

Final Report

for

“Method”; The March Madness Assistant

Isabelle Congdon, Sophia Hall, Alexis John, Noel Mathew, Emily Tibbens

Case Western Reserve University

Report Overview

- 1 Abstract
- 2 Introduction and Background
- 3 Proposed Solution and Objectives
- 4 Success Criteria
- 5 Technical Descriptions
- 6 Key End Users Features and Functionalities
- 7 Results Verification
- 8 Future Work
- 9 Access to Project
- 10 Individual Reports
 - Isabelle Congdon
 - Sophia Hall
 - Alexis John
 - Noel Mathew
 - Emily Tibbens

1 Abstract

“Method” is a user interface in which we help the users create *their* perfect bracket based on statistics, trends, and other data that is important to them. Users will rank a series of team statistics and figures that we will then use to make the predictions for them based on the factors they see as most important. Users can also see how their bracket is holding up during the course of the tournament. In addition, this application will include internal sports news so users can be up to date on the tournament through our news page, as well as compete with their friends by creating a team, and post messages for other users on the site to view.

2 Introduction and Background

March Madness is the men’s Division I basketball single elimination tournament where 68 teams around the country compete for the title of national champions. Before the tournament begins, people across the country are strategically designing their brackets to predict the winner of the tournament. In addition, sports broadcasters and other professionals are sharing their predictions of wins in each round based on statistics of how the teams have performed in the regular season. Every year, basketball fans around the world enter their own picks into multiple bracket competitions. The very first March Madness bracket pool is said to have been in 1977, where 88 people filled out their predictions in a Staten Island bar (2). Since then, more teams have been added to the bracket, and a lot more people have joined the madness. In 2021, 16.2 million brackets were officially submitted for the NCAA March Madness bracket challenge (1). With 9.2 quintillion possible bracket variations, it is not surprising that no one in the history of the tournament has created a perfect bracket.

Most fans use personal knowledge, research, intuition, and even mascots or team colors to help them fill out their bracket every year. There are many websites and applications that host bracket challenges, provide statistics, and give tournament results. The official March Madness tournament website offers a similar experience that will assist fans in making their bracket selections with a few of the most popular statistics. These websites however, are geared towards those who are familiar with the tournament, and have been keeping up with how the teams have been doing all season. This is where “Method” comes in.

3 Proposed Solution and Objectives

3.1 Proposed Solution

For this project, we intended to create a new product which can be used to make the March Madness Bracket Challenge easier and more accessible for people who aren’t familiar with it. The goal is that amateur basketball fans can use our web application to take the randomness out of picking teams, and avoid having to keep up with team statistics all season. We intended for our target audience to be anyone who wants to participate in the bracket challenge, from someone who knows nothing about basketball but wants to enter their office pool, to

someone who competes every year and wants some reassurance on the teams they have already selected. We envisioned “Method” to be more intuitive and user friendly, so those with less experience can make more confident and informed decisions about their picks.

The main feature we envisioned for our website was the creation of a bracket prediction for each user based on criteria that the user selects. We sought to provide the user with a list of team statistics to choose from, which we would then use in our algorithm to create the bracket. We also aimed to provide the user with a combination of statistics that we found to create the most accurate bracket. In addition to making brackets, we wanted to offer users the option to join teams and compete against other users. We also wanted users to be able to keep up with the latest tournament news and communicate with other users through a news page. Finally, we envisioned that the website would provide users with the results of each game in the tournament, and automatically update their bracket scores based on those results.

3.2 Objectives

Objective	Description	Status
<i>Home Page</i>	The first page seen when a user visits the website; has two different views depending on whether the user is logged in	Completed
<i>User Profile</i>	Users can create an account with a username and password and use the username and password to log in everytime they visit the website	Completed
<i>View Bracket Prediction</i>	A page where the user can view the combination of stats that we predict to be most accurate, and the bracket we have generated based on those stats	Completed
<i>Create Bracket</i>	A page where users can select up to 5 stats and the website provides them with a bracket based on the chosen stats, which they can then name and save	Completed
<i>View Brackets</i>	A page where users can view	Completed

	all of the brackets they have created. They can also delete or edit their brackets from this page.	
<i>Edit Bracket</i>	A page where the user can edit the chosen stats or edit the name of their bracket	Completed
<i>View Scores</i>	Allows the user to view updated results of each game in the tournament	Completed
<i>Create Team</i>	A page where the user can create a team to compete against other users	Completed
<i>Join Team</i>	A page where the user can enter a join code and join an existing team	Completed
<i>View Team</i>	A page where the user can view their team information, as well as the brackets of the other users in the team	Completed
<i>News Page</i>	A page where users can view recent news relating to the tournament, as well as make blog posts and view other user's posts	Completed
<i>Start Tournament</i>	A function used by the admin to lock brackets and teams and provide bracket scores after the tournament has started	Completed

4 Success Criteria

Based on our goals and objectives we defined the following deliverables to measure success criteria:

1. Accuracy of Data Collection

All locally stored data, such as data from recent tournaments or initial matchups from each year will be manually checked to ensure accuracy of our web scraping results.

2. Verification of Bracket Construction

In order to verify that the bracket construction algorithm created accurate brackets based on the stats it was given, we will run a test to make sure that the algorithm results match with the manually constructed brackets.

3. General Front End Operations

The first deliverable we were looking to achieve was the homepage and user login page. To measure the success of these pages, we check that users can navigate from the homepage to the user profile page where they are able to either create or login to an account. Next, we check that the user can navigate to the team page where they can either create or join a team to compete against friends. Our next deliverable is the bracket page where users are able to build a bracket given the functionalities given to them such as the ranking stats feature. Finally, the last deliverable we used to measure success is the functionality of the news page. We consider this page a success if the user can view live twitter feeds to see applicable sports updates.

4. Latency

In order to ensure that the website is user-friendly, we defined a load time of less than 3 seconds for each page as the ideal latency, since none of our functions were too data-heavy.

5 Technical Descriptions

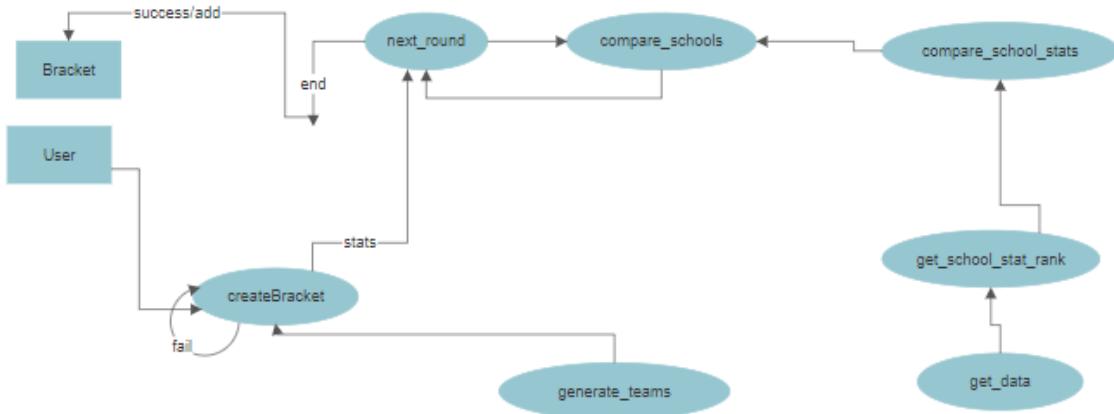
1. Bracket Generation

In order to collect team stat data for division I men's college basketball teams, we scraped *sports-reference.com*. Due to the length of time it takes our code to continuously scrape the website to get data for each team, we are storing the past 5 years of each team locally. We then wrote the function to receive any school name and scrape the individual team page for a given year. The function will then output all team stats for that year.

The function `get_school_stat_rank` allows us to get the school's overall ranking for a given stat category. We then use this function to build another function called `compare_school_stats`. This function will compare the two competing schools and output the highest ranking school for the given statistic.

Using the rankings the users implemented, and the `compare_school_stats` function we built a new function, `compare_schools`, that returns the school projected to win based on the set conditions. We will use this function to build each user's individual bracket.

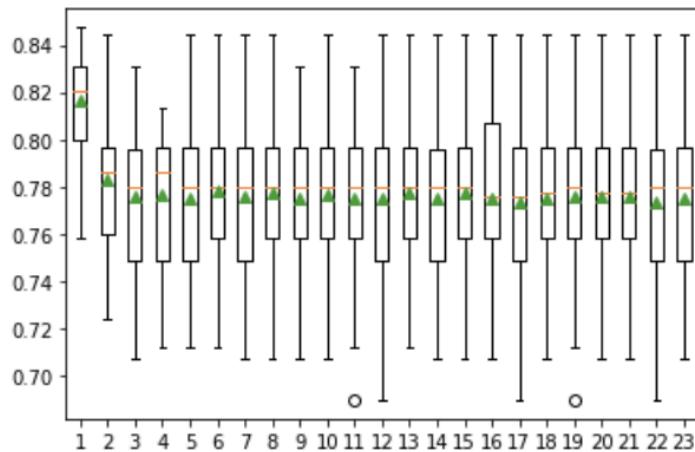
The bracket generation functionality is mapped below



2. Statistic Selection

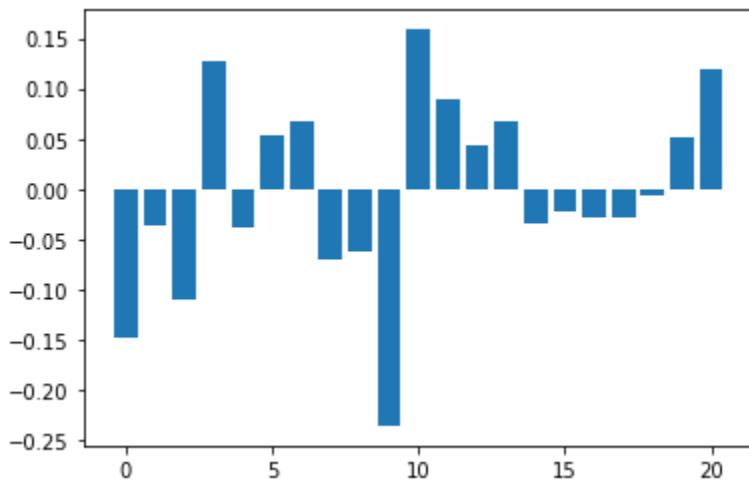
One technical constraint we looked into was deciding how many statistics to use. We used sklearn and the feature selection model recursive feature elimination that removes features until the number of specified features is reached. Our model started using all 23 statistics and removed a feature each time to give us the most accurate model. We found that using only one of the 23 total statistics, to predict if a team will advance is the most accurate model. However, letting users only choose one statistic will only create 23 distinct brackets. This will not allow for the competitive environment we are trying to create. Therefore, we decided to allow the user to choose up to 5 statistics when generating their brackets. This allowed us to increase the chances that users have unique individual brackets, since the number of brackets that can be generated from combinations of anywhere between 1 and 5 stats is 27893. Below is the output and a graph from the feature selection model.

```
>1: 0.814
>2: 0.781
>3: 0.774
>4: 0.775
>5: 0.775
>6: 0.773
>7: 0.774
>8: 0.773
>9: 0.773
>10: 0.774
>11: 0.773
>12: 0.772
>13: 0.772
>14: 0.772
>15: 0.774
>16: 0.773
>17: 0.774
>18: 0.772
>19: 0.772
>20: 0.773
>21: 0.773
>22: 0.774
>23: 0.773
```



3. Analyzing Predictiveness of Statistical Categories

To determine which statistical categories formed the most predictive combination a machine learning predictor was built. The predictor followed a binary classification model and used a logistic regression algorithm so that each of the statistical categories, when used as a feature in the model, could have an associated coefficient which could be easily extracted for determining feature importance. The coefficients of each statistical category are shown below and ordered according to how predictive each statistic is.



```
[('FTA', 0.15865628861714032),
 ('2P', 0.12713550668257806),
 ('PTS', 0.11944326327979166),
 ('FT%', 0.09008130071197853),
 ('3P', 0.06784002497236052),
 ('DRB', 0.0672393168125548),
 ('2P%', 0.053143582389198016),
 ('PF', 0.051103043526913625),
 ('ORB', 0.04350391653450242),
 ('TOV', -0.007037767530835757),
 ('AST', -0.02244139901725985),
 ('BLK', -0.02787533078412643),
 ('STL', -0.027914456387680402),
 ('TRB', -0.03537368269001054),
 ('FGA', -0.03655219533441392),
 ('2PA', -0.0383630903055797),
 ('3P%', -0.06270785691640311),
 ('3PA', -0.06997112091352461),
 ('FG%', -0.10954570660217015),
 ('FG', -0.14771712491056987),
 ('FT', -0.23590009122231262)]
```

4. Implementation

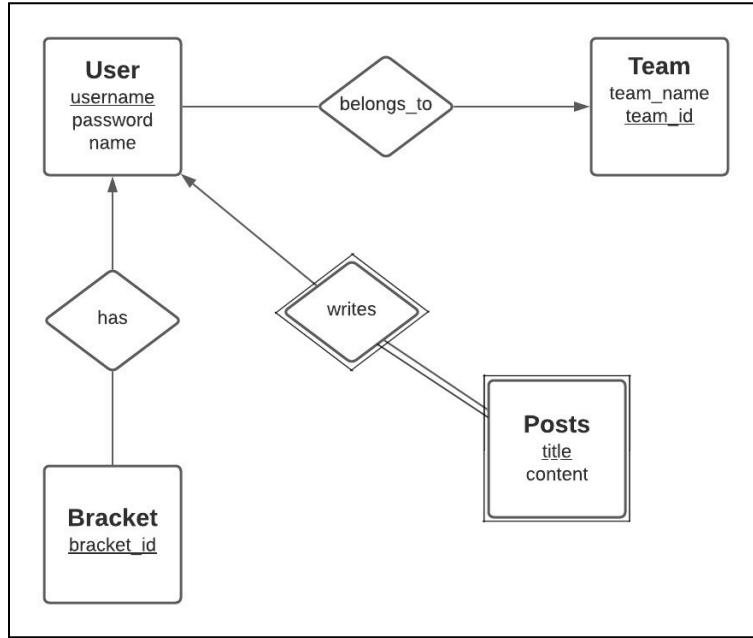
To implement our web application, we used Django, a high-level python web framework. Django allowed us to handle HTTP requests from webpages, forward them to the appropriate view, and create an HTTP response to send to the appropriate HTML file. All of our backend code was implemented in python. The bracket prediction model was built in python using python library packages such as pandas and requests. We used the Beautiful Soup python library for web scraping. The database was implemented with Django's default DBMS, SQLite.

Front end implementation was done using HTML and CSS for the overall design of the webpages. HTML code was used to design the layout of the web pages and create a comfortable user interface. The CSS code was used for the overall styling of the pages in regards to our color scheme and the March Madness theme as well as making sure all the tabs and sections were in order.

5. Backend Database

We used SQLite as our database to store information. The database contains four entities: user, team, bracket, and post. Each user has the attributes username, password, and name, where username is the primary key for the user entity. The user entity has a relationship with the team entity, which has the attribute team_name and primary key team_id. Many users can belong to a team, but a user cannot belong to more than one team. The user entity will also have a relationship with the bracket entity, which has the primary key bracket_id. Each user can have multiple brackets, but each bracket can belong to one

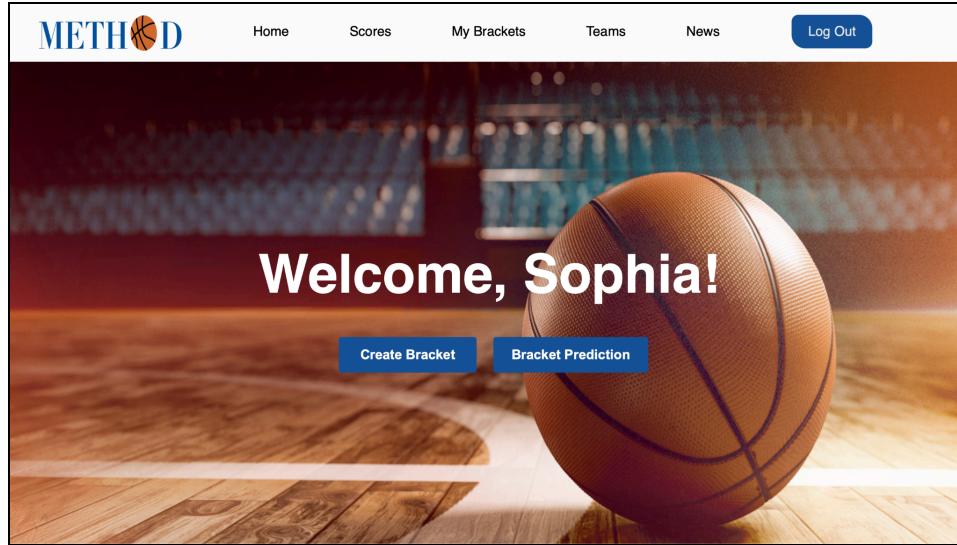
user. The Post entity has a relationship with the user entity and has the attributes body and title. The Post entity is a weak entity, so it has the primary key (user_id, title). Each user can make multiple posts, but each post can only belong to one user. The ER model for the database is shown below.



All of our queries were simple, such as finding all of the brackets that belong to a specific user or all of the users that belong to a specific team. Therefore, we did not need to write any SQL code for our queries and instead just implemented them directly in Django and python. Using Django models also allowed us to directly delete, create, and update entries in the database without writing the code in SQL.

6. Front End HTML/CSS

- Much of the front end web design was developed from scratch with HTML and CSS code and no outside frameworks or applications. Here is the design for our website home page:



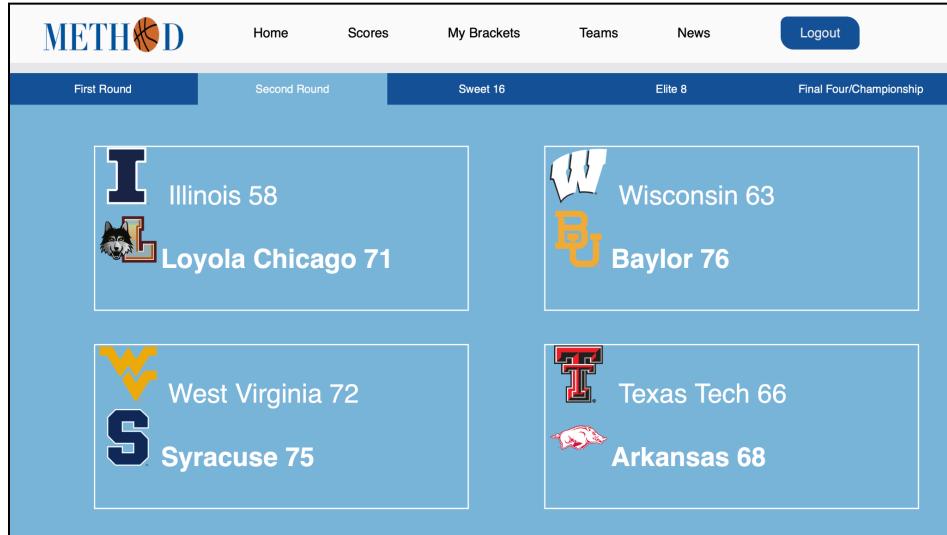
The website logo in the top left corner was designed by a member of our team.

- Our view bracket page was created using an HTML template found on <https://blog.codepen.io/2018/02/16/need-make-tournament-bracket/>. The colors and layout were adjusted to fit with the aesthetic of our other web pages. We used the Django method based views to insert the appropriate team names into the HTML file depending on the bracket the user generated.

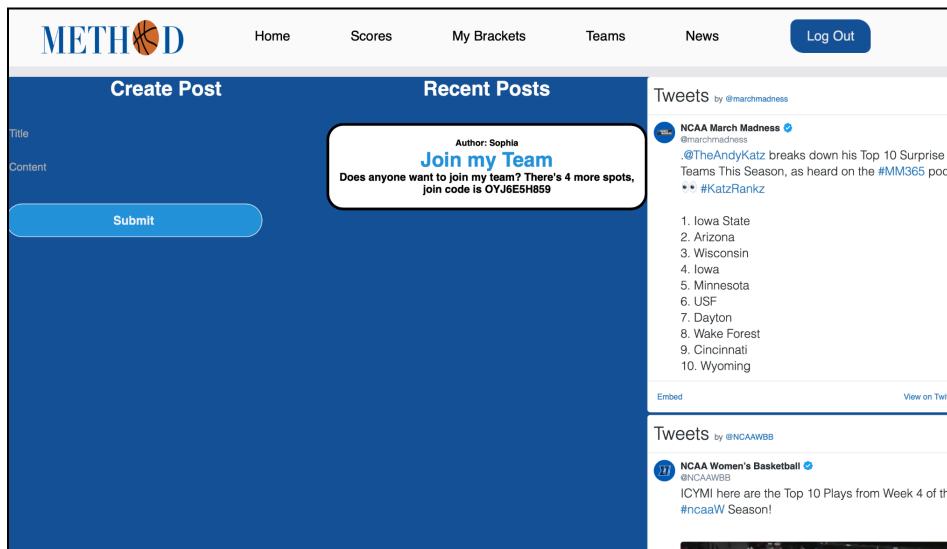


- We created the tabs on the scores page based on an HTML template for creating full page tabs: https://www.w3schools.com/howto/howto_js_full_page_tabs.asp. The team names and scores are updated automatically by web scraping from the NCAA website scores page found here <https://www.ncaa.com/march-madness-live/scores>. This web page also served as an inspiration for the layout of our scores page. The team logos are automatically

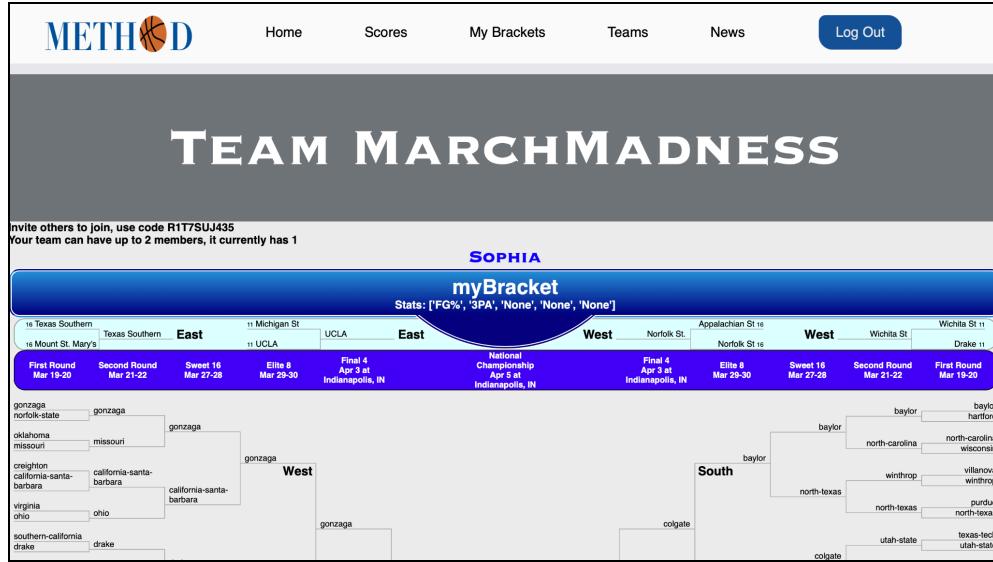
downloaded by web scraping this page as well, and then they are manually coded into the HTML file for this page.



- We also created the news page using our own CSS and HTML template. The news page uses RSS feeds from twitter in order to show tweets by @marchmadness, @ncaawbb, and @ncaa. These RSS feeds will automatically update with new information each time the page is refreshed

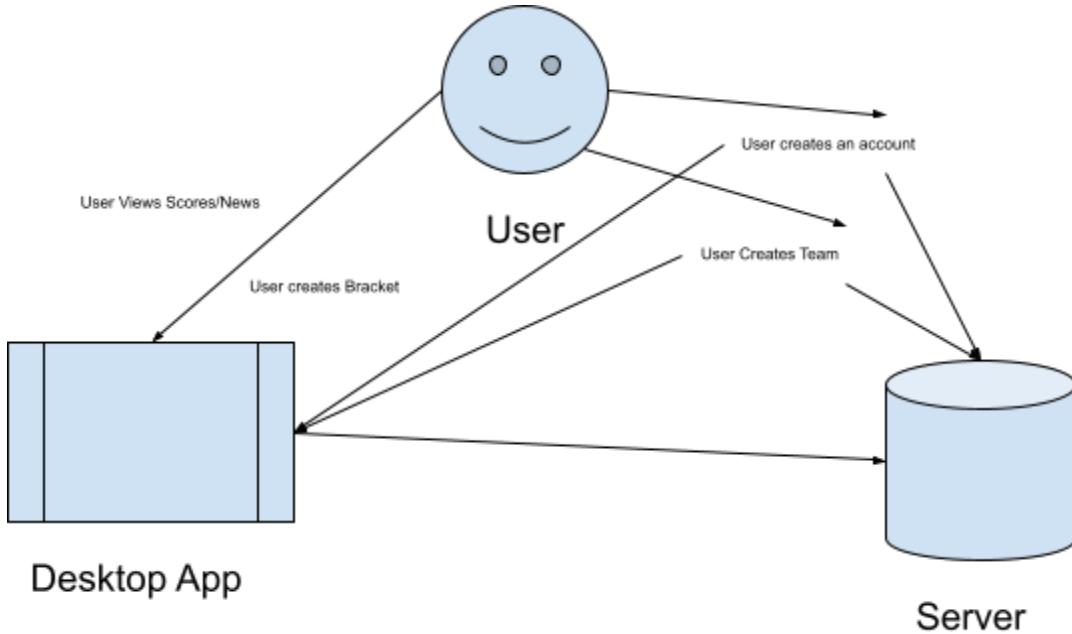


- The team page was created using our own CSS and HTML template as well. We used the Django method-based views and HTML code to display the team information. Each team member's chosen stats and bracket are displayed on the page.



7. Software Diagram

The following diagram illustrates how the various components of the website interact together.



6 Key End Users Features and Functionalities

In this section, we will discuss in greater detail the key features of our website.

6.1 Home Page

Our home page has two different views: a view for when the user is not logged in, and a view for when the user is logged in. When the user is not logged in, the home page has the navigation bar with our website logo and three tabs (“Home”, “Scores”, and “News”) for the user to navigate to other pages. There is a “login” button in the top right corner that an existing user can use to log in. The center of the home page includes a short description of the website, as well as a sign up button for users who don’t have an existing account yet.

When the user is logged in, the home page looks pretty similar. The navigation bar shows the website logo, along with five tabs (“Home”, “Scores”, “My Brackets”, “Teams” and “News”) for the user to navigate to other pages. There is a “logout” button in the top right corner which logs the user out and takes them back to the home page view for users who are not logged in. The center of the home page includes a welcome message specific to each user, as well as a create bracket button and a bracket prediction button.

6.2 User Profile

Users can view select pages of our website without logging in, but to use the main features of the website, they must create an account. The sign up page prompts users to enter their name, username, password, and confirm password. Once a user signs up, they are redirected to the login page. The login page prompts the user to enter their username and password. If a login attempt is successful, users are taken to the home page view for users who are logged in. If users without an existing account accidentally press the login button, there is a link above the “login” button that takes them to the sign up page.

We implemented constraints making sure that no two users can have the same username. If a user tries to log in with the wrong username, the website will notify them that the username does not exist. If the user enters the wrong password, the website prints an error message that the password is incorrect. If a user tries to sign up for an account with a username that is already in use, the website notifies the user that the username already exists and will not allow another account to be created under that username.

6.3 View Bracket Prediction

This page can be accessed from the “Bracket Prediction” button on the home page when a user is logged in. The page has the same navigation bar as the home page. The view bracket prediction page provides users with the combination of statistics that we found to create the most accurate bracket. It also provides the bracket that was generated from those statistics. If users scroll to the

bottom of the page, there is a “Create Your Own Bracket” button that users can use to take them to the create bracket page.

6.4 Create Bracket

Users can access the create bracket page from a button on the home page after logging in, or a button at the bottom of the view bracket prediction page. The create bracket page has the same navigation bar as the home page. The top of this page prompts users to select up to 5 statistics to create their bracket. There are 5 option selection boxes, each with 22 options to select from. The first selection box requires the user to select one of the 21 statistics listed, but the other 4 boxes provide the user with the option to select “None”. When the user is done selecting their stats, they can press the submit button, which fills in the bracket with teams. If the user doesn’t like that bracket, they can continue to change stats and press submit until they have a bracket they like. If the user scrolls down to the bottom of the page, they can enter a name for their bracket and press the “save” button to save their bracket.

If a user tries to generate a bracket without selecting either 5 stats or the “None” option, the bracket will remain blank. If they try to select the same stat for multiple option boxes, the website will print an error message telling the user to select a different stat for each category. If the user tries to save a bracket without selecting stats or pressing submit to generate a bracket, the website will notify them that they must select stats and generate a bracket before pressing save.

6.5 View Brackets

The view brackets page can be accessed from any page when the user is logged in, using the “My Brackets” tab in the navigation bar. It has the navigation bar at the top, with the same tabs as the home page. The page lists all of the users’ brackets in the order that they were created, from least recent to most recent. Each bracket has a “remove” and “edit” button underneath it. If the remove button is pressed, the bracket is deleted from the database and removed from the view brackets page. If the edit button is pressed, the website redirects the user to the edit bracket page, where they can edit their bracket. Once they have edited and saved the bracket, the updated bracket will be displayed on the view brackets page.

6.6 Edit Bracket

This page can be accessed by pressing the “edit” button on the view brackets page. The edit bracket page has a similar layout to the create bracket page. It prompts the user to change any of their statistics and generate a new bracket. The original stats are shown in each option selection

box. If the user changes any of the stats and presses the submit button, the bracket is updated. The user can continue to change stats and generate new brackets until they have a bracket they like. If the user scrolls down to the bottom of the page, they can press the “update bracket” button to save their bracket. If they want to change the name of their bracket, they can enter a new name in the box at the bottom of the page. Users have the ability to change only the name of their bracket and not the chosen stats, or vice versa. If the user is done updating their bracket, they can use the back button at the top left of the page to navigate back to the view brackets page where the updated bracket will be displayed.

6.7 View Scores

The view scores page provides users with updated results of each game in the tournament. This page has the navigation bar at the top with the same tabs as the home page. There is a second navigation bar below that with 5 tabs: “First Round”, “Second Round”, “Sweet 16”, “Elite 8”, “Final Four/Championship”. Each tab displays the results, including the score, of the games that took place in that round. This page can be viewed regardless of whether a user is logged in or not.

6.8 Team Page

The team page can be accessed from any other page with the “Teams” tab in the navigation bar. This page has the navigation bar with the same 5 tabs as the home page. This page displays four buttons: “View Team”, “Create a Team”, “Join a Team”, and “Leave Team”. The user can select any of these buttons to take them to the corresponding page. If they select the leave team button, they will be removed from their team if they were on one. If they were not on a team, nothing will happen.

6.9 Create Team

The create team page can be accessed by pressing the “Create a Team” button on the team page. This page prompts users to enter a team name, the number of members they want on the team, and the bracket they want to use for the team. The bracket can be selected using a drop down menu containing the names of all of the user’s brackets. The user can then press the submit button to create the team, or the cancel button to take them back to the team page. The submit button redirects the user back to the team page, and they can see their team information using the “View My Team” button.

6.10 Join Team

The join team page can be accessed by pressing the “Join a Team” button on the team page. This page prompts the user to enter a join code and the bracket they want to use for the team. The bracket can be selected using a drop down menu containing the names of all of the user’s brackets. The user can then press the submit button to join the team, or the cancel button to take them back to the team page. The submit button redirects the user back to the team page, and they can see their team information using the “View My Team” button.

If the user enters a join code that is not associated with any team, the website will notify them that the join code is invalid. If the user tries to join a team that already has the maximum amount of members, the website will display an error message telling the user that the team is full and will not allow the user to join.

6.11 View Team

The view team page can be accessed from the “View My Team” button on the team page. This page has the navigation bar with the same 5 tabs as the home page. On this page, users can view their team name at the top of the page. Below the team name, the join code is listed. Below that, users can see how many members their team has, as well as the maximum amount of members it can have. Then, users can see the brackets of every team member. Each bracket has the user’s name and the statistics they chose listed above it.

6.12 News Page

The news page can be accessed from any other page with the “News” tab in the navigation bar. This page has the navigation bar with the same 5 tabs as the home page. It provides users with Twitter feeds from three different March Madness related accounts. Users can also view blog posts that other users have made. This page can be viewed regardless of whether a user is logged in or not. However, if a user is logged in, they have the ability to create a blog post and post it to the website. If they are not logged in, they can only view other user’s posts.

6.13 Start Tournament

The start tournament function can be accessed by an admin. The function allows the admin to enter a date that the tournament begins. If a user is accessing the website before the tournament has begun, they have full access to the website’s functionality and features. However, if a user accesses the website after the tournament has started, some features of the website are “locked”. Once the tournament starts, the user can no longer access the “Create Bracket” button from the home page or from the view bracket prediction page. Users also can no longer edit or delete brackets, as those buttons have been removed from the view brackets page. Finally, users cannot

leave, join, or create teams after the tournament has started. The team page will instead show the user's team information, or nothing if the user never joined a team. Once the tournament has started, users will be able to view the scores of their brackets. The scores will be shown under each bracket's name on the view brackets page and on the team page.

7 Results Verification

The best way to verify our results will be to use Method for the upcoming 2022 March Madness tournament. However, since that is not currently possible, we used the previous 2021 tournament initial bracket to test the functionality of Method.

5. Accuracy of Data Collection

All locally stored data, such as data from recent tournaments or initial matchups from each year were manually tested to ensure accuracy of computed brackets. All of the data was found to be accurately stored. **PASS**

6. Verification of Bracket Construction

To verify that the bracket construction pipeline accurately computed the brackets according to inputted combinations of statistical categories, five randomly selected combinations of statistical categories were used to manually construct brackets and test against their calculated counterparts. The calculated and manually constructed brackets were in agreement. **PASS**

7. General Front End Operations

We created a “test” user account and went through the features of the website, making sure everything worked correctly. We went through and made sure that buttons and forms on each webpage worked the way we intended them to. We checked to make sure each page functioned the way we wanted it to. We tested the create bracket page by selecting stats and creating a bracket, and making sure the bracket was generated and saved to the database when the save button was pressed. We also tested to make sure that a bracket wouldn’t be saved to the database without selecting statistics and a bracket name. Similar testing was done with the teams page to make sure that teams were created correctly, and that users cannot join teams that are full. **PASS**

8. Latency

For each of the webpages, the load time was inspected in a Safari browser having a latency of 3 seconds or below. Additionally, testing of bracket construction based on recent tournament data determined that functions had a latency of at most 2 seconds when using the caching system. **PASS**

8 Future Work

In the future we would like to do more statistical analysis and better train our model. Right now, *Method* allows users to rank five different team statistics one through five. Expanding both the number and depth of attributes to rank would improve the user experience and hopefully the accuracy of the brackets. There are a few improvements we would like to make on the front end to create a more robust user experience. These improvements include a more collaborative news page that allows users to talk and message with each other. In our proposal, we stated that we would like to add automatic bracket scoring that updates as the tournament progresses. Unfortunately, we were unable to have the bracket scores update automatically throughout the tournament, since it was impossible to test this because the tournament is not currently happening. This is something we plan to implement moving forward.

9 Access to Project

Link to team GitHub Repository: <https://github.com/srh95/CSDS-395-Senior-Project>

10 Individual Reports

Isabelle Congdon

Project Responsibilities

My responsibilities during this project centered around the front end. I built the teams, blog, and news functionality for the site. Because of this I also worked a lot on the dependencies for the database as well, in order to allow for groups to be created as well as posts. I developed the following pages along with their functionalities. These pages included:

- joinTeam.html
- leaveTeam.html
- createTeam.html
- userTeams.html
- teams.html
- News.html
- userNews.html
- Database design for news/posts
- Exception testing for violations within stated pages

Along with their corresponding functions in the views.py file.

Achieving Objectives

The successful completion of my responsibilities came from reading/watching online videos about the Django framework and website usability. My previous data science classes allowed me to figure out how to set up our database to support a function for users to create/join teams as well as create posts. Python was a familiar language to me due to many of my data

science classes, but during the course of this project I was able to use it in a way that was completely new and unfamiliar.

When a user creates a team and presses submit, this creates a code that is sent and stored in the team object that other users can then use to join that same team. This code also acts as a way to link all players of a team together in order to display everyone's brackets on the teams page. In addition when a user creates/joins a team they must specify which bracket they would like to enter to compete against their friends. All of these attributes needed to be saved and connected to the user object. I was able to successfully achieve and find the best way to accomplish this through previous database classes and personal experiences.

HTML/CSS design was a brand new world for me however I was able to successfully complete my pages due to the abundance of information and examples available on the internet. Throughout the course of the project my confidence in these skills grew and I was able to eventually accomplish a much more complex/aesthetic UI for my pages than I had originally set up.

Issues

One of the biggest challenges I faced during this project was the fact that I had never used GitHub, Django, HTML/CSS, or a IDE like Pycharm/Visual Studio before. As a data science major I was used to completing 99% of my work in a Jupyter notebook and had never built anything like this before. This provided an extra level of difficulty when developing my parts but was extremely rewarding in the end. I was able to experience a dynamic, collaborative, development project. Some of the biggest issues I faced was with picking the type of model to use for Teams. While Django does have a built in group function, due to the simplicity and dependencies of our database I decided it was better to build a new custom model. This allowed me to connect to users. The biggest challenge faced, however, was assigning a connection from Bracket to User. I tried multiple versions of this relationship out in an attempt to find the best way and eventually settled on letting users pick their "favorite bracket" to be stored as an instance in the User object. This allowed me to reference and display that user's bracket on the teams page without having to parse the entire database of brackets for one that matched the same name.

Lessons Learned

The best lesson I learned from this project was how to use GitHub and how to work on a software project as a team. While it was quite intimidating at times I was eventually able to become comfortable and confident creating/merging branches and using everything it has to offer. Second, I was able to get some experience in designing web pages using HTML/CSS. While I was familiar with some aspects of HTML from web-scraping projects I had worked on in the past, this was the first time I built an entire page from scratch. This was a very enjoyable part for me and I see myself continuing to develop these skills even further after the conclusion of this project. Lastly, being able to use Python other than in the context of a Jupyter notebook was a very big challenge but has since led me to feel extremely comfortable in building

functions/dependencies. Working through the development of the teams pages as well as the news/blog was the perfect combination of things I both did and did not know and I believe I am a much more well rounded computer/data scientist now than at the beginning.

Insights

Overall this project was extremely insightful. While it first highlighted some weaknesses I may have had in my skill sets around GitHub/Web Development my team was extremely supportive in answering questions and helping each other out. I am very proud of how this project developed and how much we all learned and were able to accomplish.

Sophia Hall

Project Responsibilities

My project responsibilities consisted of designing some of the front-end of the website, as well as the backend functionality of the website. I set up the project using Django and python and used Django's default database, SQLite, to implement a majority of the database.

- Developed the front end for the two home page views using HTML and CSS (excluding the navigation bar)
- Created the front end and back end functionality for the create bracket page, view bracket page, and edit bracket page
- Created the front end and back end for the scores page and implemented web scraping to automatically update the scores
- Implemented the backend and functionality for login and signup pages
- Created tables in database for user, bracket, and teams and linked them together accordingly using foreign keys
- Created the basic functionality for the join team and create team pages (then handed off to Isabelle to finish)
- Worked extensively with Noel to merge the create bracket model with the Django project and front end for the create bracket page
- Created the bracket prediction page
- Implemented a function to lock the teams, create bracket, and edit bracket page when the tournament starts and created dual views for pages depending on whether tournament started or not
- Worked with Noel to implement the scoring function
- Extensively tested the website front end and exception cases

Achieving Objectives

I used Django and Python once before for another project, but I am still pretty new to both and had to do a decent amount of research to implement the pages I was responsible for. I was able to easily set up the project, but many of the objectives required for this project were

quite different than my previous project. With my previous project, we used CSS and HTML templates we found online. So, this was my first time actually having to write CSS and HTML code from scratch. I accomplished this by watching a lot of web development videos on YouTube, as well as reading CSS and HTML tutorials and documentation pages online.

I also implemented a majority of the tables in our database. I am currently taking a databases class, so I was able to apply my knowledge of ER diagrams, relational schemas, and SQL to create the database. I linked the teams relation to specific users by using the team's join code as a foreign key for the user relation, allowing each user to be associated with a specific team tuple in the database when they joined a team. The bracket relation was implemented similarly, with each bracket having a foreign key "user_id" connecting it to a specific user.

Issues

One of the main issues I had with this project was designing the front end for my web pages. Before working on this project, I had never used CSS before and had minimal experience with HTML. I spent a decent amount of time researching different ways to implement the layouts for the pages that I was responsible for using CSS and HTML. One of the other issues I had was just trying to plan out how the website would be used during the actual March Madness Tournament. It was hard to know how to implement some of the website features, because we couldn't exactly test how the features would work during the tournament. For example, I struggled with working on bracket scores that would automatically update as the tournament progressed. I originally wanted to use the results of web scraping for the game data to score the brackets and update the HTML files using Javascript, but it was impossible to know in what format the data would come in when the tournament was actually occurring. So, I had to settle for creating a function that would display the final score of a bracket once the tournament started. I was able to do this because we were using 2021 data, so we already knew the exact results of the tournament.

Lessons Learned

Throughout the course of this project, I learned a lot about what goes into developing a web application, such as front end development, backend functionality, and integrating the database with the web framework. As stated previously, I did not have experience with CSS and HTML. After spending the whole semester working on the front end design of the pages I was responsible for, I now feel comfortable with writing my own CSS files and integrating the code with HTML files. Web scraping was also something that was completely new to me. I was tasked with working on the view scores page, which we envisioned to automatically update the team names and scores as the tournament progressed. I was able to follow an online tutorial for using the Beautiful Soup python library and an HTML parser to pull data from HTML files. This allowed me to successfully pull the team names, game scores, and team logos from the official NCAA March Madness website. I also think I have improved my coding skills in general and learned how to handle coding with a team through the use of Github. Many of my previous projects have been medical imaging related, and most of the code I work on by myself. With this

project, I learned how to share code through Github and handle merge conflicts when working on the same parts as other members of the team.

Insights

I greatly enjoyed working on this project and feel like I gained a lot of valuable experience from it. I am a biomedical engineering and computer science double major, so in the past a lot of my coding has been more related to medical imaging research. I am planning on doing a masters in computer science, so I was hoping to get more experience with development to help me decide which area of CS I want to focus on in my masters. Through this project, I have discovered that I'm more interested in the software engineering side of things. Although I liked doing the front end design and development, I really enjoyed coding the backend and functionality of the web pages, as well as setting up the database. I also gained valuable experience with working with a team on a software project. I did not have a lot of experience working with a team this large for that long of a time period, and it was nice to see us go from just the planning stage to having a fully functional web application by the end of the semester. I am also excited to see how the website functions during the actual March Madness Tournament, and how accurate the brackets end up being with the 2022 data.

Alexis John

Project Responsibilities

My project responsibilities was contributing to front-end development of the website which consisted of the user interface. I spent the majority of my time developing the style of the website using both HTML and CSS. The bracket template was also created using HTML and CSS.

- Developed the style template for the pages
- Researched the best case user interface
- Researched connecting frontend with backend
- Worked on home and user page design and functionality
- Developed bracket design for team and user pages
- Helped developed functionality for user brackets

Achieving Objectives

In order to create an interactive and functional website I did a decent amount of research on different styling techniques and how to implement CSS with bracket design. I also referred back to prior projects I have worked on, where I worked with web design. I also spent time referring to videos online or different sites to see what was the best way to design the website to make it as user-friendly and accessible as we could.

I used Visual Studio Code with a live server to test my code and make sure the pages looked the way I wanted them to and made changes accordingly. Using a live server was a more convenient method for me to see the minor changes that needed in regards to the CSS, but once the backend was connected with the frontend running it from the server was the easiest way to implement changes.

Issues

One of the biggest challenges on the front-end team was creating a user-friendly interface that was comparable with the original NCAA March Madness Bracket challenge. HTML and CSS was something that had to be taught with lots of research throughout the process. I spent a lot of time finding ways to create a creative and functional website with functional web pages. Finding relevant resources as well as styling techniques online were helpful in assisting with creating a user-friendly website. Along with styling in general, creating a bracket template was difficult to manage in CSS, the HTML component was very feasible but styling the bracket to match the webpage was time consuming and required a number of hours of research and trial and error.

Along with the styling errors, making something unique to the original NCAA March Madness Bracket was also challenging. Most users are familiar with a regular bracket and simply choosing their teams of choice so making an application that would be worthwhile and competitive with that of the original bracket was also a minor issue that was addressed. Another issue I had was managing screen size compatibility since the code was implemented fine with pure HTML code on Visual Studio Code but it was hard to adapt to the Django framework especially when merging with the Python backend.

Lessons Learned

Throughout this project I was able to strengthen my frontend abilities and gain more insight to web design and style techniques. I was able to learn about different methods that could be easily implemented and new website design techniques. In addition, although I was not on the backend tram, I learned a lot about the algometric approach for the project. I learned about new packages and frameworks that could possibly be implemented into future projects I work on. I also learned about web scraping which is something I have not done before but now I know how to web scrape data from various websites and see how to implement it into projects. This project also helped me gain skills in working with a team and collaborating on all aspects of the project.

Insights

This was a very exciting project to be a part of because it was a topic and event I truly enjoyed. Basketball has been a huge part of my life and March Madness has always been an exciting time for me every year. Being able to create a web application different from the original NCAA bracket was an exciting experience. Having the ability to share this application with others feels accomplished and seeing how much my team has achieved as well as my own

personal contributions is truly rewarding. In addition, I learned a great deal more about software engineering and new frameworks and applications I may use in the future that I now have exposure to.

Noel Mathew

Project Responsibilities

My project responsibilities were centered around creating the backend functionality. These responsibilities started with establishing the designs of how data would be collected via web scraping and then processed to be used in the website to construct brackets and ended with implementation of this through functions. After testing these functions, they were modified during the merging process to be displayed on the user interface. Additionally, I performed an analysis on which statistical categories were the most predictive in constructing a bracket through construction of a binary classification predictor that utilized a logistic regression model with each statistic as a feature and then extracting the coefficients for this model to determine feature importance.

Specifics of responsibilities:

- Data Exploration of Sports-Reference site
- Data Collection via Web Scraping Scripts
- Conversion of Web Scraping Scripts to Functions
- Data Cleaning and Formatting
- Latency testing of functions directly reliant on web scraping
- Design and Implementation of caching system for switching between web scraping and using locally stored data and alteration of relevant functions
- Design and Implementation of algorithm for comparison of teams
- Bracket Construction Functionality based on User-Selected Stats using Team Comparison Algorithm
- Research and Implementation of ML predictor for determining most predictive stats
- Editing function outputs to match front-end specifications (worked with Sophia extensively on this)

Achieving Objectives

The process of achieving my responsibilities was primarily done through many hours of cycling through the steps of research, solution design, and trial-and-error implementation until a working solution was found. Oftentimes, many of my initial solutions were revisited after the initial production cycle of the solution due to various reasons. To promote efficiency during this revisit process thorough documentation was made a priority.

The research phase of my work cycle was primarily in the form of searching online for potential solutions to my problems and implementing online tutorials to fully understand the material before consideration of viability within the project necessities during the design phase. While it is hard to identify specific courses that provided the exact knowledge for the solutions I implemented, the skills of online research and solution design developed through my coursework were highly integral to achieving my objectives.

Issues

One of the biggest issues that occurred during the development of the website was the high latency of the initial functions used for construction of brackets based on user-selected stats. Initially, the team comparison function was constructed to acquire the necessary data using web scraping. The latency of this function in isolation was low but when the function was repeated 62 times in succession on all of the teams within a bracket, the total runtime of bracket construction ended up being over 10 minutes per bracket. After exploring each component of the pipeline, it was determined that the issue was the rate limiting technologies present on sports-reference.com that limited the number of automated requests our functions could make in succession. To overcome this issue, I designed and implemented a caching system so that our functions would primarily utilize locally stored CSV files of the necessary data from tournaments in recent years and use web scraping otherwise. Construction of these CSV files required significant time spent doing data exploration before using web scraping for collecting the data and both manual and automated data cleaning afterwards. Subsequently, the functions had to be altered to utilize the aforementioned caching system.

Lessons Learned

Through my time spent on this project, I was able to gain firsthand experience performing some of the tasks of a data scientist or engineer such as data exploration or cleaning. This was the first time I had been exposed to these work processes which lead to learning how to use commonly used data scraping and processing tools such as HTTP requests and pandas. Additionally, I learned about different types of machine learning classifiers to implement my binary classification model along with how to implement it using the sklearn library. Apart from the hard skills that I developed while working on this project, I was able to exercise and improve soft skills such as data design and technical communication.

Insights

Working on this project was extremely insightful towards what I want my career to be. Through working on the various phases of construction of a data pipeline, this helped me decide that I want to pursue a career in data science or engineering. Working on the backend and collaborating with members of the front end team was both a challenging and rewarding experience.

Emily Tibbens

Project Responsibilities

My project responsibilities primarily sided with back-end development, data management, and document preparation. Below is a more specific outline of my personal responsibilities.

- Create a comprehensive list of only current NCAA Men's Division I programs
- Collect and clean team stat data for the previous five seasons
- Make a csv of team stats of all teams that have competed in the March Madness tournament in the past 5 years (except 2020 due to COVID-19) to store locally
- Identify how many statistics to use
- Create Method logo
- Assisting in the results verification process and bracket creation for the 2021 season

Achieving Objectives

To achieve my responsibilities, I relied heavily on knowledge I have gained in previous classes. As a data science major, I have experience coding in python, but I have taken few advanced coding courses. On the back-end team I leveraged the things I did know to be as helpful and productive as possible. To clean the data, I implemented the functions Noel built along with the list of current division I schools I made to eliminate unnecessary data and obtain the correct team statistics for the past five years. Most of this was done using the python packages pandas and numpy. Noel and I primarily wrote the python scripts in Google Collaboratory as it allowed us to share and both work on the scripts. For the functions and parts of code I didn't know how to initially implement, I made sure to at least assist in the research of packages and implementation techniques.

Issues

The biggest issue I encountered while working on this project was the statistical analysis. Our objectives weren't exceptionally clear and that made it difficult to figure out what was in scope and what wasn't. To begin, using sklearn and the feature selection model, I found that using only one of the 23 total statistics, to predict if a team will advance is the most accurate model. This created a new issue because letting users only choose one statistic will only create 23 distinct brackets. After contemplating this for a while, our team decided to increase the number of statistics. Therefore, we increased the number to 5.

Lessons Learned

While I mostly stayed in my comfort zone and mostly cleaned and analyzed data, I was able to assist and learn many new things. I have only ever built a website with a very simple HTML. While I didn't help much on front end development, I was able to learn while at team meetings. Being on the back-end team as someone who wouldn't

consider themselves a coder was challenging, and beneficial as I will most likely be coding and implementing a lot of things I learned throughout the duration of this project post graduation. This project also helped me gain experience working with a team, and working together to meet deadlines.

Insights

This project was very enjoyable, and it was fun to be able to take part in building something that I myself would use. It was also very rewarding to be able to see the part that I helped with assist in developing something that people can actually use. While I am unlikely to have to build something in my future career. The individual work, and experience I gained while being a part of this project will no doubt be beneficial in the long run.