

NFL Prediction

A Deep Learning Model

Introduction

In this project, we implemented a deep learning model to predict the number of wins an NFL team will achieve in a season. This approach uses neural networks to analyze, and train based of historical data and then predict the wins. While sports predictions are commonly used, often in combination with sports betting, these Models are not accessible to the public. Therefore, me and my partner, we decided to implement our own Model with the keras/tensorflow library.

Dataset

To train our Deep Learning Model we had to find historical data for all NFL teams. On the Pro Reference Football website, we were able to find offense and defense statistics for each NFL team in the last 50-60 seasons.

We used 13 input features for each NFL Team in a specific season. In the following screenshot you can see a small part of the data set. The input features go from “Completions”, “PassAttempts” till “SafetyByTeam”.

Rk	Season	Team	W	Completions	PassAttempts	PassYD	PassTD	Int	RushYD	RushTD	PointsAgainst	XPA	XPM	FGA	FGM	SafetiesByTeam
1	2023	WAS	4	407	636	3725	24	21	1592	14	518	35	32	24	19	0
2	2022	WAS	8	344	554	3472	24	16	2143	9	343	28	24	30	25	1
3	2021	WAS	7	356	550	3441	21	15	2061	13	434	29	25	33	28	0
4	2020	WAS	7	389	601	3465	16	16	1611	18	329	32	30	34	27	1
5	2019	WAS	3	298	479	2812	18	13	1583	9	435	22	21	30	25	0
6	2018	WAS	7	311	509	3021	16	15	1774	12	359	26	25	29	26	0

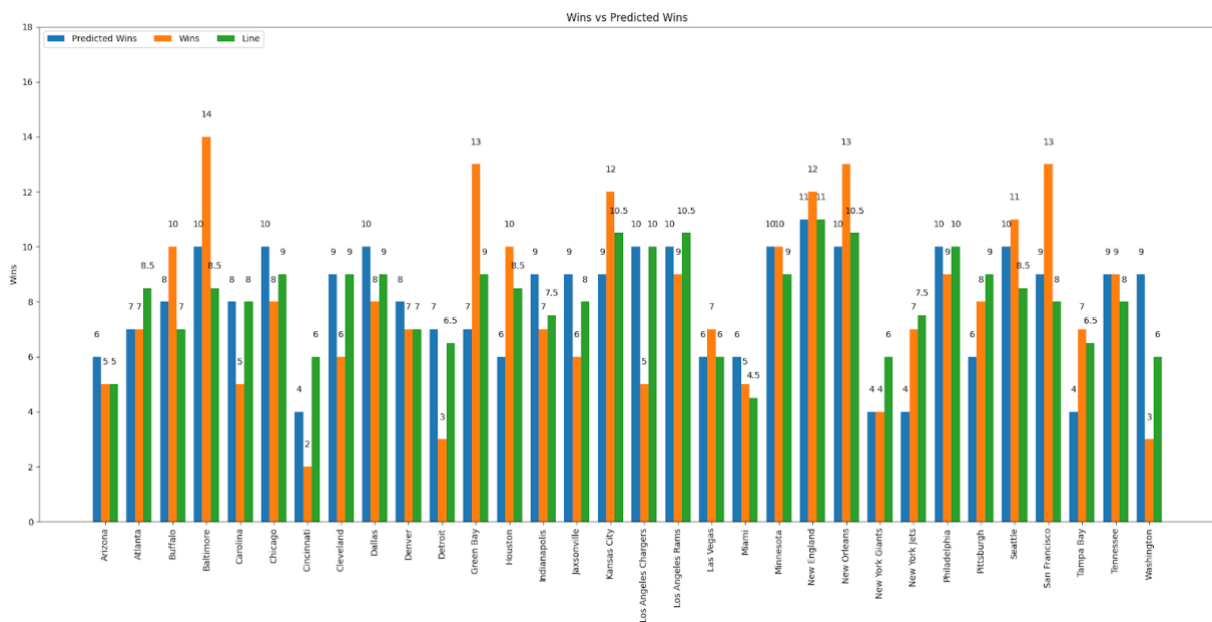
In total we got 1706 Datapoints. Each Datapoint having those 13 input features. We used 80% of that data to train our Model and 20% to test it.

Model and methods

We implemented a Multi-Layer-Perceptron with a total of 9 Layers. 6 hidden Layers, 2 Drop out Layers and 1 output layer. In our first layer we used 10 perceptrons and between Layer 2 and 6 we have each 64 perceptrons and the output Layer has 17 perceptrons. That makes a total amount of parameters (weights + biases) of 18,589 trainable parameters. After each hidden dense layer we used the relu activation function. The output layer is followed by a softmax activation function, as we want to classify the result between 0–16 wins (17 perceptrons in the output).

Results

Our model achieves an accuracy of around 30%, which may initially seem low. However, considering the unpredictable factors in sports predictions, such as key player injuries or weather conditions, this level of accuracy is noteworthy. These elements are challenging to simulate and incorporate into training data, making precise predictions very difficult. However, looking at our results we see that most of our predictions are very close to the actual wins.



This was for the 2019 NFL season. Despite some outliers, the blue line generally aligns closely with the orange one.

In addition to implementing our model, we aimed to evaluate its effectiveness for betting purposes. To do this, we tested the model using betting data from the 2019 NFL season for all teams. That is the following result:

```
1/1 0s 23ms/step
1/1 0s 26ms/step
1/1 0s 25ms/step
1/1 0s 24ms/step
1/1 0s 34ms/step
1/1 0s 24ms/step
1/1 0s 24ms/step
1/1 0s 40ms/step
1/1 0s 29ms/step
Initial Budget was 32000, Final Budget was 39257.622007622005, for a gain/loss of 7257.622007622005 for a percent gain of 22.680068773818764
fabi NFL Prediction$:
```

With the predictions of our model we were able to make a 22% increase of our initial budget. We went from \$32,000 to \$39,257.