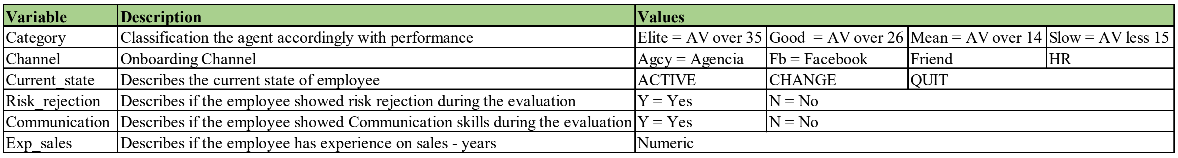
Business Cases

The company has some challenges due to the high turnover index and staff gap in some branches. The company is focused on bank loans and recently included PFA in the business. The company has 780 branches, and the main objective is covering each branch at least with one PFA agent, some branches are allocated in big shopping centres, and they need more than one agent. The company needs 1,696 agents according with the business strategy and currently it has only 1,126 agents.

The PFA agent needs a special training and certification to be able to offer services and financial education. Each client that signs a PFA contract is consider as one sale. As result of special requirements, the cost, time, and training to fill these positions are expensive for the company. Due to the lack of staff in some branches the company set 2 new campaigns: social media focused on Facebook and other using the current staff (the employee could recommend an applicant, when the new employee makes the first sale the employee gets bonus).

Information about dataset

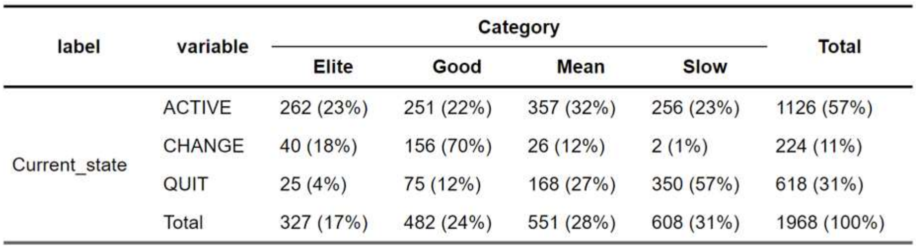
The initial dataset has 37 variables and 1,968 records. The variables used in the study were:



Variable	Description	Values			
Category	Classification the agent accordingly with performance	Elite = AV over 35	Good = AV over 26	Mean = AV over 14	Slow = AV less 15
Channel	Onboarding Channel	Agcy = Agencia	Fb = Facebook	Friend	HR
Current_state	Describes the current state of employee	ACTIVE	CHANGE	QUIT	
Risk_rejection	Describes if the employee showed risk rejection during the evaluation	Y = Yes	N = No		
Communication	Describes if the employee showed Communication skills during the evaluation	Y = Yes	N = No		
Exp_sales	Describes if the employee has experience on sales - years	Numeric			

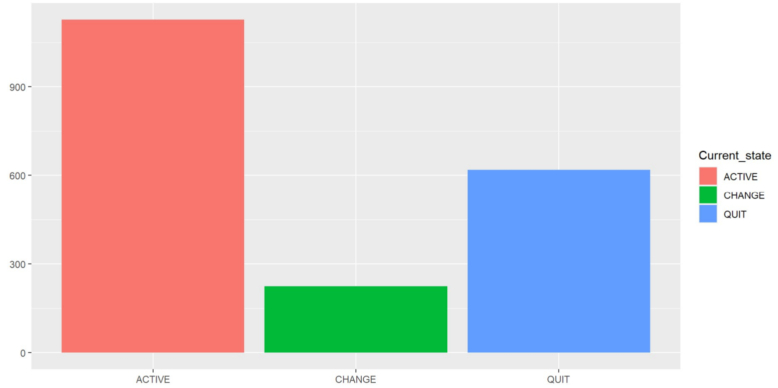
Descriptive Data

Table 1 and chart 1 show the current situation regarding personnel. The company has 1,126 active agents, 11% switched areas, and 31% quit.

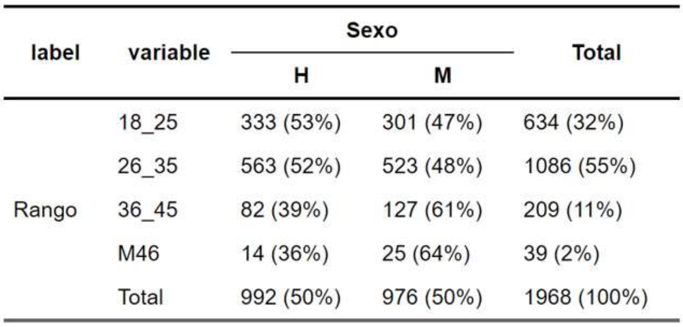


label	variable	Category				Total
		Elite	Good	Mean	Slow	
Current_state	ACTIVE	262 (23%)	251 (22%)	357 (32%)	256 (23%)	1126 (57%)
	CHANGE	40 (18%)	156 (70%)	26 (12%)	2 (1%)	224 (11%)
	QUIT	25 (4%)	75 (12%)	168 (27%)	350 (57%)	618 (31%)
	Total	327 (17%)	482 (24%)	551 (28%)	608 (31%)	1968 (100%)

Table 1



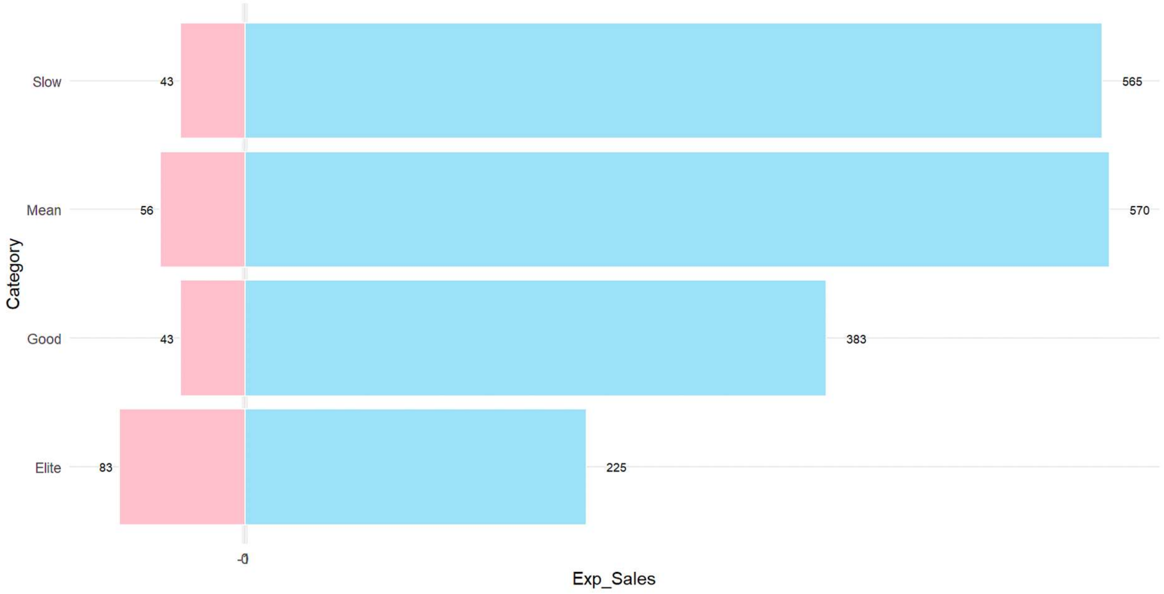
The dataset has 87% of people younger than 36 years old and similar gender distribution.



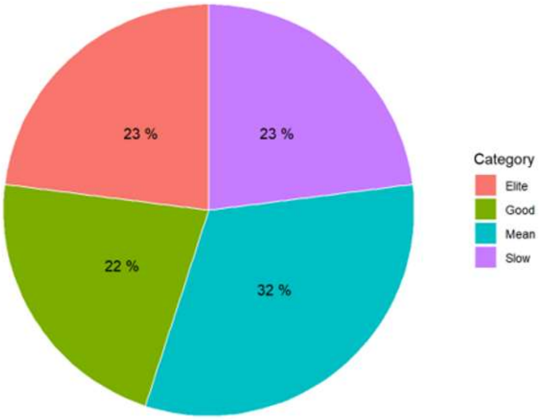
label	variable	Sexo		Total
		H	M	
	18_25	333 (53%)	301 (47%)	634 (32%)
	26_35	563 (52%)	523 (48%)	1086 (55%)
Rango	36_45	82 (39%)	127 (61%)	209 (11%)
	M46	14 (36%)	25 (64%)	39 (2%)
	Total	992 (50%)	976 (50%)	1968 (100%)

Table 2

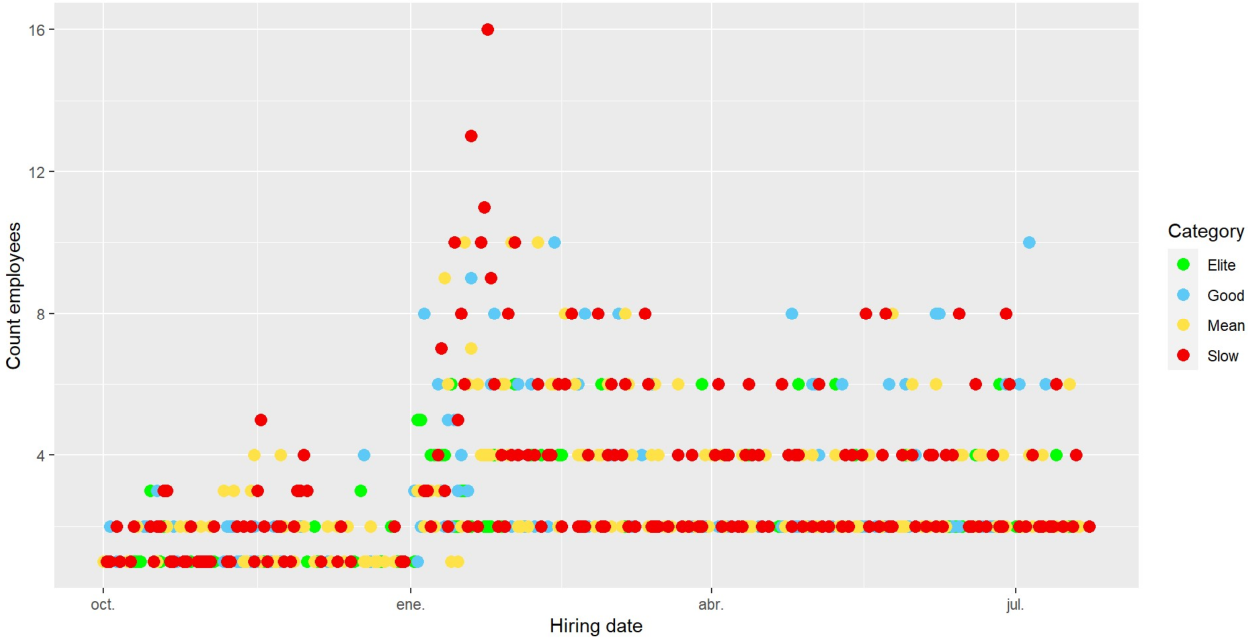
The following picture shows the distribution of agents with sales experience, where the blue bar is a subset of agents with experience and the pink bar represents staff without experience.



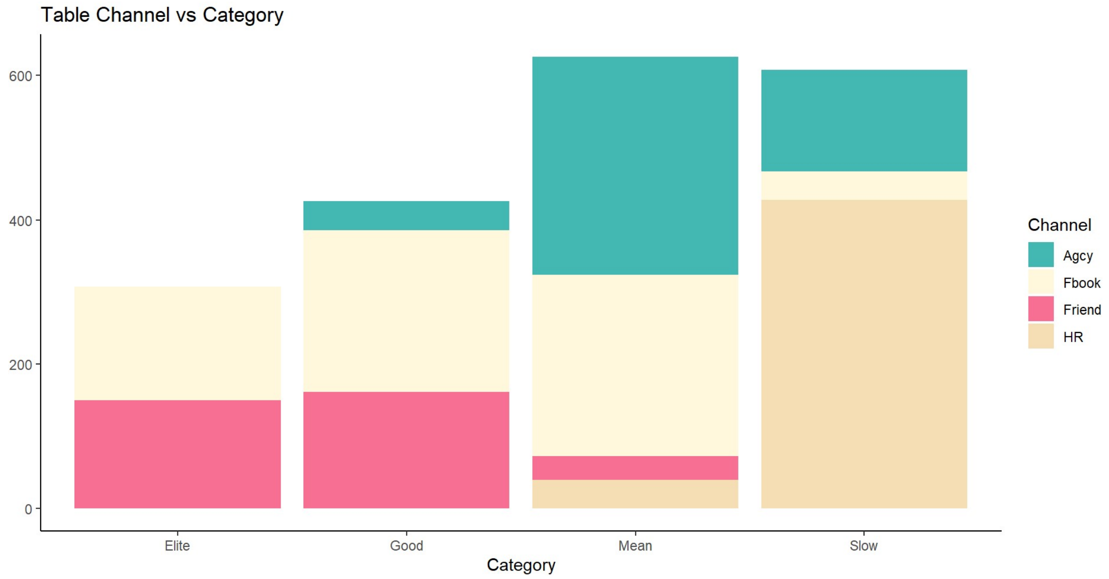
The dataset also shows that 23% of the current active sales force is categorized as slow sellers, which means their production is low than the average. The 32% is generating only the average number of sales. That means 55% of the current staff have a poor production.



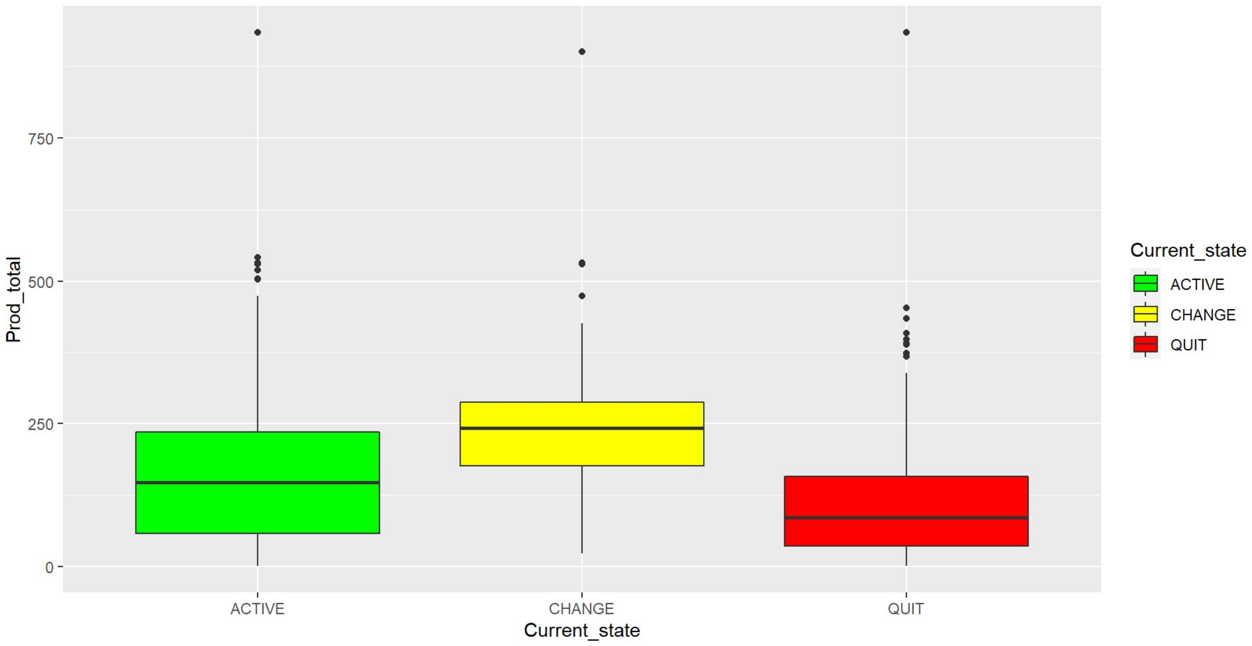
The following chart reveals that less than four employees are hired per day, and on exceptional occasions, more than eight are hired. However, the quality of human talent is low since most of the employees hired are categorized as Slow.



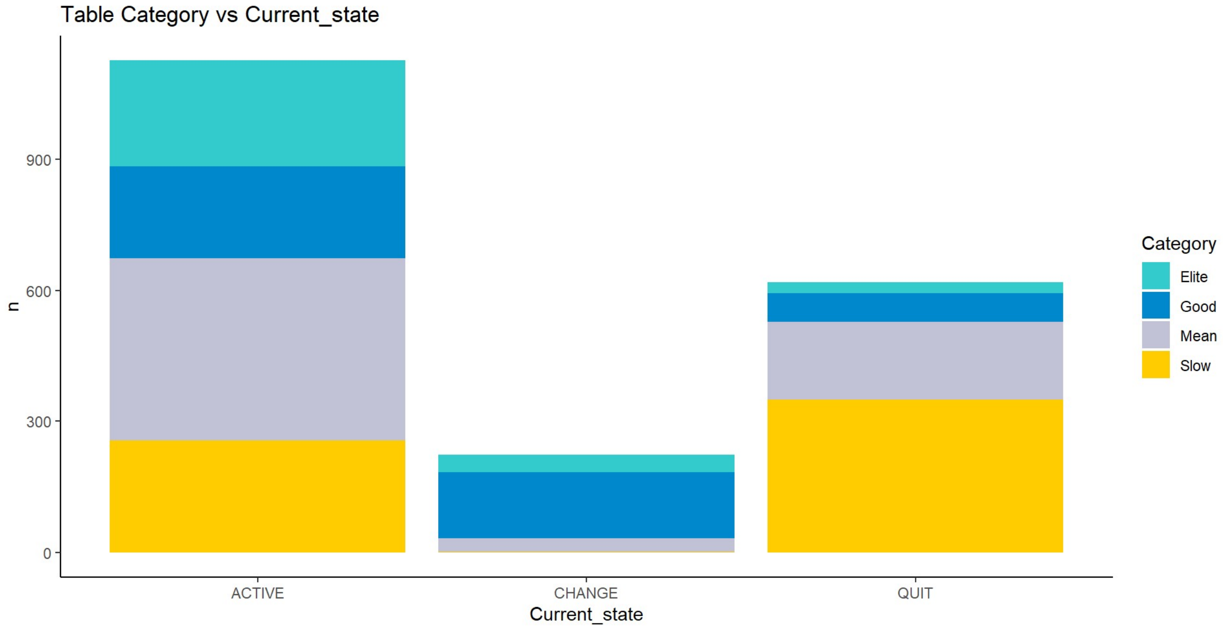
In addition, the following chart shows that agents with a category equal to Elite or Good were onboarding by a Friend or Facebook. The agents categorized as Slow came mainly from HR, and many agents coming from Agency are categorized as Mean.



Additionally, the chart below shows a small group of employees who are changing to other areas of the company; this group is exclusive of agents with good performance.



The following chart shows the proportion of well-performing agents switching areas. It represents a possible talent flight. On the other hand, the agents who are resigning are agents with a low performance showing a degree of functional rotation.



Research questions

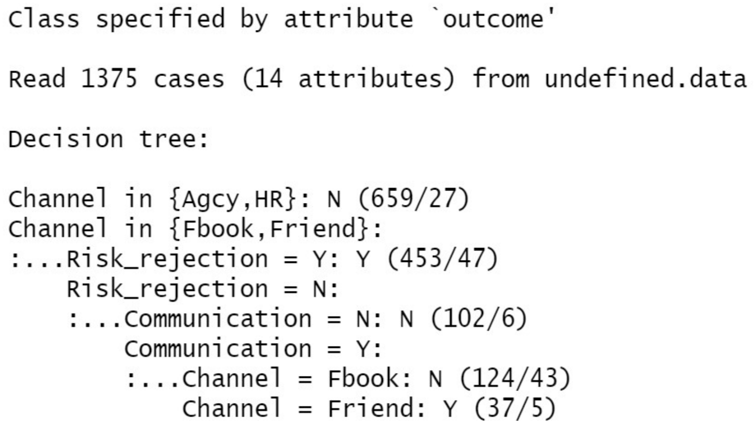
The study aims to demonstrate whether there is a positive or negative influence between having children and staff performance. In addition, DS techniques could sort which features are crucial to choosing quality staff and predicting turnover. Furthermore, ML could assist know which channels are reliable to attract quality human talent.

Analysis

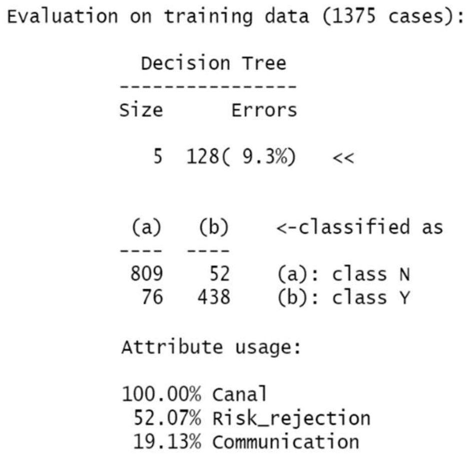
Initially, using the decision tree model could be possible to find which features are desirable to select quality human talent. It also helped to know what channel is attracting more quality staff. One variable was created to indicate if the agent has the desired profile using the employee category, if it was Good or Elite the variable marked it as a desirable prospect. Lastly, to sort out why the agents are more likely to switch was useful to use the decision tree model. One variable was added to indicate if the agent switch areas.

Results

The model used 1,375 records with 14 attributes to maximize its accuracy. The model generated a decision tree (Figure 1) with 9.3% of margin error and size five, using three attributes, including Channel, Risk_rejection, and Communication, with the following distribution.



```
Class specified by attribute `outcome'

Read 1375 cases (14 attributes) from undefined.data

Decision tree:

Channel in {Agcy,HR}: N (659/27)
Channel in {Fbook,Friend}:
:...Risk_rejection = Y: Y (453/47)
    Risk_rejection = N:
    :...Communication = N: N (102/6)
        Communication = Y:
        :...Channel = Fbook: N (124/43)
            Channel = Friend: Y (37/5)
```



```
Evaluation on training data (1375 cases):

          Decision Tree
        ----------------
        Size      Errors

           5  128( 9.3%)   <<


         (a)   (b)    <-classified as
        ----  ----
         809    52    (a): class N
          76   438    (b): class Y

        Attribute usage:

        100.00% Canal
         52.07% Risk_rejection
         19.13% Communication
```

The first node splits the dataset in two subsets according with recruitment channel; 48% came from Agency and HR and 52% came from Facebook and Friend. Node two shows if the person came from Agency and HR, they are most likely “not desirable”. Also, node six indicate if the person come from Facebook or Friend, Risk_rejection and Community as N then classify as “not desirable”. However, node four represent the 33% of the dataset, and indicates if person come from Facebook or Friend and shows Risk_rejection as Y, they are most likely as a desirable staff. Node 7 indicates if person shows Risk_rejection as N and Communicacion as Y and channel is Friend, they have desirable profile.

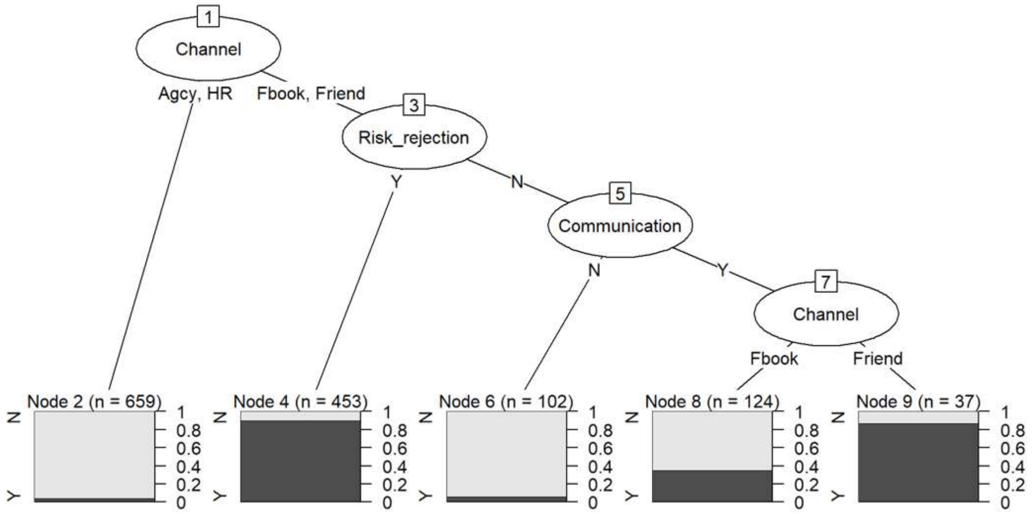
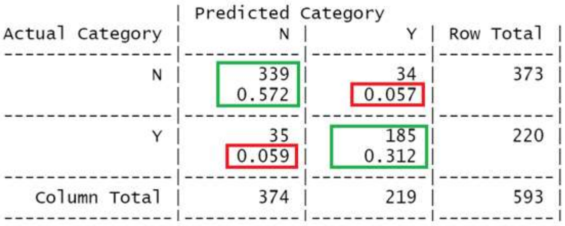


Figure 1

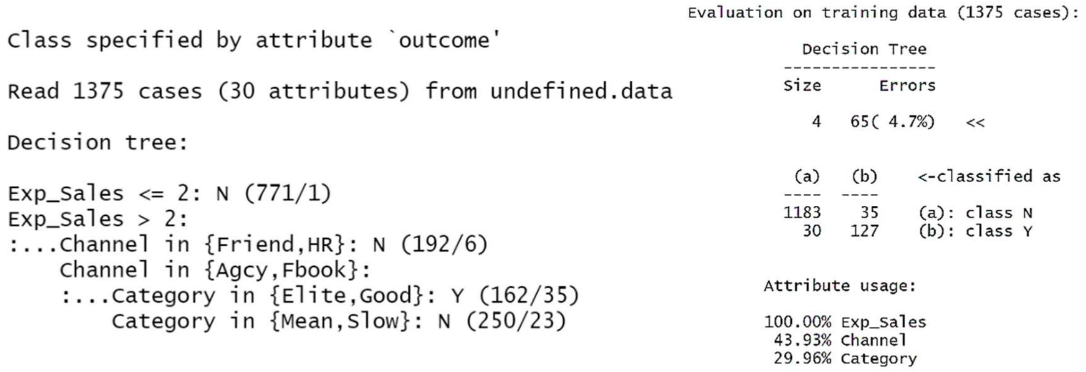
Additionally, Table 3 indicates the performance of the model using test data, the accuracy of the model is 88.4%, and a margin of error of 11.6%. The model would help classify profiles as desirable and undesirable with an accuracy greater than 88%. The possibility of rejecting a good prospect is less than 6%. Rejecting the prospects with a lower profile could help the company to have a higher quality workforce with a possible positive impact on production.



```
                | Predicted Category
Actual Category |          N |          Y | Row Total |
----------------|------------|------------|-----------|
              N |        339 |         34 |       373 |
                |      0.572 |      0.057 |           |
----------------|------------|------------|-----------|
              Y |         35 |        185 |       220 |
                |      0.059 |      0.312 |           |
----------------|------------|------------|-----------|
   Column Total |        374 |        219 |       593 |
----------------|------------|------------|-----------|
```

Table 3

The model used a dataset with 1,375 records. The decision tree (Figure 2) generated had a margin error of 4.7% and size four, using Exp_sales, Channel, and Category.



```
Class specified by attribute `outcome'

Read 1375 cases (30 attributes) from undefined.data

Decision tree:

Exp_Sales <= 2: N (771/1)
Exp_Sales > 2:
:...Channel in {Friend,HR}: N (192/6)
    Channel in {Agcy,Fbook}:
    :...Category in {Elite,Good}: Y (162/35)
        Category in {Mean,Slow}: N (250/23)
```

```
Evaluation on training data (1375 cases):

          Decision Tree
        ----------------
        Size      Errors

           4   65( 4.7%)   <<


          (a)   (b)    <-classified as
         ----  ----
         1183    35    (a): class N
           30   127    (b): class Y


        Attribute usage:

        100.00% Exp_Sales
         43.93% Channel
         29.96% Category
```

The first node based on years of sales experience splits the dataset. Node six represents 12% of the dataset and indicates if the staff have more than two years of experience in sales, came from an agency or Facebook, and were qualified as an Elite or Good seller are more likely to switch areas. However, if the person has experienced sales <= two years or came from HR or recommended by a Friend or agent was categorized as Mean or Slow, then the person is more likely to not switch areas.

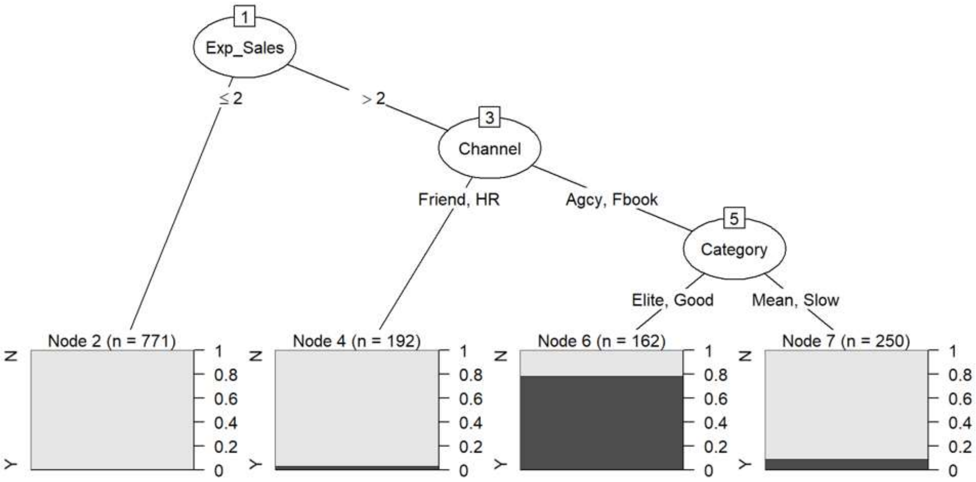
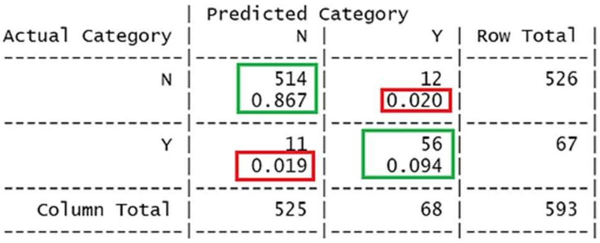


Figure 2

In addition, Table 4 shows the performance of the model using test data has an accuracy of 96.1%, and a margin of error of 3.9% that is better than 4.7% initial. The model shows a specific profile of agents who are switching areas confirming the initial concern about talent flight.



```
                | Predicted Category 
Actual Category |         N |         Y | Row Total | 
----------------|-----------|-----------|-----------|
              N |       514 |        12 |       526 | 
                |     0.867 |     0.020 |           | 
----------------|-----------|-----------|-----------|
              Y |        11 |        56 |        67 | 
                |     0.019 |     0.094 |           | 
----------------|-----------|-----------|-----------|
   Column Total |       525 |        68 |       593 | 
----------------|-----------|-----------|-----------|
```

Table 4

Suggestions for performance improvements

Regarding attracting quality staff, from the descriptive analytics and the decision tree, it is observed that the candidates recommended by a friend and those who arrived through Facebook have a desirable profile for the company. Thus, the company could focus its efforts on these recruitment channels.

Additionally, the company could implement the model to filter applicants' profiles, prioritizing the values for Risk_rejection and Community. Hiring agents with adequate performance maximize sales and drive an adjustment of required positions; an agent with good performance could produce the same as two agents marked as slow category. Additionally, the company could generate a new payment and rewards strategy to increase the agent's engagement.

According to the second decision tree, a talent flight is detected among the group's companies. The recommendation is to review the working conditions and benefits to which employees get access when they switch areas. It allows the company to modify or update benefit and payment schemes to increase engagement. Then, the costs of training and certification of the employees could convert an investment; retaining the talent would influence sales performance.

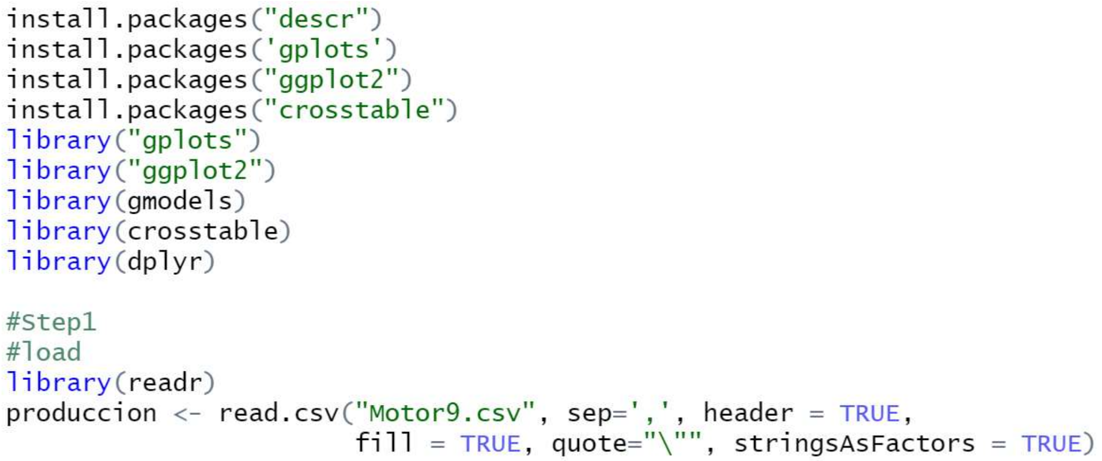
Conclusion

The best practices suggest implementing an AI project first be sure about data quality and quantity. Second, identify the problem to solve as something meaningful and with enough data. Moreover, the company requires to assess its readiness before to start the project. The use of multiple metrics, costs, efficiencies, and impacts of people management constitutes extensive information, which requires a good focus on data analysis to obtain relevant evidence. The challenge is select the correct data and ask the right questions to add value and improve fundamental aspects for the company.

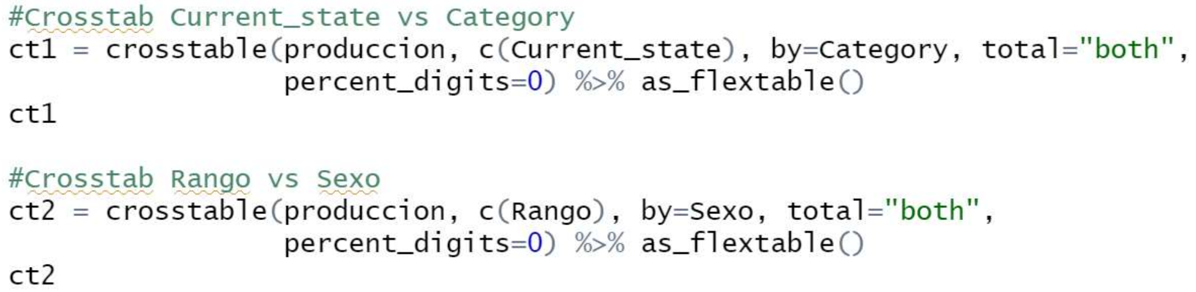
AI applied in HR Analytics helps HR managers to make decisions based on mathematics, statistics, and modelling, which allows obtaining evidence and predicting patterns in the behaviour, performance, and results of people in the organization. AI entails improving the company's capabilities from the recruitment process unto increasing financial benefits, generating a strategic advantage and differential value for the company. Finally, setting HR data value in the organization could be very productive for talent management and business results. Therefore, regardless of size or activity, organizations could design AI-based HR Analytics models tailored to their needs.

APENDIX

Crosstabs

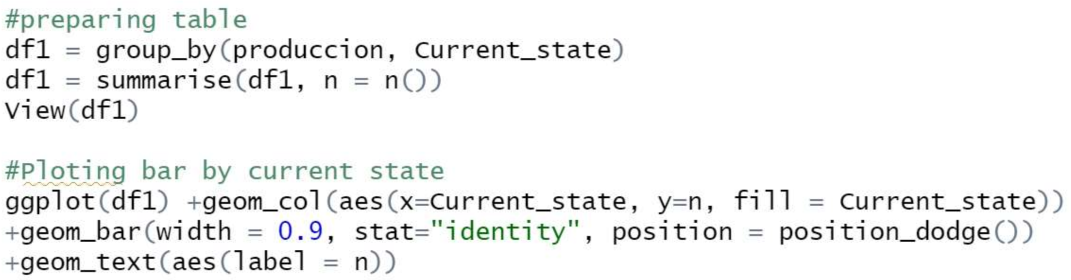


```
install.packages("descr")
install.packages('gplots')
install.packages("ggplot2")
install.packages("crosstable")
library("gplots")
library("ggplot2")
library(gmodels)
library(crosstable)
library(dplyr)

#Step1
#load
library(readr)
produccion <- read.csv("Motor9.csv", sep=',', header = TRUE,
                       fill = TRUE, quote="\"", stringsAsFactors = TRUE)
```

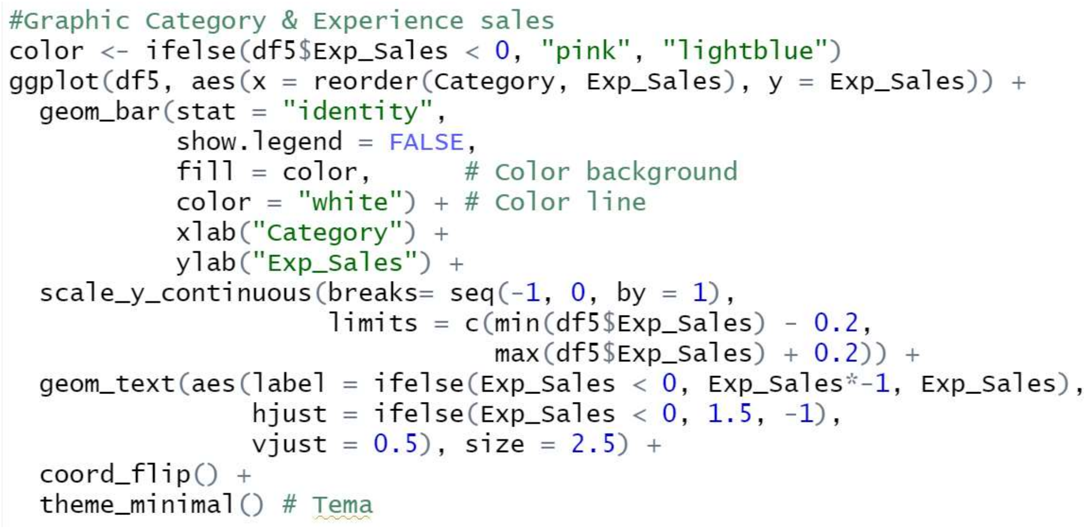


```
#Crosstab Current_state vs Category
ct1 = crosstable(produccion, c(Current_state), by=Category, total="both",
                 percent_digits=0) %>% as_flextable()
ct1

#Crosstab Rango vs Sexo
ct2 = crosstable(produccion, c(Rango), by=Sexo, total="both",
                 percent_digits=0) %>% as_flextable()
ct2
```

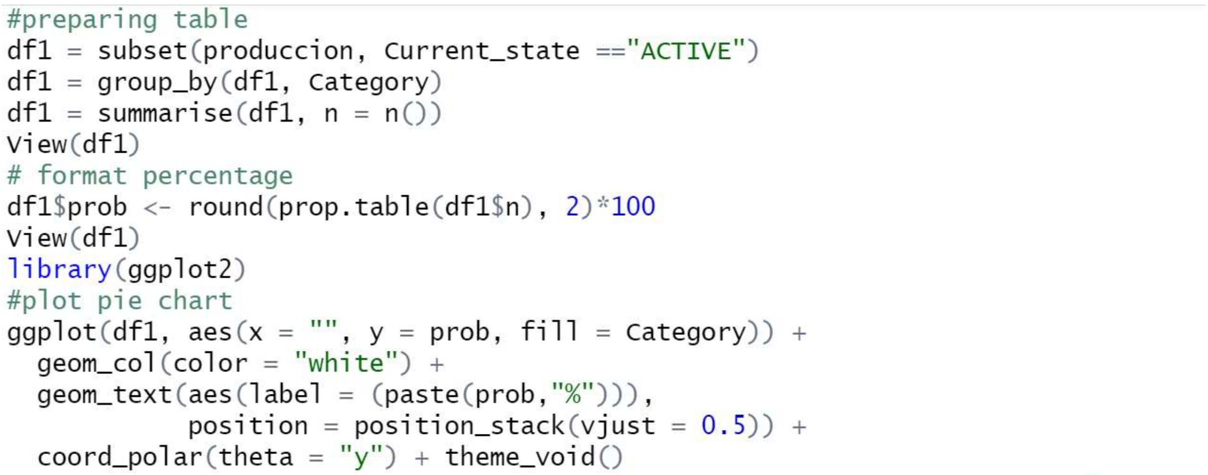
Charts



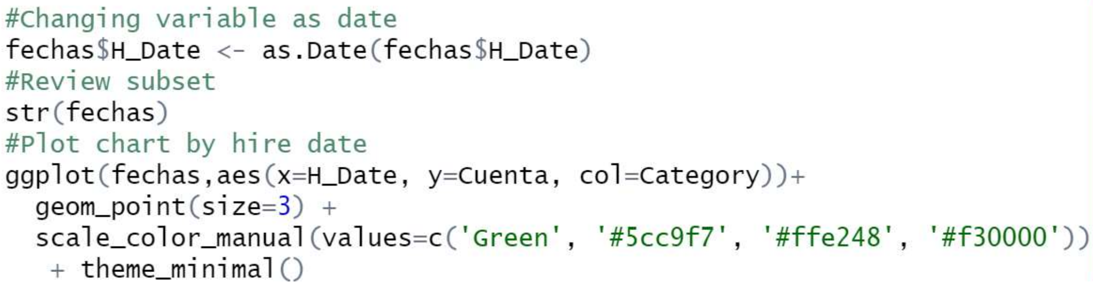
```
#preparing table
df1 = group_by(produccion, Current_state)
df1 = summarise(df1, n = n())
View(df1)

#Ploting bar by current state
ggplot(df1) +geom_col(aes(x=Current_state, y=n, fill = Current_state))
+geom_bar(width = 0.9, stat="identity", position = position_dodge())
+geom_text(aes(label = n))
```



```
#Graphic Category & Experience sales
color <- ifelse(df5$Exp_Sales < 0, "pink", "lightblue")
ggplot(df5, aes(x = reorder(Category, Exp_Sales), y = Exp_Sales)) +
  geom_bar(stat = "identity",
           show.legend = FALSE,
           fill = color,      # Color background
           color = "white") + # Color line
           xlab("Category") +
           ylab("Exp_Sales") +
  scale_y_continuous(breaks= seq(-1, 0, by = 1),
                     limits = c(min(df5$Exp_Sales) - 0.2,
                                max(df5$Exp_Sales) + 0.2)) +
  geom_text(aes(label = ifelse(Exp_Sales < 0, Exp_Sales*-1, Exp_Sales),
                hjust = ifelse(Exp_Sales < 0, 1.5, -1),
                vjust = 0.5), size = 2.5) +
  coord_flip() +
  theme_minimal() # Tema
```

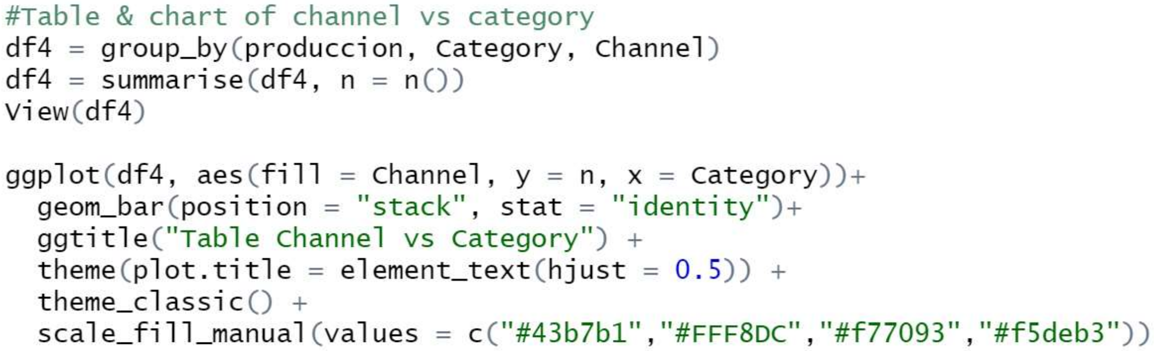


```
#preparing table
df1 = subset(produccion, Current_state =="ACTIVE")
df1 = group_by(df1, Category)
df1 = summarise(df1, n = n())
View(df1)
# format percentage
df1$prob <- round(prop.table(df1$n), 2)*100
View(df1)
library(ggplot2)
#plot pie chart
ggplot(df1, aes(x = "", y = prob, fill = Category)) +
  geom_col(color = "white") +
  geom_text(aes(label = (paste(prob,"%"))),
            position = position_stack(vjust = 0.5)) +
  coord_polar(theta = "y") + theme_void()
```

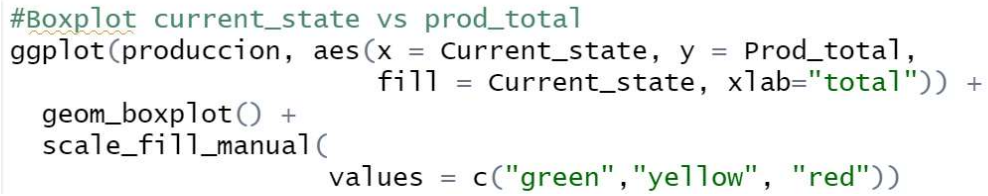


```
#Changing variable as date
fechas$H_Date <- as.Date(fechas$H_Date)
#Review subset
str(fechas)
#Plot chart by hire date
ggplot(fechas,aes(x=H_Date, y=Cuenta, col=Category))+
  geom_point(size=3) +
  scale_color_manual(values=c('Green', '#5cc9f7', '#ffe248', '#f30000'))
   + theme_minimal()
```

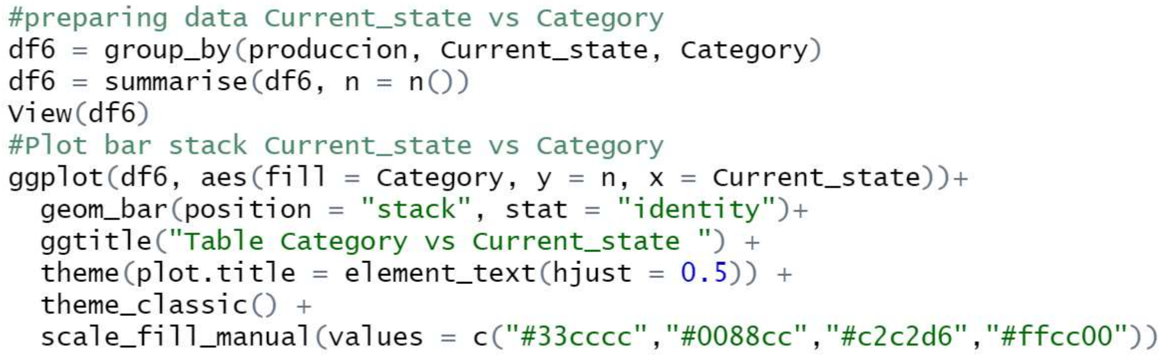
```
#Changi
fechas$
#Review
str(fec
#Plot c
ggplot(
  geom_
  scale
   + th
```



```
#Table & chart of channel vs category
df4 = group_by(produccion, Category, Channel)
df4 = summarise(df4, n = n())
View(df4)

ggplot(df4, aes(fill = Channel, y = n, x = Category))+
  geom_bar(position = "stack", stat = "identity")+
  ggtitle("Table Channel vs Category") +
  theme(plot.title = element_text(hjust = 0.5)) +
  theme_classic() +
  scale_fill_manual(values = c("#43b7b1","#FFF8DC","#f77093","#f5deb3"))
```

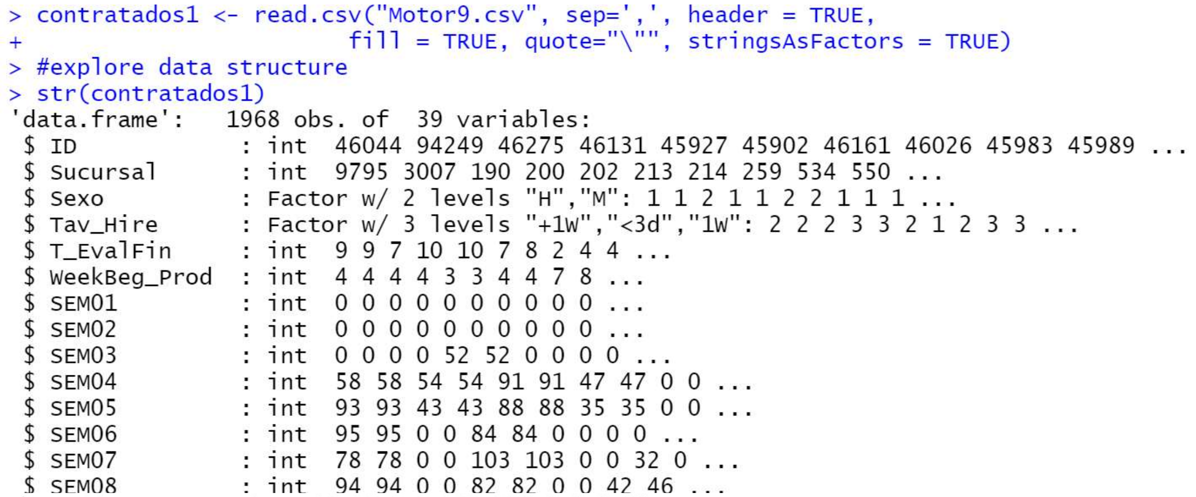


```
#Boxplot current_state vs prod_total
ggplot(produccion, aes(x = Current_state, y = Prod_total,
                       fill = Current_state, xlab="total")) +
  geom_boxplot() +
  scale_fill_manual(
                    values = c("green","yellow", "red"))
```

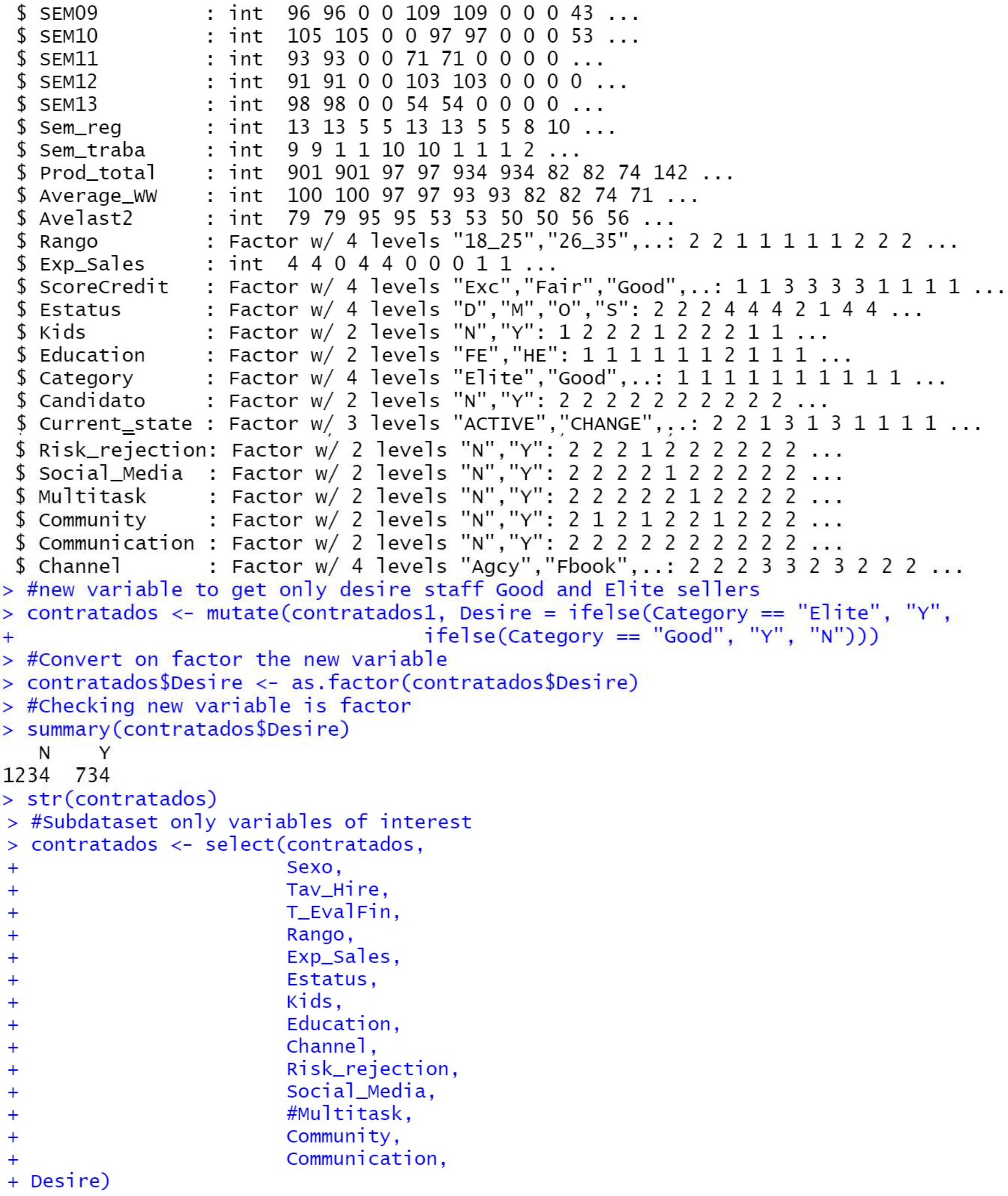


```
#preparing data Current_state vs Category
df6 = group_by(produccion, Current_state, Category)
df6 = summarise(df6, n = n())
View(df6)
#Plot bar stack Current_state vs Category
ggplot(df6, aes(fill = Category, y = n, x = Current_state))+
  geom_bar(position = "stack", stat = "identity")+
  ggtitle("Table Category vs Current_state ") +
  theme(plot.title = element_text(hjust = 0.5)) +
  theme_classic() +
  scale_fill_manual(values = c("#33cccc","#0088cc","#c2c2d6","#ffcc00"))
```

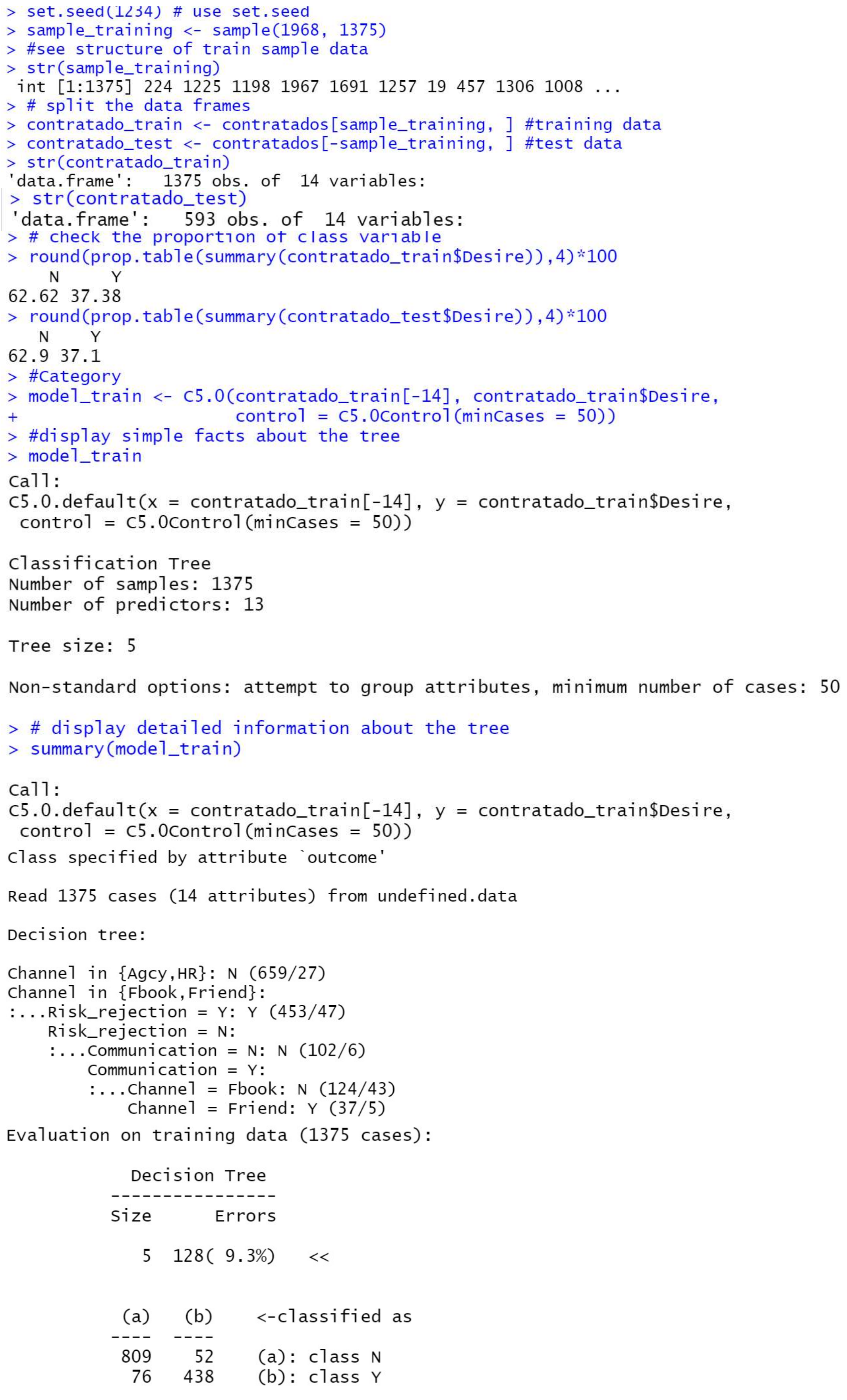
Decision tree to classify profiles



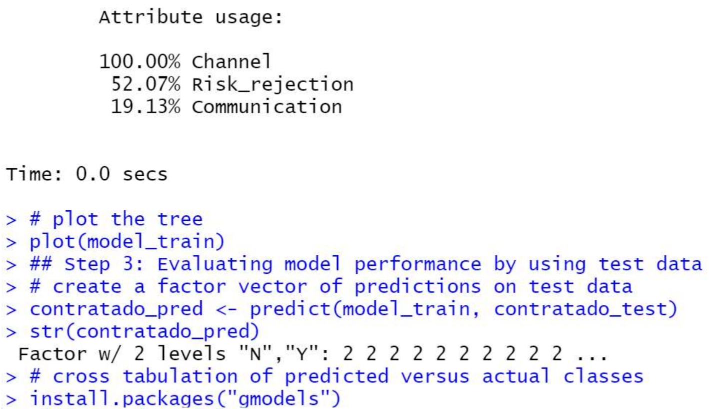
```
> contratados1 <- read.csv("Motor9.csv", sep=',', header = TRUE,
+                          fill = TRUE, quote="\"", stringsAsFactors = TRUE)
> #explore data structure
> str(contratados1)
'data.frame':   1968 obs. of  39 variables:
 $ ID           : int  46044 94249 46275 46131 45927 45902 46161 46026 45983 45989 ...
 $ Sucursal     : int  9795 3007 190 200 202 213 214 259 534 550 ...
 $ Sexo         : Factor w/ 2 levels "H","M": 1 1 2 1 1 2 2 1 1 1 ...
 $ Tav_Hire     : Factor w/ 3 levels "+1W","<3d","1W": 2 2 2 3 3 2 1 2 3 3 ...
 $ T_EvalFin    : int  9 9 7 10 10 7 8 2 4 4 ...
 $ WeekBeg_Prod : int  4 4 4 4 3 3 4 4 7 8 ...
 $ SEM01        : int  0 0 0 0 0 0 0 0 0 0 ...
 $ SEM02        : int  0 0 0 0 0 0 0 0 0 0 ...
 $ SEM03        : int  0 0 0 0 52 52 0 0 0 0 ...
 $ SEM04        : int  58 58 54 54 91 91 47 47 0 0 ...
 $ SEM05        : int  93 93 43 43 88 88 35 35 0 0 ...
 $ SEM06        : int  95 95 0 0 84 84 0 0 0 0 ...
 $ SEM07        : int  78 78 0 0 103 103 0 0 32 0 ...
 $ SEM08        : int  94 94 0 0 82 82 0 0 42 46 ...
```



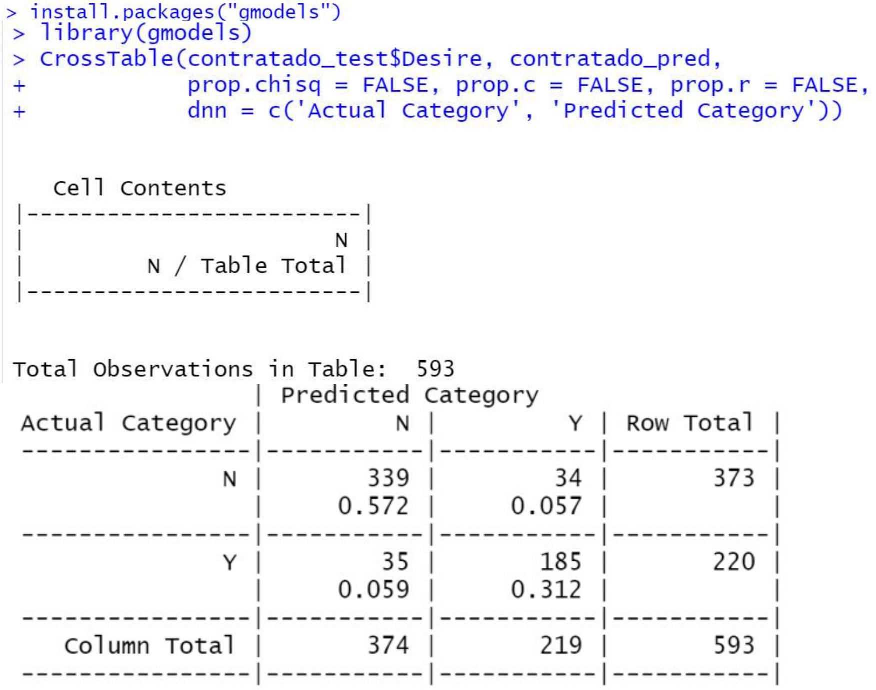
```
 $ SEM09        : int  96 96 0 0 109 109 0 0 0 43 ...
 $ SEM10        : int  105 105 0 0 97 97 0 0 0 53 ...
 $ SEM11        : int  93 93 0 0 71 71 0 0 0 0 ...
 $ SEM12        : int  91 91 0 0 103 103 0 0 0 0 ...
 $ SEM13        : int  98 98 0 0 54 54 0 0 0 0 ...
 $ Sem_reg      : int  13 13 5 5 13 13 5 5 8 10 ...
 $ Sem_traba    : int  9 9 1 1 10 10 1 1 1 2 ...
 $ Prod_total   : int  901 901 97 97 934 934 82 82 74 142 ...
 $ Average_WW   : int  100 100 97 97 93 93 82 82 74 71 ...
 $ Avelast2     : int  79 79 95 95 53 53 50 50 56 56 ...
 $ Rango        : Factor w/ 4 levels "18_25","26_35",..: 2 2 1 1 1 1 1 2 2 2 ...
 $ Exp_Sales    : int  4 4 0 4 4 0 0 0 1 1 ...
 $ ScoreCredit  : Factor w/ 4 levels "Exc","Fair","Good",..: 1 1 3 3 3 3 1 1 1 1 ...
 $ Estatus      : Factor w/ 4 levels "D","M","O","S": 2 2 2 4 4 4 2 1 4 4 ...
 $ Kids         : Factor w/ 2 levels "N","Y": 1 2 2 2 1 2 2 2 1 1 ...
 $ Education    : Factor w/ 2 levels "FE","HE": 1 1 1 1 1 1 2 1 1 1 ...
 $ Category     : Factor w/ 4 levels "Elite","Good",..: 1 1 1 1 1 1 1 1 1 1 ...
 $ Candidato    : Factor w/ 2 levels "N","Y": 2 2 2 2 2 2 2 2 2 2 ...
 $ Current_state : Factor w/ 3 levels "ACTIVE","CHANGE",..: 2 2 1 3 1 3 1 1 1 1 ...
 $ Risk_rejection: Factor w/ 2 levels "N","Y": 2 2 2 1 2 2 2 2 2 2 ...
 $ Social_Media  : Factor w/ 2 levels "N","Y": 2 2 2 2 1 2 2 2 2 2 ...
 $ Multitask     : Factor w/ 2 levels "N","Y": 2 2 2 2 2 1 2 2 2 2 ...
 $ Community     : Factor w/ 2 levels "N","Y": 2 1 2 1 2 2 1 2 2 2 ...
 $ Communication : Factor w/ 2 levels "N","Y": 2 2 2 2 2 2 2 2 2 2 ...
 $ Channel       : Factor w/ 4 levels "Agcy","Fbook",..: 2 2 2 3 3 2 3 2 2 2 ...
> #new variable to get only desire staff Good and Elite sellers
> contratados <- mutate(contratados1, Desire = ifelse(Category == "Elite", "Y",
+                       ifelse(Category == "Good", "Y", "N")))
> #Convert on factor the new variable
> contratados$Desire <- as.factor(contratados$Desire)
> #Checking new variable is factor
> summary(contratados$Desire)
   N    Y
1234  734
> str(contratados)
> #Subdataset only variables of interest
> contratados <- select(contratados,
+                      Sexo,
+                      Tav_Hire,
+                      T_EvalFin,
+                      Rango,
+                      Exp_Sales,
+                      Estatus,
+                      Kids,
+                      Education,
+                      Channel,
+                      Risk_rejection,
+                      Social_Media,
+                      #Multitask,
+                      Community,
+                      Communication,
+ Desire)
```



```
> set.seed(1234) # use set.seed
> sample_training <- sample(1968, 1375)
> #see structure of train sample data
> str(sample_training)
 int [1:1375] 224 1225 1198 1967 1691 1257 19 457 1306 1008 ...
> # split the data frames
> contratado_train <- contratados[sample_training, ] #training data
> contratado_test <- contratados[-sample_training, ] #test data
> str(contratado_train)
'data.frame':   1375 obs. of  14 variables:
> str(contratado_test)
'data.frame':   593 obs. of  14 variables:
> # check the proportion of class variable
> round(prop.table(summary(contratado_train$Desire)),4)*100
    N     Y
62.62 37.38
> round(prop.table(summary(contratado_test$Desire)),4)*100
   N    Y
62.9 37.1
> #Category
> model_train <- C5.0(contratado_train[-14], contratado_train$Desire,
+                     control = C5.0Control(minCases = 50))
> #display simple facts about the tree
> model_train
Call:
C5.0.default(x = contratado_train[-14], y = contratado_train$Desire,
 control = C5.0Control(minCases = 50))

Classification Tree
Number of samples: 1375
Number of predictors: 13

Tree size: 5

Non-standard options: attempt to group attributes, minimum number of cases: 50

> # display detailed information about the tree
> summary(model_train)

Call:
C5.0.default(x = contratado_train[-14], y = contratado_train$Desire,
 control = C5.0Control(minCases = 50))
Class specified by attribute `outcome'

Read 1375 cases (14 attributes) from undefined.data

Decision tree:

Channel in {Agcy,HR}: N (659/27)
Channel in {Fbook,Friend}:
:...Risk_rejection = Y: Y (453/47)
    Risk_rejection = N:
    :...Communication = N: N (102/6)
        Communication = Y:
        :...Channel = Fbook: N (124/43)
            Channel = Friend: Y (37/5)
Evaluation on training data (1375 cases):

	    Decision Tree
	  ----------------
	  Size      Errors

	     5  128( 9.3%)   <<


	   (a)   (b)    <-classified as
	  ----  ----
	   809    52    (a): class N
	    76   438    (b): class Y
```

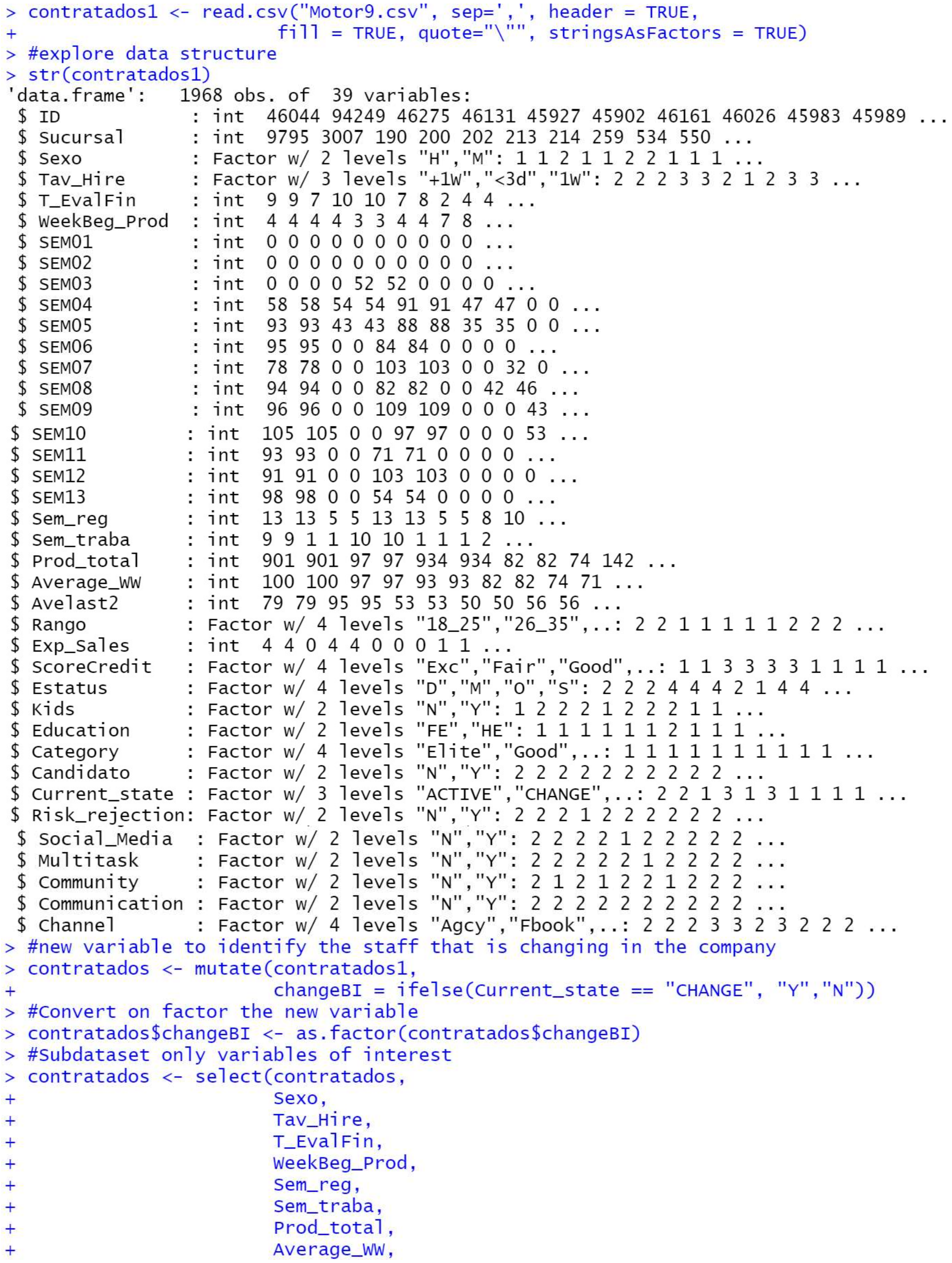


```
        Attribute usage:

        100.00% Channel
         52.07% Risk_rejection
         19.13% Communication


Time: 0.0 secs

> # plot the tree
> plot(model_train)
> ## Step 3: Evaluating model performance by using test data
> # create a factor vector of predictions on test data
> contratado_pred <- predict(model_train, contratado_test)
> str(contratado_pred)
 Factor w/ 2 levels "N","Y": 2 2 2 2 2 2 2 2 2 2 ...
> # cross tabulation of predicted versus actual classes
> install.packages("gmodels")
```

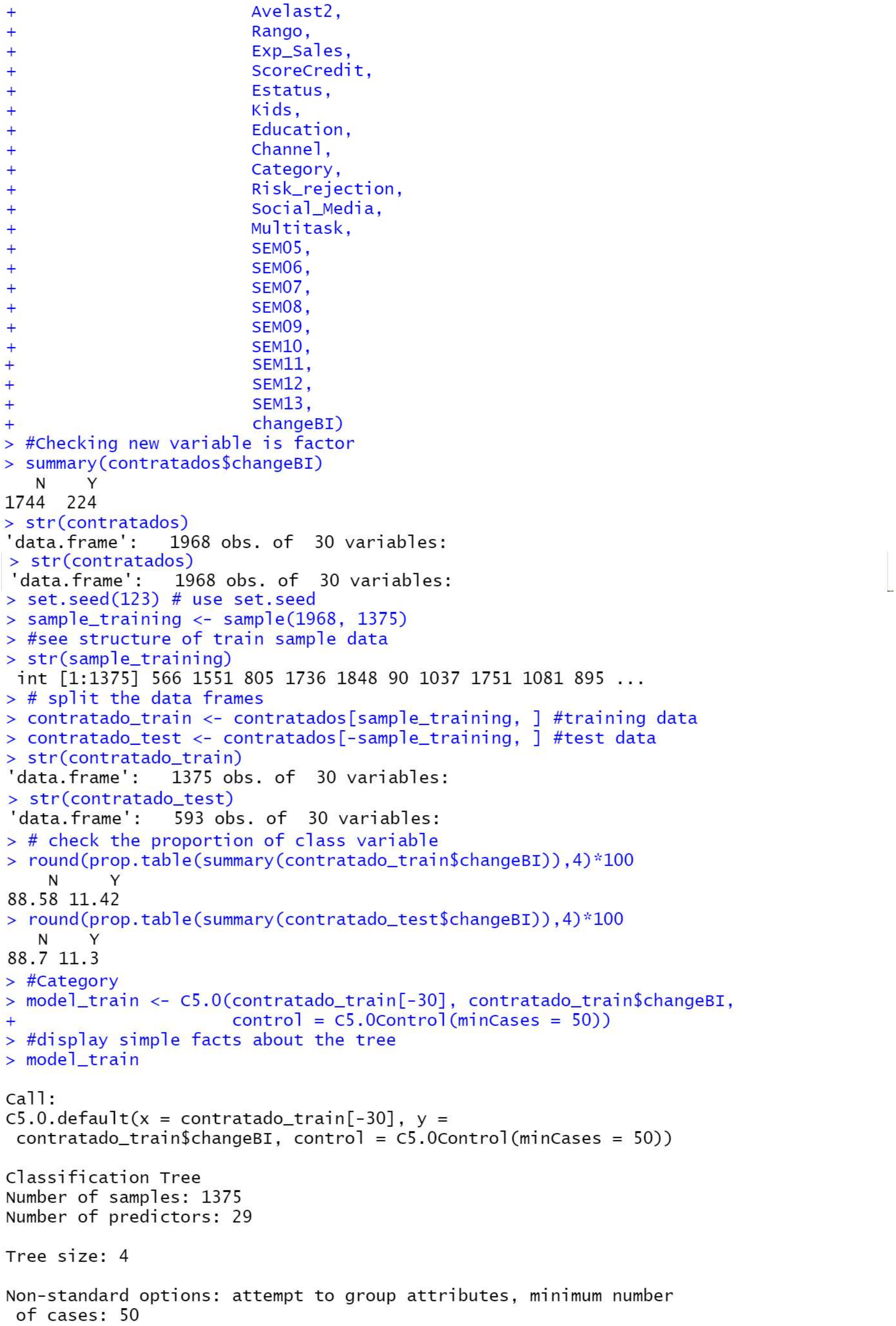


```
> library(gmodels)
> CrossTable(contratado_test$Desire, contratado_pred,
+            prop.chisq = FALSE, prop.c = FALSE, prop.r = FALSE,
+            dnn = c('Actual Category', 'Predicted Category'))


   Cell Contents
|-------------------------|
|                       N |
|         N / Table Total |
|-------------------------|


Total Observations in Table:  593
                | Predicted Category
Actual Category |         N |         Y | Row Total |
----------------|-----------|-----------|-----------|
              N |       339 |        34 |       373 |
                |     0.572 |     0.057 |           |
----------------|-----------|-----------|-----------|
              Y |        35 |       185 |       220 |
                |     0.059 |     0.312 |           |
----------------|-----------|-----------|-----------|
   Column Total |       374 |       219 |       593 |
----------------|-----------|-----------|-----------|
```

Decision tree – switching areas



```
> contratados1 <- read.csv("Motor9.csv", sep=',', header = TRUE,
+                          fill = TRUE, quote="\"", stringsAsFactors = TRUE)
> #explore data structure
> str(contratados1)
'data.frame':   1968 obs. of  39 variables:
 $ ID             : int  46044 94249 46275 46131 45927 45902 46161 46026 45983 45989 ...
 $ Sucursal       : int  9795 3007 190 200 202 213 214 259 534 550 ...
 $ Sexo           : Factor w/ 2 levels "H","M": 1 1 2 1 1 2 2 1 1 1 ...
 $ Tav_Hire       : Factor w/ 3 levels "+1W","<3d","1W": 2 2 2 3 3 2 1 2 3 3 ...
 $ T_EvalFin      : int  9 9 7 10 10 7 8 2 4 4 ...
 $ WeekBeg_Prod   : int  4 4 4 4 3 3 4 4 7 8 ...
 $ SEM01          : int  0 0 0 0 0 0 0 0 0 0 ...
 $ SEM02          : int  0 0 0 0 0 0 0 0 0 0 ...
 $ SEM03          : int  0 0 0 0 52 52 0 0 0 0 ...
 $ SEM04          : int  58 58 54 54 91 91 47 47 0 0 ...
 $ SEM05          : int  93 93 43 43 88 88 35 35 0 0 ...
 $ SEM06          : int  95 95 0 0 84 84 0 0 0 0 ...
 $ SEM07          : int  78 78 0 0 103 103 0 0 32 0 ...
 $ SEM08          : int  94 94 0 0 82 82 0 0 42 46 ...
 $ SEM09          : int  96 96 0 0 109 109 0 0 0 43 ...
$ SEM10          : int  105 105 0 0 97 97 0 0 0 53 ...
$ SEM11          : int  93 93 0 0 71 71 0 0 0 0 ...
$ SEM12          : int  91 91 0 0 103 103 0 0 0 0 ...
$ SEM13          : int  98 98 0 0 54 54 0 0 0 0 ...
$ Sem_reg        : int  13 13 5 5 13 13 5 5 8 10 ...
$ Sem_traba      : int  9 9 1 1 10 10 1 1 1 2 ...
$ Prod_total     : int  901 901 97 97 934 934 82 82 74 142 ...
$ Average_WW     : int  100 100 97 97 93 93 82 82 74 71 ...
$ Avelast2       : int  79 79 95 95 53 53 50 50 56 56 ...
$ Rango          : Factor w/ 4 levels "18_25","26_35",..: 2 2 1 1 1 1 1 2 2 2 ...
$ Exp_Sales      : int  4 4 0 4 4 0 0 0 1 1 ...
$ ScoreCredit    : Factor w/ 4 levels "Exc","Fair","Good",..: 1 1 3 3 3 3 1 1 1 1 ...
$ Estatus        : Factor w/ 4 levels "D","M","O","S": 2 2 2 4 4 4 2 1 4 4 ...
$ Kids           : Factor w/ 2 levels "N","Y": 1 2 2 2 1 2 2 2 1 1 ...
$ Education      : Factor w/ 2 levels "FE","HE": 1 1 1 1 1 1 2 1 1 1 ...
$ Category       : Factor w/ 4 levels "Elite","Good",..: 1 1 1 1 1 1 1 1 1 1 ...
$ Candidato      : Factor w/ 2 levels "N","Y": 2 2 2 2 2 2 2 2 2 2 ...
$ Current_state : Factor w/ 3 levels "ACTIVE","CHANGE",..: 2 2 1 3 1 3 1 1 1 1 ...
$ Risk_rejection: Factor w/ 2 levels "N","Y": 2 2 2 1 2 2 2 2 2 2 ...
 $ Social_Media  : Factor w/ 2 levels "N","Y": 2 2 2 2 1 2 2 2 2 2 ...
 $ Multitask     : Factor w/ 2 levels "N","Y": 2 2 2 2 2 1 2 2 2 2 ...
 $ Community     : Factor w/ 2 levels "N","Y": 2 1 2 1 2 2 1 2 2 2 ...
 $ Communication : Factor w/ 2 levels "N","Y": 2 2 2 2 2 2 2 2 2 2 ...
 $ Channel       : Factor w/ 4 levels "Agcy","Fbook",..: 2 2 2 3 3 2 3 2 2 2 ...
> #new variable to identify the staff that is changing in the company
> contratados <- mutate(contratados1,
+                        changeBI = ifelse(Current_state == "CHANGE", "Y","N"))
> #Convert on factor the new variable
> contratados$changeBI <- as.factor(contratados$changeBI)
> #Subdataset only variables of interest
> contratados <- select(contratados,
+                        Sexo,
+                        Tav_Hire,
+                        T_EvalFin,
+                        WeekBeg_Prod,
+                        Sem_reg,
+                        Sem_traba,
+                        Prod_total,
+                        Average_WW,
```



```
+                       Avelast2,
+                       Rango,
+                       Exp_Sales,
+                       ScoreCredit,
+                       Estatus,
+                       Kids,
+                       Education,
+                       Channel,
+                       Category,
+                       Risk_rejection,
+                       Social_Media,
+                       Multitask,
+                       SEM05,
+                       SEM06,
+                       SEM07,
+                       SEM08,
+                       SEM09,
+                       SEM10,
+                       SEM11,
+                       SEM12,
+                       SEM13,
+                       changeBI)
> #Checking new variable is factor
> summary(contratados$changeBI)
   N    Y 
1744  224 
> str(contratados)
'data.frame':	1968 obs. of  30 variables:
```

```
> str(contratados)
'data.frame':	1968 obs. of  30 variables:
```

```
> set.seed(123) # use set.seed
> sample_training <- sample(1968, 1375)
> #see structure of train sample data
> str(sample_training)
 int [1:1375] 566 1551 805 1736 1848 90 1037 1751 1081 895 ...
> # split the data frames
> contratado_train <- contratados[sample_training, ] #training data
> contratado_test <- contratados[-sample_training, ] #test data
> str(contratado_train)
'data.frame':	1375 obs. of  30 variables:
> str(contratado_test)
'data.frame':	593 obs. of  30 variables:
> # check the proportion of class variable
> round(prop.table(summary(contratado_train$changeBI)),4)*100
    N     Y 
88.58 11.42 
> round(prop.table(summary(contratado_test$changeBI)),4)*100
   N    Y 
88.7 11.3 
> #Category
> model_train <- C5.0(contratado_train[-30], contratado_train$changeBI,
+                     control = C5.0Control(minCases = 50))
> #display simple facts about the tree
> model_train

Call:
C5.0.default(x = contratado_train[-30], y =
 contratado_train$changeBI, control = C5.0Control(minCases = 50))

Classification Tree
Number of samples: 1375 
Number of predictors: 29 

Tree size: 4 

Non-standard options: attempt to group attributes, minimum number
 of cases: 50
```

```
Class specified by attribute `outcome'

Read 1375 cases (30 attributes) from undefined.data

Decision tree:

Exp_Sales <= 2: N (771/1)
Exp_Sales > 2:
:...Channel in {Friend,HR}: N (192/6)
    Channel in {Agcy,Fbook}:
    :...Category in {Elite,Good}: Y (162/35)
        Category in {Mean,Slow}: N (250/23)
Evaluation on training data (1375 cases):

	    Decision Tree
	  ----------------
	  Size      Errors

	     4   65( 4.7%)   <<


	   (a)   (b)    <-classified as
	  ----  ----
	  1183    35    (a): class N
	    30   127    (b): class Y


	Attribute usage:

	100.00% Exp_Sales
	 43.93% Channel
	 29.96% Category


Time: 0.0 secs

> # plot the tree
> plot(model_train)
> ## Step 3: Evaluating model performance by using test data
> # create a factor vector of predictions on test data
> contratado_pred <- predict(model_train, contratado_test)
> str(contratado_pred)
 Factor w/ 2 levels "N","Y": 1 1 1 1 1 1 1 2 2 1 ...
> # cross tabulation of predicted versus actual classes
> install.packages("gmodels")
Error in install.packages : Updating loaded packages
> install.packages("gmodels")
> CrossTable(contratado_test$changeBI, contratado_pred,
+            prop.chisq = FALSE, prop.c = FALSE, prop.r = FALSE,
+            dnn = c('Actual Category', 'Predicted Category'))


   Cell Contents
|-------------------------|
|                       N |
|         N / Table Total |
|-------------------------|

 
Total Observations in Table:  593 

 
                | Predicted Category 
Actual Category |         N |         Y | Row Total | 
----------------|-----------|-----------|-----------|
              N |       514 |        12 |       526 | 
                |     0.867 |     0.020 |           | 
----------------|-----------|-----------|-----------|
              Y |        11 |        56 |        67 | 
                |     0.019 |     0.094 |           | 
----------------|-----------|-----------|-----------|
   Column Total |       525 |        68 |       593 | 
----------------|-----------|-----------|-----------|
```