CS 221 Project Final Report

Upamanyu Dass-Vattam, Campbell Hoskins

December 2023

1 Introduction

In the 2023 season, the Los Angeles Clippers are scheduled to travel a league-high 51,000 miles [1]. Their 82-game season will include stretches with back-to-back games, cross-country travel, and grueling overnight flights. Recent breakthroughs in polysomnography have quantified the negative effect that fatigue, a consequence of such demanding schedules, has on human cognitive and athletic performance. This project seeks to predict "scheduled losses" in the NBA Schedule, considering various factors related to team fatigue such as travel, rest, and game frequencies, with the aim of assisting stakeholders and oddsmakers in understanding when a team might underperform due to schedule-induced factors.

The significance of this project is emphasized by emerging research on the large impact that sleep and travel schedules have on athletes' performance and well-being. A narrative review focusing on NBA teams shows that the condensed game schedule and frequent air travel disrupt sleep patterns, affecting physical and mental health [4]. This disruption in sleep has been less explored in the context of the grueling demands of an NBA season. Our project uses these findings, as we aim to predict the performance impacts arising from these challenges.

A different study conducted at Stanford University with its men's varsity basketball team highlights the positive effects of sleep extension on athletic performance, including improved reaction times, mood, and reduced daytime sleepiness [3].This study provides evidence supporting the hypothesis that enhanced sleep can significantly boost performance metrics specific to basketball, such as sprint times and shooting accuracy. Inherently, this finding suggests that fatigue would cause a reduction in performance in NBA games.

These insights form a crucial backdrop to our project's objective: predicting game outcomes based on fatigue factors. By integrating such knowledge into our analysis, we aim to expose the lack of fatigue consideration by odds makers when creating the betting spreads and shed light on how the rigors of NBA scheduling might predispose teams to "scheduled losses." Our findings could serve as a valuable resource for the NBA in optimizing schedules to minimize these losses, thus promoting fairer competition and potentially enhancing the overall well-being of the players. Understanding and mitigating the impacts of travel and rest schedules on performance not only benefits the league and its stakeholders but also contributes to the broader conversation on athlete health and peak performance in professional sports.

2 Literature Review

The Mahscore, developed by Dr. Cheri Mah, predicts NBA game outcomes based on factors like travel, recovery time, and game frequency [2]. Our model shares similarities with Mahscore in considering fatigue and travel-related factors but differs in scope. While Mahscore focuses on individual player fatigue, our model assesses team-wide dynamics, incorporating broader factors like cumulative travel distance and game frequency. The exact methodology used in the Mahscore is not disclosed, but our logistic regression approach likely parallels their predictive modeling techniques.

Another project conducted by Josh Weiner aimed at predicting NBA game outcomes using various machine learning techniques, including feature engineering and model selection like Logistic Regression and RandomForestClassifier [5]. Weiner's approach is similar to ours since it uses statistical models to predict game outcomes. However, our project specifically uses a set of features focusing on team fatigue and travel-related factors, which sets it apart. Weiner's approach competes directly against the factors taken into account by odds makers when making game spreads, whereas our model uses features overlooked by the odds makers to gain an edge over the spread predictions.

Our project contributes to the growing field of sports analytics by offering a unique perspective on predicting NBA game outcomes. It aligns with the trend of employing machine learning techniques in sports predictions but stands out by focusing on team-level fatigue factors, offering a fresh angle compared to the predominantly player-centric or general team performance models in existing literature.

3 Data

3.1 Datasets

Our project used two primary datasets. The first, a dataset from Kaggle, includes vital game information such as the home and visitor teams, game location, date, start time, final scores, and overtime for NBA games from 2011 onwards. The second dataset, sourced from a hobbyist's collection, provides gambling odds from 2012 onwards such as the date of the game, home and away teams, and the betting spread. The betting spread and whether it was covered is used to determine the success of our model's predictions. The combined datasets span over a decade of NBA games, providing data for over 12,000 NBA games.

3.2 Data Preprocessing

Cleaning Data

First, we merged the two dataframes. We did this by merging based on the date and home/away teams of the game. Next, we standardized the representations of the teams - for example, the Lakers could have been represented as the L.A Lakers or Los Angeles Lakers, and some teams(like the Charlotte Hornets) were originally in other locations.

We then filtered the games. Every game that did not have spread data, which could be because it wasn't captured or because it was a preseason or exhibition game, was removed. Every game which is considered a 'PUSH' by Vegas, i.e the game matches the spread exactly, was removed as well. This is simply to prevent a class imbalance from categorizing these games as either covered or not, and thus we have exactly the same number in each class.

Feature Engineering

Once all data was compiled, we used the raw data to engineer the features used in our model. To create each feature, the following processing was done:

1. **Date**: Conversion 'Date' column from string to a Python datetime object.
2. **Start (ET)**: Transformation of game start time into a float representing time in hours.
3. **Previous Game OT**: Binary feature indicating whether the previous game went into overtime.
4. **Miles Traveled**: Calculation of the distance traveled by a team to a game. This was done by calculating the distances from latitude-longitude coordinates between every arena, and then mapping each game based on its arena and the teams' previous game's arena.
5. **Spread**: Betting line for the home team from a separate dataset and merging it with the main dataset.
6. **Home Status**: Binary feature indicating whether the team was playing at home or away.
7. **Previous Game Points**: Total points scored in the previous game.
8. **Days from Last Game**: Number of days since the team's last game.
9. **Change in Timezones**: The change in time zones between consecutive games for a team.
10. **Label (Covered)**: Binary factor showing whether or not the team covered.

Normalization
We normalized features like 'Previous Game Points' and 'Miles Traveled' using z-score normalization.

4 Baseline

For our baseline model, we implemented a coin flip system, which predicted that the home team would cover if a random variable if above 0.5 and would predict the team did not cover if the variable was less than 0.5. This was essentially a random guess, which actually represents the challenge of picking against the spread because odds makers do their best to make it a 50/50 guess.

5 Main Approach

5.1 Approach 0: Logistic Regression

Our initial approach involved employing a logistic regression model, leveraging the various features engineered from our dataset. The choice of logistic regression was driven by its efficiency in binary classification tasks and its interpretability, which is crucial for understanding the influence of different features on the model's predictions.

The logistic regression model functions by applying a logistic function to a linear combination of the input features to predict the probability that a given input belongs to a certain class. In our case, the model predicts the probability of a team covering the spread in a game.

Model Training and Optimization: The model was trained using gradient descent, an optimization algorithm that iteratively adjusts the weights of the features to minimize the cost function, in this case, the binary cross-entropy loss. This loss function measures the difference between the predicted probability and the actual class (whether the team covered the spread or not).

5.2 Approach 1: Neural Network

5.2.1 Linear Neural Network

After testing logistic regression, we decided to build a neural network for predicting whether or not a team covers the spread. Initially, we predicted for every game whether the team covered the spread or not. This was accomplished by using each game as a datapoint and whether the team covered or not as a label.

The model is comprised of three linear sections, and a prediction head. At each of the linear sections, we have a dense layer with ReLU activation, a batch normalization layer, and a dropout layer. After these layers, the model incorporates a global maxpooling layer, and finally a dense neuron which outputs the probability of a team covering the spread.

The model were optimized by variants of stochastic gradient descent. SGD is an algorithm which starts at a random point on the function and travels along its slope until it finds a local minimum. In this case, the function is the binary cross entropy loss function, which calculates the probabilities of the binary classes and penalizes the loss based on the distance from the expected value. Some variants considered were RMSProp, which is root mean squared propagation, and AdamW which is a variation of SGD that decays weights independent of gradient descent.
The model outputs a floating point value between 0 and 1, where anything above 0.5 is categorized as the team covers the spread, and below 0.5

5.2.2 Hyperparameter Selection

There were numerous hyperparameters we tested. We decided on the actual variant of SGD and the associated learning rate, as well as the number of neurons in each dense layer and the dropout rates. To select hyperparameters, we used Bayesian hyperparameter optimization.
Bayesian optimization essentially builds a model approximating the probability representation of the objective function, in this case binary cross entropy loss. It starts off with an initial set of hyperparameters(in this case randomly sampled from a space we gave it), and trains a model with those parameters. It then uses Bayes theorem on the model to build a surrogate model for the loss

function, and then updates each hyperparameter one by one using an approximation function. After that, it trains another model, and repeats this problem until stopped by number of iterations or some other cutoff.

5.3 Timeseries Prediction Data Revision

Upon reflection of our single game data prediction, we decided that the most important aspect of modeling this problem is the features the model sees relating to the games. Until this point, we had been iterating through the dataframe and finding cumulative factors of fatigue such as miles traveled for the past x days. However, we decided that our cutoffs may have been arbitrary and not representative of the true factors of the game.

Thus, we decided it was better for the nature of the problem to model this as a timeseries prediction problem. To do so, we altered the way the model interacted with the data. Instead of calculating factors for the previous games in a timespan, we simply passed a sequence of the previous games and their factors. This way, the model is able to make inferences on the sequence of previous games instead of being limited to the factors we calculate.

Initially, we tried using Long Short-Term Memory layers in place of the dense layers and used the previous 20 games in the season. LSTMs are a type of recurrent neural network, and is made of cells that remember values over arbitrary time intervals and have three gates controlling the flow of information in and out. This ability to output information selectively makes it useful to maintain long term dependencies, and thus are used in problems like these where there is a time component.

We also implemented our own attention layers. This was a simple weight matrix matching the shape of the input, which is updated with the goal of identifying correlations between the features of the game, and thus 'pay attention' to certain features more.

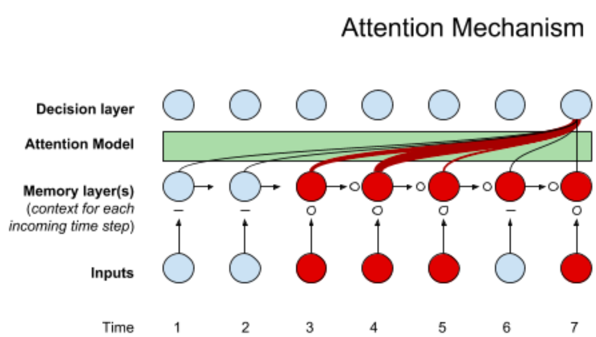


Figure 2: Attention Explained. Source: analyticsvidhya.com

However, we found that neither of these changes had any significant effects on the results, and went back to the linear neural network from above. We chose to do this because the model trained significantly slower, so to allow for rapid experimentation, it made sense to stick with the simpler model.

6 Results & Analysis

In this section, we present the outcomes of our two primary models: Logistic Regression and Neural Network. Our analysis focused on predicting NBA game outcomes and challenging the efficiency of Las Vegas odds in accounting for these elements.

6.1 Results

The Logistic Regression model provided a fundamental understanding of the relationship between fatigue factors and game outcomes. However, the model's performance was modest, with the following statistics:

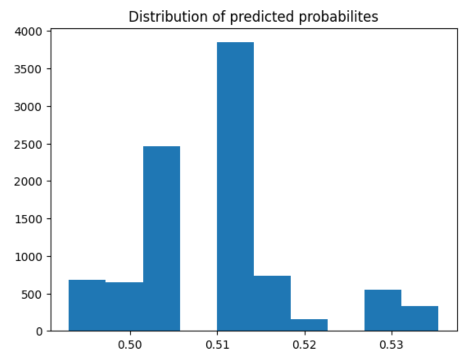
Set	**Accuracy**	**Precision**	**Recall**	**Confusion Matrix**
Train	0.5031326324731867	0.5031326324731867	1.0	$\begin{bmatrix} 0 & 4679 \\ 0 & 4738 \end{bmatrix}$
Validation	0.4855018587360595	0.4855018587360595	1.0	$\begin{bmatrix} 0 & 692 \\ 0 & 653 \end{bmatrix}$
Test	(Test Accuracy)	(Test Precision)	(Test Recall)	$\begin{bmatrix} 0 & 1357 \\ 0 & 1335 \end{bmatrix}$

Table 1: Logistic Regression Performance Metrics

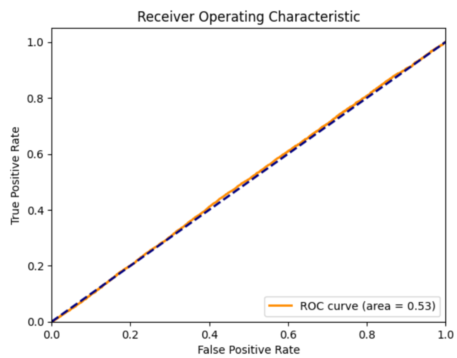
The Neural Network model was developed to enhance prediction accuracy and handle the complexity of the data more effectively. The model's architecture included three linear sections with ReLU activation, batch normalization, and dropout layers, followed by a global maxpooling layer and a dense neuron for output. The key statistics for the Neural Network model are:

Set	**Accuracy**	**Precision**	**Recall**	**Confusion Matrix**
Train	0.4986726133588192	0.5034177724165662	0.26424651751794004	$\begin{bmatrix} 3444 & 1235 \\ 3486 & 1252 \end{bmatrix}$
Validation	0.5033457249070632	0.47875354107648727	0.25880551301684535	$\begin{bmatrix} 508 & 184 \\ 484 & 169 \end{bmatrix}$
Test	0.5033432392273403	0.498546511627907	0.25692883895131086	$\begin{bmatrix} 1012 & 345 \\ 992 & 343 \end{bmatrix}$

Table 2: Neural Network Performance Metrics



Probability Distribution of the Neural Network Model on the Training Set



ROC Curve for the Neural Network on the Training Set

Above is the Receiver Operating Characteristic evaluating the performance of the system across different thresholds. The ROC graph plots the True Positive Rate against the False Positive Rate at various threshold settings. Each point represents a different threshold where the classification decision is made. Our curve being close to the diagonal line means the system is essentially a random guess, which supports our result that our model was not effective at predicting.

6.2 Analysis

Our initial hypothesis suggested that Las Vegas odds might not sufficiently consider fatigue factors in setting the spread. However, the performances of both models indicated no significant advantage over the spread, suggesting that our data did not capture all the elements necessary to outperform the established odds.

To refine our approach, we could broaden our data scope to include more fatigue factors such as amount of sleep and quality of nutrition. Additionally, we could pivot to include other basketball performance factors and not just fatigue-related factors.

7 Error Analysis

We performed two main types of error analysis. First and foremost, we performed a y-permutation bootstrapping analysis of the results. To do this, we randomly shuffled the labels of the data and trained the model on this permuted data. Next, we got predicted probabilities for each game. We then repeated this process 100 times to get a distribution for each game. Using this distribution, we calculated the mean and standard deviation to determine whether or not our model's predictions were statistically significant or within the realm of chance.

	Spread Covered	Not Covered
Statistically Significant	0	0
Not Statistically Significant	6726	6729

Table 3: Dismal Results

As shown above, all of our predictions could theoretically be attributed to random chance. Thus, we believe our model did not learn any significant patterns in the data, and its predictions were only slightly better than random chance due to overfitting on the dataset.

Next, we stratified the datapoints by the probabilities and examined the accuracy by these groupings. As you can see in the probability distribution image, there appears to be a small group of games with probabilities predicted under 0.5. We examined the accuracy of these games and found that there was no significant difference in the accuracy of the predictions in these games versus the accuracy of the predictions in the rest of the games, meaning that this is likely just noise learned rather than being an indicator of some predictability on not covering the spread.

8 Future Work

Our experiments show that it is clearly extremely difficult to predict a team's performance against the expected based off of publicly available fatigue related data. However, it would be interesting to see what access to limited data could do. In her study, Dr. Ma was integrated with teams' daily routines and would have had access to data around sleep, flight times, nutrition, and more. It would be interesting to see how those other factors could swing potential differences to make the model more useful.

Another interesting exercise would be to try to beat Vegas at all. Bookies, and especially with legalized sports gambling, have high powered teams dedicated to algorithms and simulations which they use to calculate spreads in a way that maximizes their profits. It would be interesting to see whether even the most powerful open source models with all the available open access data could beat their spreads generated by proprietary models and systems at all.

9 Code

https://github.com/UpamanyuDassVattam/CS221FinalProject

References

[1] Crean, Tim. “Clippers Won’t Love Crazy Travel Schedule for 2023-24 NBA Season.” *ClutchPoints*, ClutchPoints, 21 Aug. 2023, clutchpoints.com/clippers-news-la-wont-love-crazy-travel-schedule-2023-24-nba-season.

[2] Holmes, Baxter. “Which Games Will Your Team Lose Because of the NBA Schedule?” *ESPN*, ESPN Internet Ventures, www.espn.com/nba/story/$_{/}id/25117649/nba - schedule - alert - games - your - team - lose - 2018 - 19.Accessed7Dec.$2023.

[3] Mah, Cheri, et al. “The Effects of Sleep Extension on the Athletic Performance of Collegiate Basketball Players.” *Sleep*, U.S. National Library of Medicine, June 2011, pubmed.ncbi.nlm.nih.gov/21731144/.

[4] Singh, Meeta, et al. “Urgent Wake up Call for the National Basketball Association.” *Journal of Clinical Sleep Medicine: JCSM: Official Publication of the American Academy of Sleep Medicine*, U.S. National Library of Medicine, Feb. 2021, pubmed.ncbi.nlm.nih.gov/33112229/.

[5] Weiner, Josh. “Predicting the Outcome of NBA Games with Machine Learning.” *Medium*, Towards Data Science, 15 July 2022, towardsdatascience.com/predicting-the-outcome-of-nba-games-with-machine-learning-a810bb768f20.