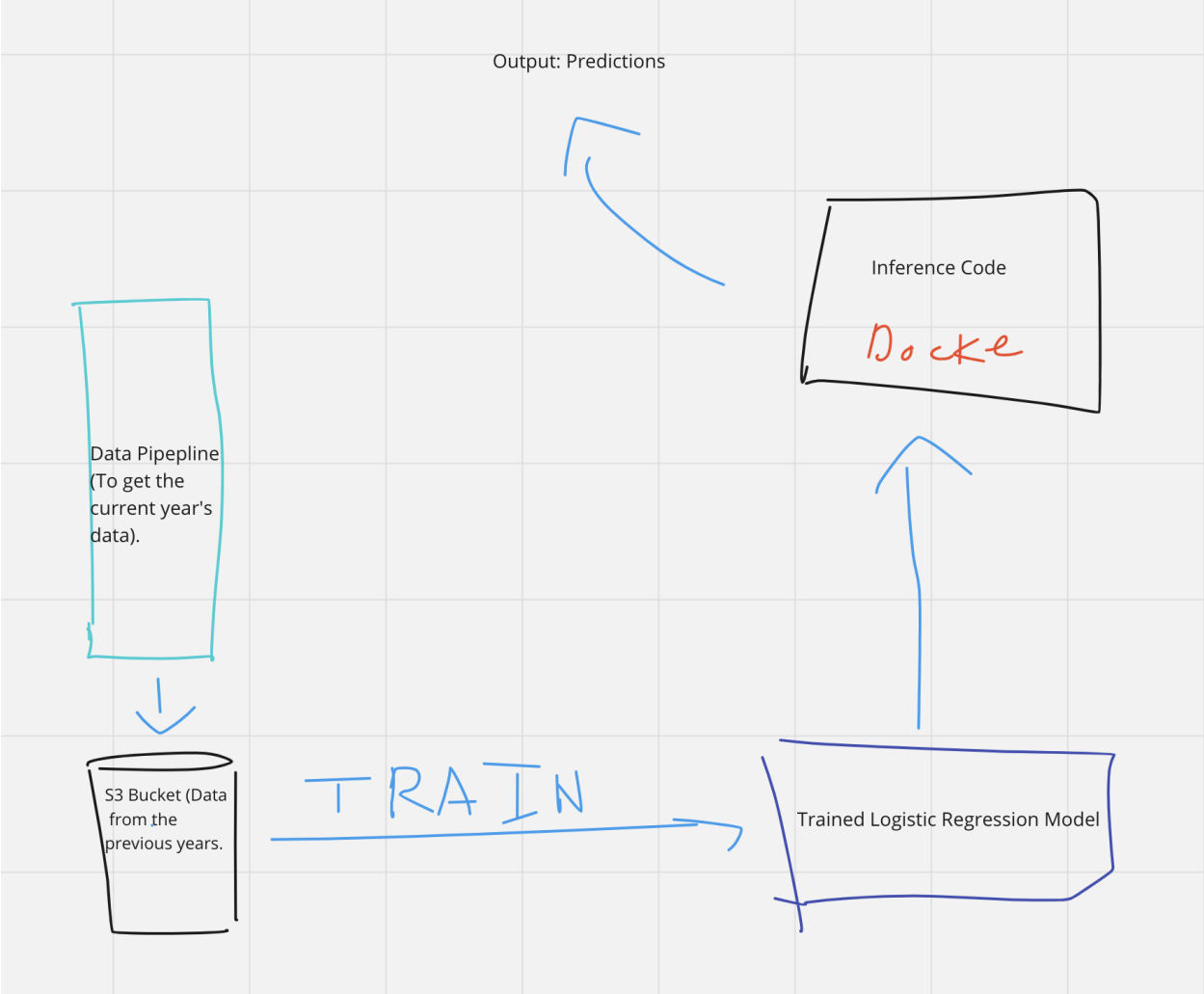


Above is my deployment architecture. It may not seem much, but it is based on an article on how to predict March Madness games. Here is the link to Part 1 and Part 2. I have also attached the Jupyter Notebook that can be found in this folder. Rather than explain why I chose to follow this exercise, I will explain the steps on how it is done - a summary of the article.

First, we will create an S3 bucket to do our work. This is where we will store our data. Next, we will create a Jupyter Notebook instance to where we will do our modeling and testing. Afterwards we will just copy our code in Step 9, but we will change our model. We will still use

the Logistic Regression approach, but we will use the built-in one. The tutorial uses XGBoost, but the process is still the same. This should not take long aside from the setup, so after testing and showing some results, it is time for the deployment part.

First we will form the CloudFormation pre-work and move around our files. Then, we will use a created EC2 key pair to start building the application. We then match it to 'march_madness'. Now we execute the CloudFormation Script with the new files we created in the first step of deployment. After fitting in the appropriate parameters to the stack, we will add our tags that we created in Part 1. Afterwards, we need to change our application so that it can handle new data. And of course, the last step is to test the website.

**Pre-Deployment Checklist**

1) Problem - How do we create the perfect March Madness bracket? We could just coin-flip this or pick the higher seed, but we know that is not going to win. Since it has never been done before due to the astronomical odds of even coming close, you would get a sense of pride and accomplishment when you do. If you are a college basketball fanatic who watches all Division I games, then you do not exist. Only that kind of person would have a chance at predicting the whole tournament with absolute perfection.
2) Data - Our data contains all the regular season games played in a given year along with the March Madness tournament that year. We also have important information on every team which includes their conference. We designate a TeamID for every team which can be easily translated among the datasets.

3) Algorithm - We used a Logistic Regression model with the default parameters. Given select features and every game played in that regular season including tourney play, we train that model using 70% of that data with 30% as our validation set. We then use that model to predict the tournament.
4) Results - Well, we check if hyperparameter tuning will actually help. Since it did not, we use the basic one. We viewed the results using past tournaments and found out that it hovered around 55% accuracy. In depth, we saw that Round 1 predictions were not the greatest, but the model has done well in predicting the Final 4.