Section One:

Name, Programme, Year: Michael Buchar, Software Engineering, U0

Why did you choose MAIS 202? I wanted to learn more about machine learning and its applications in our society. I received a small introduction to this topic in my high school and strived to learn more about it.

Hobbies: Rocketry, Cycling, Table Tennis, Finance

Goals for the Future: To explore as many opportunities as possible and to stay creative.

Section Two (Medium Blog):

As a part of my MAIS 202 final project, I applied machine learning to sports, specifically soccer, or football as Europeans like to call it. My program predicts the outcome (win, draw, loss) of upcoming Bundesliga (1. German football league) and Premier League (1. English football league) games using six machine learning models. I have achieved a predicted accuracy score of around 70%, yet I am currently waiting for actual games to be played on March 30th to validate these results.

Regarding my inspiration for this project, I am an avid soccer fan and follow these two leagues with passion. I have numerous friends who bet on sports games and I wanted to analyze if the outcomes of this unpredictable sport can be somewhat calculated. However, the beauty of football is that it is impossible to predict a hundred percent of the time the outcome of a game and more often than not underdogs win. For instance, in season 2015-2016, Leicester City won the entire Premier League even though they started the season as complete outsiders. This project stems from my curiosity to discover how well can artificial intelligence predict the outcomes of sports games and, additionally, whether machine learning can be used for betting purposes.

The first obstacle one encounters when undertaking any machine learning project is finding a suitable dataset. After conducting extensive research, I found a valuable website (*http://www.football-data.co.uk*) that stores all the data and analysis needed for the prediction algorithm. It includes full-time results, half-time results, shots, shots on target, fouls, corners, bookings, betting odds and many more. Specific datasets used in my programs can be found in my GitHub repository (link at the end of this blog). Concerning machine learning libraries, I used the scikit-learn library (*https://scikit-learn.org/*), which is likewise very convenient. I decided to take the approach of training my data on a bunch of models and then evaluating their results. Because this is a classification problem, I decided to use Logistic Regression, XGB Classifier, K-Neighbors Classifier, Linear SVC, Random Forest Classifier, and Multinomial Naive Bayes. Moreover, NumPy, Pandas, Matplotlib, and similar libraries confirmed to be helpful for data manipulation and visualizations.

Now let's dismantle my algorithms. Firstly, I have to mention that I am rather new to the language of Python and data manipulation took me longer than I would have liked; I had a somewhat noisy dataset. Secondly, preprocessing the data is utterly important and it is crucial

to input significant features in the prediction algorithm. One of the problems I faced was figuring out how to represent the “strength of a team.” Initially, I wanted to use the position in the table as a statistic, but my research showed that features called *Home/Away Attacking/Defensive Strength* prove to be more effective in this case. Taking *Home Attacking Strength* as an example, it is calculated as the sum of total home goals scored divided by the number of games at home (assuming that the number of games at home is equal to the number of games away for simplicity) divided by the average of the total of home goals scored.

Furthermore, another challenging part of this project was implementing the past n matches statistics since a team’s performance is correlated to how well they did in the past games. The implementation takes into account past shots, corners and goals in the last n games. After that, the program computes which number it should use as n (number of past games) in order to maximize the accuracy score. Moreover, I implemented a home team advantage by subtracting the away statistic from the home statistic creating features called *Past Goal/Corner/Shot Difference*. This caused a problem for the Multinomial Naive Bayes model since it does not take negative values as inputs. Last but not least, I decided that the program itself optimizes each model’s parameters (number of estimators, number of neighbors, etc...) to maximize the accuracy score.

My favorite part of the project came after all the hard work, which is predicting the outcome of future games and comparing it to my friends’ forecasts. In addition to that, I am planning on adding additional improvements over the summer such as adding betting odds as inputs and comparing my predictions with Google. I am curious to see if any of my models will do better than my friends, who are long-term fans and know the leagues quite well. The match day is on March 30th.

GitHub repository: **https://github.com/michaelbuchar/Bundesliga-Predictor**