**Predicting MLS Results**

Alexandre Moreira Batista

300344513

Armando Defendi Rossi

3003340211

CSIS 4495-001

Stephen Chiong

Douglas College

**Predicting MLS results**



**Executive Summary:**

This project aims to give sports fans accurate predictions for Major League Soccer (MLS) games using robust machine-learning models. By utilizing web scraping, database management, machine learning analysis, and an Android application, this project will deliver valuable predictions to users without needing them to pay exorbitant fees to sports betting companies. This project will cover all aspects of the Data Analytics program at Douglas College. The primary beneficiaries will be soccer fans with access to accurate game predictions at no cost.

**Table of Contents**

[1. Introduction 4](#_Toc130797460)

[A. Statement of the Problem 4](#_Toc130797461)

[B. Significance of the Study 4](#_Toc130797462)

[C. Technologies 4](#_Toc130797463)

[D. Project Implementation and timelines 4](#_Toc130797464)

[2. Actual Report – Progress Report 3 5](#_Toc130797465)

[A. Project Implementation 5](#_Toc130797466)

[i. Complete Steps 5](#_Toc130797467)

[a) Development and testing of database architecture to receive data. 5](#_Toc130797468)

[b) GitHub configuration for Project 7](#_Toc130797469)

[c) Development of Python scripts for web scraping 9](#_Toc130797470)

[d) Stage 01 - Development of the Android App 10](#_Toc130797471)

[e) Web scrapping for Machine Learning 12](#_Toc130797472)

[f) Plots and Descriptive Analysis 16](#_Toc130797473)

[g) Data Cleaning and Preparation 19](#_Toc130797474)

[h) Modeling data to features engineering 21](#_Toc130797475)

[i) Creating Machine Learning Models 24](#_Toc130797476)

[3. Software Design Architecture 26](#_Toc130797477)

[4. References 28](#_Toc130797478)

# Introduction

## Statement of the Problem

The sports betting industry has seen significant growth recently, with large companies and start-ups competing for customers and revenue. While many companies offer predictions for game results, access to this valuable information often comes at a cost. This project aims to provide a cost-effective solution for all sports fans to access accurate predictions for MLS games.

## Significance of the Study

This project aims to deliver predictions that improve the odds ratio when determining the winner of a game. By utilizing techniques such as web scraping, database management, machine learning, and analytics, this project will also identify features that impact the accuracy of predictions. Additionally, this project will deliver an Android application, providing users with easy access to past and future game predictions.

## Technologies

The project will make use of the following technologies:

* Programming Language: Python for web scraping and data prediction.
* Database: MySQL
* Front-end and Backend: Android Studio

## Project Implementation and timelines

* Development and testing of database architecture to receive data - 1 week.
* Development and testing of Python scripts for web scraping - 2 weeks.
* Research on modelling and machine learning uses in soccer results prediction - 2 weeks.
* Preparation and setup of a virtual machine in the cloud (Azure) to run the database and modelling - 1 week.
* Data wrangling in Python - 1 week.
* Data modelling (machine learning to predict and check prediction accuracy) - 2 weeks.
* Development of the Android App - 4 weeks.
* Testing the entire ecosystem and fixing any bugs - 1 week.

# Actual Report – Progress Report 3

## Project Implementation

### Complete Steps

#### Development and testing of database architecture to receive data.

After conducting experiments on Azure SQL (as illustrated in Figure 01), our team decided to utilize MySQL instead, as presented in Figure 02. This web-based solution is easily accessible via Python scripts (for web scraping) and our Android application.

During the development of our Android application and data collection from the MLS website, we observed frequent updates to the website, posing a significant challenge to our web scraping approach. Due to the use of JavaScript in constructing pages, changes occurred rapidly, necessitating a switch to a more stable data source. Thus, we opted to leverage ESPN's website, which offers comparable data. We made this transition when updating our database with the latest scores of recent matches. As a result, scores are now updated one day after the matches are played.

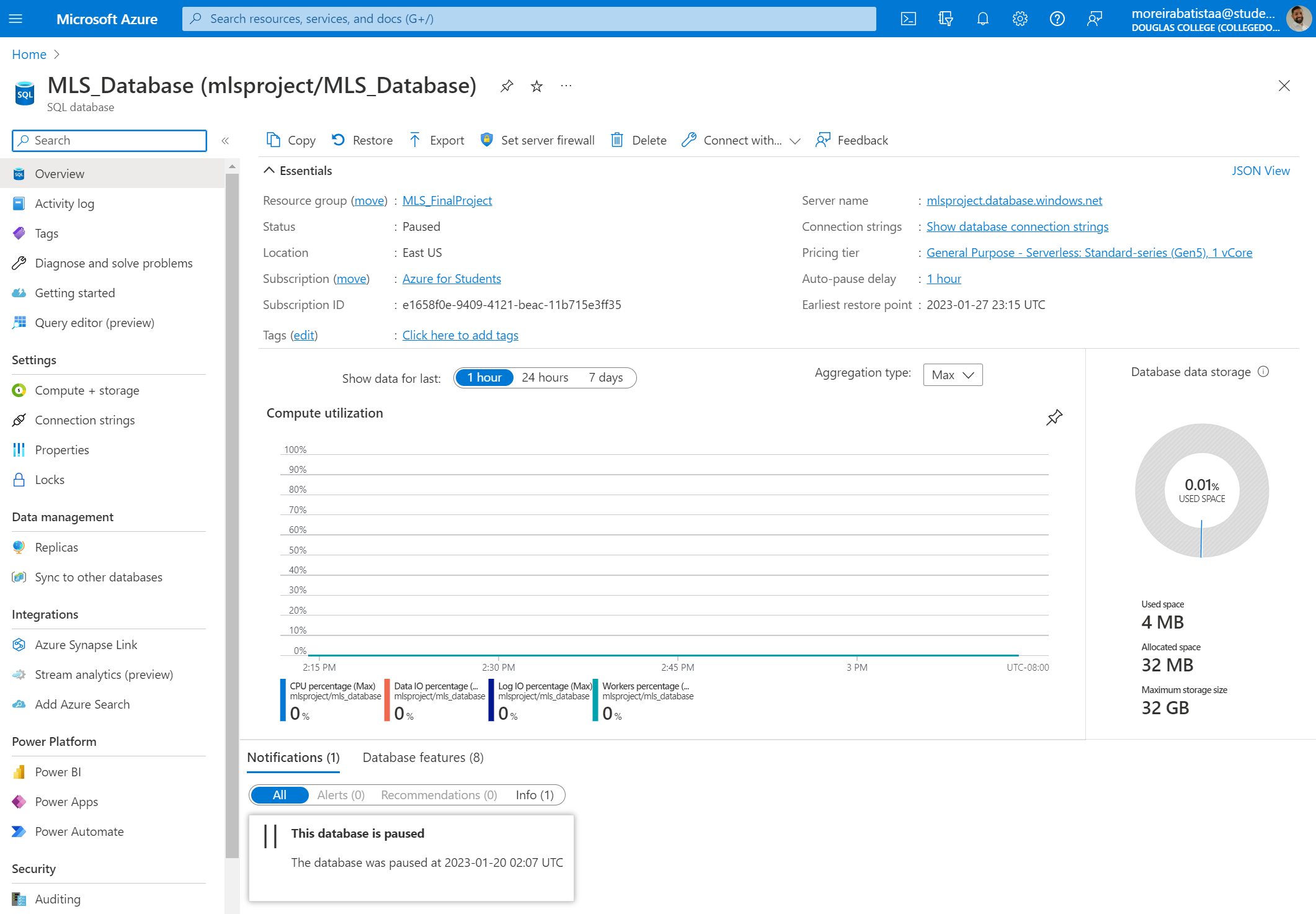


Figure 1 - Azure SQL Server

Graphical user interface

Description automatically generated with low confidence Graphical user interface

Description automatically generated with medium confidence

Figure 2 - MySQL structure

The first tables were created, and the connection tested.

Database configuration: Alexandre on 16-20/Jan/2023  
Database creation of tables/fields: Armando on 16-20/Jan/2023

#### GitHub configuration for Project

A repository was created to manage the versioning and save all the scripts and codes online as shown on figure 03.

GitHub creation: Armando on 16-20/Jan/2023

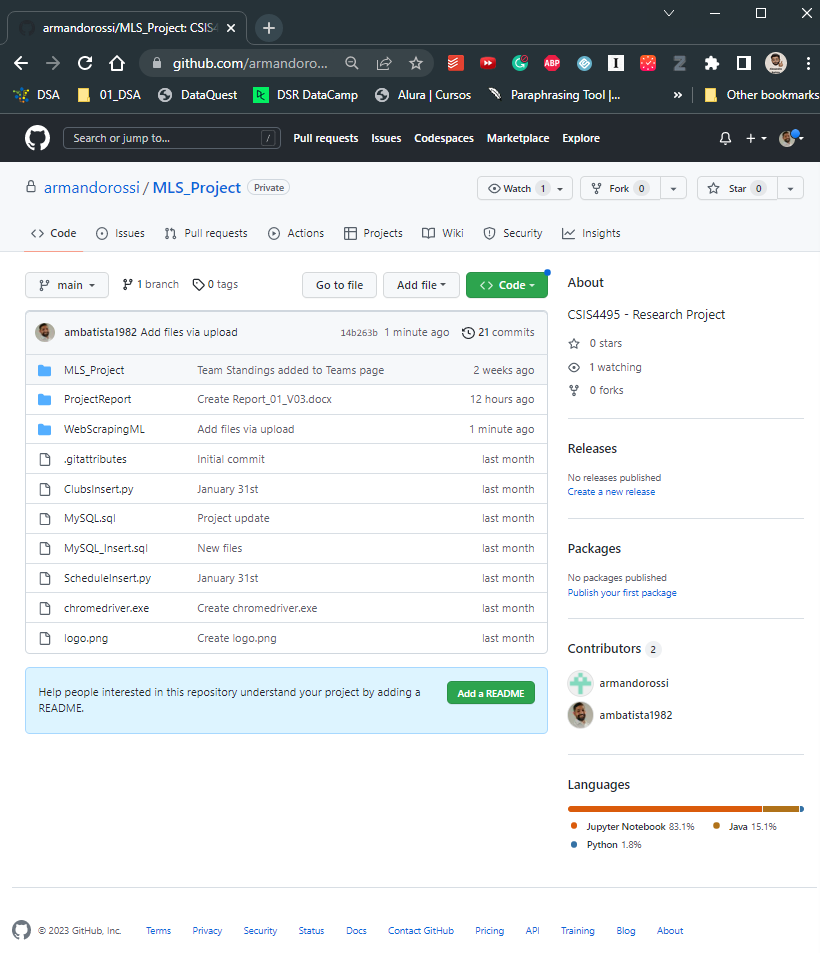


Figure 3 - GitHub directory

#### Development of Python scripts for web scraping

Using certain Python libraries, the scripts were created to extract schedule, team, and standings information from both the MLS Soccer and ESPN websites.

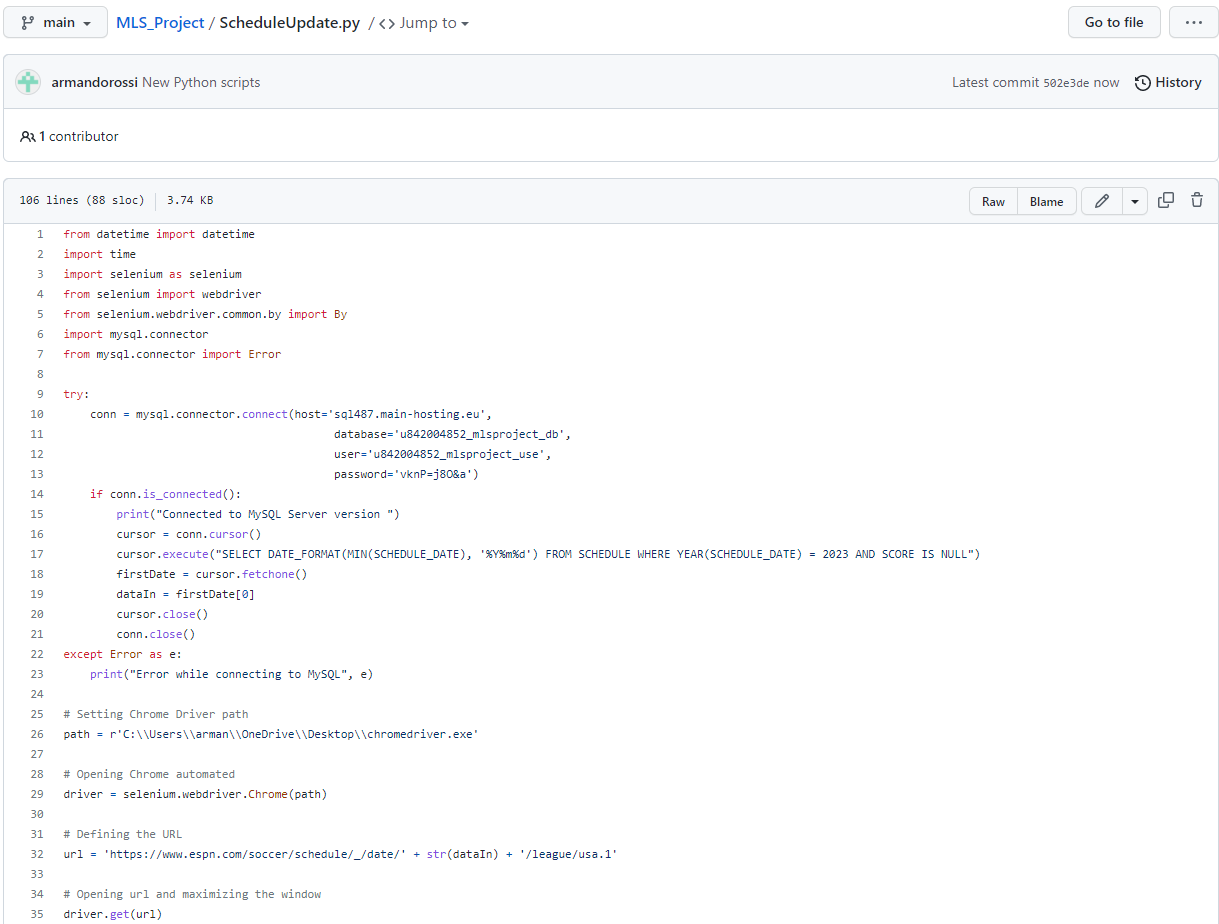


Figure 4 - Python script

A recently developed Python script has been updated to incorporate the latest changes. This updated version can now effectively insert, and update schedules based on the date provided, retrieving schedule information from the ESPN website.

Graphical user interface, application

Description automatically generated

Figure 5 - Schedule table details

Python scripts: Armando on 23-26/Jan/2023

#### Stage 01 - Development of the Android App

The first visuals were developed, including the Login, Register, Schedule, and User administration page layout.  
Graphical user interface, text, application, email

Description automatically generated Graphical user interface, text, application, email

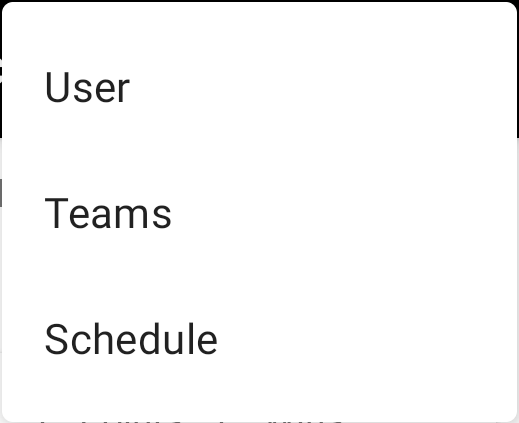
Description automatically generated  
*Figure 6 - Login page* *Figure 7 - Registration page*

Graphical user interface, application

Description automatically generated Graphical user interface, application

Description automatically generated  
*Figure 8 - User page* *Figure 9 - Schedule page*

To accommodate the upcoming modifications, the application has undergone a revamp. A fresh menu has been introduced to enable users to switch between the User (admin access only), Schedule, and the newly developed Team screen. The Team screen showcases a comprehensive list of all MLS teams and their standings. It also allows users to choose a specific year to review each team's standings.

 Graphical user interface, application

Description automatically generated

Figure 10 – Menu Figure 11 – Teams page

The Teams page now features an added functionality that enables users to view the upcoming scheduled matches for their teams by simply clicking on them.

*Graphical user interface, application

Description automatically generated  
Figure 12 – Team schedule page*

Graphical user interface, text, application

Description automatically generated  
*Figure 13 – GitHub repository*

App Development: Armando on 23/Jan to 13/Mar/2023

#### Web scrapping for Machine Learning

To enhance the machine learning model's prediction accuracy, it was concluded that supplementary data beyond game outcomes was required. Consequently, Python scripts were developed to extract data from the FB Ref website.

The struture of the Machine Learning Scripts was create using the following steps:

1. Import the additional Libs
2. Defining the standard Site Adresses
3. Creating Function to Extract the Data from the site
4. Creating SQL Functions to Query, Insert and Truncate tables
5. Function GetInsert (grouping toguether all Functions)
6. Testing Functions
7. Get and Inserting Data from web to Database (2015-2023)

*Graphical user interface, text, application

Description automatically generated*  
*Figure 14 – Python web scrapping for ML*

Text

Description automatically generated  
*Figure 15 – Python web scrapping for ML*

Graphical user interface, text, application

Description automatically generated

*Figure 16 – Python web scrapping for ML*

Web scraping Development: Alexandre on 23/Jan to 26/Feb/2023

#### Plots and Descriptive Analysis

Graphical user interface, text

Description automatically generated with medium confidence

*Figure 17 – Python web scrapping for ML*

A picture containing text

Description automatically generated

*Figure 18 – Python web scrapping for ML*

Graphical user interface, application, table

Description automatically generated

*Figure 19 – Python web scrapping for ML*

Chart, line chart

Description automatically generated

*Figure 20 – Python web scrapping for ML*

#### Data Cleaning and Preparation

Text

Description automatically generated

*Figure 21 – Python web scrapping for ML*

A picture containing calendar

Description automatically generated

*Figure 22 – Python web scrapping for ML*

Table

Description automatically generated

*Figure 23 – Python web scrapping for ML*

Text

Description automatically generated

*Figure 24 – Python web scrapping for ML*

Data analysis and cleaning: Alexandre on 26/Feb to 12/Mar/2023

#### Modeling data to features engineering

Text

Description automatically generated  
*Figure 25 – Python for ML*

Graphical user interface, application, table

Description automatically generated  
*Figure 26 – Python for ML*

A picture containing table

Description automatically generated  
*Figure 27 – Python for ML*

#### Creating Machine Learning Models

Text

Description automatically generated  
*Figure 28 – Python for ML*

Text

Description automatically generated  
*Figure 29 – Python for ML*

The dataset has been divided in 4 groups:

G01 => Training data: 2016-2018 and Testing Data: 2019

G02 => Training data: 2016-2019 and Testing Data: 2020

G03 => Training data: 2016-2020 and Testing Data: 2021

G04 => Training data: 2016-2021 and Testing Data: 2022

The scores shown in the picture above are the results of the machine learning classifier models on the Group 01 of the Split dataset. The goal of the classification task was to predict the classes of the samples, which belong to one of three classes: 'H', 'A', or 'D'. This is an example of a multi-class classification problem.

Multi-class classification can be more challenging than binary classification because there are more possible classes that the model needs to distinguish between. In this case, the main difficulty was finding the optimal hyperparameters for the machine learning classifier models. Hyperparameters are parameters that are set before training the model and affect the learning process and the model's ability to generalize to new data.

In addition to hyperparameter tuning, other techniques can be used to improve the performance of multi-class classifiers. For example, ensemble methods, such as bagging and boosting, can combine multiple classifiers and improve their performance. This option will be tested in the next steps of the project.

Next steps:

* 13. Measure accuracies, create Predictions and export to Mobile Applications.

# Software Design Architecture

**1. Front-end**

* Android application for users to access the sports predictions and other features.
* User interface design to allow easy navigation and interaction.

**2. Back end**

* Web scraping module to collect data on MLS games, including past results and upcoming fixtures.
* Machine learning module to analyze the data and generate accurate predictions for future games.
* Database management module to store and manage the data collected from web scraping and machine learning modules.
* Analytics module to identify the features that impact the accuracy of predictions.
* API layer to expose the prediction data and other features to the Android application.

**3. Integration and Deployment**

* Integration of the front-end and back-end modules to create a seamless user experience.
* Deployment of the application on cloud platforms or other hosting environments to ensure scalability and availability.

The software design architecture depicted above adopts a client-server style, in which the Android application serves as the client while the back-end modules function as the server. The system will gather and analyze data by leveraging web scraping, machine learning, and database management techniques, enabling it to generate precise predictions for MLS games.

Furthermore, the analytics module will pinpoint the factors that influence the precision of the predictions. The Android application will offer effortless access to past and future game predictions and other functionalities. The system will be deployed on cloud platforms or other hosting environments to ensure scalability and availability.

# References

Major Soccer League. Schedule & Scores: <https://www.mlssoccer.com/schedule>

FB Ref – Soccer References Databases: <https://fbref.com/en/comps/22/history/Major-League-Soccer-Seasons>

ESPN. Schedule & Scores: <https://www.espn.com/soccer/schedule/_/league/usa.1>