École nationale de la statistique
et de l'analyse de l'information

Graphical Models

Application Project

Bayesian Network to predict the probability of a pass being completed in football

Léon Kourtis

Professor :
Eftychia Solea

2022/2023

Introduction

With the rise of digital technology and the digital revolution, the world of sports is not spared from evolutions. Indeed, the digital transformation is reaching the field of sport in all its dimensions. It started with the video analysis of performance but begins to reach more and more the field of data science. The latter, with the automation of data acquisition, allows progress in tactical analysis but also in physical data and recruitment. The stakes are the improvement of performance, the recruitment of players thanks to statistics, the prevention of fatigue and injuries, the follow-up of players in the training center etc...

Today we gonna see further tactical data. Indeed, the main components of football is passing networks that allow you to keep the possession of the ball or to score a goal. The importance of the passes is such that it is necessary to have a cohesion between the eleven players present on the field. Data analysis can provide insights and can explain certain aspects of the game that increase the overall performance of the team.

Today the question would be on the passes and what affects them. The first idea that we can have are that these passing networks depends on the players involved in it, by the passer and the receiver. Moreover, it depends also on the difficulty of the pass : by difficulty we can think about the location of the pas ser and of the future receiver. We can think also of the distance or the angle of the pass, and the number of opponents between them. Furthermore, if the opposite team applicates a pressing on the team, or the action of trying to cut the trajectories of short passes by isolating the ball carrier

The Bayesian Networks seems to be a good way to model our problem. It permits a compact and intuitive representation to make easy understanding of the dependancies in a model. Moreover it has efficient inference and handles incertainity.
One objective is also to create a metric "expectedpass" the same way we can see in the litterature with the "expectedgoals"[1]. The expectedpass would be the probability of a pass being succeed. For example, a pass involving two players really close from each other would have a expectedpass of 0.95, that means 95% of chance to succeed the pass.

In this paper, we will first detail the variables used in a Bayesian model and their relationships, then we will define the probabilies of the distributions and estimating them and finally we will make predictions and test the performance of our model. We will also propose another network which performs better predictions for an estimation of "expectedPasses".

Similar analysis could have been done on other team sports data, such as Basketball or Volleyball. Our choice is on football analysis, which is more complete and have more references on data analysis compared to other sports.

1 The data

The data used in this project is the data of Montpellier Hérault Sport Club during the actual season in football in France 1st division, currently called Ligue 1. At the date of the project, the team has played 17 championship games and the data we will use corresponds to the data of this project.
I have already worked on similar data during my internship with the data of last year's season. I worked on visualisations about the variables that influences a ball possession.
The data is not available in the internet, and although I don't have instructions about it, it may be confidential, here I will share results and how I obtained them.
The data corresponds to all the events during a game, with encoding for passes,shots,crosses,fouls,etc...
It comes from the society Opta StatsPerform.

2 The Bayesian Network

2.1 The variables

First, we will define the variables and their relationships : we will need to define the variables that we want to include in our network and will specify the relationships between them. This will involve identifying which variables are dependent on which others and how they are related.

As we mentionned in the introduction , there is several variables that could interest us. Knowing that our data is composed of 3800 observations, we have to be carefull not taking too much variables so as to avoid overfitting.

Here is the choice we did for the variables :

— **Players (Passer/receiver) :** It corresponds to the players with 11 modalities. Here we took the eleven players that played the most for the club this season, based on [2]. It corresponds of Maouassa, Nordin, Savanier, Ferri, Wahi, Omlin, Sacko, Cozza, Jullien, Esteve and Chotard. Let us note that they use to play regularly together : this would allow us to investigate the passing network of a core group of players who have played together frequently and may have established more cohesive patterns of play.
Thus, we will define next two variables corresponding to the players : the passer and the receiver. The passer corresponds to the player attempting the pass and the receiver the player that recovers or tries to recover it. It makes sense that these two variables have an impact on the pass success.

— **Location of the passer :** It corresponds to the location on the pitch of the passer when he starts the pass. It will corresponds to two variables corresponding to the coordinates on the pitch (x,y) that could go from 0 to 100.

— **Location of the receiver :** It corresponds to the location on the pitch of the receiver when he receives the pass or the location the pass is not succeed. It will corresponds to two variables corresponding to the coordinates on the pitch (x,y) that could go from 0 to 100.

— **Outcome :** This is the variable of interest. It corresponds to 0 if the pass does not arrive to the receiver and 1 when pass succeeded.

We made the choice not to take further variables for less complexity and to avoid overfitting. For example we could have taken angle and distance of the pass but we believe it does not provide really much more information comparing to the location. Moreover we could quantify the pressing with metrics like Passes Per Defensive Actions (PPDA) that could give insights on the actual opponent pressing.[3]

2.2 Links between variables

Variables being defined, we have to define the links between them.

First, it looks obvious that all variables are linked to the outcome variable. The probability of a pass being succeed depends on the location of the passer and the receiver, as much as the identity of them (they don't have the same pass quality).

Then, the position of the passer depends on the passer himself. The goalkeeper "Omlin" will be more at the defensive area and never in offensive area. We can have the same reflexion for the receiver and his position.

Finally, because of the cohesion between players, the identity of the receiver depends on the passer.

2.3 The graphical model

In our case we will use a Bayesian Network. A Bayesian network is a directed acyclic graph. It represents a joint probability distribution over a set of variables V.

We take the choice of a directed graph because the relationship between all variables are not the same and because of dependancies mentionned before. Such a model permits to observe interdependancies between all the variables involved and does not assume independancy like a Linear Regression model. Moreover, we can perform marginal inference in a bayesian network which helps for the interpretation.
Here is a representation of the graph using the packages *bnlearn* and *visNetwork :*

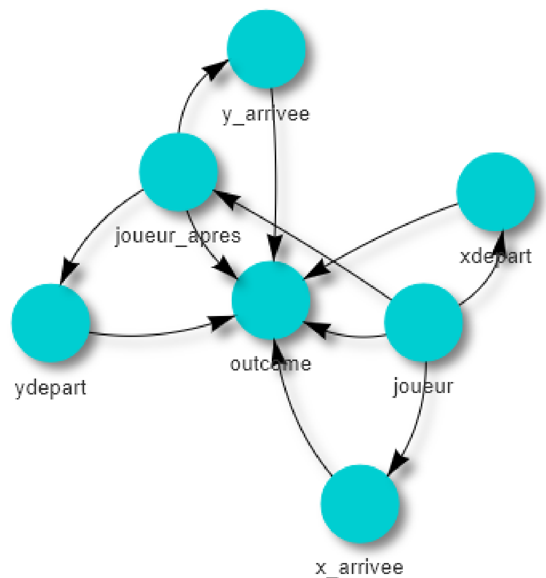

Figure 1 – Bayesian network to model pass outcome

We can state some properties about this model . First, the Bayesian Network will typically satisfy conditional independence restrictions which enables computations of updated probabilities for states of unobserved variables to be made very efficiently [4]. The computations are based on exploiting conditional independencies in an undirected graph

The choice of a Bayesian Network is also because it is more intuitive for non-statisticians (like football team's staff) than machine learning or deep learning techniques.

2.4 Probability distributions

Now we will define the probability distributions for each node : Once we have defined the structure of the network, we will need to specify the probability distributions for each node. This will involve specifying the probability of each possible value of the node given the values of its parents.
In our model, we want to model outcome as factor, so all the variables should be factors. Indeed we need pre-processing in the data using the *cut* function in R for the continous variables.
Here, player variables are already coded as factors containing 11 levels. The location variables should be discretized because they are continous with values between 0 and 100 (data in %).
I made the choice of 5 intervals for the y value and 6 for the x values. Indeed the x value corresponds to the length of the pitch which is larger than the height of the pitch hence my choice. The intervals on x and on y measure the same size.

After this data pre-processing, we have our 7 discretized variables. We will use the package *bn-learn* for our computations.
However, we can give informations about the laws modeled by this packages when fitting our bayesian network. Indeed, the probability distributions that are used in a Bayesian network depend on the type of variable that the node represents. For example, the node representing a binary outcome (such as success or failure) would typically be modeled using a Bernoulli distribution. The other nodes representing categorical variables with more than two categories would typically be modeled using a multinomial distribution.

3 Parameter estimation

Then, with the function bn.fit of the package *bnlearn* we will fit our model with our data. To measure the performance of our model, we separate the data between train and test. The test data will be the data of the two last games of MHSC.

This will estimate the parameters of our network : in a Bayesian network, this would involve estimating the conditional probabilities for each node in the network based on the train data that is available.

Thus, we fit our model based on our network and our data. We will use the 'bayes' method of bn.fit that uses Bayesian parameter estimation to learn the structure and parameters of the Bayesian network [5].
Bayesian parameter estimation involves using probability distributions to represent uncertainty about the parameters of the model. It estimates the parameters by computing the posterior distribution over the parameters given the data and a prior distribution over the parameters. The posterior distribution is then used to make predictions about the outcome of interest. This method can be more computationally intensive than other methods, but it can also be more robust and lead to more accurate models.

4 Results and Performance

Here we will be interested by a set of results that could interest the team's analysts. Based on our graph, we can compute some conditionnal probabilities that could interest us :

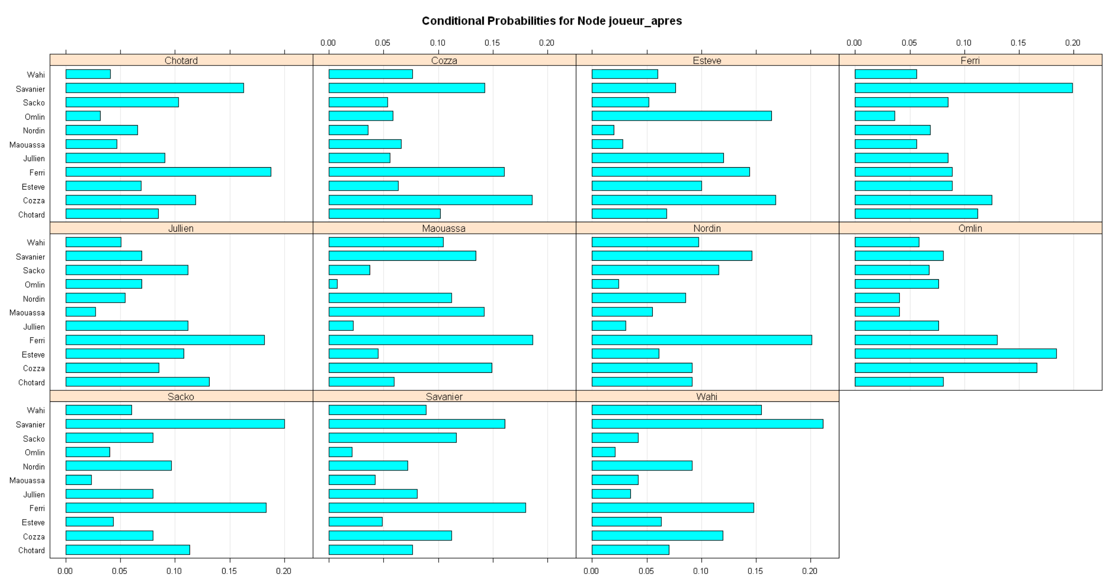

FIGURE 2 – Conditionnal probabilities on receiver

For exemple, in this graph, we can observe conditionnal probabilities for the node "joueur_apres" which is the receiver. This node depends on the node "joueur" which corresponds to the passer. We can make some interpretations : for exemple, if the passer is Ferri, the probability Savanier to be the receiver is 20%. Mathematically, this is written as $\mathbb{P}(receiver = \text{Savanier } | passer = \text{Ferri}) = 0.2$. For the interpretation, that means that if the eleven players are present together on the pitch, if Ferri has the ball, there is a probability of 0.2 of passing it to Savanier, a high value which is not surprising because they play often together in midfield.

Thus, we can watch for other conditionnal probabilities such as the player locations. Let's see for the passer location on the length of the pitch :

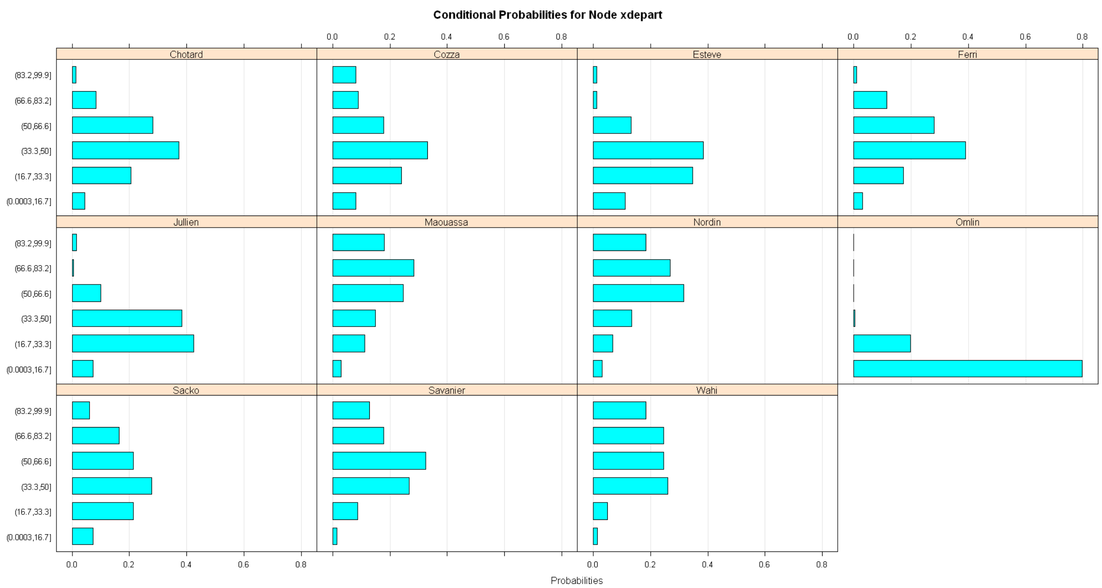

FIGURE 3 – Conditionnal probabilities on x_location

Here, due to our graphical model, the x location of the passer depends of the player. We can see for exemple that the probability the ball passed is in the first sixth of the pitch and knowing the passer is Omlin is close to 80%. This is due to the fact the goalkeeper won't go in attacking areas. We see also that Esteve and Jullien have more probabilites to do their passes in defensive areas, because their position is defenders. Nevertheless, we can see that Jullien probability of passing in offensive areas is smaller than Esteve, that shows maybe the fact that Esteve has the tendancy to take more risks than Jullien.

Now, we want us to observe information about the variable of interest : the **outcome** of the pass. Here conditionnal probabilities showed in a barplot can't be computed because of the number of dependancies and modalities. However, we will be interested about the predictions of our model.
For this, we will compute the predictions of our fitted model applicated to our test data. Then, we will see the confusion matrix and calculate some relevant indicators. The predictions are computed using the *bayes-lw* method, where the predicted values are computed by averaging likelihood weighting simulations performed using all the available nodes as evidence. This method can make predictions even when the network has not seen a particular combination of values in the training data, that is why we have chosen it.
We perform the predictions on the games against Marseille and Lorient which are the two last games of Montpellier. We have the following confusion matrix comparing the predictions of our model with the actual values :

	0	1
0	67	83
1	137	308

Table 1 – Confusion matrix : Column = Predictions, Rows = Actual values

We notice that successful passes are often well predicted while failed passes much less. Regarding the accuracy of the model, we have an accuracy of 66% which is not really good but not bad. We can also be interested about the F1 score which is a metric giving more importance to true positive observations :

$$F_1 = \frac{TP}{TP + \frac{FN+FP}{2}} = 0.74$$

Nevertheless, even with a F1-score of 74%, we can have some doubts about the performances of our network for the prediction. Indeed, when we look at the predictions in-depth, we realize that there is a majority of predictions really close to 0.5. This can be explained because we need more data and this may prone to overfitting. Because of the number of the modalities, the model is really complex and in our test data we may have a big number of new observations that our algorithm don't know what to predict for it.

For interpretations on conditionnal probabilities, this actual model is more intuitive but might not be as performant as another model with a simpler structure.

5 A new model

Regarding the remarks we mentionned in the last section, we will provide a better model that may be more appropriate for the predictions and the performances.
Unlike our precedent approach, now we will not consider intuitive dependancies but we will find the network that matches the best with our train dataset.
For this, we will use a **Max-Min Hill-Climbing algorithm**.The MMHC algorithm was proposed by Tsamardinos et al. [6] in 2006. The basic idea is to use conditional independence test to find the Parents and Children node sets of each node and reduce the search space. Then use the hill climbing search to search for the highest score in the structure space.
This is the graph we obtain at the end of the algorithm :

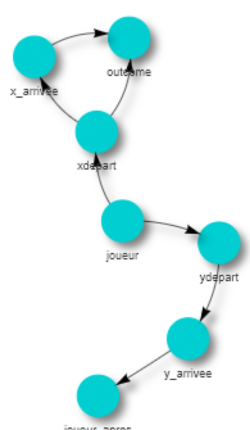

Figure 4 – Bayesian network with MMHC algorithm

This model is less intuitive than the other. Regarding the variable of interest, we see that it depends especially on the location of the passer, which is not surprising. The variable outcome is not directly connected to the variable player and that was our goal at the beginning, "passer" here will have an influence on "passer x location" which is a causal variable. We can see if the performances are increasing based on this model.

	0	1
0	81	69
1	27	418

Table 2 – Confusion matrix of our second model : Column = Predictions, Rows = Actual values

Here we see better performances with an accuracy of 83%. Moreover, watching at the predictions we see probabilities that are not only close to 0.5, 0 or 1. Finally this may be a better network for the problematic of a pass being succeed.
Thus, an indicator of expectedpass could be made with this fitted model. It will correspond to the probability of a pass being completed having the value of the predictors

6 An exemple of utilisation

Here we will use our second model so as to explain the utility of such a model.
It could be used to measure the difficulty of the pass attempts but also for the post-game analysis, watching which passes could have been done in important moments so as to improve passing team patterns.

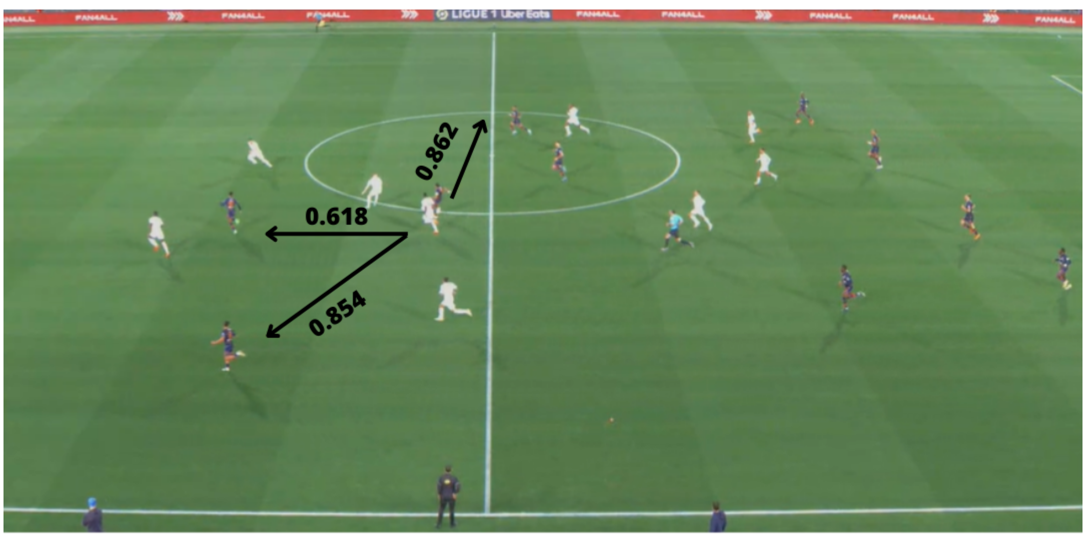

Figure 5 – Annotated of MHSC action against Marseille : 02/01/23, 10th minute

Here is an exemple of the last game of MHSC at the 10th minute. Savanier (MHSC player playing in blue and orange), has the possession and the ball. Based on our algoritm, we can predict the probability he succeed his pass to another player. For example, the probability the pass to Nordin to the right succeed is 86,2%. The probability the pass to Ferri to the left succeed is 85.4%. The probability the pass to Wahi at the top suceed is 61.8%.

Figure 6 – Following of MHSC action against Marseille : 02/01/23, 10th minute

We see that in reality in this second figure that he has chosen to pass the ball to Nordin. The analysis of the game showed that it was the best progressive pass to do because the probability of succeed was high.
Nevertheless we can make some criticisms about our model. For example it does not take the account of the defenders between the passer and the receiver. If we look to the pass to the left, we see that it cannot be succeed in practice and there is a probability of 85.4% of succeeding it with our model.
However the results are encouraging for a model without a big dataset available and with few variables : Bayesian Network seems to be a really good way to model expectedPasses in our conditions. Moreover, one advantage about this modeling is that it is possible also to predict the "expectedPass" without having all the conditionnal variables value, so we could have an estimation of the "expectedPass" even if the player was not in the 11 players studied.

Conclusion and perspectives

In this paper, we defined a first graphical model that represents the intuitive dependancies between variables such as location of the passer, location of the receiver, identity of the passer, identity of the receiver and the outcome binary variable. The goal was to create a network capable of explaining the relationships between the variables in the data but also to predict the outcome variable for new data.

After fitting our model, we interpreted some results like conditionnal probabilities of receiver knowing the passer. This gave us interesting informations for the analysis of the game, we also analyzed conditional probabilities on the locations.

However, our model showed its limits in predictions : the accuracy of our model is far from excellent and a large part of the test dataset were predicted randomly due to overfitting and not enough data in the fitted algorithm.

For the predictive way, another network was performed using MMHC algorithm that creates a new graph structure. When we fit this structure, we get a better accuracy and better predictions. These predictions helps us to create a metric that we can call "expectedPass" which is the probability of a pass being succeeded knowing the passer, its location, the receiver and its location. Such a metric is useful for the post-game analysis where we can see which passes could have been done but also which simple passes were failed and which difficult passes were suceed.
We showed finally that the Bayesian Network worked really well, even with not a large dataset and it is possible also to predict the "expectedPass" without having all the conditionnal variables value.

Nevertheless, our model can be improved in different ways. For exemple, we could have performed cross-validation for a better estimation. Moreover we could have added other variables, such as the number of opponents between the passer and the receiver, or a pressure indicator like PPDA. We could have added also the body part where the pass was made (right or left foot, head).
We noticed also that it is difficult to create a metric such as an expectedPass that depends directly on the player. Indeed, most of the metrics in the litterature like expectedGoals does not depend on the player himself but only on the characteristics of the shot (the pass in our case) and takes much more data, that's why we can affirm that the results of our finale model here are really good for our problematic and the data we have.
We could also take other modeling ideas, such as neural networks or modeling the players and the actions in an undirected graph.

Finally, this model can be viewed as a starting point for other improved models, when Bayesian Networks are not often used in football analytics.

References

[1] https://dataglossary.wyscout.com/xg/#:~:text=Expected%20goals%20(xG)%20is%20a,Location%20of%20the%20assist

[2] understat.com

[3] https://dataglossary.wyscout.com/ppda/

[4] Søren Højsgaard , David Edwards , Steffen Lauritzen, Graphical Models with R, 2012

[5]bnlearn.com

[6]Tsamardinos, I., Brown, L.E. Aliferis, C.F. The max-min hill-climbing Bayesian network structure learning algorithm. Mach Learn 65, 31–78 (2006)

[7]Ievoli, R., Gardini, A. Palazzo, L. The role of passing network indicators in modeling football outcomes : an application using Bayesian hierarchical models. AStA Adv Stat Anal (2021)

R code

```
library(bnlearn)
library(iniestat)
library(dplyr)
library(visNetwork)

#### Data creation

MHSC <- iniestat::read_f24(dir_path = "C:/Users/leonk/Documents/PGM/Données/")
data <- MHSC %>%
  data.frame() %>%
  select(c(id, period_id, min, sec,team_id, type_id, event_id, qualifier_id, match_day, outcome,
           keypass, assist, x, y, away_team, home_team, value,timestamp,player_id))%>%
  subset(select=-c(away_team,home_team))

equivalent_player_name <- tribble(~"player_id",~"joueur",
                                  "186796","Omlin",
                                  "102636", "Savanier",
                                  "477711", "Chotard",
                                  "477717", "Esteve",
                                  "477724", "Wahi",
                                  "210200", "Sacko",
                                  "214473","Maouassa",
                                  "428989","Cozza",
                                  "107641","Ferri",
                                  "114546","Jullien",
                                  "235050","Nordin"

)

data <- data %>%
  left_join(equivalent_player_name, by= "player_id")

data <- data%>%filter(team_id == 147)
data <- arrange(data,match_day,period_id,min,sec,timestamp)

passes <- data %>%
  select(c(id,joueur, match_day,type_id, qualifier_id,
           x, y,outcome, min))%>%
  filter(type_id == 1)%>%
  distinct(id,.keep_all = TRUE)%>%
  mutate(xdepart = x)%>%
  mutate(ydepart = y)%>%
  select(-c(x,y))

tofilt <- data %>%
  select(c(id,joueur, match_day,
           x, y,value,type_id, qualifier_id,outcome))%>%
  filter(type_id == "1")

x_arrivee <-  data%>%
  select(c(id, qualifier_id,value,type_id))%>%
  filter(qualifier_id =="140")%>%
  mutate(x_arrivee = as.numeric(value))%>%
  subset(select=-c(qualifier_id,value,type_id))

```

Figure 7 – Data manipulation 1

```r
y_arrivee <-  data%>%
  select(c(id, qualifier_id,value,type_id))%>%
  filter(qualifier_id =="141")%>%
  mutate(y_arrivee = as.numeric(value))%>%
  subset(select=-c(qualifier_id,value,type_id))

longueur <- data%>%
  select(c(id, qualifier_id,value,type_id))%>%
  filter(qualifier_id =="212")%>%
  mutate(longueur = as.numeric(value))%>%
  subset(select=-c(qualifier_id,value,type_id))

angle <- data%>%
  select(c(id, qualifier_id,value,type_id))%>%
  filter(qualifier_id =="213")%>%
  mutate(angle = as.numeric(value))%>%
  subset(select=-c(qualifier_id,value,type_id))

passes <- passes %>%
  left_join(y_arrivee, by= "id")%>%
  left_join(x_arrivee, by= "id")%>%
  left_join(longueur, by = "id")

event_apres <- function(identifiant){
  w <- which(match(data$id,identifiant)==1)[1]
  newid <- NULL
  while (data$id[w]==identifiant){
    w <- w + 1
    newid <- data$id[w]
  }
  newid
}

joueur_apres <- c()
for (i in 1:length(passes[,1])){
  idj <- event_apres(passes$id[i])
  w  <- which(match(data$id,idj)==1)[1]
  j <- data$joueur[w]
  joueur_apres <- rbind(joueur_apres,j)

}
passes <- cbind(passes,joueur_apres)
passes <- passes %>%
  left_join(angle, by= "id")

passes <- passes %>%filter(!is.na(joueur))%>%
                   filter(!is.na(joueur_apres))
```

Figure 8 – Data manipulation 2

```r
#### The first model

#Function to vizualise the network
plot.network <- function(structure, ht = "400px"){
  nodes.uniq <- unique(c(structure$arcs[,1], structure$arcs[,2]))
  nodes <- data.frame(id = nodes.uniq,
                      label = nodes.uniq,
                      color = "darkturquoise",
                      shadow = TRUE)
  edges <- data.frame(from = structure$arcs[,1],
                      to = structure$arcs[,2],
                      arrows = "to",
                      smooth = TRUE,
                      shadow = TRUE,
                      color = "black")
  return(visNetwork(nodes, edges, height = ht, width = "100%"))
}

#The network with a manually-defined adjacency matrix
variables <- c("joueur","joueur_apres","outcome","xdepart","x_arrivee","ydepart","y_arrivee")
network_structure <- empty.graph(variables)

amat(network_structure) <- matrix(c(0,1,1,1,1,0,0,
                                    0,0,1,0,0,1,1,
                                    0,0,0,0,0,0,0,
                                    0,0,1,0,0,0,0,
                                    0,0,1,0,0,0,0,
                                    0,0,1,0,0,0,0,
                                    0,0,1,0,0,0,0),
                                  nrow=7, byrow=TRUE)
plot.network(network_structure)

plot.network(res)

#pre-processing for discrete BN
passes_data <- passes%>% select(joueur,outcome,xdepart,ydepart,y_arrivee,x_arrivee,joueur_apres,match_day)
passes_data$xdepart <- as.numeric(passes_data$xdepart)
passes_data$ydepart <- as.numeric(passes_data$ydepart)
passes_data$x_arrivee <- as.numeric(passes_data$x_arrivee)
passes_data$y_arrivee <- as.numeric(passes_data$y_arrivee)
passes_data$joueur <- as.factor(passes_data$joueur)
passes_data$joueur_apres <- as.factor(passes_data$joueur_apres)
passes_data$xdepart <- cut(passes_data$xdepart, breaks = 6)
passes_data$x_arrivee <- cut(passes_data$x_arrivee, breaks = 6)
passes_data$ydepart <- cut(passes_data$ydepart, breaks = 5)
passes_data$y_arrivee <- cut(passes_data$y_arrivee, breaks = 5)
passes_data$outcome <- as.factor(passes_data$outcome)


#train and test separation
train <- passes_data%>%filter(match_day %in% c(1,2,3,4,5,6,7,8,9,10,11,12,13,14,15))%>%
  select(joueur,outcome,xdepart,ydepart,y_arrivee,x_arrivee,joueur_apres)
test <- passes_data%>%filter(match_day %in% c(16,17))%>%
  select(joueur,outcome,xdepart,ydepart,y_arrivee,x_arrivee,joueur_apres)
```

Figure 9 – First model code

```r
#Fitting data
fittedbn <- bn.fit(network_structure, data = train,method = 'bayes')

summary(fittedbn$outcome)

#Conditionnals
bn.fit.barchart(fittedbn$xdepart)
bn.fit.barchart(fittedbn$joueur_apres)
bn.fit.barchart(fittedbn$x_arrivee)
bn.fit.barchart(fittedbn$joueur)

#predictions
predictions <- predict(fittedbn,node = "outcome",test, method = 'bayes-lw',n=1000,prob = T)
mc = table(test$outcome, predictions)
mc


#### The second model
res <- hc(train)
plot.network(res)
fittedbn2 <- bn.fit(res, data = train,method = 'bayes')

predictions2 <- predict(fittedbn2,node = "outcome",test, method = 'bayes-lw',n=1000,prob = T)

mc = table(test$outcome, predictions2)
mc


#Queries and predictions
predict(fittedbn2,node = "outcome",test[388,], method = 'bayes-lw',n=1000,prob = T) #The action of Savanier in part 6
t <- test[388,]
t$joueur_apres <- factor("Wahi",levels = levels(test$joueur_apres))
t$y_arrivee <- factor("(60,80]",levels = levels(test$y_arrivee))
t$x_arrivee <- factor("(66.7,83.3]",levels = levels(test$x_arrivee))
predict(fittedbn2,node = "outcome",t, method = 'bayes-lw',n=1000,prob = T) #What if he tried to pass to Wahi?


cpquery(fittedbn2,event = (outcome ==1),evidence = (c(joueur_apres == "Jullien" & xdepart == "(50,66.6]" & ydepart == "(60,80]" & y_arrivee == "(
```

FIGURE 10 – Second model code