define problem scope and fails to mention relevant fields of engineering . Minimal or no consideration of the impact of engineering interventions on society and environment
IV	**Understanding the Design Process and Problem Solving:** Ability to explain the design process including the importance of needs, specifications, concept generation and to develop an approach to solve a problem.	Demonstrates a comprehensive ability to understand and explain a design process. Considers multiple approaches to solving a problem, and can articulate reason for choosing solution	Demonstrates an ability to understand and explain a design process. Considers multiple approaches to solving a problem, which is justified and considers consequences.	Demonstrates some ability to understand and explain a design process. Considers a few approaches to solving a problem; doesn't always consider consequences.	Demonstrates minimal or no ability to understand and explain a design process. Considers a single approach to solving a problem. Does not consider consequences.

Build (PO4,PO5, PSO1)	V	**Implementing Design Strategy and Evaluating Final Design:** Ability to execute a solution taking into consideration design requirements using appropriate tool (software/hardware);	Demonstrates a skillful ability to execute a solution taking into consideration all design requirements using the most relevant tool.	Demonstrates an ability to execute a solution taking into consideration design requirements using relevant tool.	Demonstrates some ability to execute a solution but not using most relevant tool.	Demonstrates minimal or no ability to execute a solution. Solution does not directly attend to the problem.
Test & Deploy (PO4, PO5, PSO1)	VI	**To evaluate/confirm the functioning of the final design. To deploy the project on the target environment**	Demonstrates a skillful ability to evaluate/confirm the functioning of the final design skillfully, with deliberation for further Improvement after deployment.	Demonstrates an ability to evaluate/confirm the functioning of the final design. The evaluation is complete and has sufficient depth.	Ability to evaluate/confirm the functioning of the final design, but the evaluation lacks depth and/or is incomplete.	Demonstrates minimal or no ability to evaluate/confirm the functioning of the final design.

Ethical responsibility (PO8)	VII	**Proper Use of Others' Work:** Ability to recognize, understand and apply proper ethical use of intellectual property, copyrighted materials, and research.	Always recognizes and applies proper ethical use of intellectual property, copyrighted materials, and others' research.	Recognizes and applies proper ethical use of intellectual property, copyrighted materials, and others' research.	Some recognition and application of proper ethical use of intellectual property, copyrighted materials, and others' research.	Minimal or no recognition and/or application of proper ethical use of intellectual property, Copyrighted materials, or others' research.
Team Skills (PO9)	VIII	**Individual Work Contributions and Time Management: Ability to carry out individual Responsibilities and manage time (estimate, prioritize, establish deadlines/ milestones, follow timeline, plan for contingencies, adapt to change).**	Designated jobs are accomplished by deadline; completed work is carefully and meticulously prepared and meets all requirements.	Designated jobs are accomplished by deadline; completed work meets requirements.	Designated jobs are accomplished by deadline; completed work meets most requirements.	Some Designated jobs are accomplished by deadline; completed work meets some requirements.

I X	**Leadership Skills**: Ability to lead a team. (i) Mentors and accepts mentoring from others. (ii) Demonstrates capacity for initiative while respecting others' roles. (iii) Facilitates others' involvment. (iv) Evaluates team Effectiveness and plans for improvements	Exemplifies leadership skills.	Demonstrates leadership skills.	Demonstrates some leadership skills at times.	Demonstrates minimal or no leadership skills.
X	**Working with Others:** Ability to listen to, collaborate with, and champion the efforts of others.	Skillfully listens to, collaborates with, and champions the efforts of others.	Listens to, collaborates with, and champions the efforts of others.	Sometimes listens to, collaborates with, and champions others' efforts.	Rarely listens to, collaborates with, or champions others' efforts.

Project Presentation (P10)	XI	**Technical Writing Skills** Ability to communicate the main idea with clarity. Ability to use illustrations properly to support ideas (citations, position on page etc)	Main idea is clearly and precisely stated. Materials are seamlessly arranged in a logical sequence Illustrations are skillfully used to support ideas	Main idea is understandable. Material moves logically forward, Illustrations are properly used to support ideas	Main idea is somewhat understandable. Materical has some logical order and is somewhat coherent or easy to follow. Illustrations are for the most part properly used to support ideas	Main idea is difficult to understand. Material has little logical order, and is often unclear, inconherent. Illustrations are used, but minimally support ideas. (not properly cited etc)

XII	**Communication Skills for Oral Reports** Ability to present strong key ideas and supporting details with clarity and concision. Maintain contact with audience, and ability to complete in the allotted time	Presentation logically and skillfully structured. Key ideas are compelling, and articulated with exceptional clarity and concision. Introduction, supporting details and summary are clearly evident and memorable, and ascertain the credibility of the speaker Presentation fits perfectly within time constraint.	Presentation has clear structure and is easy to follow. Key ideas are clearly and concisely articulated, and are interesting. There is sufficient detail to ascertain speaker's authority, and presentation includes an introduction and summary. Presentation fits within time constraint, though presenter might have to subtly rush or slow down.	Presentation has some structure. Key ideas generally identifiable, although not very remarkable. Introduction, supporting details and/or summary may be too broad, too detailed or missing. Credibility of the speaker may be questionable at times. Presentation does not quite fit within time constraint; presenter has to rush or slow down at end	Presentation rambles. Not organized; key ideas are difficult to identify, and are unremarkable. No clear introduction, supporting details and summary. Speaker has no credibility. Presentation is unsuitablably short or unreasonably long.

Project management (PO11,PSO2)	XIII	Use of software project management principles and tools (versioning, time schedules etc)	Employ all the appropriate tools or engineering techniques. Clearly demonstrates mastery of several areas of the curriculum.	Employ the appropriate tools or engineering techniques	Employ some management tools	Does not make use of any tool
Lifelong Learning (PO12,PSO2)	XIV	**Extend Scope of Work:** Ability to extend the project through implementation in other study areas	Demonstrates a skillful ability to explore a subject/topic thoroughly, discusses the road map to extend the project in other areas.	Demonstrates an ability to explore a subject/topic, and shows possible areas in which project can be extended	Demonstrates some ability to explore a subject/topic, providing some knowledge of areas in which project can be extended	Demonstrates minimal or no ability to explore a subject/topic, and does not discuss future work clearly mentioning other areas

Table 3 : Rubrics for Project

111. Rubrics Evaluation:

	PO/ PSO	PO1,PO2, PO6, PO7, PSO1			PO3, PSO1	PO4, PO5, PSO1	PO4, PO5, PSO1	PO8	PO9			PO10		PO11, PSO2	PO12 ,PSO2
Batch	**Rubrics**	R1			R2	R3	R4	R5	R6			R7		R8	R9
	Roll No	CI	CII	CIII	CIV	CV	CVI	CVII	CVIII	CIX	CX	CXI	CXIII	CXIII	CXIV

Table 4 : Rubrics Evaluation