For my project, I used a number of college basketball stats to try and predict NCAA tournament success. I made a neural network, with the X being a collection of relevant team stats such as Adjusted Offensive and Defensive Efficiency, three point shooting efficiency, as well as simple ones such as wins. The y for my neural network was the number of games they played in the NCAA, meaning the more games they played the more successful the team was. I chose games played instead of wins to differentiate teams that didn't make the tournament from teams that lost in the first round. My goal was to create an effective model and then apply it to the set of stats from teams during the 2020 season, when there was no tournament, to try and predict the outcome for those teams. Of course since there was no way to calculate the accuracy of the model with the 2020 data, I also ran the model through a k-fold cross validation method to estimate the accuracy of it.

I started off the project by writing the methods within the 'data_setup' module to get a usable version of my data within rust. I uploaded the data as a csv, so my 'read_csv' method was used to read the files and turn them into rather complicated hashmaps. I would later make a method that transformed these hashmaps into arrays for use in the model, but I kept the initial hashmap format to allow the ability to grab specific teams' stats at ease. I later created a test for this method and the hashmap it produces. The hashmap created by this method contained a tuple of 2 vectors as its values, one of which contained info on the team such as tournament seed and athletic conference, while the other contained all of the stats. I used an iterator and closure to split the csv into these two vectors. The 'create_x' and 'create_y' methods transform the given hashmap into arrays, using another method 'sort_hash' to ensure they contain a universal order. I did use chatGPT to help with the helper function 'sort_hash' since the code for sorting a vector made little sense to me: https://chatgpt.com/share/675e7aa7-2958-8007-91e4-539c5ab5a1df. The create_x function also implements z-score normalization on the dataset to prepare for the neural network. The create_y method was used to create an array that contained the number of tournament games played by each team.

As for the neural network, I created a very similar one to the one I used in homework 9, changing spots where it was necessary and altering the parameters to improve results. I then created the 'k_folds_cv' method to split up the training dataset into k folds and apply the NN to it. In this method, I shuffled the indices of the dataset using the rand crate, then found the '.select()' method on the ndarray code book to allow me to grab specific rows from the array:

https://docs.rs/ndarray/latest/ndarray/struct.ArrayBase.html#method.select. After creating k different NNs and testing the accuracy of them, the method computed the average accuracy and returned it. The last function I made was the 'apply_model' method which simply trained a NN and tested it on a set of data without calculating the accuracy, which I used to predict the 2020 tournament outcome based on the incomplete season.

The model I created ended up being very inaccurate and could not create a good prediction for the 2020 data. The model accuracy produced by the k-fold cross validation typically hovered between 5 and 15 percent. Despite there likely being issues within the creation of my NN, I do believe a lot of the blame for the lack of accuracy in my model can be attributed to the nature of the craziness of the NCAA tournament and the lack of strong relationship between specific stats and tournament success.