# Predictive Analytics in NFL Betting: A Machine Learning Approach

Samuel Salama, Blake Ziegler, Cole Bradfield, Aaron Eads

## Abstract

Our project aims to investigate whether a neural network is capable of accurately predicting the scores of NFL games, and how it can be implemented into a comprehensive betting system to identify profitable opportunities.

**Question:** Can an artificial neural network predict or approximate the scores and outcomes of NFL games effectively enough to enhance a betting strategy?

## Introduction

The National Football League or NFL, is a major sport organization within the US. Our project intends to predict NFL games for optimal betting odds and the amount of money to bet for a game. Using data from previous games from past years, we focus on these main data points: the time and the season of a game, who the teams are and if they are home or away and each teams score, and with the score the spread of points and how over/under the score is, the favored team,, and then what the weather was like, with temperatures and wind speed.

Compared to earlier models that may use individual player scoring and how other betting sites are betting.(1) Ours will be more useable in the long term since it doesn't rely on anything but previous data and non individual data. Also we use coding with neural networks over the traditional method of using google excel and pay for data.(1)

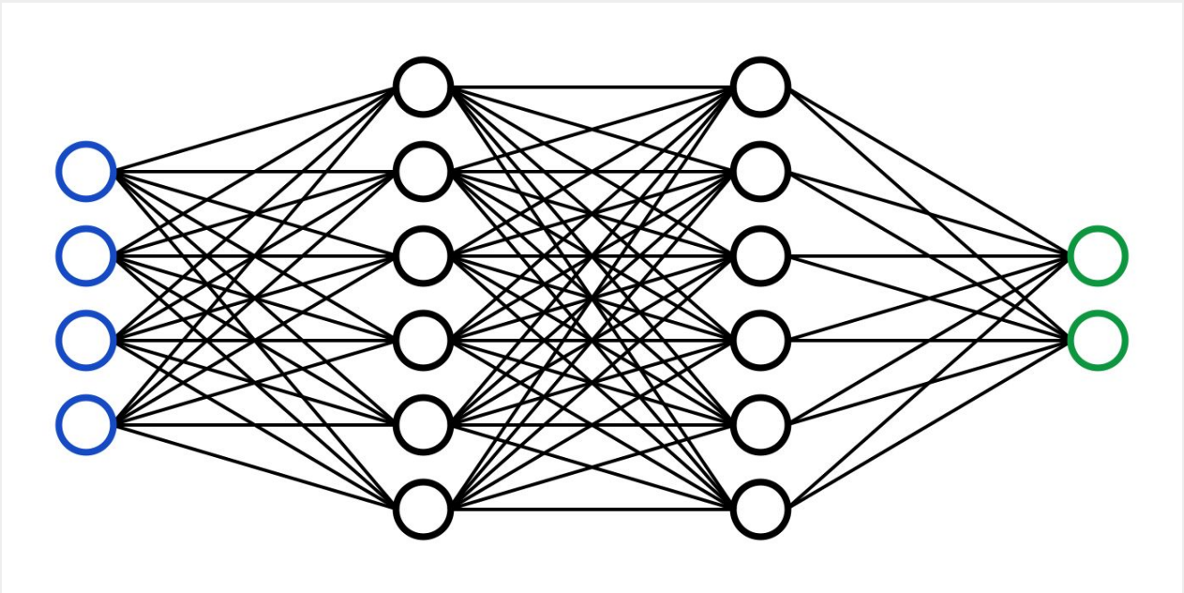
## Methodology

### Data & Feature Engineering

Data prior to the 2010 NFL season was excluded to focus on recent trends, and incomplete or redundant columns were removed to streamline the dataset. Missing values in weather-related features were imputed using mean values grouped by stadium and schedule week. Non-predictive columns, such as stadium identifiers, were dropped to reduce noise.

Core predictive features included game-level metrics like total points, point differentials, and a binary flag for home team favorites. Team-specific metrics, such as rolling averages for points scored and allowed, win streaks, and spread cover rates, were also created. Dynamic power ratings were updated throughout the season to reflect team performance and validated against point differentials for predictive reliability.

### Neural Network



**Primary Components:**

- Initialization
- Forward Propagation
- Backward Propagation
- Activation
- Training
- Prediction

**Algorithms:**
He Initialization
Mean Squared Error (Cost)
Gradient Descent
ReLU Activation

**Libraries:**
NumPy
Pandas
MatPlotLib

Our neural network utilizes a 9-32-16-1 structure. We found better results with a high number of hidden layers because of the high dimensional nature of the data. Additionally, the learning rate was set at a standard 0.1, and ReLU was the chosen activation function was used.

### Betting System

Our system combines data processing, predictive modeling, and risk management to identify profitable betting opportunities. Game data is processed to extract features like power ratings, recent team performance, and spread calculations, which are fed into a neural network to predict game spreads.

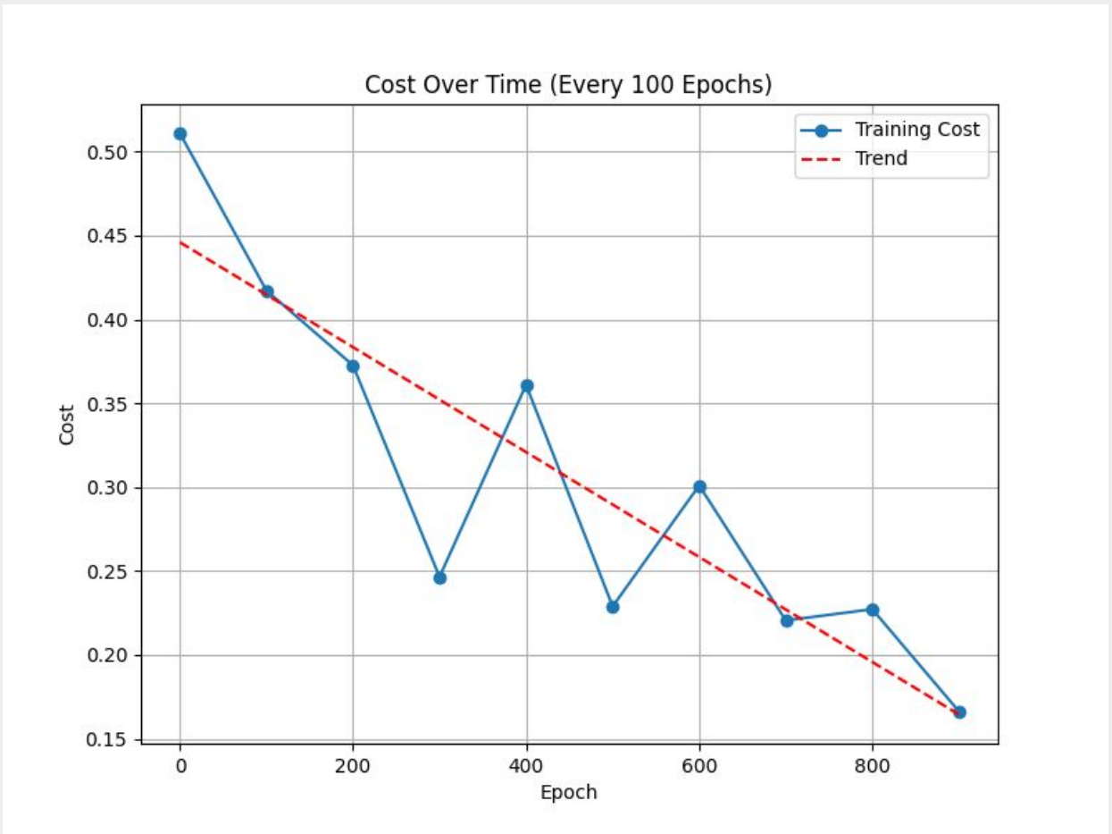
Predictions are compared to market spreads to calculate edges, with bets validated based on edge size, spread ranges, and team performance. Stake sizes are calculated dynamically as a percentage of the bankroll, adjusted for edge strength, while ensuring risk controls with minimum and maximum limits.

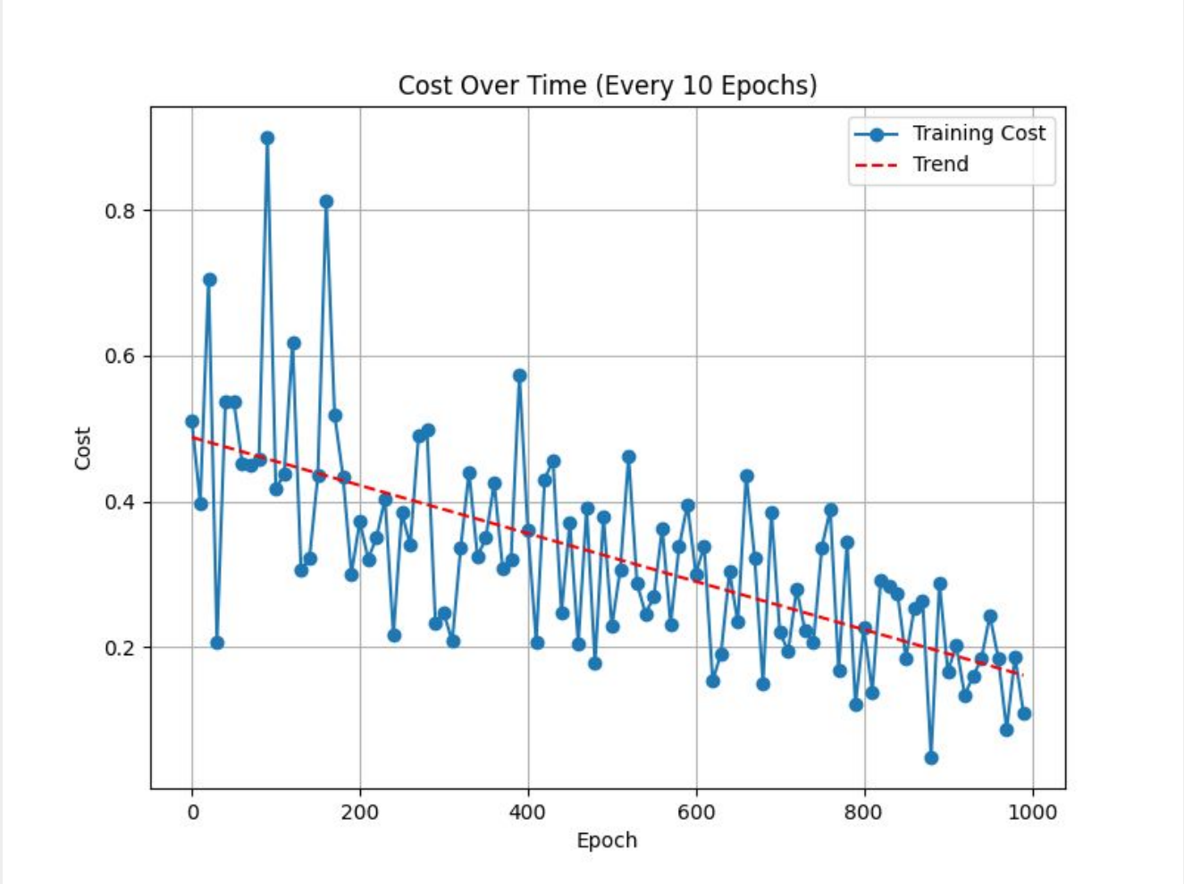
Profit and loss are evaluated using game outcomes and sportsbook odds (-110), providing a systematic, data-driven approach to NFL betting.

## Results

Our results show both that both the neural network and its broader implementations in the betting system made accurate and informed conclusions based on the data.

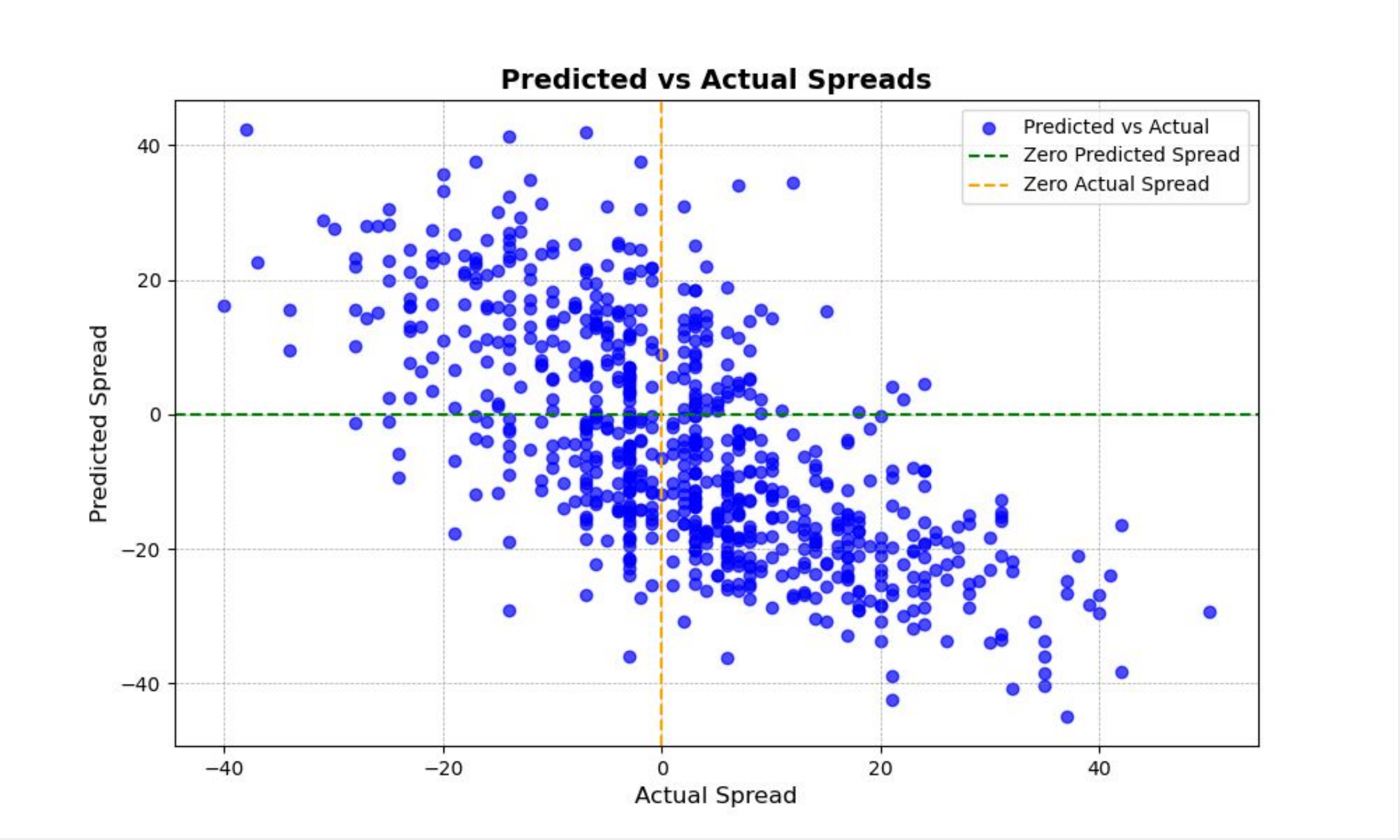
### Neural Network Tests





```
Random Test Results:
Predicted: -4.18, Actual: -4.00
Predicted: -1.32, Actual: -5.50
Predicted: -4.07, Actual: -3.50
Predicted: -4.05, Actual: -6.00
Predicted: -12.67, Actual: -12.50
Predicted: -10.09, Actual: -10.50
Predicted: -2.50, Actual: -8.00
Predicted: -7.38, Actual: -7.50
RMSE Score: 2.51
```

### Betting System Tests



The system analyzed 500 NFL games spanning multiple seasons and identified 59 value betting opportunities that met the defined criteria for edge, spread, and team performance. All 59 bets were placed, resulting in 39 wins and a 66.10% win rate. The system generated a total profit of $4334.54 with an ROI of 29.00%, outperforming market benchmarks (e.g., underdog cover rate of 50.6%). The average bet size was $253.37, with a total of $14,948.73 staked across all bets after starting with $10,000. These results demonstrate the system's ability to identify profitable opportunities in NFL betting.

## Conclusion

Overall, our model has improved the accuracy of sport betting with ours having 66.10% win rate over old sport betting rules from 1981 to 2000 which was on average around 50%(2), so we have a a 16% increase in accuracy compared to previous betting rules. We also have a prediction accuracy within 3 points which is our B-goal outlined in the proposal. Our core milestones, including the implementation of the feature engineering pipeline, neural network with backpropagation, and a dynamic power rating system, were completed. Our process of finding the rolling averages, streaks, and spread validations allowed for the model ot consistently identify value betting opportunities.

Improvements to the code, would first involve adding more data points from previous games, as that will help neural network. Improvements within the neural network regarding both forward and backward propagation could have been made.

In summary, our model met and overpassed the key objectives we set at the beginning,achieving a measurable improvement in sports betting accuracy and profitability.

## References

(1) https://oddsjam.com/betting-education/how-to-build-an-nfl-betting-model-in-excel
(2) https://www.researchgate.net/figure/Success-Rates-of-Simple-Betting-Rules-in-the-NFL-This-table-shows-the-success-rate-of_tbl2_24075972
(3) https://www.geeksforgeeks.org/implementation-of-neural-network-from-scratch-using-numpy/
(4) https://realpython.com/python-ai-neural-network/