This report will detail the aspects of a machine learning/data analytics project completed for EE551 – Engineering Programming: Python. The project involves utilizing a dataset to create multiple machine learning models, analyzing the results and drawing any pertinent conclusions. To start, the dataset was downloaded from Kaggle1. The dataset contains statistics for the first 10 minutes of approximately 10,000 games.2 There are 19 features per team (38 in total for the two teams) and the target value (‘blueWins’ being 1 if the blue team won or 0 if the blue team lost). The goal will be to build machine learning models which can predict the outcome of the game given the aforementioned features at the 10-minute mark.

To begin, the data was downloaded into a CSV file which will be read using the pandas Python library. Then, the initial exploratory data analytics was completed to understand the main characteristics of the dataset. Many features were removed from the set due to being repetitive or unnecessary. The game ID, for example, will not aid in predictions as it is an entirely unique token given as a specific game identifier. Furthermore, a feature such as ‘redFirstBlood’ will not add any significance given that it will be the opposite of ‘blueFirstBlood’ (if the blue team has the first kill of the match, the red team cannot also have it). Due to the nature of the game being blue team against red team, it can be assumed that some features may be highly correlated and as such a correlation matrix was created. This led to the additional removal of these features and the creation of a final dataframe for use in the machine learning aspect of the project.

The first step for the machine learning portion was to split the dataframe between testing and training data. The training data will be used to actually train the algorithms, while the testing data can be utilized to understand the model’s accuracy. Three models were then created (with scikit-learn3) using three algorithms: decision tree, random forest, and artificial neural network. The decision tree classifier creates a tree-like structure of decisions and possible outcomes and the random forest algorithm is an ensemble learning method that creates multiple decision trees. Random forests are used to combat the tendency of a single decision tree to over-fit data. The neural network is composed of an input layer, an output layer, and zero or more hidden layers which loosely resemble the biological neurons in a human brain. These models were created firstly with the default parameters (measuring the time for creation) and then a function called ‘GridSearchCV’ was used to test various possible parameter combinations. The initial and final accuracies were recorded for further analysis.

After implementing the models the final step is to analyze the results obtained. To understand feature importance, a library called SHAP4 is used. This library finds the average impact each feature had on the output model magnitude. Using this, it can be seen that the features which had the most impact are ‘blueGoldDiff’ and ‘blueExperienceDiff’. A simple conclusion which could be inferred is that in order for the blue team to win the game, the team members should attempt to earn as much gold as possible while limiting the amount of gold earned by the red team (to increase the differential). Furthermore, the initial and final accuracies and the time for model creation were also analyzed. A simple chart was created using plotly5 as a visualization tool to aid in comparison. It can be seen that the increase in accuracy was extremely high for all models after altering the parameters, specifically for the neural network which saw an almost 15% increase in model accuracy. It is also important to understand the time taken for creation as different projects may have different requirements. The artificial neural network, for example, took approximately 1 second to create, while the decision tree took less than 100ms, a 10-fold difference. The accuracy for the neural network was not much higher than the accuracy for the other two models, meaning the excess time spent was not well justified.

**Sources**

1. <https://www.kaggle.com/>
2. <https://www.kaggle.com/bobbyscience/league-of-legends-diamond-ranked-games-10-min>
3. <https://scikit-learn.org/stable/index.html>
4. <https://github.com/slundberg/shap>
5. <https://plotly.com/>