Thesis Project

Applied Artificial Intelligence 180 credits

Non-Basketball Features in the NBA: Machine Learning and SHAP Analysis of Player Contracts and Team Priorities

Data Science 15 credits

Halmstad June 2, 2025

Emil Falk

Supervisor: Yuantao Fan

HALMSTAD UNIVERSITY

Abstract

The National Basketball Association (NBA) is the world's biggest basketball league, with 30 teams across the United States and Canada, which generates billions in annual revenue. An NBA players impact is usually determined based on performance on the court. However, off-court factors influence aspects of the NBA, such as player contracts. This study investigates how non-basketball features, such as off-court factors like Instagram followers and player loyalty, affect player contracts. By using various Machine Learning models, this study analyzed the model's predictive results trained on a dataset containing only basketball features and compared the results on a dataset containing both basketball and non-basketball features. Furthermore, this study was assisted by the Explainable AI model SHAP to examine how the most valuable teams versus the least valuable teams in the NBA prioritized these non-basketball features. SHAP's reliability was also assessed for this specific problem. The results showed that incorporating non-basketball features significantly improves the predictive performance of many Machine Learning models, but not Deep Learning models performance in this study. The SHAP analysis revealed that there are differences between highly valuable teams and low-value teams. Highly valuable teams pay for every feature on average more than low-value teams, and if a player were an All Star, it is more likely that this player will be paid more on a highly valuable team. The SHAP assessment test demonstrated its functionality in this case. However, in a general context, SHAP reliability cannot be proved in this study. These results highlight the role of non-basketball features in NBA salaries and offer insights into the application of explainable AI in salary prediction.

Keywords Non-Basketball Features, Explainable AI, Machine Learning Models, National Basketball Association

Sammanfattning

National Basketball Association (NBA) är världens största basketliga, med 30 lag i USA och Kanada, vilket genererar miljarder i årliga intäkter. En NBA-spelares inverkan bestäms vanligtvis baserat på prestationer på planen. Faktorer utanför planen påverkar dock aspekter av NBA, såsom spelarkontrakt. Denna studie undersöker hur icke-basketrelaterade funktioner, vilket innebär faktorer utanför planen som Instagramföljare och spelarlojalitet, påverkar spelarkontrakt i NBA. Genom att använda olika maskininlärningsmodeller analyserade denna studie modellernas prediktiva resultat tränade på en datauppsättning som endast innehöll basketrelaterade funktioner och jämförde resultaten på en datauppsättning som innehöll både basketrelaterade och icke-basketrelaterade funktioner. Dessutom använde sig denna studie av Explainable AI-modellen SHAP för att undersöka hur de mest värdefulla lagen kontra de minst värdefulla lagen i NBA prioriterade dessa icke-basketrelaterade funktioner. SHAP:s tillförlitlighet utvärderades också för detta specifika problem. Resultaten visade att inkluderingen av icke-basketrelaterade funktioner avsevärt förbättrar den prediktiva prestandan för många maskininlärningsmodeller, men inte Deep Learning-modellers prestanda i denna studie. SHAP-analysen visade att det finns skillnader mellan mycket värdefulla lag och lag med lågt värde. Högvärdiga lag betalar i genomsnitt mer för varje funktion än lågvärdiga lag, och om en spelare vore en All Star är det mer troligt att den spelaren kommer att få mer betalt i ett högvärdigt lag. SHAP-bedömningstestet visade dess funktionalitet i detta fall. Men i ett generellt sammanhang kan dock inte SHAP-tillförlitlighet bevisas i denna studie. Dessa resultat belyser rollen av icke-basketrelaterade funktioner i NBA-löner och ger insikter i tillämpningen av förklarbar AI i löneprognoser.

Nyckelord Icke-Basket funktioner, Förklarbar AI, Maskininlärningsmodeller, National Basketball Association

Contents

List of Figures

List of Tables

Listings

Chapter 1

Introduction

1.1 Background

The National Basketball Association (NBA) is the world's biggest basketball league with a revenue of 11.34 billion USD during the 23/24 season [1]. The NBA contains 30 teams that each year compete for the national championship. What differs NBA from European basketball tournaments is that in Europe, teams are allowed to spend money on player salaries based on their own financial situation. However, in the NBA they use a hard cap system, meaning that every team gets a limited amount of money they can spend each season on player salaries [2].

The NBA salary cap is determined by many things. However, the two things that influence the salary cap the most by far are the Basketball Related Income (BRI) deal and the Collective Bargaining Agreement (CBA) deal. The BRI deal is the league revenue from tickets, TV and radio deals, merchandise sales, sponsorships, and other incomes related to basketball operations specified in the BRI deal. The other deal, the CBA deal, is an agreement between the NBA and the National Basketball Players Association (NBPA). The NBPA works like a union for NBA players and is run by current NBA players. The CBA deal determines how much of the BRI revenue should be allocated to the teams. The salary cap is therefore designed to be a product of the league's financial health [3].

With the NBA salary cap determined, there are still ways for teams to exceed the salary cap. The exception arises when signing some specific free agents (players without a contract). Teams can exceed the salary cap if they re-sign their own free agents if the player has played at least three consecutive seasons with the team. Players that are drafted and playing their first year (rookies) in the league also get a fixed salary negotiated by the CBA in which teams are allowed to exceed the salary cap in order to sign the rookie.

The salary cap is therefore designed so that every team gets the same amount of money to spend on players. This will promote the competitive balance by pre-

venting some wealthier teams from exclusively spending money on star players. However, some argue that teams from smaller markets cannot compete against teams from bigger markets. One instance is that exceeding the salary cap might be more beneficial for some teams than others. Penalties might be easy for some teams to pay but difficult for other teams, meaning some teams can develop multiple star players and exceed the salary cap, compared to smaller teams, which are only able to retain less developed star players. Former NBA player and analyst Avery Johnson says that small-market teams cannot compete against the large sum of money a good player can earn in a large market from endorsement opportunities and that smaller teams must bring in undervalued players in free agency and convince their own draftees to sacrifice external money in order to re-sign with the team [4].

Further analysis has also been done on the contract structures in the NBA and how teams are using data to operate the business. Players, together with their agents, are in more control of their own path in the NBA. NBA Agents are experts in contract laws and salary cap intricacies, and together with their knowledge in market trends, skilled agents are securing favorable terms for their clients today. Besides negotiating salaries, agents also advocate for additional benefits such as performance incentives and endorsement opportunities while highlighting their client's non-basketball related traits, such as leadership traits [5].

The investment in basketball analytics is also increasing every year in the NBA to improve the team's operations. In 2009, 10 data analysts were working across teams in the NBA. In 2023, that number had increased to 132 data analysts [6]. One study found that teams that invested more in data analytics in general tended to win more games [7].

1.2 Problem Statement

The goal with this project is to predict NBA players salaries based on a set of features, non-basketball features, and basketball-related features using Machine Learning (ML). This study will research whether the addition of non-basketball features can improve the predictive accuracy of various ML models. Consider a dataset of N players, denoted $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{N}$, where:

- $\mathbf{x}_i = [x_{i1}, x_{i2}, \ldots, x_{ip}] \in \mathbb{R}^p$ is the feature vector for player i, with p features partitioned into basketball features ($\mathbf{x}_i^B$) and non-basketball features ($\mathbf{x}_i^{NB}$) subsets.
- $y_i \in \mathbb{R}$ is the salary of player i.

A predictive model $f : \mathbb{R}^p \to \mathbb{R}$ maps features to salaries, such that $\hat{y}_i = f(\mathbf{x}_i)$. The output from the best performing model f will then be evaluated with an Explainable AI model and compared with smaller and bigger teams based on the analysis in **Section 1.1**. The Explainable AI model will also be validated to accu-

rately quantify the influence of some features. Therefore, the research questions are stated as:

- Can non-basketball features improve the predictive accuracy of machine learning models for NBA player salary prediction?
- Can Explainable AI (XAI) values assist in revealing differences in the importance of non-basketball features for player salaries between small and big market teams?
- How effectively can an XAI model be validated to quantify the influence of specific features on NBA players salaries in a predictive model?

1.3 Purpose and goals

The primary purpose of this project is to research whether non-basketball features, such as marketability, agents role, endorsement potential, etc., play a big role when NBA teams are making decisions about player signings during the free agency period. The research will also analyze disparities between small market teams and big market teams when evaluating the importance of non-basketball features. To understand if non-basketball features play a big role when a player signs an NBA contract, Machine Learning will be used to see if non-basketball features enhance the predictive accuracy for some common Machine Learning models compared with only using basketball features.

Another purpose of this project is to provide interpretable insights and conclusions on how smaller markets and bigger markets operate by leveraging XAI values, and then evaluate the reliability of using XAI, ensuring that the insights derived are robust and trustworthy.

1.4 Requirements

1.4.1 Functional Requirements

This project should use public NBA player data that includes averaged basketball features. Basketball features are metrics based on how players actually perform on the court. Points per game, rebounds, and assists are typical basketball features where, if a player scores fewer points than their average points per game in an NBA game, that feature will be decreased. The project should also include non-basketball features. These non-basketball features can be manually scraped or included in the data, but they must be information known to the public. The criteria for teams categorized as a small market or big markets will be based on the official team valuations for the season the player data is collected from.

The data must be preprocessed for Machine Learning model compatibility. This will include handling missing values, normalizing the dataset for some machine

learning models, but also encoding some text features such as a players team.

This project shall implement multiple machine learning models and deep learning models to predict NBA players salaries. This will include simpler traditional model, and deep learning models. These systems will each be trained and tested with one dataset containing only basketball features and another dataset containing both basketball features and non-basketball features. Each model will be evaluated by some regression metrics and then be compared with each dataset.

The system will produce XAI values for the best-performing model according to both the primary regression metric trained with the dataset containing basketball and non-basketball features. The purpose is to quantify the actual contributions of non-basketball features when determining a players contract. This project should provide different sets of XAI values, one for the more valuable teams in the NBA and one for the less valuable teams in the NBA, and then visualize feature importance and analyze the comparison of non-basketball features between small market teams and big market teams. The XAI values will also be used to provide interpretable reports and findings about the role of non-basketball features in NBA contracts. The system should also implement a method to evaluate the reliability of using XAI values to interpret the importance of different features by using methods such as a stability analysis or a sensitivity analysis.

1.4.2 Non-Functional Requirements

The machine learning models and the dataset are required to be able to operate within a Python Ecosystem. This project must be able to support Python-based libraries such as NumPy, Pandas, TensorFlow, and scikit-learn. The project must be able to operate on a simple and user-friendly platform such as Jupyter Notebook. It must also be able to operate on a local machine without a graphics processing unit (GPU) to ensure accessibility. This project should be executable on a standard hardware such as a laptop with at least 16 Random-Access Memory (RAM) gigabytes (GB).

The system should then be able to compute XAI values and train each model with a dataset containing up to 1 000 NBA players and up to 50 features. The system should also be designed so that the addition of new features, such as non-basketball features, does not affect the code structure and only requires minimal code modification. The system should also ensure that results are reproducible for every machine learning and deep learning model. This project will make the system modular, meaning that each model or the XAI computation can be updated separately without affecting the other models.

At the end of the project, this model will remain public. The model relies on publicly available data, open-source code, and open-source libraries. I think there are

competitive advantages with this model if it is implemented successfully. The usage of non-basketball features and their contributions to an NBA contract is not widely researched. There is a risk that the model will still be in development after the project, so one ethical concern could be a misuse of potentially inaccurate predictions from users. However, the models predict estimated "fair values" for players based on basketball features and non-basketball features, and not negotiated outcomes so therefore, the tool should be treated as a tool that predicts estimated fair values. As long as there is empathy for that, the model will remain public.

1.5 Related Work

The research on NBA salary predictions using machine learning models is limited, but there has been research conducted about the NBA salaries using machine learning that is useful for this study. Most of these studies have focused a lot on how basketball features correlate to an NBA players salary. One study used a linear approach to predict NBA contracts. Although the explained variance (R^2) reached 0.5908 [8]. One study found clear evidence that the relationship between an NBA player's performance statistics and their salary is non-linear and concluded that there is a necessity to apply non-linear models and algorithms when predicting NBA contracts [9]. Non-linear models like tree models are widely researched in this field. One study used Random Forest, which is a tree-based model, and achieved an R^2 score of 0.8818 [10]. Another study used Gradient Boosting and achieved an R^2 score of 0.74 [11].

The research on Deep Learning models for tasks like salary prediction has also been conducted, but not in an NBA context. One study predicted salary grades for graduates graduating from a university by using Multi-Layer Perceptrons (MLP), which outperformed traditional Machine Learning models [12]. Another study used a Graph Convolutional Network (GCN) in order to classify salary ranges based on the job posting, which outperformed traditional models [13]. Convolutional Neural Networks (CNNs) have been used in regression tasks. One study used different types of CNNs to predict the orientation and direction of straight arrow signs, where the two best CNN models, VGG16 and VGG19, achieved an R^2 of 0.9984 and 0.8483. This study highlighted the effectiveness of using CNNs in regression tasks, but also the importance of choosing a task-specific architecture for the CNN model [14].

Many studies contain a section for potential improvements to more accurately predict NBA contracts. The study that achieved an R^2 score of 0.8818 concluded that future work could be the inclusion of a player's off-court appeal to achieve a higher prediction score [10].

Research about how non-basketball features influence salaries is not widely re-

searched. However, research about the importance of external factors in sports is widely conducted. Agents are one feature that has been researched. One study looked at the football market and concluded that an agent's role is to negotiate contracts but also find external marketing and endorsement deals, among others [15]. The influence of the draft picks' role in the NBA has also been researched. One study confirmed that a higher draft pick usually leads to higher paychecks [16]. Therefore, teams might be willing to take a risk on players drafted at the higher end of the draft when signing a new contract. Research on All-Star players, meaning good players in the league, has also concluded that big market teams tend to go after those players when adding new players during free agency [17]. How social media followers impact salaries has also been researched. One study concluded that Instagram followers are one of the biggest factors in an NBA player's salary, meaning that players with more Instagram followers generally have a higher salary [18].

XAI is something that is researched in basketball. The research that incorporated XAI had the purpose to explain their non-interpretable model's predictions and not beyond that. One study predicted NBA players outcomes based on their college stats and looked at the important features using SHAP [19]. Another study predicted NBA game outcomes and used SHAP to explain the reasons for the predictions in order to find the "key winning factors" and the "key losing factors" [20]. Research that aims to explain salaries and differences in sports markets is hard to find, which highlights the potential for XAI to bridge gaps in sports salary research.

The contributions for this study are to novel empirical evidence on the impact of non-basketball features for NBA player salaries. This study will also use XAI to reveal market-specific differences in feature importance, advancing XAI role in sports economics but also more transparency from the output in every models, including traditional models to see an absolute contribution for every feature. This study will also explore using Deep Learning models in an NBA contract context.

Chapter 2

Technical Background

2.1 Statistical Models

2.1.1 Multiple Linear Regression

Multiple Linear Regression (MLR) is a statistical method where the model explains the linear relationship between a continuous dependent variable y with multiple independent variables $x_1, x_2, \ldots, x_n$. So the variable y can be described as a linear combination of the independent variables plus a random error. What differs Multiple Linear Regression from Linear Regression is that Linear Regression includes just one independent variable, while Multiple Linear Regression includes at least two independent variables.

Multiple Linear Regression for i observations can be described as:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_i x_i + \epsilon \tag{2.1}$$

Where β_i are the regression coefficients that describe the response in variable y for every change in variable x_i. The goal is to estimate and optimize β by minimizing the sum of squared errors between the predicted values and the actual values. The loss function Multiple Linear Regression uses is the Residual Sum of Squares (RSS), and β is optimized by minimizing RSS through a method called Ordinary Least Squares (OLS) [21]. OLS estimates $\hat{\beta}$ by minimizing:

$$\text{RSS} = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \tag{2.2}$$

2.1.2 Ridge Regression

Ridge Regression is a regularized version of Linear Regression. The core difference between Ridge Regression is that it adds penalties on the RSS, ensuring that the coefficients will shrink to prevent issues like overfitting. This is done by including

a $L2$ penalty when optimizing β [21].

$$\text{RSS}_{\text{ridge}} = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2 + \lambda \sum_{j=1}^{p} \beta_j^2 \tag{2.3}$$

Where $\lambda \geq 0$ is the regularization parameter that can be tuned in order for optimal performance.

2.1.3 Lasso Regression

Least Absolute Shrinkage and Selection Operator (Lasso) Regression is a regularized version of Linear Regression. The core difference between Lasso Regression is that it adds penalties on the RSS, ensuring that some coefficients will shrink to 0, meaning that the penalty term $L1$ performs variable selection [21].

$$\text{RSS}_{\text{lasso}} = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2 + \lambda \sum_{j=1}^{p} |\beta_j| \tag{2.4}$$

Where $\lambda \geq 0$ is the regularization parameter that can be tuned in order for optimal performance.

2.2 Tree-Based models

2.2.1 Decision Tree

Decision Tree (DT) is a Machine Learning model that predicts and makes decisions in a way that can be represented as a hierarchical tree structure with different nodes. The first node, the root node, splits the dataset based on a feature and threshold that best separates the data according to a specific splitting criterion (Usually Mean Squared Error (MSE) for regression tasks). Then each split gets travels through an edge to an internal node where more splits occurs until a stopping condition is met where the data reach a leaf node that provides the prediction.

Decision trees can be described mathematically for regression problems. There are two steps when building a regression tree [22].

- Divide the prediction space, meaning the space for possible values for $X_1, X_2, \ldots, X_i$ into J distinct regions R which do not overlap, $R_1, R_2, \ldots, R_J$.
- Select the predictor X_j and cutpoint s that result in the largest reduction in RSS.
- Continue this recursive partitioning until a stopping criterion is met or no further reduction in RSS.

Decision Trees look for the values j and s that minimize the equation:

$$\sum_{i:x_i \in R_1(j,s)} \left(y_i - \hat{y}_{R_1}\right)^2 + \sum_{i:x_i \in R_2(j,s)} \left(y_i - \hat{y}_{R_2}\right)^2 \tag{2.5}$$

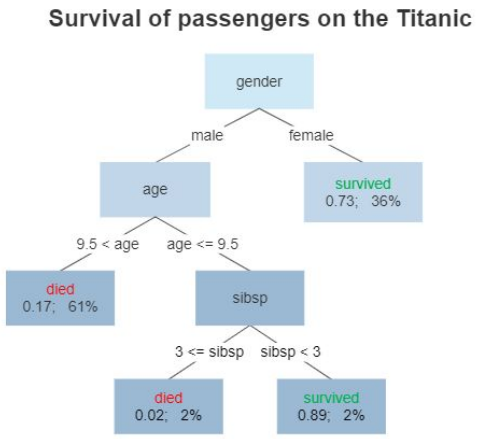

Figure 2.1: A visual representation of a Decision Tree containing different feature thresholds to reach a prediction.

2.2.2 Random Forest

Random Forest (RF) is a tree-based algorithm based on an ensemble method called bagging. Ensemble methods are where multiple decision trees, called weak learners, make their prediction based on an aggregated score, for regression tasks, usually the mean score [23]. Bagging introduces random sampling from the training data. This creates multiple random subsets where each subset is intended for training a weak learner [24]. Random Forest uses bagging but also introduces random feature selection in each subset [25].

Random Forest can mathematically be expressed as [26]:

$$\{h(x, \Theta_k), k = 1, \ldots, K\} \tag{2.6}$$

Where:

- $\mathbf{x} \in \mathbb{R}^p$ is the input vector, with p being the number of features.
- Θ_k are independent identically distributed (i.i.d.) random vectors that introduce randomness into the tree construction through bagging and random feature selection.
- $h(\mathbf{x}, \Theta_k)$ is the prediction of the k-th tree for input $\mathbf{x}$:
  - For regression, $h(\mathbf{x}, \Theta_k)$ outputs a numerical value.
- K is the number of trees in the forest.

2.2.3 XGBoost

Extreme Gradient Boosting (XGBoost or XGB) is a model built on the foundation of Gradient Boosting. Gradient Boosting is, like Random Forest, an ensemble method that constructs weak learners, usually Decision Trees. Instead of building parallel independent trees, Gradient Boosting builds trees sequentially. Each decision tree corrects the errors from the previous trees by using a gradient descent loss function. At each iteration, Gradient Boosting approximates the negative gradient of the loss function and fits a weak learner to it [27].

Mathematically, Gradient Boosting can be described by given a dataset $\{(x_i, y_i)\}_{i=1}^n$, the goal is to approximate a function $F(x)$ that minimizes a loss function. The squared loss $L(y, F(x)) = \frac{1}{2}(y - F(x))^2$ is a common loss function for regression tasks. The model is built as [28]:

$$F(x) = \sum_{t=0}^{T} \eta_t h_t(x) \tag{2.7}$$

Where $h_t(x)$ is a weak learner, η_t is the learning rate, and T is the number of iterations.

The algorithm proceeds as follows:

1. Initialize $F_0(x) = \frac{1}{n}\sum_{i=1}^n y_i$.
2. For $t = 1$ to T:
    - Compute the negative gradient:

$$-g(x_i) = -\frac{\partial L(y_i, F(x_i))}{\partial F(x_i)} = y_i - F_{t-1}(x_i) \tag{2.8}$$

    - Fit a regression tree $h_t(x)$ to the negative gradients $\{-g(x_i)\}_{i=1}^n$.
    - Update the model: $F_t(x) = F_{t-1}(x) + \eta_t h_t(x)$.

XGBoost extends this framework by introducing a regularized objective that explicitly penalizes model complexity to prevent overfitting. While Gradient Boosting uses first-order gradients, XGBoost minimizes by using second-order approximations, including L1 and L2 regularization [29]:

$$\mathcal{L}^{(t)} = \sum_{i=1}^{n}\left[g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i)\right] + \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} w_j^2 + \alpha \sum_{j=1}^{T} |w_j| \tag{2.9}$$

Where g_i, h_i are first- and second-order gradients, and $\gamma T + \frac{1}{2}\lambda \sum w_j^2 + \alpha \sum |w_j|$ penalizes complexity with L2 (λ) and L1 (α) regularization.

2.3 Deep Learning models

2.3.1 Multilayer Perceptron

A Multilayer Perceptron (MLP) is a feed-forward Artificial Neural Network (ANN) that consists of multiple layers of interconnected neurons that provide an output by learning complex, non-linear relationships in the input data. An MLP has three types of layers. One input layer that receives the input features. One or more hidden layers that learn the patterns in the data by transforming the input through weighted connections, biases, and activation functions. Then an output layer providing the models output or prediction. All layers and neurons are connected in an MLP [30].

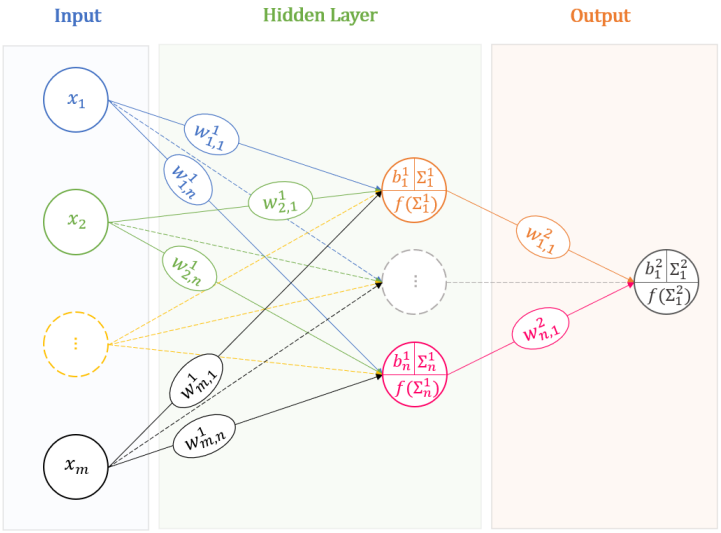

Figure 2.2: A visual representation of a Multilayer Perceptron.

An MLP learns a function $F(x)$ by minimizing a loss through backpropagation. Given a dataset $\{(x_i, y_i)\}_{i=1}^{n}$, MLP works by:

1. For a neuron j in layer l, a weighted sum and an activation, such as Sigmoid or ReLU, depending on the task, are computed:

$$z_j^{(l)} = \sum_i w_{ij}^{(l)} a_i^{(l-1)} + b_j^{(l)}, \quad a_j^{(l)} = \text{activation}(z_j^{(l)}) \tag{2.10}$$

   Where $w_{ij}^{(l)}$, $b_j^{(l)}$ are weights and biases, and $a_i^{(l-1)}$ are activations from the previous layer (input layer: $a_i^{(0)} = x_i$).
2. The loss between predicted output $\hat{y} = F(\mathbf{x})$ and true label y is computed with a loss function, for instance, MSE.

3. The gradients of L w.r.t. parameters are computed by using the chain rule:

$$\frac{\partial L}{\partial w_{ij}^{(l)}} = \frac{\partial L}{\partial z_j^{(l)}} \cdot a_i^{(l-1)} \tag{2.11}$$

The parameters are updated via gradient descent:

$$w_{ij}^{(l)} \leftarrow w_{ij}^{(l)} - \eta \frac{\partial L}{\partial w_{ij}^{(l)}} \tag{2.12}$$

Where η is the learning rate.

2.3.2 1D Convolutional Neural Network

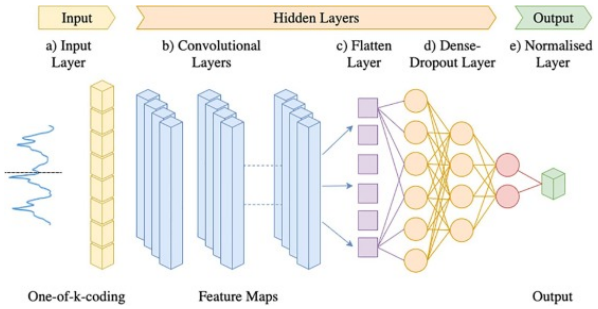

Figure 2.3: A visual representation of a 1D CNN with convolutional layers.

A 1-dimensional (1D) Convolutional Neural Network (CNN) is a deep learning algorithm designed for sequential data in order to find local patterns such as trends in the data [22]. In the application of NBA contracts, a pattern could be that a player with higher values in feature x_i and feature x_j is paid more.

The architecture of a 1D CNN consists of convolutional layers and pooling layers. The convolutional layers consist of multiple filters, also known as kernels. These kernels are then used to extract features from the input data by taking the dot product between the entire set of input data and all kernels [31]. Padding can also be introduced to not lose information during the convolution by preserving the spatial dimensions [32].

Pooling layers in the CNN are also used to reduce the input feature map while retaining the most important information. By defining a pooling window along with a stride, which is the number of steps the pooling window moves, the operation extracts the pooling value from each region. Common pooling techniques are Max Pooling, where the maximum value in the region is extracted, or Average Pooling, where the average value in the region is extracted [33].

2.4 Statistical Test

2.4.1 Shapiro-Wilk Test

The Shapiro-Wilk Test is a statistical test used to inspect if a sample of data is normally distributed and measures how closely the sample data reflects a normal distribution. The Shapiro-Wilk test computes a test statistic W to measure how closely the sample reflects a normal distribution. The formula is:

$$W = \frac{\left(\sum_{i=1}^{n} a_i x_{(i)}\right)^2}{\sum_{i=1}^{n}(x_i - \bar{x})^2} \tag{2.13}$$

Where:

- n: Sample size.
- $x_{(i)}$: The i-th order statistic.
- x_i: The i-th sample value.
- $\bar{x}$: The sample mean.
- a_i: Constants derived from the expected values of order statistics for a standard normal distribution, dependent on n.

W ranges from 0 to 1, where a closer value to 1 indicates that the sample matches a normal distribution [34].

2.4.2 Paired T-Test

The paired t-test is a statistical test used to compare the means of two related groups by looking at the distribution of the differences between the observations. This determines if the mean difference is statistically significant. The paired t-test assumes that the differences are approximately normally distributed.

Given paired observations $(X_1, Y_1), (X_2, Y_2), \ldots, (X_n, Y_n)$, the difference is defined for each pair: $D_i = X_i - Y_i$. Then the mean differences are calculated: $\bar{D} = \frac{1}{n}\sum_{i=1}^{n} D_i$, and the standard deviation of differences:

$$s_D = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(D_i - \bar{D})^2} \tag{2.14}$$

These values are used to compute the T-statistic:

$$t = \frac{\bar{D}}{s_D/\sqrt{n}} \tag{2.15}$$

which is then compared with a p-value [35].

2.4.3 Wilcoxon Signed-Rank Test

The Wilcoxon signed-rank test is a statistical test that does not assume any distributions between the samples. It works by calculating the absolute differences for each pair of observations. Then the differences are ranked and assigned to the original signs, and then the positive and negative ranks are summed to compute a test statistic W. These test statistics are then compared with a critical value from a Wilcoxon signed-rank table, which determines if the critical value is lower than a specific p-value [36].

2.5 Regression Metric models

2.5.1 R-Squared

The coefficient of determination or R^2 measures how well the model explains the variance of the target variable, meaning the data it's trying to predict. It ranges from 0, meaning the model does not explain anything, to 1, meaning a perfect explanation [22].

$$R^2 = 1 - \frac{\sum(y_i - \hat{y}_i)^2}{\sum(y_i - \bar{y})^2} \tag{2.16}$$

Where $\bar{y}$ is the mean of the actual salaries, y_i is the actual salary at index i, and $\hat{y}_i$ is the prediction at index i.

2.5.2 Mean Absolute Error

Mean Absolute Error (MAE) measures the average absolute difference between predicted and actual values. This model gives a clear number, in this case the amount of money, that shows the typical error size [22].

$$\text{MAE} = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i| \tag{2.17}$$

Where y_i is the actual value, $\hat{y}_i$ is the predicted value, and N is the number of data points.

2.6 Explainable AI

2.6.1 SHAP

SHapley Additive exPlanations (SHAP) is a model-agnostic framework used for interpreting model predictions by assigning each feature an importance value for a specific prediction. Based on cooperative game theory, SHAP leverages Shapley values to fairly attribute the difference between a model's prediction and its expected output to individual features [37].

For a model f, input x, and M features, the explanation model is:

$$g(z') = \phi_0 + \sum_{i=1}^{M} \phi_i z'_i \tag{2.18}$$

Where $z' \in \{0,1\}^M$ is a binary vector, $\phi_0 = E[f(z)]$, and ϕ_i is the SHAP value for feature i, computed as:

$$\phi_i = \sum_{z' \subseteq x'} \frac{|z'|!(M-|z'|-1)!}{M!} \left[E[f(z) \mid z_S] - E[f(z) \mid z_{S\setminus\{i\}}] \right] \tag{2.19}$$

Here, $z' \subseteq x'$, $|z'|$ is the number of non-zero entries, and $E[f(z) \mid z_S]$ is the expected output given features in S (non-zero indices in z'). SHAP values sum to the prediction:

$$f(x) = E[f(z)] + \sum_{i=1}^{M} \phi_i \tag{2.20}$$

A feature that is constant across all instances (i.e., $x_i = c$ for some constant c) contributes negligibly to the model's prediction, as it does not vary to affect the output. A constant feature has a SHAP value $\phi_i \approx 0$, as its fixed value yields zero marginal contribution, i.e., $E[f(z) \mid z_S] = E[f(z) \mid z_{S\setminus\{i\}}]$ [37].

Chapter 3

Method

3.1 Development Process

The development process will be divided into multiple parts, and each step plays a crucial role in producing a result for this project. The development steps in this project are sequential, meaning the steps have to follow a specific order to implement this project correctly. The steps are:

- Determine Non-Basketball Features
- Data Scraping and Data Entry
- Data Preprocessing
- Model Development
- Testing, Comparing, and Evaluating the Models
- Implement Explainable AI
- Comparative Analysis of the Explainable AI
- Evaluation of Explainable AI

3.1.1 Determining Non-Basketball Features

The process will begin by determining which non-basketball features this project will use to find an answer to the problem statement. The non-basketball features will be determined based on findings in previous research **Section 1.5**, but also the background analysis in **Section 1.1**.

3.1.2 Data Scraping and Data Entry

Then the project need to scrape the data where an algorithm will be developed to automatically extract data by using a browser automation tool. Websites with public data where this tool is not able to operate on will manually extract data through manual data entry.

3.1.3 Data Preprocessing

First, the different files containing the different features will be merged together. Some Regular Expression (Regex) might be required if merging the files is dependent on the feature name of the player. Every string feature will be transformed into a categorical label. Then, the data will be divided into a train and a test set based on split ratios used in similar research. The train and test sets will be duplicated. One train-test set will include only basketball features, and the other train-test set will include both basketball and non-basketball features. For the 1D CNN, the datasets will require reshaping into a three-dimensional space. The data will then be inspected to find rows containing missing values. If the data contains missing values, these rows will be dropped. The column containing player names will be removed as that is not a relevant feature for this case.

3.1.4 Model Development

The development of the statistical models, machine learning models, and deep learning models described in Chapter 2 will mostly be developed by using public libraries. Therefore, examining the existing libraries and determining the modifications for this problem will be the most important part of this step. The models will then perform hyperparameter tuning, ensuring the optimal parameters for the specific problem. To ensure a robust performance for the unseen test data.

3.1.5 Testing, Comparing, and Evaluating the Models

All the models will be trained, validated, and tested with the data, and an output will be produced. The model will produce an output as well with regression metric scores mentioned in **section 2.4.1 and 2.4.2**. To evaluate the models performance, multiple experiments will be conducted. First, a statistical test will be used to understand if an improvement was statistically significant or happened by chance. Further analysis could also be done to evaluate the models performance. If the dataset is considered small, the dataset need to add synthetic tabular rows to make sure the models performance are not due to overfitting.

3.1.6 Implement Explainable AI

An explainable AI model will be used to explain the model output from the best model testing the dataset with non-basketball features. This model will also be developed using open source libraries. So again, examining the existing libraries and determining the modifications for this problem will be the most important part of this step.

3.1.7 Comparative Analysis of the Explainable AI

These explainable AI values will be used to see if those values can reveal differences in the importance of non-basketball features for player salaries between

small and big market teams.

3.1.8 Evaluation of Explainable AI

This explainable AI model will then be evaluated in order to understand the reliability and effectiveness of this specific model's values in identifying the influence on salaries.

3.2 Technical Design

3.2.1 Determining Non-Basketball Features

The non-basketball feature will be based on former NBA player and analyst Avery Johnson's analysis of contributing factors to why it is harder for smaller teams to compete with bigger teams in **section 1.1**. The analysis on how agents influence contracts and how other immaterial traits highlight players from **Section 1.1** will also be used to determine non-basketball features. The related work section of non-basketball features in **section 1.5** will also contribute to implementation of non-basketball features. Table **3.1** displays the non-basketball features used for this project.

Non-Basketball Feature	Short Explanation
Agent	An NBA player's agent
All Star	A player accolade
Draft Position	Which position player was drafted
Number of teams	NBA teams the player played for
Years with team	Years with player's current NBA team
Instagram Followers	Number of Instagram followers
Resigned	If player resigned or not

Table 3.1: Non-Basketball Features that will be used in this project.

3.2.2 Data Scraping and Preprocessing

This project will develop a tool that can manually extract data from sites containing NBA data. This tool will be able to operate on websites with publicly available NBA data and store the data in a comma-separated value (CSV) file. The use of CSV ensures compatibility and consistency with the Python library Pandas. The automation tool will be developed with multiple open-source libraries specialized for this kind of task. The approach for this project will be to develop an automation tool using Selenium and BeautifulSoup. Selenium provides a flexible and scalable solution for web scraping while minimizing manual effort. BeautifulSoup is chosen because it simplifies the navigation and the parsing of Hyper Text Markup Language (HTML) content which will allow easy extraction of structured data.

Every 28 basketball features will be scraped from the official NBA website containing players from the season 2020-2021 [38]. One study predicted different machine learning models on the same data. The salary data will be scraped from HoopsHype, a site with public official NBA data and historic salaries [39]. The non-basketball features *Agent and Resigned* will be scraped from a website called Spotrac, which is a site that specializes in the business side of the four major sports in the United States (American football, Basketball, Ice hockey, Baseball) [40]. NBA players Instagram followers will be scraped from *Popularbasketballers* [41]. The other non-basketball features from websites where automation is infeasible will be manually entered through Data Entry into the CSV file. *Wikipedia* can provide public data for the non-basketball features *Draft Position, Number of Teams, All Star, and Years with Team*.

The CSV files containing the NBA data will then be inspected. There will be one file for the basketball features, one file for agents, one file for the resigned feature, and one file for the Instagram followers. These CSV files containing the NBA data will be merged together based on a common key, in this case, the players name. Regex will be applied to make sure every name has the same format when merging the CSV files. If a player has any special characters, such as Ć, this is transformed into a regular C. Every column that does not include numerical values will either be dropped or replaced with a categorical label. Not a Number (NaN) values will be dropped. The data will also be divided into a train and test set with an 80:20 ratio, since that is the most common split for these type of tasks. Some models require normalization of the data, which will be done in some cases to ensure uniformity. This will be done by using StandardScaler. For the 1D CNN, the train and test sets will be reshaped from a 2-dimensional tabular data into a 3-dimensional format. The train and test sets used for testing performance with basketball features will have the non-basketball features dropped, while the data for testing the non-basketball features will have every feature included.

Since the dataset is relatively small (354 rows), the experiment will use Data Augmentation to make sure that the models performance was not limited by the smaller dataset. This will also be done to ensure the models is not overfitted to the small dataset. This will also be done to confirm that inclusion of non-basketball features are necessary and the potential increase of non-basketball features are not a result of a small dataset. The data augmentation dataset will be implemented by using Conditional Tabular GAN (CTGAN) which is a generative model specifically designed for tabular data. This model is publicly available from the Python library called Synthetic Data Vault (SDV). The CTGANSynthesizer that this project will use can handle mixed data types and imbalanced columns, which is suitable for this project [42].

3.2.3 Model Development and Testing

The models used for this project will be based on how previous research has solved similar problems. **Section 1.5** mentioned that one common approach is to either use a statistical model or a Tree-based model. This project will use *Multiple Linear Regression, Ridge Regression, Lasso Regression, Decision Tree, Random Forest, XGBoost*. Deep learning models like *MLP* will be implemented, and a *1D CNN* will be implemented to explore the effectiveness of 1D CNNs for regression tasks.

Open source libraries such as *scikit-learn*, *TensorFlow*, and *PyTorch* will be used to develop the models. These libraries will be implemented for simplicity and flexibility, and support a wide range of algorithms, and are tailored for performance. This project will also require some hyperparameter tuning to optimize the performance. The approach for this is to use grid search to find the optimal hyperparameters based on a predetermined hyperparameter distance. K-fold cross-validation will be used in order for the model to generalize well to unseen data. For this project, we will use five folds in the cross-validation task.

This project will include an experiment to answer the problem statement question: *Can non-basketball features improve the predictive accuracy of machine learning models for NBA player salary prediction?* To answer this, the models will first be tested with the dataset containing only basketball features and be evaluated with regression metric models described in **section 2.4.1 and 2.4.2**. The output will then be compared with the result from the models tested on the dataset containing both basketball and non-basketball features. Through this, we will find out if non-basketball features improve the predictive accuracy of machine learning models for NBA player salary prediction. If non-basketball features improve the predictive accuracy of machine learning models, in terms of both R^2 and *MAE*, a statistical test will be necessary to claim this. If so, the distribution of the differences will be explored by using a Shapiro-Wilk test, and if the differences are normally distributed, a paired t-test will be done on every model where the MAE values improve after incorporating non-basketball features. If the differences are not normally distributed, a Wilcoxon test will be performed instead.

Additional tests will also be done to make sure that a potential improvement of non-basketball features is not a result of a relatively small dataset. Synthetic tabular rows will be included to the dataset only containing basketball features, this will lead to an addition of 7000 samples in the dataset. If the models yield a similar result as the models trained on the original, smaller dataset, we can show that the original dataset is sufficient for this project and that the non-basketball features provide predictive improvements.

To confirm that the inclusion of non-basketball features improves the model, we will test the theory in multiple steps for those models whose improvement was

statistically significant. This project will test three, five, and seven non-basketball features. For every level of non-basketball features, this project will test every possible combinations of the non-basketball features. Technically, this will be done by using the library *Itertools* which is a module that is a part of Pythons standard library which calculates the number of combinations using the binomial coefficient. The results will then be evaluated in terms of the average *MAE* calculated from every *MAE* from every combination of non-basketball features to make sure if the inclusion of more non-basketball features can improve the predictive accuracy of the ML models.

3.2.4 Explainable AI

The Explainable AI (XAI) model used in this project will be the open-source model SHAP to interpret the best-performing model's predictions, focusing on non-basketball features. SHAP is chosen because the library is compatible and flexible with other libraries used in this project, such as scikit-learn and TensorFlow. SHAP is also a state-of-the-art XAI model, providing insights into complex models in an interpretable way. SHAP's ability to showcase informational visualizations and absolute values for exactly how much a feature contributed to the output on average makes this model relevant for this project. These SHAP values will be used to draw conclusions in order to answer the problem statement question: *Can XAI values assist in revealing differences in the importance of non-basketball features for player salaries between small and big market teams?* To determine big market teams, the project will group the SHAP value for the five biggest teams according to the NBA market valuations in 2021. The five most valuable teams in the NBA year 2021 are *New York Knicks, Golden State Warriors, Los Angeles Lakers, Chicago Bulls, and Boston Celtics*. The SHAP values from the five most valuable teams will be compared with the SHAP values from the five least valuable teams in the NBA year 2021. These teams are *Memphis Grizzlies, New Orleans Pelicans, Minnesota Timberwolves, Detroit Pistons, and Orlando Magic* [43].

SHAP will then be evaluated in order to understand the reliability and effectiveness of SHAP's values in identifying the influence of non-basketball features on player salaries. This will be done by setting a feature in the data to a constant neutral value, in this case, the mean of a random feature, and then inspecting the SHAP values to see if the SHAP values yield around 0 for that feature. **Section 2.5.1** provides information on why a SHAP value should be around 0 in this case. This will be done to answer the problem statement question: *How effectively can an XAI model be validated to quantify the influence of specific features on NBA players salaries in a predictive model?* If the SHAP value yields around 0, then the result suggests that SHAP is functioning correctly in this specific case.

Chapter 4

Implementation

The source code and related materials for this project are available for public use in a dedicated GitHub repository [44]. The project was implemented in Jupyter Notebook 7.2.2, using Python 3.12.7. The computer used in this project runs on a macOS Sonoma 14.4 system, Darwin 23.4.0, with a 64-bit architecture. This computer uses the Central Processing Unit (CPU) Apple M3 Pro and has 18 GB RAM with a 512 GB Solid State Drive (SSD).

Eight ML was implemented in this project. All models were implemented using public libraries. For the models Multiple Linear Regression, Ridge Regression, Lasso Regression, Decision Tree, Random Forest, and Multilayer Perceptron, the library Sci-Kit Learn version 1.5.2 were used. XGBoost was implemented using the public library XGBoost version 2.1.4. The 1D CNN was implemented using TensorFlow, which also integrated the Application Programming Interface (API) Keras. The TensorFlow version used was 2.19.0, and the Keras version used was 3.8.0. Every model except Multiple Linear Regression used Grid Search from Sci-Kit learn to find optimal hyperparameters. Every model used a random seed $= 42$ to ensure reproducibility. The hyperparameter configuration was:

- **Multiple Linear Regression:** No hyperparameters were used, using the default settings from the library for both datasets.
- **Ridge Regression:** Both datasets used an Alpha Value of 100.0. Alpha values 0.01, 0.1, 1.0, and 10.0 were also tested.
- **Lasso Regression:** Both datasets used an Alpha Value of 100.0. Alpha values 0.01, 0.1, 1.0, and 10.0 were also tested.
- **Decision Tree:** The basketball feature dataset used a maximum depth of 7 and minimum samples split of 10. The hyperparameters for the dataset containing both features were adjusted to 5 and 5, respectively. A maximum depth of 3, 10, and None was also tested, and a minimum sample split of 2 was also tested.
- **Random Forest:** The model used for the basketball feature dataset had no maximum depth and 200 estimators, while the model containing the full dataset had a maximum depth of 5 and 200 estimators. 100 estimators and

a maximum depth of 10 were also tested.

- **XGBoost:** Both datasets used a learning rate of 0.1 and maximum depth of 5, along with 200 estimators for the dataset containing basketball features and 100 estimators for the full dataset. A learning rate of 0.01 and a maximum depth of 3 and 7 were also tested.
- **Multilayer Perceptron:** Both datasets used two hidden layers of 50 neurons each, a learning rate of 0.01, and a maximum of 1000 iterations. A learning rate of 0.001 was also tested. Using one hidden layer with 50 or 100 neurons was also tested.
- **1D Convolutional Neural Network:** The 1D CNN model used grid search to find the optimal hyperparameters. The model first use a dense layer to project project the input features into a high-dimensional space and then uses two convolutional layers. The output is then passed through fully connected layers with dropout to prevent overfitting, and the model is optimized using Adam with mean squared error loss. **Listing 4.1** describes the hyperparameters tested and used in the model.

```
param_grid = {
    'model__learning_rate': [0.0001, 0.001, 0.01],
    'model__filters_1': [16, 32, 64],
    'model__filters_2': [8, 16, 32],
    'model__dense_units': [20, 50, 100],
    'model__dropout_rate_1': [0.2, 0.3],
    'model__dropout_rate_2': [0.2, 0.3],
    'model__dropout_rate_3': [0.3, 0.4],
    'batch_size': [16, 32]
}

# Best 1D CNN params
{'batch_size': 16,
 'model__dense_units': 50,
 'model__dropout_rate_1': 0.3,
 'model__dropout_rate_2': 0.3,
 'model__dropout_rate_3': 0.4,
 'model__filters_1': 16,
 'model__filters_2': 8,
 'model__learning_rate': 0.01}
```

Listing 4.1: Hyperparameter search distributions and best parameters for 1D CNN model.

Chapter 5

Results

5.1 Model Results

After running the experiment, the results showed that all statistical models and all Tree-Based models saw an improvement in both R^2 and MAE. For the Deep Learning models, only the Multilayer Perceptron saw a slight improvement, while the 1D CNN performed worse in terms of R^2. **Table 5.1** shows the reported results in this experiment.

Model	Dataset	MAE	R^2
MLR	Basketball Features	$ 4 210 336	0.6694
MLR	All Features	$ 3 722 742	0.7482
Ridge	Basketball Features	$ 4 197 284	0.6927
Ridge	All Features	$ 3 747 815	0.7522
Lasso	Basketball Features	$ 4 184 810	0.6737
Lasso	All Features	$ 3 684 783	0.7499
DT	Basketball Features	$ 3 672 115	0.6404
DT	All Features	**$ 2 898 829**	0.7613
RF	Basketball Features	$ 3 318 613	0.7551
RF	All Features	$ 2 965 484	0.8089
XGB	Basketball Features	$ 3 516 959	0.7043
XGB	All Features	$ 2 910 528	**0.8184**
MLP	Basketball Features	$ 3 436 726	0.7232
MLP	All Features	$ 3 380 124	0.7316
1D CNN	Basketball Features	$ 3 723 348	0.6296
1D CNN	All Features	$ 3 576 049	0.5945

Table 5.1: Comparative Analysis for the ML models used in this experiment.

5.2 Evaluation of the Models

All the ML models, except 1D CNN, saw an improvement in both R^2 and MAE. Some models saw a bigger improvement, while others, like MLP, saw minor improvements. The Shapiro-Wilk test showed that none of the differences were normally distributed at a significance level $\alpha = 0.05$, so a one sided Wilcoxon test was performed instead. The hypotheses for the Wilcoxon test were formulated as follows:

- H_0: Errors from the models only containing basketball features are not larger than the errors from models with both basketball and non-basketball features.
- H_1: Errors from the models only containing basketball features are larger than the errors from models with both basketball and non-basketball features.

At a significant level $\alpha = 0.05$, 5 %, the Wilcoxon P-value were:

Model	Shapiro-Wilk P-value	Wilcoxon P-Value	Significant
MLR	0.0002	0.0097	Yes
Ridge	0.0000	0.0030	Yes
Lasso	0.0002	0.0044	Yes
DT	0.0000	0.1283	No
RF	0.0000	0.0068	Yes
XGB	0.0000	0.0101	Yes
MLP	0.0029	0.3975	No

Table 5.2: The Machine Learning models and their respective p-values

Table 5.2 shows that the null hypothesis can be rejected for the models Multiple Linear Regression, Ridge Regression, Lasso Regression, Random Forest, and XGBoost, but can not be rejected for the models Decision Tree and Multilayer Perceptron.

The artificially generated dataset containing only basketball features were also tested on every model and evaluated by the same regression metrics. **Table 5.3** show the results yielded from the artificial dataset. The results shows that all of the models yielded a lower results than with the original dataset and the linear models together with MLP achieved the best predictive score with the augmented dataset.

Model	*MAE*	R^2
MLR	$ 4 310 212	0.6393
Ridge	$ 4 305 512	0.6407
Lasso	$ 4 310 204	0.6393
DT	$ 4 220 120	0.5227
RF	$ 4 177 958	0.5702
XGB	$ 4 159 098	0.5723
MLP	**$ 3 978 935**	**0.6426**
1D CNN	$ 4 011 558	0.5960

Table 5.3: ML models results on the augmented basketball features dataset.

The final experiment done was to see if the models react differently depending on number of non-basketball features. Each model whose improvement was statistically significant was tested with different number of non-basketball features K. **Table 5.4** displays the different observed *MAE* results at different number of non-basketball features. All models yielded the lowest *MAE* score using the full set of non-basketball features while the *MAE* decreased as the number of non-basketball features increased.

Model	$K = 0$	$K = 3$	$K = 5$	$K = 7$
MLR	$ 4 210 336	$ 3 984 595	$ 3 848 627	**$ 3 722 742**
Ridge	$ 4 197 284	$ 3 983 625	$ 3 857 485	**$ 3 747 815**
Lasso	$ 4 184 810	$ 3 938 244	$ 3 799 827	**$ 3 684 783**
RF	$ 3 318 613	$ 3 136 792	$ 3 033 319	**$ 2 965 484**
XGB	$ 3 516 959	$ 3 269 110	$ 3 136 236	**$ 2 910 528**

Table 5.4: Statistically significant ML models *MAE* results with various number of non-basketball features.

5.2.1 SHAP Values

The best overall model, out of those models where the improvement was statistically significant, in terms of *MAE* and R^2, was XGBoost. Therefore, SHAP values were computed on every feature based on the predicted salary from XGBoost. Two plots were generated, one plot containing the average SHAP values for the contracts given by the five most valuable teams in the NBA according to their team valuation. The other plot contains the average SHAP values for the contracts given by the five least valuable teams in the NBA according to their team valuation.

The SHAP plots show that both the biggest teams and the smallest team have a similar feature contribution distribution. The SHAP plot also shows that each feature contributes differently in bigger teams compared to smaller teams. The SHAP plot also provides an absolute value for how much, on average, that feature

contributed to the NBA contract.

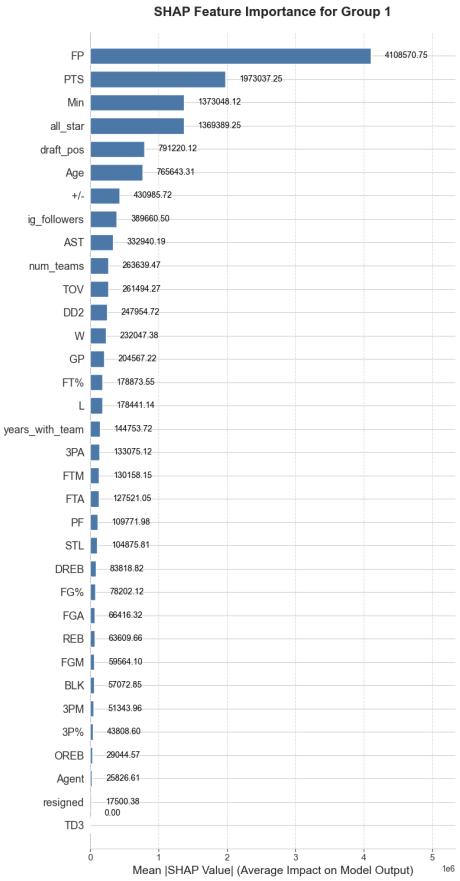

Figure 5.1: SHAP Feature Importance for bigger teams.

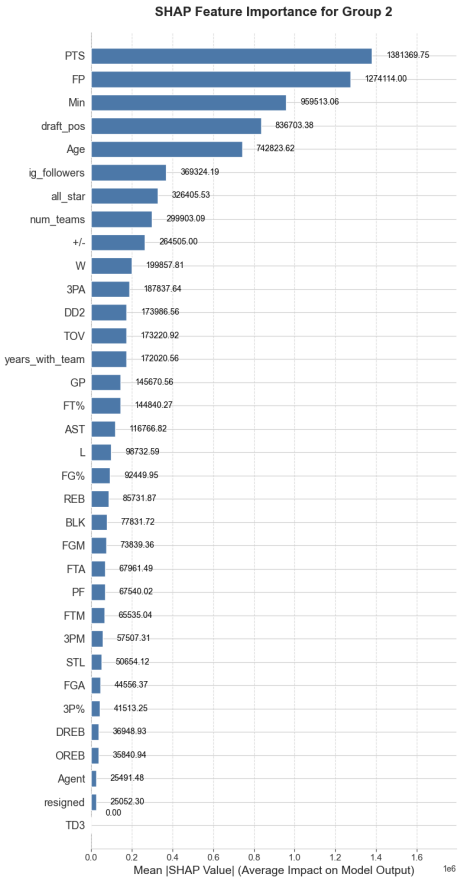

Figure 5.2: SHAP Feature Importance for smaller teams.

Figure 5.3: SHAP Feature Importance Plots for the best performing model XGBoost.

The experiment also inspected the average SHAP value for the basketball features and non-basketball features and then compared the SHAP values. **Table 5.3** shows the observed SHAP difference.

Group	Basketball SHAP value	Non-Basketball SHAP value
High Valuation	$ 423 181	$ 428 855
Low Valuation	$ 246 709	$ 293 556

Table 5.5: The average SHAP value for non-basketball & basketball features for both high valued teams and small valued teams.

The results for feature importance for the individual non-basketball features for

big and small market teams were as follows:

Big Market Teams	Feature	Small Market Teams
25 826.61	Agents	25 491.48
389 660.50	Instagram Followers	369 324.19
263 639.47	Number of Teams	299 903.09
17 500.38	Resigned	25 052.30
791 220.12	Draft Pick	836 703.38
1 369 389.25	All Star	326 405.53
144 753.72	Years with Team	172 020.56

Table 5.6: Average SHAP values for non-basketball features in Big Market Teams and Small Market Teams

To evaluate SHAP, the feature 'FP' was set to a constant value throughout the datasets. The SHAP values revealed that the feature 'FP' is highly important when predicting NBA contracts. The feature 'FP' was set to the mean of the feature for every instance in the dataset. This was done to see how SHAP reacts to a constant feature, and the expectation was that a SHAP value for feature 'FP' should be around 0.

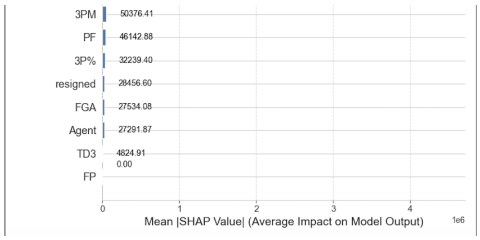

Figure 5.4: SHAP Feature Importance for bigger teams after feature 'FP' was set to a constant value.

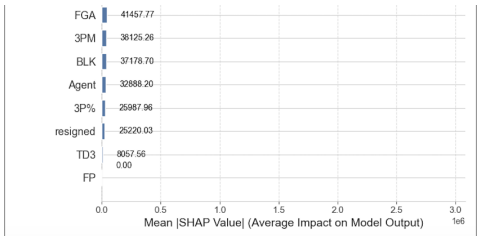

Figure 5.5: SHAP Feature Importance for smaller teams after feature 'FP' was set to a constant value.

Figure 5.6: SHAP Feature Importance Plots after feature 'FP' was set to a mean value.

The SHAP value for feature 'FP' revealed that the feature 'FP' did not contribute anything to the predicted NBA contract for both smaller teams and bigger teams.

Chapter 6

Discussion

6.1 Model Performance

All the models except 1D CNN showed an improvement in both *MAE* and R^2 scores when including non-basketball features in the dataset. Linear models used in the project excel at capturing linear relationships. This can suggest that non-basketball features have a roughly linear correlation with an NBA salary. This argument can also be strengthen by looking at how the *MAE* scored improved linearly for all linear models when choosing how many non-basketball features that was included in the dataset. All linear models showed that the improvement was statistically significant. Ridge Regression was probably improved because the model handles multicollinearity well. In the NBA, a good player should have high values on a basketball feature such as *Points-per-Game* and therefore, will attract more basketball fans, leading to an increase in a non-basketball feature such as *Instagram followers*. Ridge Regression captures this well. Lasso Regression uses L1 regularization, which sets some irrelevant coefficients to zero. If non-basketball features are correlated with salary, Lasso can capture this well. The linear models was also among the best performing models when applying data augmentation on the dataset, outperforming tree-based models. The reason for this could be the augmented data introduced more noise to the dataset that might smoothed non-linearities and diluted the dataset, and therefore, the overall feature-target relationship could have become more linear, which also explain the drastic decrease in performance from the tree-based models. However, this also tells that non-basketball features do improve linear models.

The Tree-Based models (Decision Tree, Random Forest, XGBoost) performed best on average when including non-basketball features. The linear models suggest that non-basketball features might be highly correlated with the salary. However, Tree models are better at capturing the interaction between basketball features and non-basketball features. Non-basketball features probably introduced a non-linear effect by setting different feature thresholds in the trees, leading to capturing relationships and interactions without assuming a specific functional

form. While linear models see that highly correlated basketball features and non-basketball features lead to a high salary, Tree models see that if a player have low basketball features but high non-basketball features (like popularity in social media followers) lead to a higher salary. Therefore, those models performed better. The improvement from the Decision Tree was not statistically significant. This could have been because the model in this case created splits that overfitted the training data and does not capture general patterns.

The Deep Learning models did not perform better when including non-basketball features. The MLP model barely improved, and the 1D CNN model actually performed worse. One reason the models did not improve after including non-basketball features could be due to the dataset size. Deep Learning models may require high amounts of data to learn complex patterns effectively due to their high parameter count weights in multiple layers. However, after including synthetic data in the dataset, the evaluation metrics did not improve although MLP was the best performing model with the larger dataset. This could tell that deep learning models truly benefit from large amount of data. However, the introduced data probably diluted the dataset and made the data not complex enough, which explains the results. The Dataset the models were trained on contained 28 and 35 features, respectively, for 354 instances. This dataset is generally speaking small for Neural Networks, which prevents them from fully capturing complex patterns.

1D CNNs assume local dependencies between adjacent features, meaning the model excels when the information lies in small, localized groups of features and not spread across the entire input. Treating a basketball statline as a sequence might lead to meaningless convolutions if the features are not ordered correctly. The architecture of the 1D CNN is also required to be very task-specific, as mentioned in **Section 1.5**. The 1D CNN model used in this project was tested with multiple hyperparameter combinations and different types and levels of regularization, but also different architectures. While testing different architectures, the evaluation scores showed different scores, and the model used in this project ended up with the best score. Why non-basketball features did not improve the model could have been because only non-basketball features like Instagram followers and All Star gave any meaningful convolutions since those are the only features that give a clear distinguishable idea of players with higher salaries. The other features probably just included unnecessary convolutions to the model which worsen the predictions. However, the model did improve its *MAE* result but not the R^2 result by including non-basketball features. The less meaningful convolutions by adding non-basketball features might added noise or irrelevant information, which does not improve the models ability to fit the overall structure of the data.

6.2 SHAP Performance

The experiment was constructed to explore how non-basketball features differ between smaller and bigger teams based on former NBA player and analyst Avery Johnson's comments and other findings in **Section 1.1**, as well as the related work **Section 1.5**. In general, we found that the average SHAP value for both the basketball features and non-basketball features differs between highly valuable teams and lowly valuable teams. The average SHAP value of the teams with higher values was far higher than the average SHAP value of the least valuable teams. This could indicate that highly valued teams can afford to pay their player more or that better players are generally playing for high valued teams.

Despite one source in the introduction claiming that agents have a lot of power in negotiating contracts, the non-basketball feature *Agent* showed no difference between smaller and bigger teams, meaning that every team does not pay their players based on their agent. This could be because agents have clients in many teams in the NBA, leading to a great relationship with not just the big teams, but also the small teams. However, the analysis that NBA players and their agents are in more control of their own path could still be true and it makes sense as an agents to have a relationship with every team in the league.

The feature *Instagram Followers* showed a slightly bigger impact for bigger teams than smaller teams. This could mean that bigger teams values NBA players with more Instagram followers more than smaller teams, as mentioned in **Section 1.5**, so they can grow their own fanbase but it can also mean that players in bigger teams have more followers due to the teams bigger fanbases.

The feature *Number of teams* suggests that smaller teams value how many teams a player has played for more than bigger teams. This could be about that smaller teams want to find hidden gems, or unwanted players, on the market as Johnson said in his analysis. The feature *Resigned* also suggests a small but not meaningful impact for smaller teams. This could mean that smaller teams have to pay more to retain their players compared to bigger teams.

Draft Pick shows that the position a player was drafted has a greater impact for smaller teams. This could mean that smaller teams either have more players drafted higher or that smaller teams take a bigger financial risk developing young players to compensate for their inability to recruit bigger players in Free Agency. The feature *Years with Team* shows a bigger impact for smaller teams than for bigger teams. This could mean that players in bigger teams play on average fewer years than players on smaller teams. It could also mean that smaller teams pay their own players more as a loyalty bonus or as a way to prevent big teams from signing them.

The feature *All Star* shows a big difference between bigger teams and smaller teams. This could conclude that bigger teams go for All Star players more frequently than smaller teams as mentioned in **Section 1.5**. It could also mean that All Star Players want to play for a big team rather than a small team. However, a difference that large might suggest that there are not enough All Star players in the test set.

6.3 Development Process Reflection

The development process was followed throughout this project. The research in this field was limited. Besides using the same data as another study, this project could not improve the values achieved in this study [10]. This could have been due to using different splits in the data. One novelty of this study was using deep learning models for predicting NBA contracts. With more data, deep learning models like neural networks should be able to predict as well as tree-based models. However, previous research shows that researchers tend to only go with linear models or tree-based models for this type of problem.

6.4 Societal Aspects

With AI tools being mainstream in Western society, it will continue to be a tool used in every aspect of a business, including a basketball team. Some teams might soon use models to value players. Even some companies might incorporate these tools to value their employees. The problem with AI tools is that most people cannot interpret why a prediction was made, leading to people having more confidence in AI tools than they can explain. By including SHAP in NBA salary predictions, we aim for more transparency in data-driven decisions and hope that all companies using data driven decision makers in salary negotiation aims for the same transparency. This can reveal if sensitive attributes like race or socioeconomic status indirectly influence outcomes, potentially reinforcing systemic inequities. Machine Learning models could be an efficient way to value employees, or in this case, NBA players. But data-driven valuation must be done in an ethical way.

Chapter 7

Conclusion

This project researched NBA data in order to answer three research questions. The first question was *Can non-basketball features improve the predictive accuracy of machine learning models for NBA player salary prediction?* This study found that for some common machine learning models, non-basketball features are able to improve the predictive accuracy, but for other models, such as deep learning models, non-basketball features cannot improve the predictive power in this specific case.

The second research question was *Can Explainable AI (XAI) values assist in revealing differences in the importance of non-basketball features for player salaries between small and big market teams?* SHAP in this case can reveal smaller differences between small market teams and big market teams in most of the specific non-basketball features. The SHAP values can reveal that players who have been All Stars are valued higher by bigger teams. It is also clear that highly valuable teams pay more for every feature than less valuable teams.

The final research question was *How effectively can an XAI model be validated to quantify the influence of specific features on NBA players salaries in a predictive model?* The experiment was done by setting a feature to a constant value, and in theory, a SHAP value for that feature should yield around 0. The SHAP value was indeed 0, which is consistent with expectations and suggests that SHAP is functioning correctly in this specific case. However, it does not comprehensively prove SHAP's effectiveness or reliability across all scenarios.

7.1 Future Work

While Tree-Based and Statistical models are the common approach to predict NBA salaries, deep learning models could be even more powerful. Models like Long-Short Term Memory (LSTM) could use data from multiple seasons to predict a salary for every season or 1D CNN models can use game-by-game data to predict an NBA salary.

Since non-basketball features have improved many models, even more non-basketball features can be included in the dataset in order to make accurate valuations of NBA players. More research on the business side of sports could have been conducted in order to improve the study.

Bibliography

[1] Statista, *Total league revenue of the NBA in the United States from 2005/06 to 2023/24*, https://www.statista.com/statistics/193467/total-league-revenue-of-the-nba-since-2005/, Accessed: March 4, 2025, 2025.

[2] Wikipedia Contributors, *Nba salary cap*, https://en.wikipedia.org/wiki/NBA_salary_cap, Accessed: March 4, 2025, 2025.

[3] J. Wimbish, *Nba cba 101: Everything to know about new agreement, from salary cap to free agency and beyond*, https://www.cbssports.com/nba/news/nba-cba-101-everything-to-know-about-new-agreement-from-salary-cap-to-free-agency-and-beyond/, Accessed: March 4, 2025, 2023.

[4] D. Lewis. "The small-market disadvantage: Reflecting on a speech by avery johnson," Daniel Lewis | Sports Analysis - NFL, NBA, MLB, Tennis. (Mar. 2013), [Online]. Available: http://www.daniellewissports.com/the-small-market-disadvantage-reflecting-on-a-speech-by-avery-johnson.html (visited on 04/23/2025).

[5] "Nba contract controversies: A closer look," VDG Sports. (Jan. 2025), [Online]. Available: https://vdgsports.com/nba-contract-controversies-a-closer-look/ (visited on 05/14/2025).

[6] MIT Department of Biology. "Basketball analytics: Investment in nba wins and other successes," MIT News. (Mar. 2025), [Online]. Available: https://news.mit.edu/2025/basketball-analytics-investment-nba-wins-and-other-successes-0325 (visited on 05/14/2025).

[7] H. Wang, A. Sarker, and A. Hosoi, "The effect of basketball analytics investment on national basketball association (nba) team performance," *Journal of Sports Economics*, vol. 0, no. 0, pp. 1–21, 2025, Advance online publication. DOI: 10.1177/152700252412XXXXXX.

[8] Y. Zhao, "Model prediction of factors influencing nba players' salaries based on multiple linear regression," *International Journal of Sports Science*, 2023.

[9] I. Papadaki and M. Tsagris, "Estimating NBA players salary share according to their performance on court: A machine learning approach," pp. 1–19, Nov. 2020, Preprint, accessed: March 04, 2025.

[10] A. Jain, S. Jain, N. M. Pancinovia, and J. P. George, "A non-linear approach to predict the salary of NBA athletes using machine learning technique," in *2022 International Conference on Trends in Quantum Computing and Emerging Business Technologies (TQCEBT)*, Accessed: March 04, 2025, CHRIST, Pune Lavasa Campus, India: IEEE, Oct. 14–15, 2022, pp. 1–5, ISBN: 978-1-6654-5361-5. DOI: 10.1109/TQCEBT59414.2022..

[11] Y. Wang, "NBA player salary projections based on gradient boost in 2022-23 season," *WEP Transactions on Computer Science and Intelligent Systems Research*, vol. 5, pp. 236–241, 2024, Accessed: March 04, 2025, ISSN: 2960-1800. DOI: 10.62931/tcsisr.v5i.157. [Online]. Available: https://doi.org/10.62931/tcsisr.v5i.157.

[12] J. Chen, H. Huang, X. Huang, Q. Sun, and W. Fang, "Building graduate salary grading prediction model based on deep learning," *Intelligent Automation Soft Computing*, vol. 27, no. 1, pp. 53–64, 2021. DOI: 10.32604/iasc.2021.013935. [Online]. Available: https://www.techscience.com/iasc/v27n1/41144.

[13] L. Chen, Y. Sun, and P. Thakuriah, "Modelling and predicting individual salaries in united kingdom with graph convolutional network," in *Hybrid Intelligent Systems*, ser. Advances in Intelligent Systems and Computing, A. Madureira, A. Abraham, N. Gandhi, and M. Varela, Eds., vol. 923, Springer, 2020, pp. 65–75. DOI: 10.1007/978-3-030-14347-3_7. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-030-14347-3_7.

[14] C.-I. Cira, A. Diaz-Álvarez, F. Serradilla, and M.-Á. Manso-Callejo, "Convolutional neural networks adapted for regression tasks: Predicting the orientation of straight arrows on marked road pavement using deep learning and rectified orthophotography," *Electronics*, vol. 12, no. 18, p. 3980, 2023. DOI: 10.3390/electronics12183980. [Online]. Available: https://www.mdpi.com/2079-9292/12/18/3980.

[15] R. Poli, G. Rossi, and R. Besson, "Football agents in the biggest five european football markets: An empirical research report," CIES Football Observatory, University of Neuchâtel, Tech. Rep., Feb. 2012. [Online]. Available: http://www.football-observatory.com/IMG/pdf/report_agents_2012-2.pdf.

[16] L. Qian, "Influencing factors of NBA player's salary based on mixed linear model," *Highlights in Science, Engineering and Technology*, vol. 88, pp. 260–266, 2024, Accessed: March 04, 2025. DOI: 10.54097/hset.v88i.15792. [Online]. Available: https://doi.org/10.54097/hset.v88i.15792.

[17] J. Cleary, "Nba team valuation: What drives value?" Honors Thesis, Peter T. Paul College of Business and Economics, May 2023. [Online]. Available: https://scholars.unh.edu/honors/730.

[18] C. C. Lee, R. Zhang, R. Lomotey, J. Watkins, and Y. Huang, "Examining the impact of social media following on player salary in the national basketball association: A multivariate statistical analysis," *Journal of Applied Business and Economics*, vol. 25, no. 1, pp. 280–288, 2023.

[19] D. d. A. Costa, J. M. Fechine, J. R. d. S. Brito, J. V. R. Ferro, E. d. B. Costa, and R. V. V. Lopes, "A machine learning approach using interpretable models for predicting success of ncaa basketball players to reach nba," in *Proceedings of the 16th International Conference on Agents and Artificial Intelligence (ICAART 2024)*, INSTICC, vol. 1, Rome, Italy: SciTePress, Feb. 2024, pp. 758–765, ISBN: 978-989-758-680-4.

[20] Y. Ouyang, X. Li, W. Zhou, W. Hong, W. Zheng, F. Qi, and L. Peng, "Integration of machine learning xgboost and shap models for nba game outcome prediction and quantitative analysis methodology," *PLOS ONE*, vol. 19, no. 7, e0307478, Jul. 2024, ISSN: 1932-6203. DOI: 10.1371/journal.pone.0307478. [Online]. Available: https://doi.org/10.1371/journal.pone.0307478.

[21] D. C. Montgomery, E. A. Peck, and G. G. Vining, *Introduction to Linear Regression Analysis*, 6th. Hoboken, NJ: Wiley, 2021.

[22] G. James, D. Witten, T. Hastie, and R. Tibshirani, *An Introduction to Statistical Learning: with Applications in R*, 2nd. New York, NY: Springer, 2021. DOI: 10.1007/978-1-0716-1418-1.

[23] IBM, *What is bagging?* https://www.ibm.com/think/topics/bagging, Accessed: 2025-04-25, 2023.

[24] Simplilearn, *Bagging in machine learning: A comprehensive guide*, https://www.simplilearn.com/tutorials/machine-learning-tutorial/bagging-in-machine-learning, Accessed: 2025-04-25, 2023.

[25] GeeksforGeeks, *Random forest algorithm in machine learning*, https://www.geeksforgeeks.org/random-forest-algorithm-in-machine-learning/, Accessed: 2025-04-25, 2023.

[26] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[27] GeeksforGeeks, *Ml | gradient boosting*, https://www.geeksforgeeks.org/ml-gradient-boosting/, Accessed: 2025-04-25, 2023.

[28] C. Li, *A gentle introduction to gradient boosting*, College of Computer and Information Science, Northeastern University, https://github.com/cheng-li/pyramid, Accessed: 2025-04-25, 2015.

[29] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, San Francisco, California, USA: Association for Computing Machinery, 2016, pp. 785–794. DOI: 10.1145/2939672.2939785.

[30] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA: MIT Press, 2016, ISBN: 978-0-262-03561-3.

[31] J. Zhao, X. Mao, and L. Chen, "Speech emotion recognition using deep 1d & 2d cnn lstm networks," *Biomedical Signal Processing and Control*, vol. 47, pp. 312–323, 2019, ISSN: 1746-8094. DOI: 10.1016/j.bspc.2018.08.035.

[32] GeeksforGeeks. "Cnn - introduction to padding." Accessed: 2025-04-25, GeeksforGeeks. (2021), [Online]. Available: https://www.geeksforgeeks.org/cnn-introduction-to-padding/.

[33] GeeksforGeeks. "Cnn | introduction to pooling layer." Accessed: 2025-04-25, GeeksforGeeks. (2025), [Online]. Available: https://www.geeksforgeeks.org/cnn-introduction-to-pooling-layer/.

[34] S. S. Shapiro and M. B. Wilk, "An analysis of variance test for normality (complete samples)," *Biometrika*, vol. 52, no. 3/4, pp. 591–611, 1965. DOI: 10.2307/2333709. [Online]. Available: https://doi.org/10.2307/2333709.

[35] statsTutor, *Paired t-test*, Accessed: 2025-04-30, 2025. [Online]. Available: https://www.statstutor.ac.uk/resources/uploaded/paired-t-test.pdf (visited on 04/30/2025).

[36] F. Wilcoxon, "Individual comparisons by ranking methods," *Biometrics Bulletin*, vol. 1, no. 6, pp. 80–83, 1945. DOI: 10.2307/3001968.

[37] S. Lundberg and S.-I. Lee, *Shap: Shapley additive explanations*, Accessed: 2025-01-30, 2020. [Online]. Available: https://shap.readthedocs.io/en/latest/.

[38] National Basketball Association. "Nba player traditional statistics: 2020–21 regular season." Accessed on May 15, 2025, at 10:20 AM CEST. (2021), [Online]. Available: https://www.nba.com/stats/players/traditional?Season=2020-21&SeasonType=Regular+Season (visited on 05/15/2025).

[39] "2021/22 salaries of all nba players," HoopsHype. (2021), [Online]. Available: https://hoopshype.com/salaries/players/2021-2022/ (visited on 04/29/2025).

[40] Spotrac. "Spotrac: Sports contracts, salaries, caps, bonuses, & transactions," Spotrac. (2025), [Online]. Available: https://www.spotrac.com (visited on 05/16/2025).

[41] "Popular basketballers," Popular Basketballers. (2025), [Online]. Available: https://www.popularbasketballers.com (visited on 04/29/2025).

[42] The SDV Team. "CTGANSynthesizer – SDV Documentation." Accessed: 2025-05-28, DataCebo. (2024), [Online]. Available: https://docs.sdv.dev/sdv/single-table-data/modeling/synthesizers/ctgansynthesizer.

[43] Hoops Rumors Staff. “Forbes releases 2021 NBA franchise valuations.” (Feb. 2021), [Online]. Available: https://www.hoopsrumors.com/2021/02/forbes-releases-2021-nba-franchise-valuations.html (visited on 05/15/2025).

[44] E. Falk, *Bachelor’s thesis*, 2025. [Online]. Available: https://github.com/Emil1Falk/Bachelors-Thesis (visited on 04/29/2025).

Appendix A

Extra Material

A.1 Figures

- **Figure 2.1:** *A visual representation of a Decision Tree containing different feature thresholds to reach a prediction.*: A tree showing survival of passengers on the Titanic ('sibsp' is the number of spouses or siblings aboard). Based on a previous work by Stephen Milborrow to better reflect the current description of a decision tree on https://en.wikipedia.org/wiki/Decision_tree_learning. Licensed under CC BY-SA 4.0 (https://creativecommons.org/licenses/by-sa/4.0/). No changes were made to the original image.
- **Figure 2.2:** *A visual representation of a Multilayer Perceptron*: Multi-Layer Perceptron (MLP) Neural Network. From Left to right: Inputs, Weights, Perceptron Neurons in Hidden Layer, Weights and Output Layer" by Arash Yoosefdoost, 5 May 2022, sourced from https://doi.org/10.13140/RG.2.2.35394.04800. Licensed under CC BY-SA 4.0 (https://creativecommons.org/licenses/by-sa/4.0/). No changes were made to the original image.
- **Figure 2.3:** *A visual representation of a 1D CNN with convolutional layers.*: Figure from 'One-dimensional convolutional neural networks for low/...' published in ScienceDirect, https://www.sciencedirect.com/science/article/pii/S1746809421008004#f0015. Licensed under CC BY-NC-ND 4.0. Used for non-commercial purposes. No changes were made to the original image.
- Figures not mentioned here are my own figures.