Mamba Sports Analytics
Project Proposal
James Armstrong, Ryan D'Mello, Jason Li, Jesse Li

Goal

For our project, our group would like to analyze odds and projections for various sports in order to develop some sort of method/system that can consistently "beat the odds." By gathering data for current and past official game lines (such as point spread) and taking into account other factors, we hope to be able to come up with a best prediction for each game.

How we are going to achieve this:

Our methodology will require aggregating prediction data from different web sources with the goal of making a "composite" prediction from these sources. We may also introduce our own analysis in order to improve these projections. In order to achieve this, we will need to centralize all data from the sources through webpage crawling, accessing site APIs, or importing .csv files when available. We will then use these data to make many types of potential predictions, including (but not limited to) game outcome and scores, individual player statistics, and season outcomes. Once the data are collected, we will be able to investigate what potential statistical methods we can employ to come up with a historically-validated model. We seek to be able to consistently beat publically available odds, not for the purpose of gambling but rather in order to have a way of testing our models against a "gold standard" for sports projections that the public is not supposed to be able to "beat."

Regarding the analytics side of the project, We plan on utilizing different Python libraries that fulfil different analytics capabilities. For example, if we plan on performing any Natural Language Processing (NLP) task, we might use NLTK and Gensim. Moreover, if we want to implement any form of machine learning or deep learning, we could gravitate towards Scikit-learn, Keras, Tensorflow, and PyTorch. For example, if we want to make a text classifier from scratch in order to interpret live play-by-play data, making a recurrent neural network (RNN), specifically a LSTM, would likely be the best choice. Furthermore, we could also dive into question-answering, a hot topic in NLP, in which we would likely have to train our model on a large dataset like Stanford's SQUAD.

To make sure our application works if we go down the web-application route, we can use a web driver like Selenium to run automated tests.

Where we are getting our data from:

While the prediction data sources included in our final model remain to be seen, we will begin investigating and historically-validating projection models from websites such as FiveThirtyEight.com (which conveniently makes all data available publically in an online repository) and espn.com. We will then need to also use data from VegasInsider.com, which reports sports betting information from multiple reputable sportsbooks, for comparison with our own predictions, once we have created our composite prediction.