

NBA PLAYER POSITION PREDICTION

Hackathon 2020 – Team 24

Team Members:

Jemima Esinu Denteh

Andre Afful

David Selorm Gebe

Player Line-up Recommendation

**TABLE OF CONTENT**

[INTRODUCTION 3](#_Toc51278246)

[PROBLEM STATEMENT 3](#_Toc51278247)

[RESEARCH QUESTIONS 4](#_Toc51278248)

[RESEARCH OBJECTIVES 4](#_Toc51278249)

[AIM OF THE PROJECT 4](#_Toc51278250)

[METHODOLOGY 4](#_Toc51278251)

[DATA DESCRIPTION 5](#_Toc51278252)

[DATA CLEANING 6](#_Toc51278253)

[DATA ANALYSIS 7](#_Toc51278254)

[EXPLORATORY DATA ANALYSIS 7](#_Toc51278255)

[THE MACHINE LEARNING PROCESS 10](#_Toc51278256)

[RESULTS 12](#_Toc51278258)

[CONCLUSION AND RECOMMENDATION 13](#_Toc51278259)

# INTRODUCTION

The National Basketball Association (NBA) is an elite male basketball league of America. It is made up of 30 teams and is one of the 4 largest professional sports leagues in the US and Canada, 29 in the United States and one in Canada. It is the world's largest professional men's basketball league. The regular season of the NBA runs between October and April with 82 matches each. The playoff tournament of the League is scheduled for June. By 2020, average annual earnings per player will make NBA players the highest paying athletes in the world. In a traditional basketball game, every team has five players, and each player has his / her roles and responsibilities.

They are defined as the point guard (PG), the shooting guard (SG), the small forward (SF), the power forward (PF), and the center (C). The point guard is the actual leader of the team on the court. This position requires substantial ball-handling skills and the ability to facilitate the team during a play. The shooting guard, as the name implies, is often the best shooter, as well as being capable of shooting accurately from longer distances. The small forward often has an aggressive approach to the basket when handling the ball. The small forward is also known to make cuts to the basket in efforts to get open for shots. The power forward and the Center make up the frontcourt, often acting as their team's primary rebounders or shot blockers, or receiving passes to take inside shots. The Center is typically the larger of the two.

Previously, players were assigned a single part to play and would only play that part and concentrate on their job, but since then, basketball has changed a lot, and every player has had two or three roles at a time these days. This is due to the nature of the sport where players move about and play fluidly.

This project seeks to build a model to predict the positions (PG, SG, SF, PF, C) of NBA players based on their performance metrics. The metrics include: Player Efficiency Rating (PER), True Shooting (TS%), a measure of how efficient a player is shooting a ball, 3-Point Attempt Rate (3Par), Free Throw rate (FTr), ORB%, DRB%, TRB%, AST%, STL%, BLK%, TOV%, USG%, OWS, WS, WS/48, OBPM, DBPM, BPM, VORP, FG, FGA, FG%, 3P, 2P, 2PA, 2P%, eFG%, FT, FTA, FT%, ORB, DRB, TRB, AST, STL, BLK, TOV, PF, PTS, height and weight.

.

# PROBLEM STATEMENT

The NBA stores tons of data season after season and game after game. Analytics have taken over the game to the point where decisions ranging from the scope of business to actual game play are being backed by the numbers, not just philosophy and the eye-test. This modern era of analytics provides valuable assets using data that could aid the coaching staff of NBA Teams today to effectively line up their players against the different grades of opponents they face night after night and week after week.  
We aim to provide an algorithm that gives the coaching staff of NBA Teams a recommended starting line-up based on the players' performance statistics over the season.

# RESEARCH QUESTIONS

1. Do a player’s physical attributes (height & weight) influence his position?
2. Which positions are given more minutes on court?
3. Which positions are more efficient scorers?
4. Which positions are more impactful?
5. Which positions contribute to more wins?
6. What is the effect of Field goal percentage (FG%) on height?
7. How does age affect player position?
8. MP (Minutes played) on Position
9. FG% on position
10. FT% (Free throw percentage) on position
11. WS (Win shares) on Position
12. OWS (Offensive Win Shares) on Position
13. DWS (Defensive Win Shares) on Position
14. PER (Player Efficiency Rating) on Position

# RESEARCH OBJECTIVES

1. Classify players into the five positions based on their performance metrics

2. To determine the influence of time on definition of positions.

# AIM OF THE PROJECT

To provide an algorithm that recommends the “best case scenario” roster line-up of a basketball team. This is achieved by taking into consideration the accumulated season performance statistics of players in the NBA, and predicts best-suited player positions to match the measured performance statistics.

# METHODOLOGY

The goal of this project is to train a neural network with this data, to try to predict the position of each player.

Ideally, in cleaning the data, the positions should have been transformed into numbers from 1 to 5 (1 for a PG, 2 for a SG, 1.5 for a PG-SG, and so on, until 5 for a C), and use the network for regression instead of classification. But we wanted to see if the network was able to predict labels as "SG-PF", so we decided to work with the categorical labels. Another reason is that this makes this study easily extensible to other areas.  
   
To carry this out, we made use of the Azure Machine Learning Designer, accessible under the Enterprise edition of the Azure Machine Learning studio.

# DATA DESCRIPTION

There are two different datasets; the first one is titled Players and the second is titled Seasons\_stats. The dataset contains data from 1950 to 2017. The Players dataset has 8 columns, 3922 and contains the bio (height, weight, college, year of birth, birth city and birth state) of players. The Seasons\_stats dataset has 53 columns, 3922 rows and contains PER, PS%, 3Par, FTr, ORB%, DRB%, TRB%, AST%, STL%, BLK%, TOV%, USG%, OWS, WS, WS/48, OBPM, DBPM, BPM, VORP, FG, FGA, FG%, 3P, 2P, 2PA, 2P%, eFG%, FT, FTA, FT%, ORB, DRB, TRB, AST, STL, BLK, TOV, PF, PTS.

The two datasets were merged into a single dataset and both have strings and numerical data. The target feature (position/POS) is categorical.

Players dataset

1. Year: NBA Season Year
2. Player: Player full name
3. Pos: Player Position (PG – Point Guard, SG – Shooting Guard, SF –Small Forward, PF-Power Forward, C – Center)
4. Age: Player Age
5. Tm: Team
6. G: Games Played
7. GS: Games Started
8. MP: Minutes Played
9. PER: Player Efficiency Rating
10. TS%: True Shooting Percentage
11. 3Par: 3-point attempts rate
12. FTr: Free Throw rate
13. ORB%: Offensive Rebound Percentage
14. DRB%: Defensive Rebound Percentage
15. TRB%: Total Rebound Percentage
16. AST%: Assist Percentage
17. STL%: Steal percentage
18. BLK%: Block percentage
19. TOV%: Turnover percentage
20. USG%: Usage Percentage
21. OWS: Offensive Win Shares
22. DWS: Defensive win Shares
23. WS: Win Shares
24. WS/48: Win Shares per 48 minutes
25. OBPM: Offensive Box Plus/Minus
26. DBPM: Defensive Box Plus/Minus
27. BPM: Box Plus/Minus
28. VORP: Value Over Replacement Player

 Seasons\_stats dataset

1. Player: player name
2. height: player height in centimeters
3. weight:  player weight in Kg
4. college: Player alma mater
5. born: year player was born
6. birth\_city: city of birth
7. birth\_state: state of birth

# DATA CLEANING

* Keep only players with more than 400 minutes for each season (with 82 games regular season, that is around 5 minutes per game. Players with less than that will distort the analysis as they do not have enough body of work to conduct an accurate analysis).
* Some Player names had to have the \* sign replaced.
* For the stats that represent total values (others, such as TS%, represent percentages), we will take the values per 36 minutes. The reason is to judge every player according to his characteristics, not the time he was on the floor and this evens the playing field by removing the minutes bias. If this is not done, the statistics may favor players who spent more minutes playing in games than players who played fewer minutes.

# DATA ANALYSIS

# EXPLORATORY DATA ANALYSIS

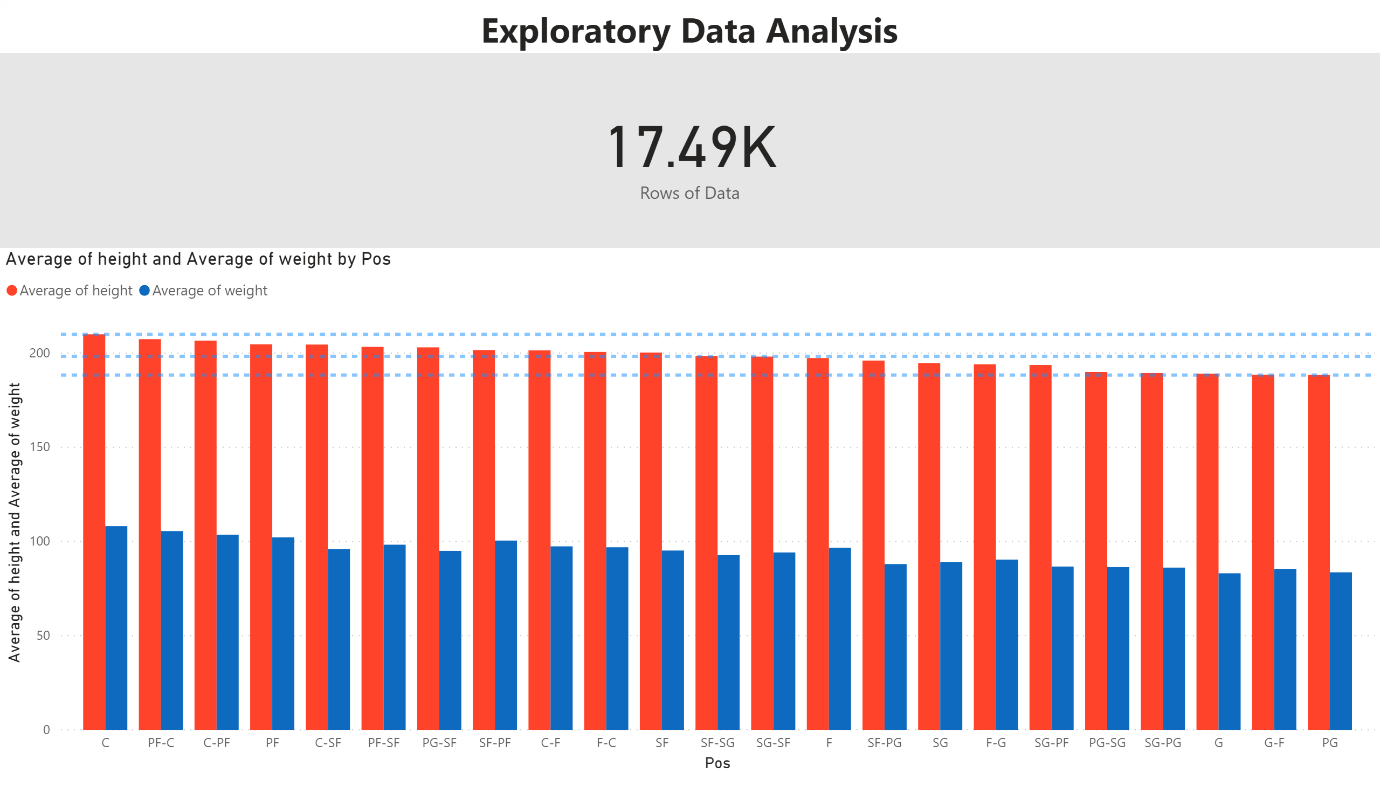
[](https://app.powerbi.com/groups/me/reports/55d8fd43-8065-465b-b2fb-184b8b1a28e5/ReportSection?pbi_source=PowerPoint)

Figure 1: The insights generated from this graph indicate that the centre position has the tallest and heaviest players

**Height/Weight on Position**

Height

* Average height for interior positions (C, PF) is above 200cm
  + Probably for better rebounding and interior defense (shot blocking)
* Average height for positions on the perimeter (PG, SG) are just around 190cm
  + Probably for better perimeter defense and mobility on the perimeter
* Average height for the positions on the wing (SF) are just around 200cm
  + Probably for better rebounding, defense and offense (Slashing to the rim)

Weight

* Average weight for interior positions (C, PF) is above 100kg
  + Probably for rebounding battles and interior defense (post defense)
* Average weight for positions on the perimeter (PG, SG) are just around 80kg-90kg
  + Probably for perimeter defense, mobility on the perimeter
* Average weight for the positions on the wing (SF) are just under 100kg
  + Probably for better versatility (rebounding and defense, offense)

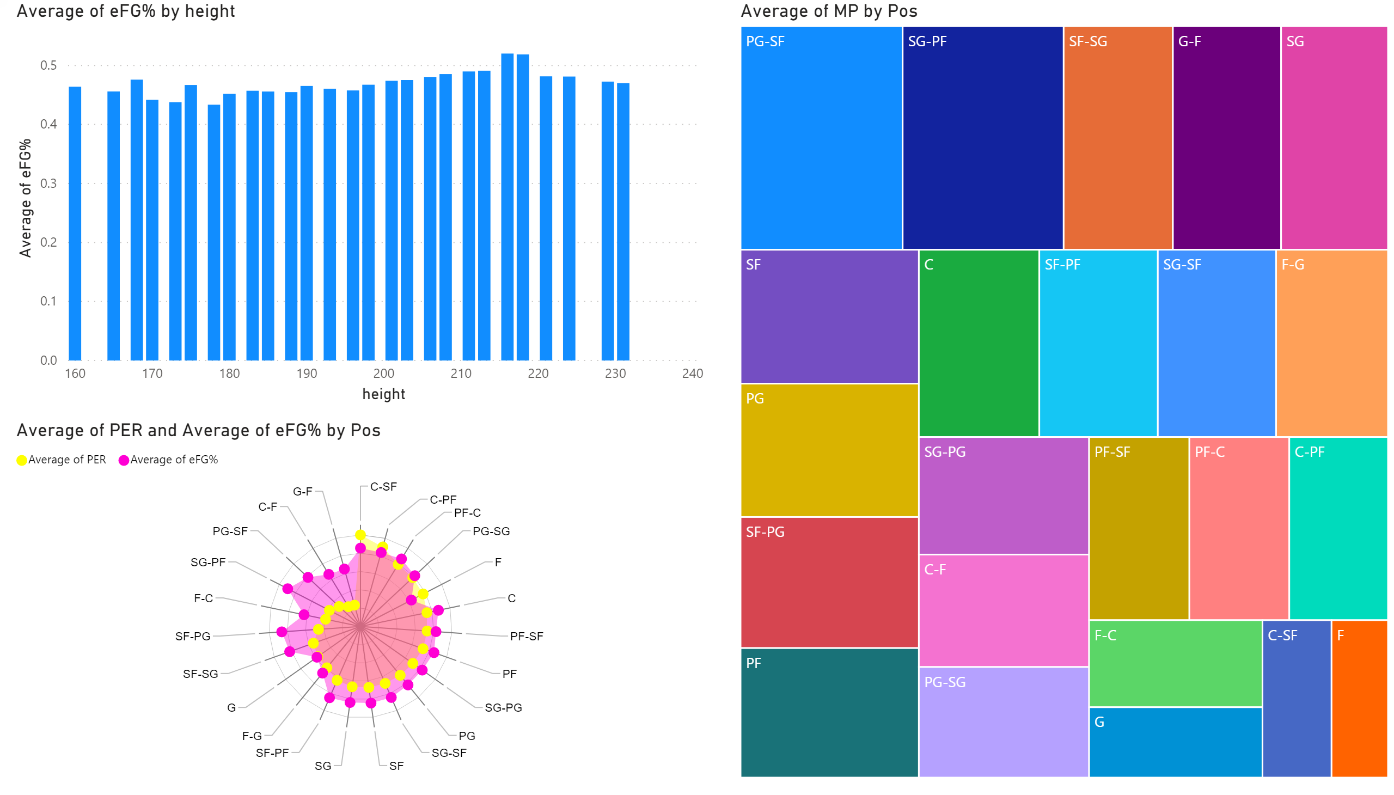
[](https://app.powerbi.com/groups/me/reports/55d8fd43-8065-465b-b2fb-184b8b1a28e5/ReportSectione19ff38f7f07d5cb9e4f?pbi_source=PowerPoint)

Figure 2: Visualization of effective Field Goal percentage by Height and Player Efficiency Rating and Minutes Played by Position.

**Efg% by height**

* Heights with over 200cm (interior players and wing players) seem to be more efficient scorers on average.
  + Because they play closer to basket compared to other players on the floor.
* Heights in the range of 190cm and below are not as efficient but the trend is not uniform either (not steadily increasing or decreasing).

Maybe because they shoot much further from the basket and they probably took lesser 3-point shots

**PER and EFG% on Position**

* Interior players record more efficiency in both scoring and overall influence on the game
  + Probably due to Rebounds, Defense, Scoring
* More versatile players record higher efficiency ratings
  + Probably for because they scoring from all over the floor (interior, free throws, 3-point range)
* It appears Guards are the least efficient positions in terms of scoring but over time the PG show more efficiency in scoring:
  + Probably for because of the reliance on 3-point shooting
  + Probably for because of a change in skill focus (shooting from 3-point range)

**MP on Position**

Dynamic players that can play multiple positions are given more minutes in games

* + Probably due to better Skillset

Players that play positions that are not so far apart are not that favored much in minutes

* + Probably because their easily replaceable by other players that are probably specialized in those same positions

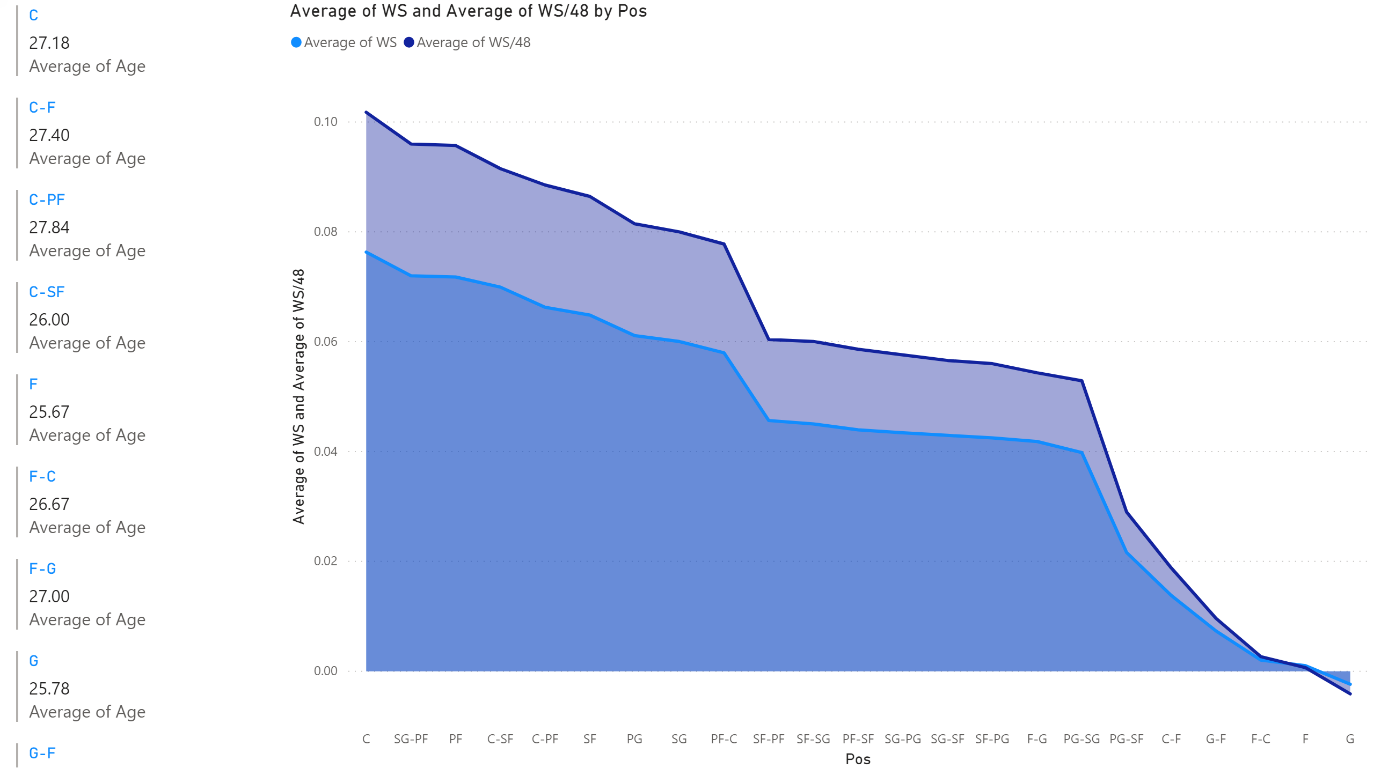
[](https://app.powerbi.com/groups/me/reports/55d8fd43-8065-465b-b2fb-184b8b1a28e5/ReportSectiona28999f252b791e7d843?pbi_source=PowerPoint)

Figure 3: Visualization of Win Shares and Win Shares per 48 minutes by Position

**WS and WS/48 by position**

The Big men dominate the court and contribute to wins more than any other position

The least position that contributes to wins in a season is the Guard position

**IMPLICATION**:

* An NBA player’s physical attributes (height and weight) influences his position.
* Heights with over 200cm (interior players and wing players) seem to be more efficient scorers on average. Because they play closer to the basket. ​
* For efficient scoring, focus your offense through your interior offensive players (PF, C).
* Dynamic players are valuable assets on the team. ​
* The Big men dominate the court and contribute to wins more than any other position. You need 'Big Men' to win games.

# THE MACHINE LEARNING PROCESS After data exploration and analysis was carried out, based on the problem statement and questions asked, a machine learning that could classify players based on their bio statistics and their in-game statistics was suggested. The algorithm was specifically identified to be a Multiclass Neural Network. The process is explained below.

# We created a machine learning workspace in an Azure Subscription. After the workspace was created, we launched the Machine Learning studio to give us access to manage our resources. To train and deploy models using the Machine Learning Designer, we had to create compute resources on which to run the training process, test the model and host the model in a deployed service. To train and deploy models using Azure Machine Learning designer, you need compute on which to run the training process, test the model, and host the model in a deployed service. Compute targets are cloud-based resources on which you can run model training and data exploration processes. We created compute instances which are development workstations that data scientists can use to work with data and models, compute clusters which are scalable clusters of virtual machines for on-demand processing of experiment code and inference clusters which are deployment targets for predictive services that use your trained models.

To train a classification model, a dataset that includes historical *features* (characteristics of the entity for which you want to make a prediction) and known *label* values (the class indicator that you want to train a model to predict) was required.

Our dataset was uploaded from the dataset page of the machine learning studio,

For azure machine learning designer, it required us to create a pipeline with our dataset.

In creating our pipeline, we specified the compute target which we had created previously. On the left side of the pane, we selected our dataset that we had had uploaded from the dataset module and dragged it unto the canvas. We then selected "Select Column in Dataset" from the Data Transformation Module. The purpose of this is to selects columns to include or exclude from a dataset in an operation. In our case, columns such as 'Player' and 'Year' were not needed in training the model, and thus were removed. We then chose 'Normalize Data' also from the Data Transformation module and this rescales numeric data to constrain dataset values to a standard range.

When training a machine learning model, it is sometimes possible for larger values to dominate the resulting predictive function, reducing the influence of features that on a smaller scale. Typically, data scientists mitigate this possible bias by *normalizing* the numeric columns so they're on similar scales. We then run the pipeline to see if the selected data transformation modules have been affected on our dataset. After we used data transformations to prepare the data, we used it to train our machine learning model. In training the model, it is commonplace to make a subset of the data available for training and the other subset available for testing.

From the Data Transformation module, we selected 'Split Data' and configured it so our splitting mode was 'split rows' and the fraction of rows in our first output dataset was 0.7, meaning 70% to be used for training and the remaining 30% for testing.

A 'Train Model' module was dragged onto the canvas, this trains a classification or regression model in a supervised manner and as such was perfect for the model. The model being trained would predict Positions and so we selected the 'Train Model' and modified its settings to set the Label Column to 'Pos' which is the column containing the positions of the players. The ‘Pos’ label the model would predict was a class, so we needed to train the model using a *classification* algorithm. Specifically, there were multiclass classes, so a multiclass*classification* algorithm was required. Under the Machine Learning Algorithms section, and under Classification, a Multiclass Neural Network module was dragged to the canvas, to the left of the Split Data module and above the Train Model module. Then connected its output to the Untrained model (left) input of the Train Model module. The Multiclass Neural Network was chosen because it creates a multiclass classification model using a neural network algorithm.

A neural network is a set of interconnected layers. The inputs are the first layer, and are connected to an output layer by an acyclic graph comprised of weighted edges and nodes.

Between the input and output layers you can insert multiple hidden layers. Most predictive tasks can be accomplished easily with only one or a few hidden layers. However, recent research has shown that deep neural networks (DNN) with many layers can be effective in complex tasks such as image or speech recognition. The successive layers are used to model increasing levels of semantic depth.

The relationship between inputs and outputs is learned from training the neural network on the input data. The direction of the graph proceeds from the inputs through the hidden layer and to the output layer. All nodes in a layer are connected by the weighted edges to nodes in the next layer.

To compute the output of the network for a particular input, a value is calculated at each node in the hidden layers and in the output layer. The value is set by calculating the weighted sum of the values of the nodes from the previous layer. An activation function is then applied to that weighted sum.

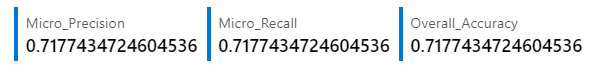
To test the trained model, it was to be used to *score* the validation dataset we held back when we split the original data - in other words, predict labels for the features in the validation dataset. From the Model Scoring & Evaluation section, a Score Model module was dragged to the canvas, below the Train Model module.



*Figure 4: Pipeline created to build and test the predictive model.*

# RESULTS

From the evaluation of the model, the primary metric was Accuracy and it had a score of approximately 0.72. We also had other metrics, precision and recall which had the same score.



Values closer to 1 suggest that the model is predicting correctly. Hence, an accuracy greater than 0.7 would have been preferred. However, from observation, and also looking at the Scored Labels, it was realized that the model predicted, for example 'C' for positions that were actually PF and vice versa.  C which stands for Centers and PF, Power Forwards have similar roles with only a slight difference, meaning players who play as Centers can alternate as power forwards and vice versa. Similar instances were observed throughout the Scored Labels where the model predicted positions that were identified to be alternate positions that could be taken up by the players by virtue of their skillset and bio statistics. From these results, it could then be inferred that the model could predict the best position of the players in which they would thrive.

# CONCLUSION AND RECOMMENDATION

Putting this into context, this could help coaches overcome situations where they need to adjust the roles of certain players in order to help them compete against teams with special skillsets. As observed, even though the model’s accuracy was not as high as one would expect, it satisfies the intended goal of the problem statement.

Going forward, the team looks forward to building an application that uses less features to make these predictions since this was a limitation.