C964: Computer Science Capstone:

Predicting NWSL Player Goal Contribution Scores with Machine Learning

Leigh Grover

12/24/2024

Part A: Letter of Transmittal

12/15/2024

Sarah Jones Simmer, Chief Operating Officer

National Women's Soccer League

292 Madison Avenue, Floor 3

New York, NY 10017

Dear Sarah Jones Simmer,

As a fan of women's sports, particularly the National Women's Soccer League (NWSL), it has been exciting to see how the league has been growing in popularity. However, a known problem is that there is still more work to be done in terms of sponsorship, marketing and fan engagement in comparison to other leagues. For example, according to a ranking of professional sports leagues by revenue in Wikipedia (n.d.), the NWSL ranks 47, and is lower than other US leagues including the MLS, NHL, NBA and NFL.

Driving fan and stakeholder engagement with web application tools that are interactive, analytical and data driven is an exciting method that will likely generate more excitement about players and games, resulting in increased popularity (Rickevicius, 2024). Unfortunately, there is a dearth of such interactive tools that is specific to the NWSL. Currently available NWSL player statistics are static and may not resonate with fans or stakeholders. It's now an exciting time to introduce products that can bolster fan and stakeholder engagement, and this project proposal aims to do so.

The following document is a project proposal for developing an application that utilizes a machine learning method to create a predictive tool for NWSL player performance in contributing to goals. The tool will utilize a Random Forest algorithm that will be created using a historical NWSL player dataset pulled from the official NWSL website for the seasons of 2023 and 2024. The model will predict a goal contribution score based off weighted goals and assists and will be predicted by features from the dataset. Furthermore, the application will engage fans by being interactive by allowing users to input statistics from players to get their predictive goal contribution score. Written and visual descriptions will inform the user's understanding of the tool to help put the received output in perspective.

In summary, the goal of this project is to create an accessible tool for fans and stakeholders to predict player performance and generate engagement in the NWSL. The hypothesis is that machine learning, specifically the Random Forest algorithm, is an effective tool for predicting a player's Goal Contribution Score (GCS) based on player data features available on the NWSL website. The objectives of this project include:

- Develop a machine learning model capable of predicting the Goal Contribution Score (GCS) using player statistics available on the NWSL website.
- Create three or more descriptive visualizations of the data used to train and test the model, as well as information about the model and its predictive behavior.
- Deploy an interactive web application to allow users to learn from three or more visualizations about the model and input player data to receive a GCS score.

There are several ways in which this proposed application could benefit the NWSL's business goals and lead to an increase in revenue and popularity.

- Fan engagement: The application provides fans with an interactive tool that has a meaningful metric to predict goal contributions for players. Having this metric could lead to more excitement around players, wanting to attend upcoming games and stir up discussions or activities with other fans.
- Player decisions: Coaching staff or other analysts could utilize this application for decisions in player usage, trades and recruiting, leading to bringing more exciting talent to the NWSL or interesting strategic decision making.
- NWSL brand strengthening: Promoting a machine learning tool demonstrates that the organization is investing in data driven methods for understanding player performance and decision making. This may attract more tech interested fans or sponsorship.

The cost for the development of this application, which involves data procurement, development, deployment and maintenance, is estimated at $4,022.87. The project can be completed in approximately two months to provide enough time for data collection and preparation, model development, application building and deployment. There is minimal risk for data sensitivity concerns, as it is all publicly available on the NWSL website. Similarly for ethical concerns, there is low risk due to the public availability of the data and the use of a large set of historical data from all NWSL players who could potentially contribute directly to scoring.

The team who will be developing and deploying the proposed application has the relevant expertise for this project. The developers have knowledge in Python, data analysis, machine learning algorithms and both descriptive and predictive methods. Furthermore, they have an appreciation of women's soccer and genuine excitement towards contributing to tools that can facilitate fan and stakeholder engagement, data driven decisions in the sport and ultimately increased popularity and revenue.

A publicly available web application that is interactive and provides meaningful predictive data that is specific to the NWSL has the potential to greatly increase engagement and more broadly, drive revenue. Thank you for considering this proposal.

Sincerely,

Leigh Grover

Senior Developer

Part B: Project Proposal Plan

Project Summary

The National Women's Soccer League (NWSL) has been growing in popularity, sponsorships and revenue, however there is still more work to be done to reach parity with other national sports leagues (Wikipedia contributors, n.d., Wetzel, 2024). The NWSL currently lacks publicly available machine learning tools that have the potential to provide meaningful insights into player performance. Data analytics has been part of sports analysis for many years and providing such tools for public use can increase fan and stakeholder engagement (Rickevicius, 2024). While player statistics, such as goals and shots on target, are publicly available as static information, an interactive tool predicting a player's total contribution to scoring could be more meaningful and engaging to a fan or stakeholder. In fact, they are more likely to be drawn to players or teams when there are compelling stories drawing them in, and this proposed application could certainly foster this to generate excitement about players. Overall, the increased engagement through interaction with the proposed application could result in more excitement developed around players and their stories, new sponsorships, increased game attendance and facilitate bringing exciting talent to the NWSL teams.

Deliverables

1. **Web Application: Predicting Player Goal Contribution Scores with Machine Learning:**
    - This application will be a Python-based machine learning tool utilizing a model developed from the Random Forest algorithm to predict a player performance metric called the Goal Contribution Score (GCS). Data used to develop the model will be from NWSL 2023 and 2024 player data.
    - Features include:
        - An interactive tool that takes in user input in the form of player stats that are easily available on the NWSL website. Once this information is submitted, the user will receive a predictive GCS and text describing how to interpret the score.
        - Visualizations such as histograms, scatterplots, and pie charts that will explain the model. This includes descriptions of the GCS data, feature importance and other meaningful data descriptions that help explain the data and model.
2. **User Guide:**
    - A user guide detailing how to access the application and operate it.

Benefits to the Client:

The proposed application will enhance the NWSL's operational and strategic capabilities by providing an easy-to-use, data driven tool for predicting goal contribution scores. This will benefit the organization in terms of:

- **Fan Engagement:** Analytical and visual tools provide fans with new ways to engage with players, teams and data. This could result in increased popularity of players, teams and the league, better game attendance, as well as new sponsorship opportunities.
- **Support Decision-Making:** Coaches and analysts can use the application to identify predictive scoring contributions by player to optimize player usage, lineups and in informing recruiting and

trades. This could result in exciting choices in game play, and bringing in new, thrilling talent to further generate fan excitement.

- **Drive Innovation:** Having a tool that utilizes machine learning will position the NWSL as a leader in utilizing data driven, predictive tools. This could be appealing to tech-savvy fans and stakeholders, potentially driving further interest in the organization.

Overall, the long-term effect of this proposed project could result in increased revenue and sponsorship for the NWSL, aligning with their business goals. The following will describe the plan for developing the project and the resources anticipated to complete it.

Data Summary

The data for this project will be procured from the NWSL website in which player statistics are made available at https://www.nwslsoccer.com/stats/players/all. The data on this site can be trusted as it is taken from the official NWSL website, and it can be assumed the league hold high standards to reporting accurate player data. The player data was copied from the 2023 and 2024 full seasons. It is appropriate for this project, because it includes the variables that will go into creating the aggregate score contribution metric, namely goals and assists. Furthermore, there are many features that will be considered as predictor variables, including shots taken, shots on target, tackles, fouls, games played, position, and teams. Lastly, it is important to use data that is easily accessible by fans to make using this application accessible.

The data will be collected by copying the player statistics directly from the NWSL website for the completed seasons of 2024 and 2023. The data will be inputted into a CSV file. From there an index column with unique identifiers will be created for each row. Rows with all zeros will be filtered for and deleted as players with no data will not be helpful to training the model.

After the data procurement, the CSV spreadsheet will be loaded into the Jupyter Notebook environment to begin preprocessing. Python is the coding language that will be used and the library, Pandas, will be used for data processing. First, the data will be explored for any missing data, and if any are found, the developers will decide whether to replace the missing data with mean or median values. Exploratory data will be conducted to understand the distribution of the data, as well as any variables that may be closely correlated to one another. The developers will review the data to verify that all variables are in the correct format to use in fitting the model. Specific data to be transformed will be categorical, including “teams” and “position.” In this stage, the aggregate goal contribution score (GCS) will be created as the dependent variable to quantify player scoring performance. Outliers will be identified and removed using the interquartile range (IQR) method as to ensure better accuracy when fitting the model. This data preprocessing and design approach ensures the data is appropriate to be fitted into the Random Forest Algorithm model.

As part of the plan for data maintenance, the project will budget for monthly developer maintenance as well as annually integrating new NWSL player data of a completed season. The new data will be included in the existing dataset and undergo the same cleaning methods. This updated dataset will be used to re-fit the model, so the model’s predictive accuracy is built from the most recent data.

This project contains no ethical or legal concerns regarding the data. The data utilized for this project is publicly available and there is no sensitive information that could have privacy concerns.

Implementation

The project development will follow the CRISP-DM methodology. This methodology is the most appropriate because of its emphasis on understanding business needs, a focus on understanding the data, modeling and then deployment. CRISP-DM also allows for utilizing Agile to promote flexibility which could be helpful allowing iterative development in project planning (Data Science Process Alliance, n.d.). The following is an outline of the project's implementation plan.

Phase 1: Business Understanding

Objective: Collaborate with NWSL stakeholders for a common understanding of the project needs, including objectives of the project and deliverables.

Tasks:

- Meet with NWSL stakeholders to identify the goals, objectives and scope of the project.
- Agree upon resources available for the project.
- Define project timeline and deliverables.

Deliverable: A document summarizing the above agreed upon points with the stakeholders plan for developing the project, such as goals, objectives, scope and deliverables.

Phase 2: Data Understanding

Objective: Collect and explore the NWSL player dataset for usability in the MI algorithm.

Tasks:

- Collect 2023 and 2024 season player data from the NWSL website and save it to a CSV file.
- Review and transform datasheet, including adding an index column and deleting rows with no data.
- Add the dataset to Jupyter Notebook and utilizing Python and the Pandas library to conduct data preprocessing.
- Explore and understand the data through exploratory statistics and visualizations.

Deliverable: A cleaned and explored dataset ready for further data processing.

Phase 3: Data Preparation

Objective: Process and transform the data to prepare it for modeling.

Tasks:

- Review data to identify variables that will be included in model fitting.
- Review data for missing data or variables that are highly correlated to each other.
- Identify and conduct variable transformations needed for model fitting.

- Encode categorical variables.
- Create the dependent variable as by summing goals (weighted by 2) and assists.
- Apply the IQR method to identify and remove outliers.

Deliverable: A dataset ready for model fitting.

Phase 4: Modeling

Objective: Develop, train, and optimize the predictive model.

Tasks:

- Import necessary Python libraries such as scikit-learn, seaborn, matplotlib and numpy.
- Split the data into sets that encompass either the independent variables or the dependent variable.
- Build the Random Forest algorithm using the prepared datasets.
- Tune hyperparameters (e.g., number of trees, max depth) to improve the model.
- Analyze the features to determine which are the most influential predictors and which can be eliminated from the model to simplify the model.
- Evaluate the model with metrics that evaluate the predictive functionality of the model, it's ability to explain the variability and generalizability to outside data.

Deliverable: A Random Forest model that has been trained, undergone hyperparameter tuning and evaluation.

Phase 5: Build the Application

Objective: Build the interactive web application and the user guide.

Tasks:

- Build an interactive user interface with Ipywidgets for user input to be captured and fed into the predictive model. The Python library, ipywidgets, will be imported.
- Integrate the results of the trained Random Forest model into the application so there is a print out of the results and an explanation.
- Create three or more visualizations to enhance the user experience and understanding of the predictive tool. The Python library, Seaborn, will be imported for this task.
- Conduct unit and system testing.
- Create the user guide and conduct peer review evaluation.

Deliverable: An interactive web application and user guide ready for evaluation.

Phase 6: Evaluation

Objective: Evaluate whether the project meets the business requirements and development plan.

Tasks:

- Review whether the goals and objectives of this project were met by the deliverables.
- Review the development plan to verify no steps were missed.
- Create a plan for any next steps and maintenance.

Deliverable: Web application and user guide that are ready for deployment.

Phase 6: Deployment

Objective: Deploy the web application and the user guide.

- Prepare the application for deployment.
    - For this project, the code and data will be uploaded to GitHub and deployed to a free hosting service called Binder.
- Setup and account with AWS and setup an AWC EC2 instance for hosting and complete all configurations.
- Transfer all application data files to the AWC EC2.
- Utilize AWS EBS for storage.
- Deploy application using AWC EC2.
- Make the user guide publicly available through the NWSL website.

Deliverable: The web application and user guide as publicly available.

Timeline

Milestone	Duration (days)	Projected start date	Anticipated end date
Business Understanding	5	01/02/2025	01/06/2025
Data Collection and Understanding	8	01/07/2025	01/14/2025
Data Processing	7	01/15/2025	01/21/2025
Model Development	10	01/22/2025	01/31/2025
Project Evaluation	5	02/01/2025	02/01/2025
User Application Deployment	8	02/02/2025	02/09/2025
User Guide	1	02/10/2025	02/14/2025

Evaluation Plan

1: Business Understanding

The Business Understanding evaluation plan is a full review of the project goals and development plan by the stakeholders. This verifies stakeholder alignment with the project goals and deliverables, and ultimately that the joint understanding of this project will benefit the needs of the business. This evaluation method is crucial before the work of developing the application.

2. Data Understanding

Evaluating this phase will be developer review of the exploratory data analysis to verify the integrity of the dataset, including no missing values, acknowledged variables for transformation and potential outliers identified.

3: Data Preparation

Evaluation of this phase focus on developer review of the preparation of variables to be fitted into the Random Forest model. This includes verifying the correct variable transformations of categorical to numeric were executed, and descriptive statistics verifying the categorical transformations make sense. Developers will also review the calculation and distribution of the GCS variable and verify that the outliers were removed.

Phase 4: Modeling

The evaluation of the Random Forest model will be done with the metrics of R-squared (R^2), Out-of-Bag (OOB) Score, and Mean Squared Error (MSE). R^2 evaluates the proportion of variance in the dependent variable that the model explains (Singh, 2023). A higher R^2, meaning close to 1, indicates that the model captures a strong relationships between the features predicting the outcome score, whereas a lower score, closer to 0, indicate a weaker relationship. The OOB score provides an estimate of the model's generalization capability by testing with unused data samples created during the bootstrapping or bagging process (Jain, V. 2024). A higher OOB score indicates that the model generalizes to outside data well and is less likely to be overfitting to the training data. Lastly, MSE will be used to measures the average squared difference between the predicted and actual values of the GCS (Singh, 2023). In general, lower MSE values indicate a better predictive performance by the model, however, the scale of the data should also be taken into account.

Phase 5: Application Development

The evaluation process of the application will focus on ensuring the functionality and integration of the user interface with the predictive model. Unit testing will be conducted to verify that the components of the interface, specifically the input forms and submit button, operate as expected and correctly capture user inputs. Integration testing will be performed to ensure that the Random Forest model integrates seamlessly with the user interface, taking in the user input into the model, and then reflecting the correct output onto the user interface. Regarding the user guide, the project team will provide peer review evaluation to verify its clarity and that it meets the project goals.

Phase 6: Evaluation

During the evaluation phase, user testing will be conducted with stakeholders to ensure the application aligns with their expectations and fulfills the project goals and objectives. Peer review will be completed

with the developers and stakeholders as the goals, objectives and development plan are reviewed to verify completeness.

Phase 7: Deployment

After deployment, user end to end testing will be completed with community members or stakeholders and the development team will receive written feedback that will be reviewed for potential improvements that could be made in the future. Website performance will also be monitored monthly by a developer for issues including verifying the load time is acceptable, whether errors or bugs are encountered that hinder the functionality of the website and scalability needs.

Resources and Costs

1. Itemize hardware and software costs.
    - There should not be any hardware costs as the proposed project and application can be developed using a standard PC and monitor that are likely found in an office. It is assumed that these are already available to the developers.
    - Regarding software expenses, the PC will need to be equipped with the software to create a CSV file, specifically Excel. This could incur an expense if the workstations do not have the software for this, however, the developers do have this available to them so it is unlikely this will be an expense. Otherwise, the developing language of Python is free to use, as well as the specific libraries that will be used in this project.

2. Estimated Labor Time and Costs

The following is a breakdown of total timeline and costs. There will be a single developer working on the project who will charge $40 per hour. A QA specialist will be brought in for some of the evaluation testing, who will charge $60 per hour.

Milestone	Developer Hours	QA Hours	Total Hours	Developer Cost	QA Cost	Total Cost
Business Understanding	6	0	6	$240	$0	$240
Data Collection and Understanding	10	0	10	$400	$0	$400
Data Processing	8	0	8	$320	$0	$320
Model Development	15	3	18	$600	$180	$780
Project Evaluation	2	0	2	$80	$0	$80
User Application Deployment	10	4	14	$400	$240	$640

Milestone	Developer Hours	QA Hours	Total Hours	Developer Cost	QA Cost	Total Cost
User Guide	2	1	3	$80	$60	$140

Total Cost of Labor: $2,600

3. Estimated Environment Costs

While the prototype of the application will be deployed on Binder, using this environment comes with storage and data limitations and so a plan will be made to transition the deployment to AWS. Specifically, Amazon Elastic Compute Cloud (Amazon EC2) will be used for hosting and Amazon Elastic Block Store (EBS) will be used for data storage in a fully virtual environment. Since the application is new, it has the potential to grow in increased networking needs, security and data storage, and so ensuring there is scalability and reliability in handling these needs is important to its ongoing maintenance (Amazon Web Services, n.d.). As this is a new application with low data needs, it is proposed to use the AWS EC2 On-Demand instance option of t2.micro, which will provide moderate computational needs, with always with the option to scale up. The cloud-based security features in the AWS service architecture are another benefit to utilizing this service, verifying it will be securely hosted and monitored against unauthorized access, which will be in addition to security measures taken by the development team including role base authorizations and managing security updates.

Based on use of the above referenced instance, the following is an annual cost estimate of using this environment based on available information on the AWS website at https://aws.amazon.com/ec2/pricing/on-demand/.

- T2.micro instance cost per hour**:** The instance costs $0.0162 per hour.
- Annual Cost Estimate: 0.0162 × 8760 hours/year= $141.912

Storage through EBS is charged separately, and those costs will be estimated based on information found at https://aws.amazon.com/ebs/pricing.

- General Purpose SSD (gp3) – Storage: $0.08/GB-month
- Annual Cost Estimate: .08 x 12 months per year = .96

Maintenance

It will be assumed that a developer will need to provide monthly monitoring and maintenance to this application and so 6 hours per month at $40 per hour will be estimated, resulting in an annual maintenance expense of $1,280.

Total Cost

Total cost estimate: $4,022.87

- There may be additional fees unanticipated that the team was unable to calculate in this estimate such as additional AWS costs, costs associated with scaling up or additional labor needed for maintenance.

Part C: Application

Web application title: **Predicting NWSL Player Goal Contribution Scores with Machine Learning**

The application may be accessed at:
https://mybinder.org/v2/gh/LeighAG/Capstone_Project_WGU.git/HEAD?urlpath=voila%2Frender%2FCapstone_LeighGrover.ipynb

Source Files: All files have been submitted via the WGU PA submission portal.

Part D: Post-implementation Report

Solution Summary

The problem discussed in this proposal is that while the NWSL has been growing in popularity, it still lacks parity with other US national sports leagues, especially in revenue. This is a concern because the NWSL is an exciting league that attracts highly talented players, and it is a business priority of the NWSL to increase not only popularity of the league but game attendance, sponsorships and revenue. Interest in the league could be helped by making available free, interactive tools that provide meaningful insights into player statistics through use of predictive models.

This application addresses the identified problem by offering a publicly accessible web-based tool powered by a machine learning model trained on NWSL player data. The interactive interface enables users to input data easily found on the NWSL website and then receive a predictive a goal contribution score. The GCS reflects the player's predicted overall impact on scoring, as well as a printout advising the user on how to interpret the score. The web application also provides visuals and documentation to enhance the user's understanding of the model and the predictive metric, which can increase perspective and trust in using the tool. The application can be easily used by fans and stakeholders to garner interest or investment in their favorite players, contribute to discussions or activities about player statistics and even play a role in recruiting talent. The web application also supports the NWSL's broader business needs of fostering interest in players which can help promote excitement around NWSL and increased sponsorship.

Data Summary

The data was collected from the NWSL website: https://www.nwslsoccer.com/stats/players/all. Specifically, this was done by copying all player data for the regular NSWL seasons for 2023 and 2024 and pasting it into a CSV file. Of note, the developer found that the variable, pass accuracy percentage, was only collected in the 2024 regular season and not 2023. Therefore, the decision was made to delete this variable from the 2024 data to ensure uniformity among the data collected.

To capture position played, it was necessary to filter the available data by forward, midfielder, defender and goalie, separately to capture this variable as it was not available on the website as a data column. A new column was created in the CSV file called "p" and was filled in by the developer with f=forward, m=midfielder, d=defender and g=goalie. Once in the CSV file, an additional column was added to provide unique ID numbers to each row of data. Also, in this stage, rows with entries that consisted of all zeros in each cell were filtered for and deleted, because these rows did not have useful data for training the model. After these steps were completed, the resulting dataset consisted of 649 rows.

Once the raw data set was completed, it included the following variables:

['Team' = Team played on, 'GP' = Games Played, 'GS' = Games Started, 'Mins' = Minutes Played, 'G' = Goals, 'A' = Assists, 'S' = Shots, 'SOT' = Shots on Target, 'KP' = Total Attacking Assists, 'Tackles' = Tackles, 'FC' = Fouls Committed, 'FS' = Fouls Suffered, 'OFF' = Total Offside, 'YC' = Yellow Cards, 'RC' = Red Cards, 'P' = Position Played]

The structure of the dataset was also reviewed, identifying the fields that would be used for calculating the dependent variable as well as potential predictor variables to use in the model. Once this phase of data collection was completed, it was saved and added to Jupyter Notebook to proceed to data design and processing. Once a notebook was started for this project, the Pandas library was imported as it has the capabilities for assisting with data removal, transformations and categorical variable encoding. Python code was executed to search for missing data and none were found.

The Random Forrest Algorithm can process categorical data, however, it needs to be in numeric format. Categorical variables that could be used as potential predictor features were identified: P=Position Played and Team = Team Played On. Both categorical variables were encoded to assigned numeric categories to the fields of team names and positions played to ensure compatibility with machine learning models. See below:

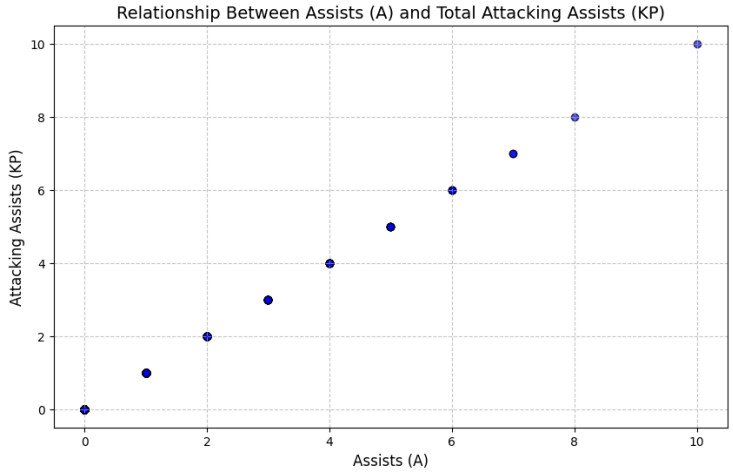
```
Team: Angel City FC, Category: 0
Team: Bay FC, Category: 1
Team: Chicago Red Stars, Category: 2
Team: Houston Dash, Category: 3
Team: Kansas City Current, Category: 4
Team: NJ/NY Gotham FC, Category: 5
Team: North Carolina Courage, Category: 6
Team: Orlando Pride, Category: 7
Team: Portland Thorns, Category: 8
Team: Racing Louisville FC, Category: 9
Team: San Diego Wave FC, Category: 10
Team: Seattle Reign, Category: 11
Team: Utah Royals FC, Category: 12
Team: Washington Spirit, Category: 13


P: D, Category: 0
P: F, Category: 1
P: G, Category: 2
P: M, Category: 3
```

The categorical variable for P = Position was labeled “Pos” and the categorical variable for Team was labeled "Teams.” Of note, during the model development and hyperparameter tuning portion of this project, it was found that “Pos,” the position played variable, was a feature that had low contribution value to the GCS. Therefore, this developer not only decided to remove this variable from the model predictor features, but the players with the position of goalie were removed from the dataset for model refitting. This was because the developer decided to narrow the data scope to those players who could more realistically provide assists and goals to give more focus to the data. Goalies are a position in which it is not expected that they would score a goal or provide an assist that led to a successful scoring opportunity.

During the exploratory data process, it was highly suspected that the variable “A”=assists was strongly correlated with the variable “KP” = total attacking assists. A correlation computation was completed along with a scatter plot, and the results indicated they were indeed highly correlated. Therefore, KP was

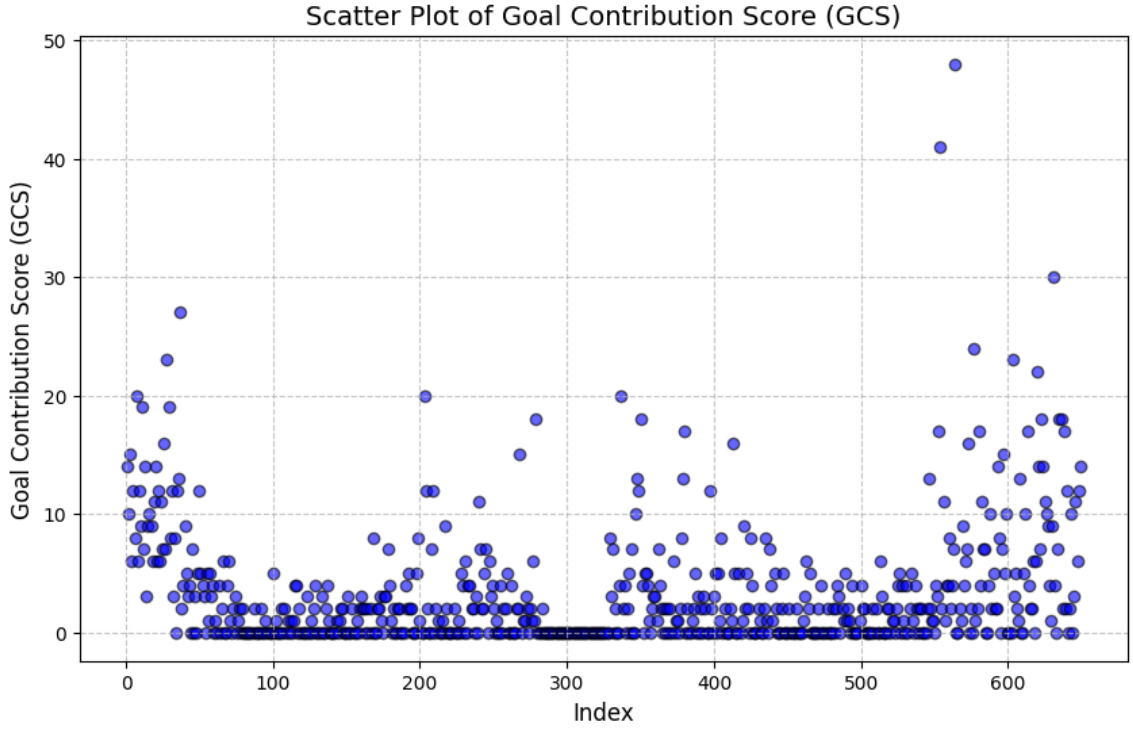
not included as a predictor, as it is too highly correlated with A, one of the variables that will go into the aggregate metric of GCS.



Pearson Correlation Coefficient: 1.00

A new variable, the planned dependent variable, was created called ‘GCS.’ This is the aggregate Goal Contribution Score, based off existing metrics that utilizes both goals and assists in understanding player contribution to scoring (FBref, n.d.). This particular metric was created by summing goals (weighted by multiplying by 2 due to significance in the total score) and assists.

Conducting exploratory data statistics on the GCS indicated that there were noticeable outliers.



The Interquartile Range (IQR) method was then used to identify and remove outliers, ensuring that extreme values did not have an effect of the accuracy of the model (Wikipedia Contributors, n.d.). This

involves identifying the extreme values of the GCS variable that fall below 25% and above 75% of the data values. After the calculation, the following is the description of the outliers that were removed:

```
Lower Bound: -6.0, Upper Bound: 10.0
count    60.000000
mean     16.466667
std       6.745913
min      11.000000
25%      12.000000
50%      14.000000
75%      18.000000
max      48.000000
Name: GCS, dtype: float64
```

The removal of the outliers and creation of a new dataset (matches_cleaned) resulted in a total of 541 rows and better distributed data.

To further prepare the dataset for model testing, two datasets were created from the matches_cleaned dataset. The first was meant to hold all possible predictors and no dependent variables. This dataset was called 'x' and contained all variables except goals, assists, attacking assists and GCS. See a sample below.

Index	Team	GP	GS	Mins	S	SOT	Tackles	FC	FS	OFF	YC	RC	P	Pos	Teams
2	Seattle Reign	23	17.0	1663	27	16	11	23	27	8	1	0	F	1	11
4	San Diego Wave FC	22	7.0	693	18	10	5	19	2	4	1	0	F	1	10
6	San Diego Wave FC	22	12.0	1205	25	15	10	10	16	3	1	0	F	1	10
8	Utah Royals FC	22	14.0	1155	18	12	11	10	8	5	1	0	F	1	12
10	NJ/NY Gotham FC	22	16.0	1307	16	8	19	26	14	0	1	0	F	1	5

A dataset called 'y' was then created from matches_cleaned that only contained the GCS variable and indices. Below is a description of the GCS variable:

```
count    541.000000
mean       2.134935
std        2.556455
min        0.000000
25%        0.000000
50%        1.000000
75%        4.000000
max       10.000000
Name: GCS, dtype: float64
```

Machine Learning

Random Forest Algorithm

To create the model predicting GSC, the random forest algorithm was chosen because of its ability to handle both categorical and continuous variables, evaluate and rank feature importance in prediction and its ability to provide better prediction accuracy through multiple decision trees and data replacement methods, such as bootstrapping (IBM, n.d.). The Random Forest model is an ensemble learning method that uses multiple decision trees to make predictions. Bootstrapping or feature bagging is used in creating random subsets of the data that involves sampling the training dataset with replacement. Thus, decision trees in the Random Forest algorithm are trained on these random subset samples. This feature makes the Random Forest Algorithm helpful when dealing with smaller datasets, as well as ensuring that the decision trees are not identical, reducing the likelihood of overfitting the data (GeeksforGeeks, 2024). In fact, when creating the Random Forest algorithm model, once can adjust many parameters, including node size, the number of decision trees, and the number of features sampled. For the model prediction, in the case of regression and with a dependent variable that is continuous, the outcome of all the decisions trees will be averaged out. In the case of classification, the prediction will be the most frequently arrived at variable (IBM, n.d.).

Model Creation

The Python library, Scikit-learn, was imported to build and create the random forest algorithm model (Ujhelyi, 2022). Other libraries added include numpy and matpotlib.plyplot. Firstly, the cleaned datasets of 'x' and 'y' were split into training data, 80%, and testing data, 20%. The following is the coding example:

```
x_train, x_test, y_train, y_test = train_test_split(x, y, random_state=11, test_size=0.2)
```

The split of the dataset resulted in 432 instances in the training dataset and 109 instances in the testing dataset.

The second step in preparation was to identify features to include in the model as predictive variables and capture them in an array to pull the selected variables from the 'x' dataset. For example, this was the final result of chosen features:

```
predictors = ["S", "SOT", "Tackles", "FC", "FS", "GP"]
```

Features were added and then removed when they were found to have a very small effect on the model's prediction performance. This was determined using the 'feature_importances_' method in the random forest model. Notably, features removed because of lack of effect were the categorical variables of teams and positions played, as well as the continuous variables of minutes played, offsides and games started. Variables on penalties, such counts of yellow and red cards, were also not features that notably effected the predictive score. The features kept as the most impactful were "S"=Shots, SOT = "Shots on Target", Tackles = "Tackles," "FC" = Fouls Committed, "FS" = Fouls Suffered, and "GP" – Games Played.

The model was initialized with key hyperparameters that the developer adjusted several times to try to get a better fitted model based on monitoring evaluation metrics such as the accuracy metric, mean squared error, OOB error and R^2 value. The finalized model is as follows:

```
#create random forest model
rf=RandomForestClassifier(
    n_estimators=150,
    min_samples_split=6,
    bootstrap=True,
    oob_score=True,
    random_state=1
)
```

The n_estimators controls the number of decision trees. The number was started at 100, then increased to 150 and then to 175. There was decline in the accuracy metric after this, and so the number was brought back to 150.

The min_samples_split signifies the minimum number of samples an internal node must have to split into further nodes. It is recommended to keep the values between 2 and 6 (GeeksforGeeks, 2021). Of note, this number was adjusted from all the numbers in this range while monitoring performance metrics and the overall better scores were achieved with 6.

Bootstrap = True, enables bootstrapping, although this is the default.

Oob_score = True, enables us to calculate the oob score for evaluation purposes (GeeksforGeeks, 2023).

Random_state=1 is setting the seed for randomization to set the same behavior when running the code.

Below is a sample of the code that fit the model using the train datasets of 'x' and 'y':

```
rf.fit(x_train[predictors].values, y_train);
```

Algorithm Justification

The goal of this project is to develop a predictive model that uses NWSL player feature statistics to estimate a Goal Contribution Score based on weighted goals and assists. To achieve this, a model was needed that could handle both continuous and categorical data while predicting a continuous target variable. Additionally, the model needed to be able to perform across a wide range of future players, ensuring generalizability beyond the training and testing datasets.

The Random Forest algorithm was a good choice because of its potential to protect against overfitting by averaging predictions from subsets of multiple decision trees. Furthermore, it's methods of multiple decision trees allow it to handle both large and small datasets effectively, which allowed the developers to have less concern about the size of the dataset. Also, the ability to fine-tune hyperparameters, such as the number of and depth of trees, provides flexibility was helpful for the developers to understand ways to optimize the model's performance. This was very important for it to be able to be useful beyond just the 2023 and 2024 data, as the hope is the model will be relevant with future NWSL stats. It was also insightful to recognize the limits to adjusting the parameters, such as observing lack of significant improvement on some evaluation metrics after adjustments. An additional advantage of Random Forest is its capability to rank feature importances, offering valuable insights into the most influential predictors of player performance. This insight can allow fans to watch key predictor variables in their favorite players, or coaches to focus on developing certain strengths in their players. It also leaves the door open for understanding other features that could be in included in the dataset to bolster the predictive value of the model. Lastly, there are various metrics that can evaluate Random Forest models so the developers, stakeholders and fans can have a deeper understanding of the predictive value of the model and how much of the variance is unexplained by the included features.

Validation

Mean square error

The Mean Squared Error (MSE) is the calculated squared error between the model prediction and the actual value based on the testing data. This measure of the average squared difference between the predicted values and the actual values provides information about how well the model is functioning in its prediction accuracies (Kumar, 2023). Generally, the smaller this number is, the better because there is less variance between the predicted and actual values, although scale of the values should be taken into consideration. The MSE from this model is: 4.98. This can be interpreted that the average squared difference between the predicted values and the actual values is about 5. This is a sizeable error considering the scale of the GCS is from 0 to 10. This indicates that improvements should be considered in developing the predictability of the model to achieve a smaller MSE.

R Squared Value

The R squared metric can be used to evaluate how well the features in the model explain the variation in the dependent variable from the model's regression line (Kumar, 2023). Specifically, it is used to measure how well the regression line in the model fits the data. Generally, the greater the value of R-Squared, the better the model is at explaining the variation of the outcome variable. The R-Squared value for this model was .16. This can be interpreted as the features used in the model are only able to explain 16% of the variance in the dependent variable score and 84% of the variance remains unexplained. This score indicates low predictability with the features and data used in the model. For future improvements, it would be helpful to find other datasets that provide additional features that may have a greater predictive affect on the model's predictions.

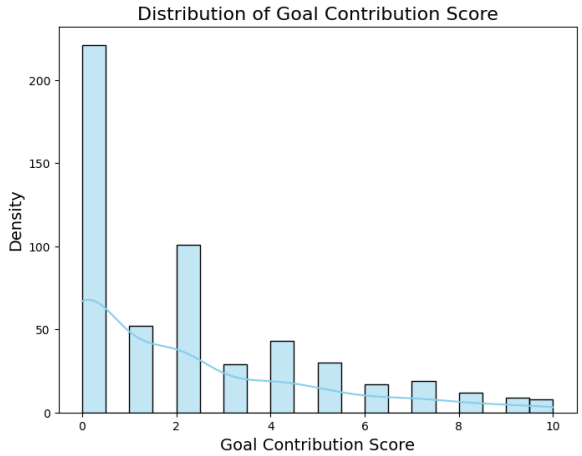
Out of bag error

The Out of Bag Error score can be a useful metric for understanding how the model performs on unused data to support how generalizable it is to outside data. It is calculated using the bootstrapped or out of bag samples that were unused in the training of the model (Geeksforgeeks, 2023). A lower the OOB error is assumed to be better, because it is the error rate of predicted scores on untested data. The OOB error score for this model was: 0.57. This can be interpreted that the model was only able to accurately predict about 43% of scores on unseen data. As found in the previous evaluation metrics, there is much room for improvement in fitting the model to be more generalizable to outside data.

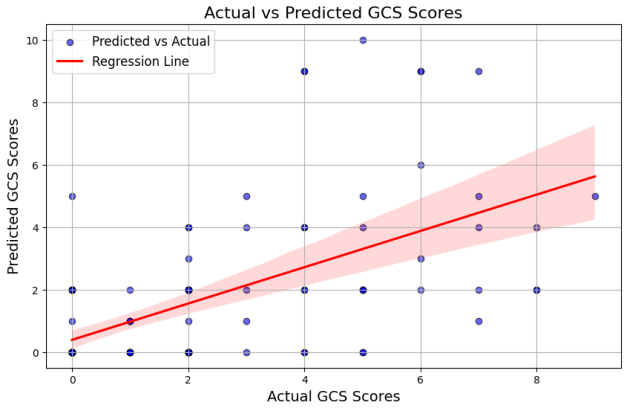
The above evaluation metrics indicate that the performance of this model show a low predictability when it comes to the GCS. There appears to be a high level of unexplained variance in the scores by the model, a proportionally high value of squared error variance from the predicted and actual values as well concerns for generalizability to outside data. These results make a case for future improvements to this model, and the team should investigate further research into this topic and explore other data sources available that may contain features that better contribute to explaining the target variable.

Visualizations

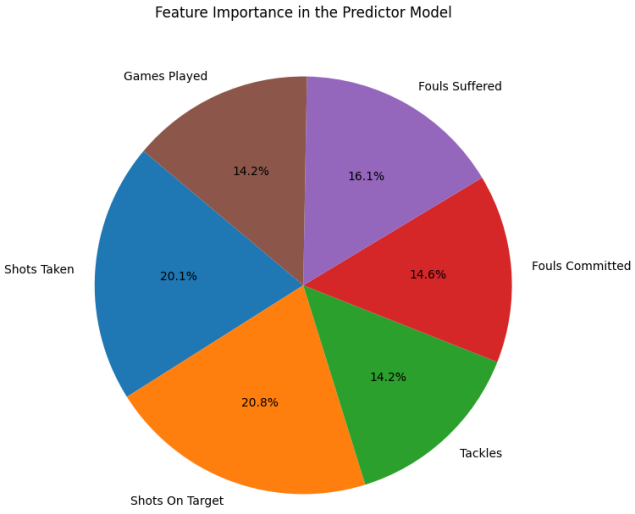
1. A histogram showing the distribution of the GCS in the data.



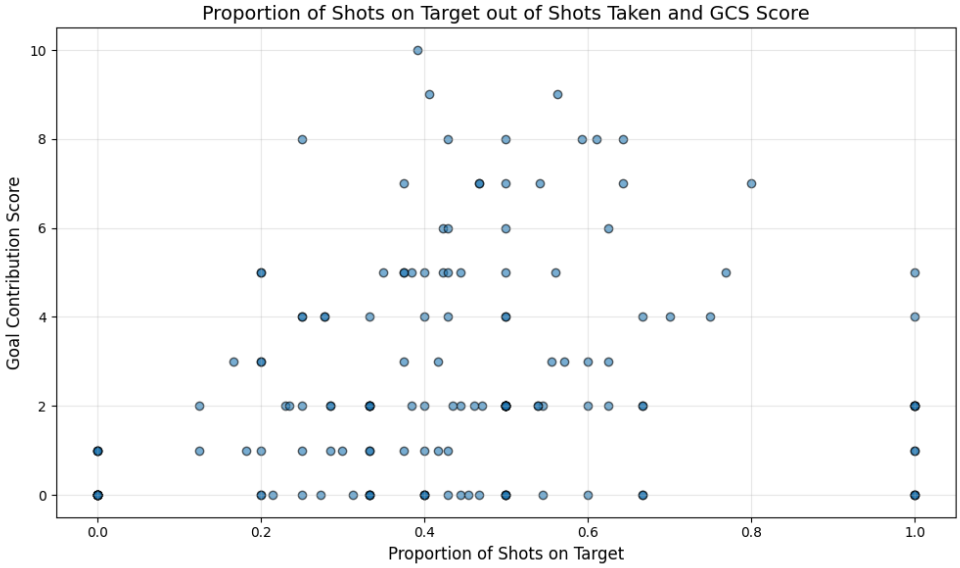
2. A scatter plot showing predicted and actual GCS values with the model regression line. Of note, the scatter plot points summing to less than the number of values in the test dataset is likely because there are values with the same predicted and actual values as there is no missing data.



3. A pie chart illustrating the feature importance percentages in the model



4. A scatter plot showing the proportion of shots on target over shots taken and GCS score. This was created as a visualization of interest because shots on target and shots taken have the highest percentage of feature predictability.



User Guide

How to Access the Web Application

1. The user only needs internet access and to open a browser.

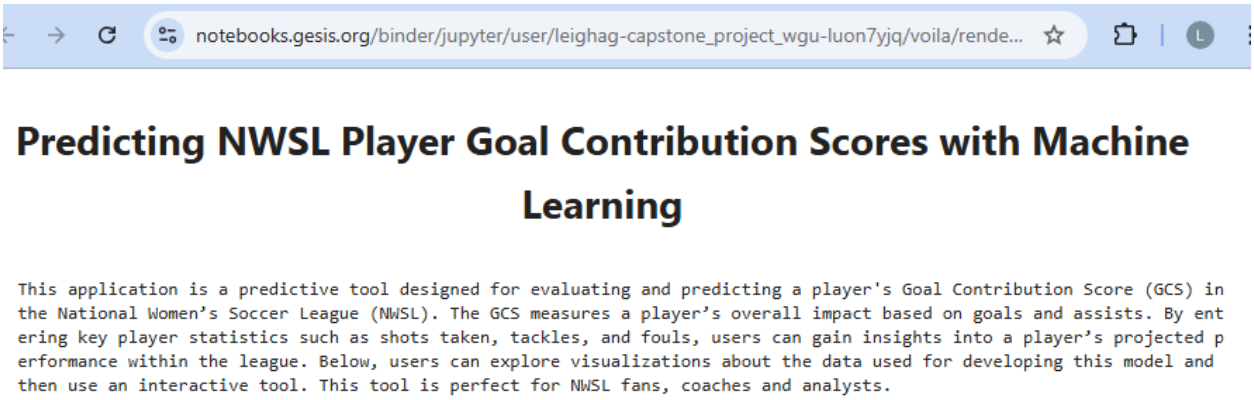
    - The application has been deployed using Binder, which builds an interactive environment from a GitHub repository that can be accessed and shared on the internet.

2. The user may copy and paste the following address into the internet browser: https://mybinder.org/v2/gh/LeighAG/Capstone_Project_WGU.git/HEAD?urlpath=voila%2Frender%2FCapstone_LeighGrover.ipynb

    - Please note that the website has a slow launch time and may take more than a few minutes to load. This is because Binder must take the necessary steps of building and launching the environment.
    - The input boxes at the bottom of the application sometimes load shortly after the rest of the application appears. Please be patient.
    - The web application may shut down if left idle for too long. If this happens, please reload the website by pasting the link into the browser.

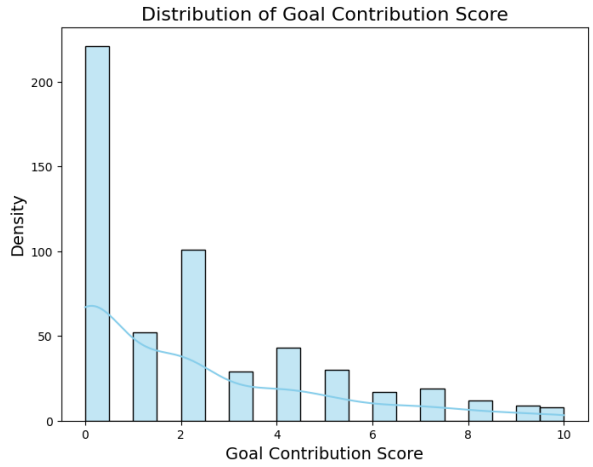
How to Use the Application:

1. Open the web application using the above link.
2. Read Introductory Paragraph that explains the purpose of the application.

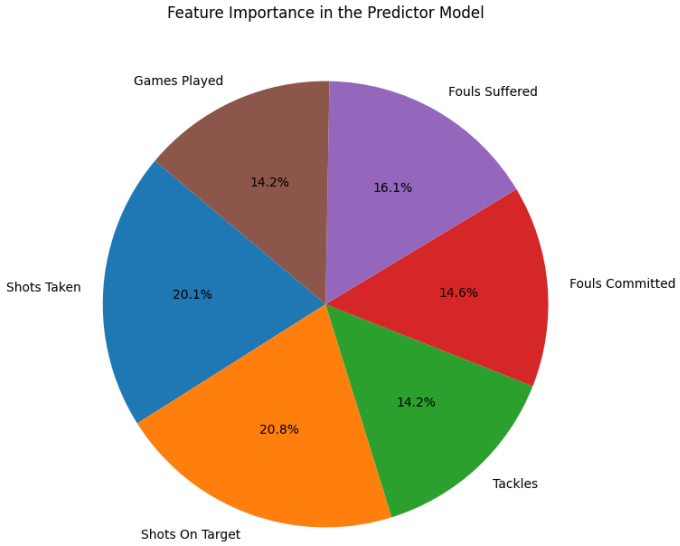


3. Review visualizations to understand the descriptive patterns in the data and model.

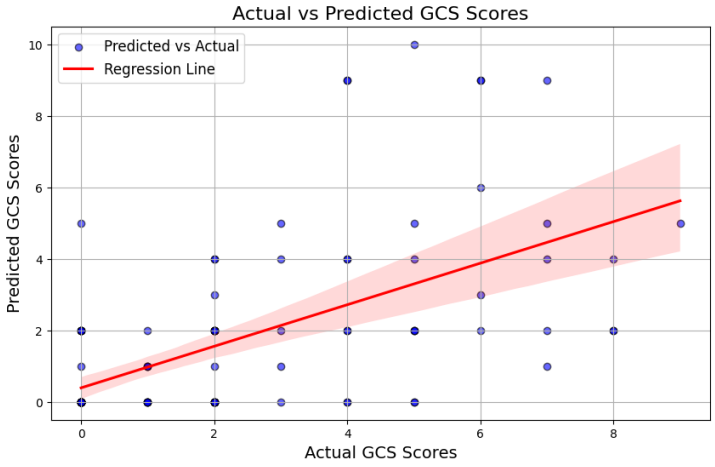
1. A histogram showing the distribution of the GCS in the data.



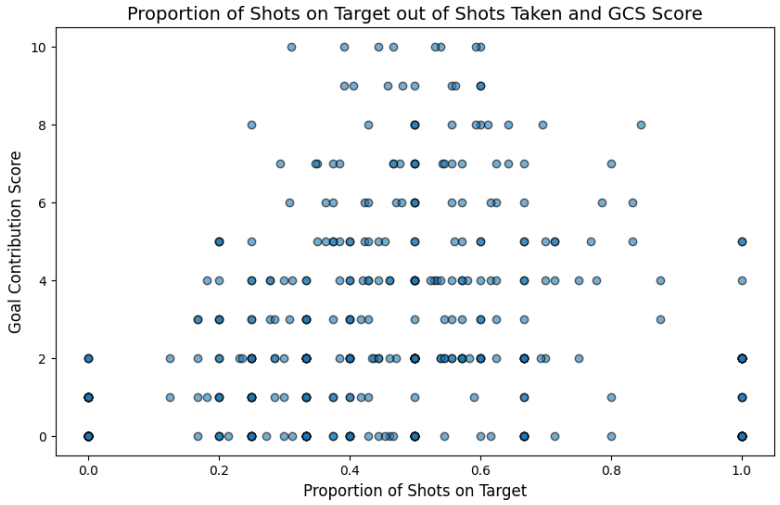
2. A pie chart illustrating the feature importance percentages in the model



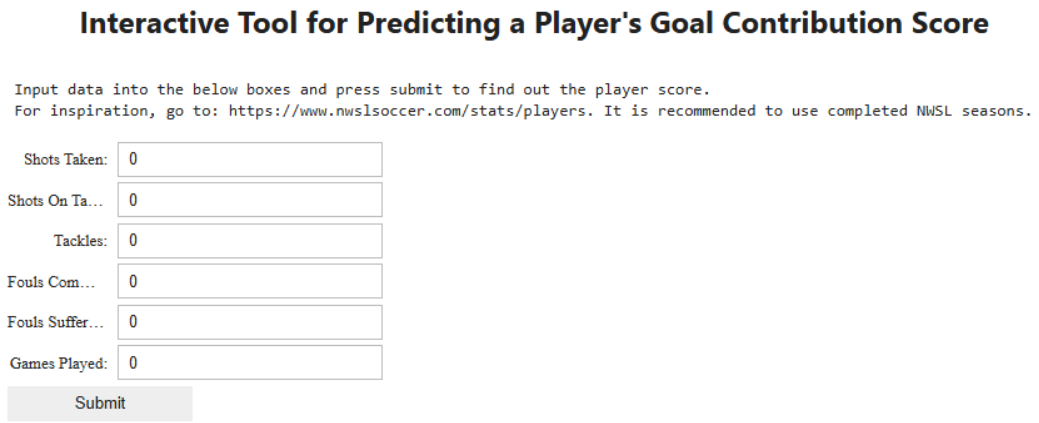
3. A scatter plot showing predicted and actual GCS values with the model regression line.



4. A scatter plot showing the proportion of shots on target over shots taken and GCS score. This was created as a visual ization of interest because shots on target and shots taken have the highest percentage of feature predictability.



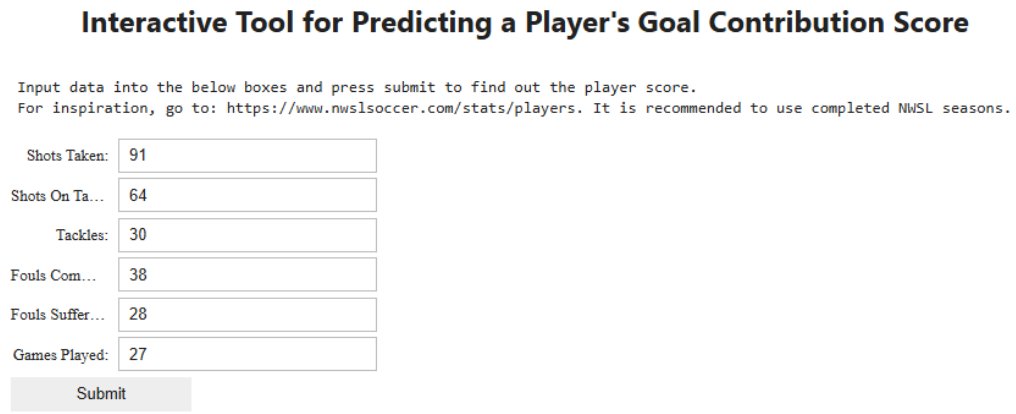
4. Navigate to the interactive tool.



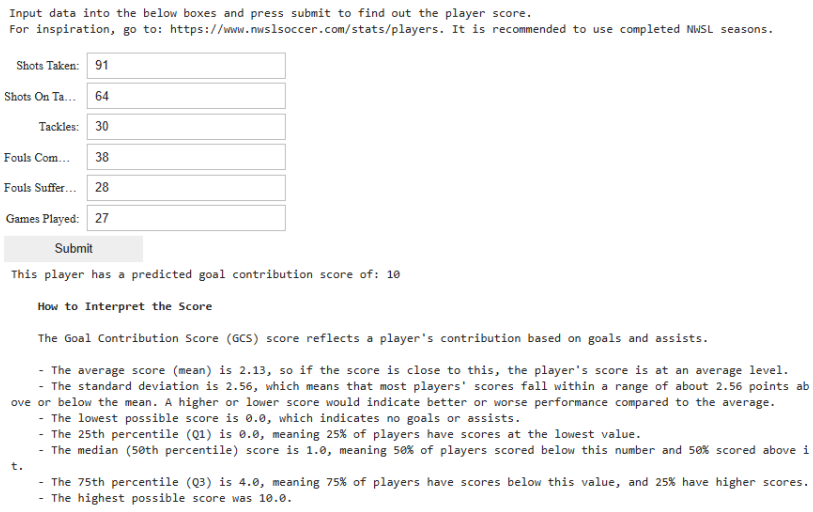
5. Pick a player and find their statistics to fill in the boxes. For this example, I will go to the NWSL website and choose 2024 Golden Boot Winner, Temwa Chawinga.

Team	Player	GP	GS	Mins	G	Pass%	A	S	SOT	KP	Tackles	FC	FS	OFF	YC	RC
KC	TEMWA CHAWINGA	27	26	2324	21	74	6	91	64	6	30	38	28	18	1	0

6. Fill in the boxes with the correct feature data.



7. Click the submit button to view results.



8. Read the accompanying descriptive information about the distribution of the GCS to provide context to the predictive score received. Go back and review the visuals for additional information about where this player's GCS score sits among the model training data.

Reference Page

Amazon Web Services. (n.d.). *Amazon EC2.* Retrieved December 20, 2024, from https://aws.amazon.com/ec2/

Data Science Process Alliance. (n.d.). CRISP-DM: The data science process. Data Science Process Alliance. https://www.datascience-pm.com/crisp-dm-2/

FBref. (n.d.). *Expected goals (xG) model explained*. Retrieved December 12, 2024, from https://fbref.com/en/expected-goals-model-explained/

GeeksforGeeks. (2023). *OOB errors for Random Forests in Scikit-learn.* GeeksforGeeks. Retrieved December 5, 2024, from https://www.geeksforgeeks.org/oob-errors-for-random-forests-in-scikit-learn/

GeeksforGeeks. (Last Updated : 28 May, 2024). Feature selection using random forest. GeeksforGeeks. https://www.geeksforgeeks.org/feature-selection-using-random-forest/

IBM. (n.d.). Random forest. IBM. Retrieved December 1, 2024, from https://www.ibm.com/topics/random-forest

Jain, V. (2024). *Understanding Random Forest and OOB error*. Medium. Retrieved December 17, 2024, from https://medium.com/@jainvidip/understanding-random-forest-0ca15aaa443c

Kumar, A. (2023). *Mean square error vs. R-squared: Which one to use?* VitalFlux. Retrieved December 17, 2024, from https://vitalflux.com/mean-square-error-r-squared-which-one-to-use/

Rickevicius, G. (2024, July 12). *How data analytics enhances fan engagement in sports*. Built In. https://builtin.com/articles/data-analytics-fan-engagement-sports

Ujhelyi, T. (2022, May 30). *Random forest in Python: A simple guide with examples*. Data36. Retrieved from https://data36.com/random-forest-in-python/

Wetzel, D. (2024, November 15). *NWSL taking a different approach than MLS in gaining an audience*. Yahoo Sports. https://sports.yahoo.com/nwsl-taking-a-different-approach-than-mls-in-gaining-an-audience-194558574.html

Wikipedia contributors. (n.d.). *Interquartile range*. In Wikipedia. Retrieved December 4, 2024, from https://en.wikipedia.org/wiki/Interquartile_range

Wikipedia contributors. (n.d.). *List of professional sports leagues by revenue*. Wikipedia. Retrieved December 20, 2024, from https://en.wikipedia.org/wiki/List_of_professional_sports_leagues_by_revenue