NFL Money Line Gambling Model

Leo Dueker

Project Overview:

This project will be using machine learning to predict money line bet winners in NFL games. The goal behind the project is to create a model which produces a long-term profit on NFL money line (straight-up winner) bets. The model will be using classifier algorithms to predict the winners of each game and then will place a theoretical $10 money line bet on the selected team depending on win probability and pick confidence. Qualified games will be selected for training the model from 1990 to 2016 (~7000 games), and testing will be done on games from 2017-2022 (~950 games). Once the testing on the 4 years of games has been done, a calculation of gross winnings, money placed, and net winnings will be done, assuming that every bet taken is $10.

Dataset and Cleaning:

Dataset: https://www.kaggle.com/datasets/tobycrabtree/nfl-scores-and-betting-data

The dataset is a collection of NFL game and gambling data from pro-football-reference, espn, nfl weather, and a few other databases. The dataset includes a variety of information including date of game, scores, point spreads, over/under lines, stadium info, weather, and more.

The dataset is fairly clean as it is, with only minor changes being required for this project's scope. Removing all games with missing fields, narrowing down the dataset to only games after 1990, changing data types, and fixing spot errors in the dataset were all done in order to clean the dataset. Extra columns (features) were created from the cleaned dataset which would be used later in the model selection and creation.

Feature Manipulation and Selection:

Initially, the dataset had 19 columns of data. Of these 19 columns, 9 were selected as predictor variables (features) and 1 was selected as the output variable. From the initial 9 features, 4 were selected using recursive feature elimination (RFE) in order to provide the most accurate and insightful model. The 4 features selected are: spread favorite, home favorite (if the home team is favored), home team average point differential, and away team average point differential.

Model Selection:

To start, 7 popular classification algorithms were tested with the data, and the 3 best performing, by ROC_AUC scoring, were chosen for the model. The 3 best performing algorithms are: Logistic Regression, XGBoost, and Decision Tree Classifier. From the 3 algorithms, a voting classifier is used to create the best possible model from the 3 algorithms selected.

Model Results:

After testing the model, the model correctly predicted 450 bets out of 637 bets made, on 935 possible games. Using the average money line odds for a given NFL point spread, the winnings are calculated based on bets won and money placed. The model won $25249.72 after placing $6370.00 in moneyline bets, resulting in a net profit of $18879.72 over 4 years of games. After the model was created, a week-by-week summary of the bets placed, bets won/lost, and winnings was created to summarize results.

From the summary dataframe, one can see that most weeks resulted in fairly reasonable returns/losses, with most weeks being +/- $40 for the week. Most of the profit was gained from just a few select weeks, with some weeks returning as much as $2000 by correctly selecting 11 out of 11 bets for the week. This trend suggests the importance of the model being a long-term use, because the large profit weeks might only be seen after many weeks of reasonable gains/losses. It is also important to note that all of the large returns from individual bets came from significant at-home underdogs, which suggests the importance of having a model that also bets on underdogs, not just teams with calculated high probabilities to win.

Project Summary and Conclusion:

Overall, the project was a success because a model was created to predict NFL money line bets and produce a profit over a long period of time. In order to improve the model in the future, one could implement usage of the “elo” rating system, which uses team and player grading to predict game outcomes. The model could also include something to handle spread betting, to maximize profit gained and bet accuracy. In order for the project to ever have true practical use, it would have to include a user-friendly method to predict future games and produce results. This would likely include wrapping the model and deploying as a web application using software like Flask and Docker. In conclusion, the model was a success because the solution met the scope and goals of the project, but there are many improvements that could be made to make the model practical.