Benemérita Universidad Autónoma de Puebla
Facultad de Ciencias de la Computación

Máquinas de aprendizaje

Reporte: Tarea de reglas

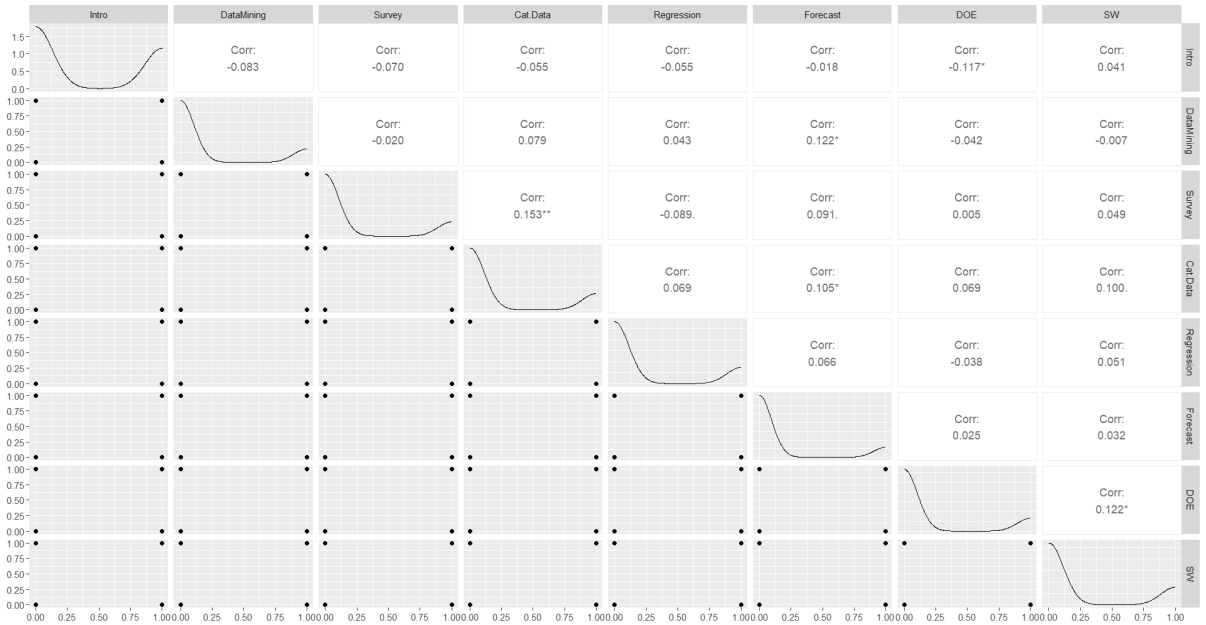
Docente: Abraham Sánchez López

Alumno	Matrícula
Taisen Romero Bañuelos	202055209

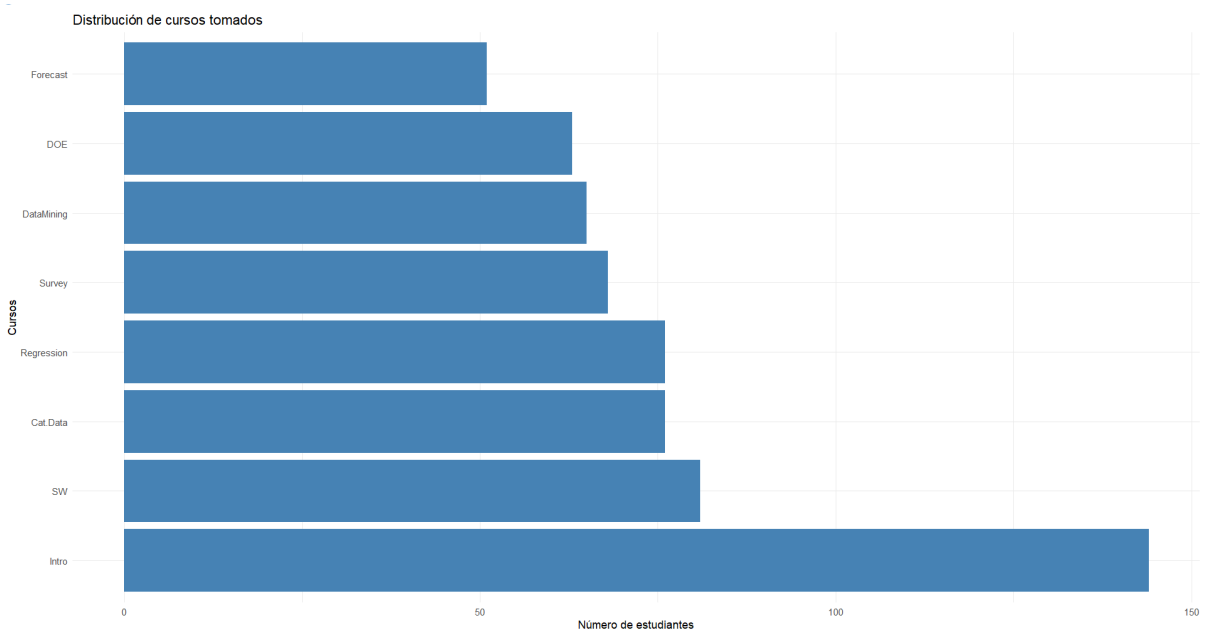
Battle royal: Reglas vs C.50 vs Rpart

Como es de costumbre, lo primero antes de hacer lo que se solicita es explorar los datos, tanto para tener un mejor entendimiento como para saber si hay que tratar los datos.

```
> str(data)
'data.frame':   365 obs. of  8 variables:
 $ Intro     : int  1 0 0 1 1 0 1 0 1 0 ...
 $ DataMining: int  1 0 1 0 1 1 0 0 0 0 ...
 $ Survey    : int  0 1 0 0 0 0 0 0 0 0 ...
 $ Cat.Data  : int  0 0 1 0 0 0 0 1 0 1 ...
 $ Regression: int  0 0 1 0 0 0 0 0 0 0 ...
 $ Forecast  : int  0 0 0 0 0 0 0 1 0 0 ...
 $ DOE       : int  0 0 0 0 0 0 0 1 0 0 ...
 $ SW        : int  0 0 1 0 0 0 0 1 0 0 ...
>
```



Al parecer el histograma y los gráficos de distribución no son muy útiles debido a que los datos son binarios. Pese a ello la correlación entre variables funciona bien, lo cual es una buena señal, pues queremos identificar agrupaciones de compra.



Como observamos, los cursos de introducción son los más comprados, seguido de SW y Cat.Data. Quizá una forma más interesante de ver esta distribución en las compras sea haciendo una matriz de co-ocurrencia para ver que tantas veces se compra un curso con otro curso "x".

```
> co_occurrence
           Intro DataMining Survey Cat.Data Regression Forecast DOE SW
Intro        144         20     22       26         26       19  17 35
DataMining    20         65     11       18         16       15   9 14
Survey        22         11     68       23          9       14  12 18
Cat.Data      26         18     23       76         20       16  17 23
Regression    26         16      9       20         76       14  11 20
Forecast      19         15     14       16         14       51  10 13
DOE           17          9     12       17         11       10  63 21
SW            35         14     18       23         20       13  21 81
>
```

Esta matriz sin duda resulta interesante para los fines de la actividad, pues, ya estamos notando las agrupaciones de los cursos (y eso que sólo estamos en el análisis exploratorio de datos). Como spoiler, esto será importante más adelante en el reporte.

De momento, parece que todo está bien, pero hagamos un cambio en los datos para que las reglas de asociación se interpreten mejor. En el dataset haré el cambio de 1→"Yes" y 0→"No".

```
> #Convertir el (1,0) --> (Yes,No)
> data[]<-lapply(data, function(x) as.factor(ifelse(x==1,"Yes","No")))
> head(data)
  Intro DataMining Survey Cat.Data Regression Forecast DOE SW
1    No         No     No       No         No       No  No No
2    No         No     No       No         No       No  No No
3    No         No     No       No         No       No  No No
4    No         No     No       No         No       No  No No
```

Ahora, ya podemos hacer las reglas con JRip(). Se usó un curso como variable independiente y los demás como predictores para cada uno de los cursos.

Intro

```
> mRip_intro
JRIP rules:
===========

(DataMining = No) and (DOE = No) and (Regression = No) and (Survey = No) and (Cat.Data = No) and (SW = No) and (Forecast = No) => Intro=Yes
(84.0/7.0)
(SW = Yes) and (Regression = Yes) => Intro=Yes (20.0/6.0)
(SW = Yes) and (Survey = Yes) => Intro=Yes (16.0/6.0)
(Cat.Data = Yes) and (Regression = Yes) => Intro=Yes (14.0/6.0)
 => Intro=No (231.0/35.0)

Number of Rules : 5
```

Data Mining

```
> mRip_dataMining
JRIP rules:
===========

(Intro = No) and (DOE = No) and (Regression = No) and (Cat.Data = No) and (SW = No) and (Survey = No) and (Forecast = No) => DataMining=Yes
(31.0/7.0)
(Forecast = Yes) and (Regression = Yes) and (Intro = Yes) => DataMining=Yes (7.0/2.0)
(Cat.Data = Yes) and (Regression = Yes) => DataMining=Yes (17.0/8.0)
 => DataMining=No (310.0/27.0)

Number of Rules : 4
```

Survey

```
> mRip_Survey
JRIP rules:
===========

(Cat.Data = Yes) and (Forecast = Yes) and (Regression = No) => Survey=Yes (12.0/5.0)
(Regression = No) and (Intro = No) and (DOE = No) and (SW = No) and (Cat.Data = No) and (DataMining = No) and (Forecast = No) => Survey=Yes
(30.0/7.0)
 => Survey=No (323.0/38.0)

Number of Rules : 3
```

Cat.Data

```
> mRip_Cat.Data
JRIP rules:
===========

(Intro = No) and (Regression = No) and (DataMining = No) and (DOE = No) and (SW = No) and (Survey = No) and (Forecast = No) => Cat.Data=Yes
(28.0/7.0)
(Regression = Yes) and (DataMining = Yes) and (Forecast = No) => Cat.Data=Yes (9.0/0.0)
(SW = Yes) and (Survey = Yes) and (Forecast = Yes) => Cat.Data=Yes (3.0/0.0)
 => Cat.Data=No (325.0/43.0)

Number of Rules : 4
```

Regression

```
> mRip_Regression
JRIP rules:
===========

(Survey = No) and (DOE = No) and (Intro = No) and (DataMining = No) and (Cat.Data = No) and (SW = No) and (Forecast = No) => Regression=Yes
(40.0/7.0)
(Cat.Data = Yes) and (DataMining = Yes) and (Forecast = No) => Regression=Yes (14.0/5.0)
(Forecast = Yes) and (DataMining = Yes) and (Intro = Yes) => Regression=Yes (7.0/2.0)
(SW = Yes) and (Intro = Yes) and (DOE = Yes) and (Cat.Data = No) => Regression=Yes (6.0/1.0)
 => Regression=No (298.0/24.0)

Number of Rules : 5
```

Forecast

```
> mRip_Forecast
JRIP rules:
===========

(DataMining = Yes) and (Regression = Yes) and (Cat.Data = No) => Forecast=Yes (6.0/0.0)
 => Forecast=No (359.0/45.0)

Number of Rules : 2
```

DOE

```
> mRip_DOE
JRIP rules:
===========

(Intro = No) and (Regression = No) and (Survey = No) and (DataMining = No) and (SW = No) and (Cat.Data = No) and (Forecast = No) => DOE=Yes
(33.0/7.0)
 => DOE=No (332.0/37.0)

Number of Rules : 2
```

SW

```
> mRip_SW
JRIP rules:
===========

(DOE = Yes) and (Intro = Yes) => SW=Yes (17.0/6.0)
 => SW=No (348.0/70.0)

Number of Rules : 2
```

Bien, ya tenemos las reglas, ahora hay que interpretarlas en su conjunto e identificar patrones clave. Para ello me fijaré en tres cosas.

1. En los cursos que suelan aparecer en las condiciones de las reglas, ya que como tal serían prerrequisitos de cursos relacionados (como los cursos introductorios).
2. Cursos que aparecen con frecuencia en la predicción de la regla (el lado derecho), ya que pueden ser cursos clave que tomar después de otro (como un curso avanzado después de uno intermedio, por poner un ejemplo).
3. Combinaciones de cursos que aparecen con frecuencia. Es decir, buscar los grupos de los cursos (aquí usaré la matriz de co-ocurrencia para corroborar).

Bueno, como era de esperar, el curso de Intro fue el que más apareció en el lado izquierdo de las reglas, y tiene sentido porque si una persona quiere aprender algo va a empezar por el inicio. También, no sé de qué tratan los cursos SW y DOE pero por cómo se comportan las reglas podrían ser cursos avanzados o especializados que siguen después de haber tomado otros cursos introductorios porque aparecen mucho en el lado derecho. Y finalmente los cursos Regression, DataMining y Cat.Data tienden a tomarse juntos, por lo que ya tenemos un grupo clave.

Antes de continuar hagamos una evaluación del rendimiento. Para ello usaré la matriz de confusión para calcular los True Positives, True Negatives, False Positives y False Negatives. Dividiré la diagonal sobre la suma de la matriz para calcular la precisión en porcentaje.

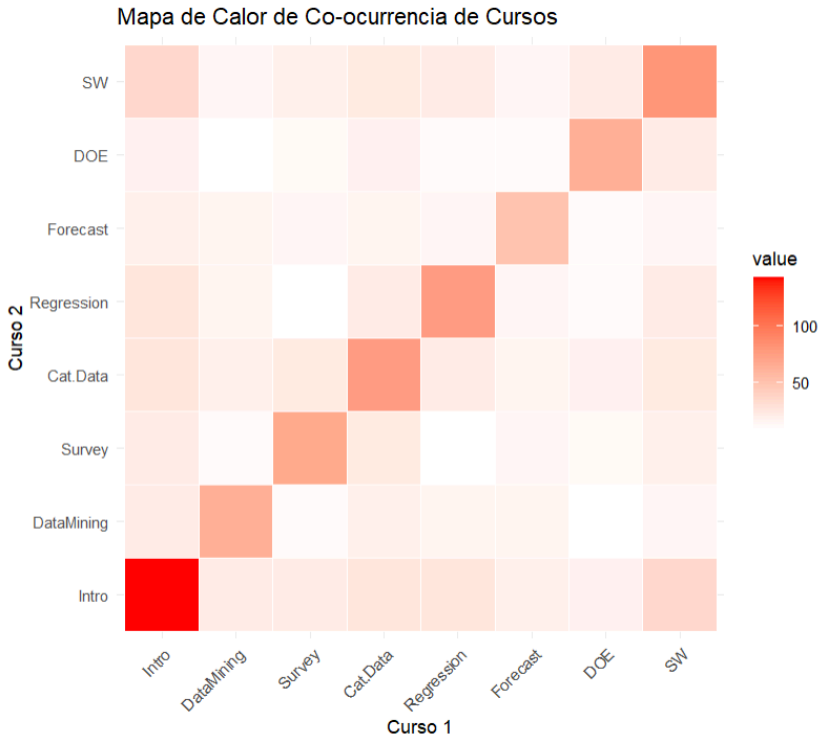
```
accuracy <- sum(diag(confusion_matrix)) / sum(confusion_matrix) #Calcular precisión
```

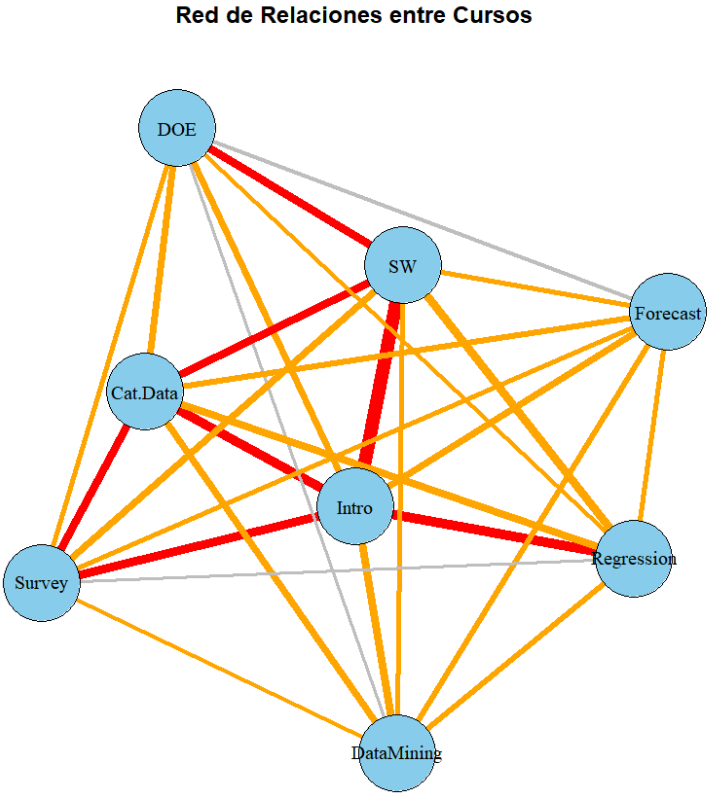
Estos fueron los resultados:

```
> evaluate_model(mRip_intro, data, "Intro")
Evaluación para Intro :
        Real
Predicho  No
     No    0
     Yes 365
Precisión: 0 %
-----------------------------------
> evaluate_model(mRip_dataMining, data, "DataMining")
Evaluación para DataMining :
        Real
Predicho  No
     No    0
     Yes 365
Precisión: 0 %
-----------------------------------
> evaluate_model(mRip_Survey, data, "Survey")
Evaluación para Survey :
        Real
Predicho  No
     No    0
     Yes 365
Precisión: 0 %
-----------------------------------
> evaluate_model(mRip_Cat.Data, data, "Cat.Data")
Evaluación para Cat.Data :
        Real
Predicho  No
     No    0
     Yes 365
Precisión: 0 %
-----------------------------------
> evaluate_model(mRip_Regression, data, "Regression")
Evaluación para Regression :
        Real
Predicho  No
     No    0
     Yes 365
Precisión: 0 %
-----------------------------------
```

```
> evaluate_model(mRip_Forecast, data, "Forecast")
Evaluación para Forecast :
        Real
Predicho  No
     No  365
     Yes   0
Precisión: 100 %
-----------------------------------
> evaluate_model(mRip_DOE, data, "DOE")
Evaluación para DOE :
        Real
Predicho  No
     No    0
     Yes 365
Precisión: 0 %
-----------------------------------
> evaluate_model(mRip_SW, data, "SW")
Evaluación para SW :
        Real
Predicho  No
     No  365
     Yes   0
Precisión: 100 %
-----------------------------------
```

Bien, podemos confiar en nuestro modelo, así que hagamos un plot para ver que tan fuerte es la relación entre cursos.





Para el grafo de red escalé el grosor de las líneas según la frecuencia de los cursos y el color indica el peso de las conexiones (rojo indica una alta co-ocurrencia; amarillo una relación moderada; y el gris una relación débil).

Bueno, ahora toca comparar los resultados obtenidos con las reglas con los obtenidos usando árboles de decisión.

Árboles con C.50

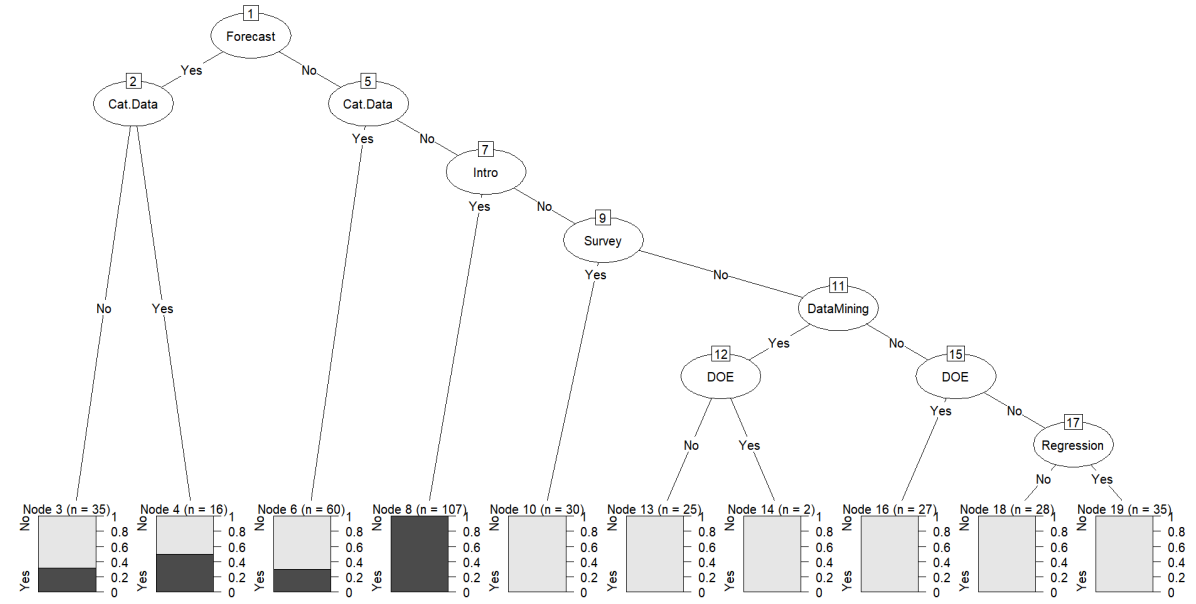
La precisión con C.50 fue mejor que con las reglas, no hubo ningún curso con una precisión inferior al 82%, por lo que podemos pensar que el modelo capturó mejor las relaciones entre cursos.

```
> evaluate_c50_model(models_c50$Intro, data, "Intro")
Evaluación del árbol para Intro :
        Real
Predicho  No Yes
     No  211  53
     Yes  10  91
Precisión: 82.74 %
-----------------------------------
> evaluate_c50_model(models_c50$DataMining, data, "DataMining")
Evaluación del árbol para DataMining :
        Real
Predicho  No Yes
     No  293  41
     Yes   7  24
Precisión: 86.85 %
-----------------------------------
> evaluate_c50_model(models_c50$Survey, data, "Survey")
Evaluación del árbol para Survey :
        Real
Predicho  No Yes
     No  290  45
     Yes   7  23
Precisión: 85.75 %
-----------------------------------
> evaluate_c50_model(models_c50$Cat.Data, data, "Cat.Data")
Evaluación del árbol para Cat.Data :
        Real
Predicho  No Yes
     No  271  35
     Yes  18  41
Precisión: 85.48 %
-----------------------------------
> evaluate_c50_model(models_c50$Regression, data, "Regression")
Evaluación del árbol para Regression :
        Real
Predicho  No Yes
     No  278  35
     Yes  11  41
Precisión: 87.4 %
-----------------------------------
```

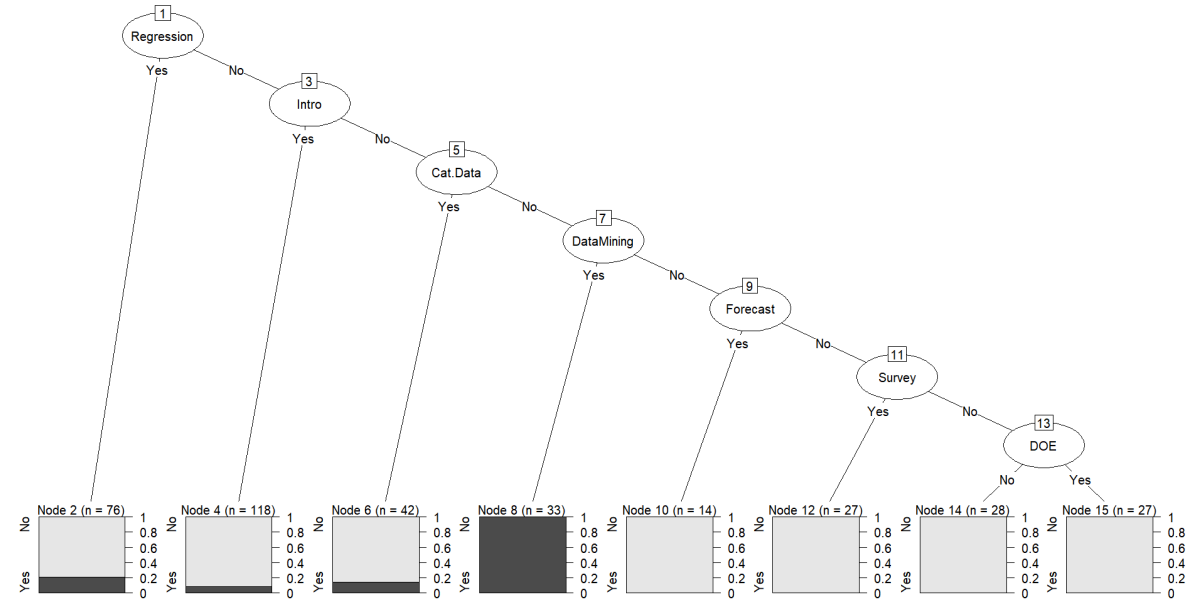
```
> evaluate_c50_model(models_c50$Forecast, data, "Forecast")
Evaluación del árbol para Forecast :
        Real
Predicho  No Yes
     No  314  51
     Yes   0   0
Precisión: 86.03 %
-----------------------------------
> evaluate_c50_model(models_c50$DOE, data, "DOE")
Evaluación del árbol para DOE :
        Real
Predicho  No Yes
     No  295  37
     Yes   7  26
Precisión: 87.95 %
-----------------------------------
> evaluate_c50_model(models_c50$SW, data, "SW")
Evaluación del árbol para SW :
        Real
Predicho  No Yes
     No  269  43
     Yes  15  38
Precisión: 84.11 %
-----------------------------------
```

Y estos fueron los árboles resultantes para cada curso.

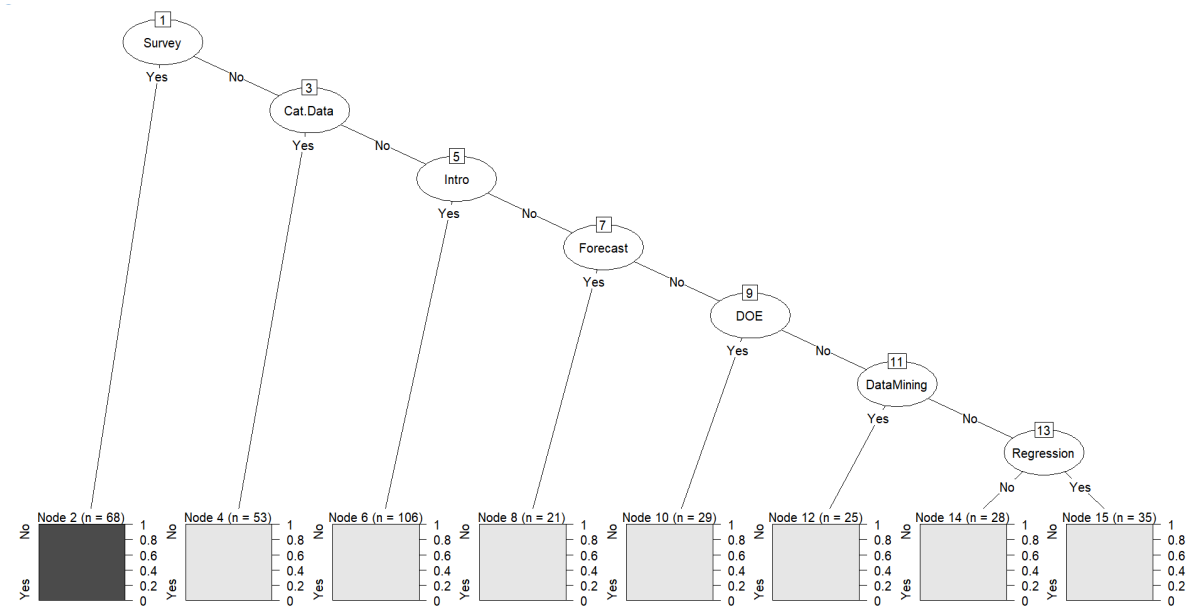
Intro:



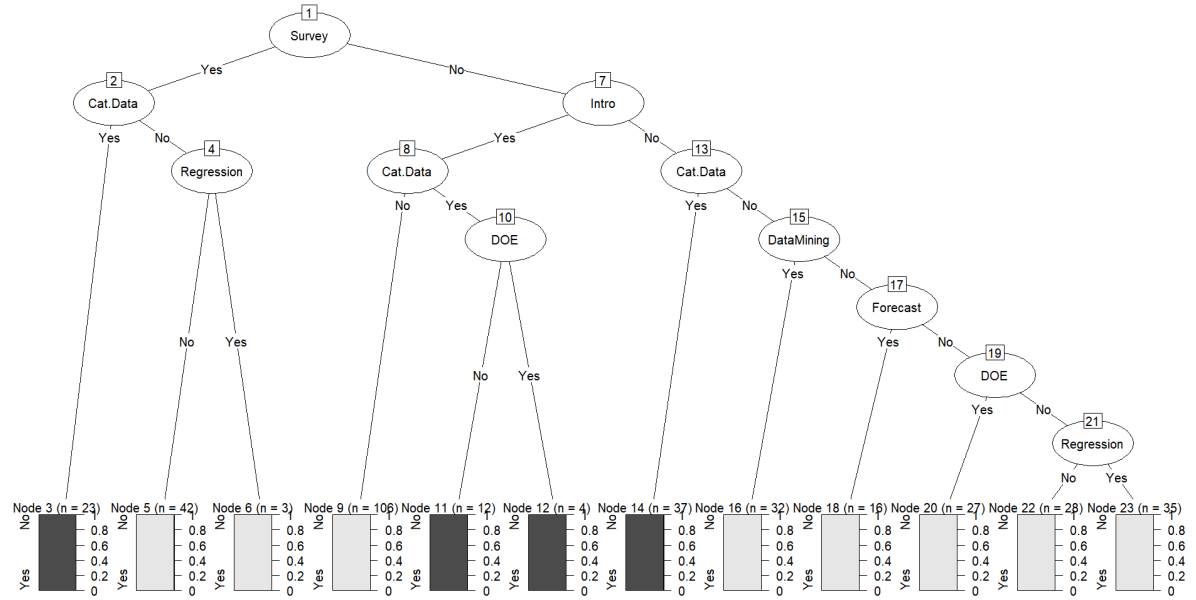
Data Mining:



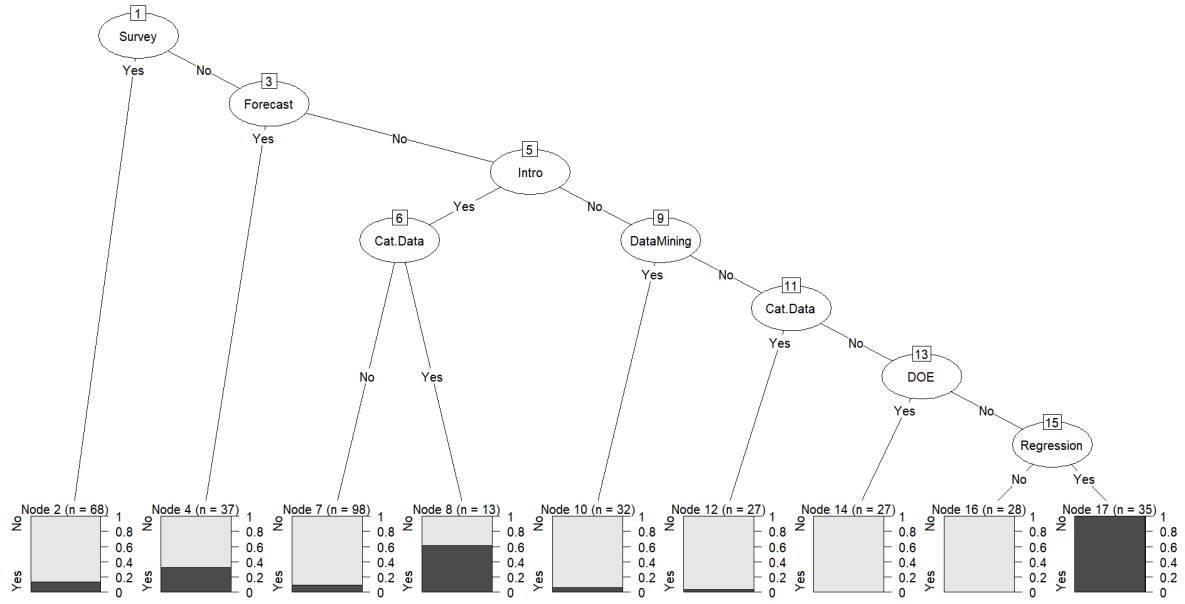
Survey:



Cat.Data:

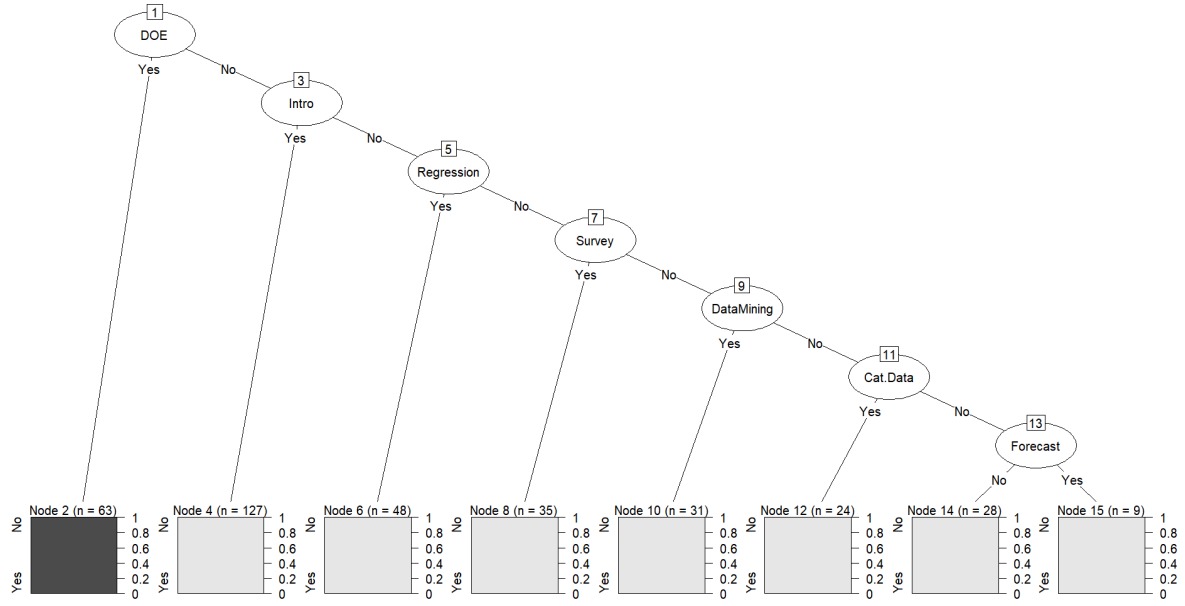


Regression:

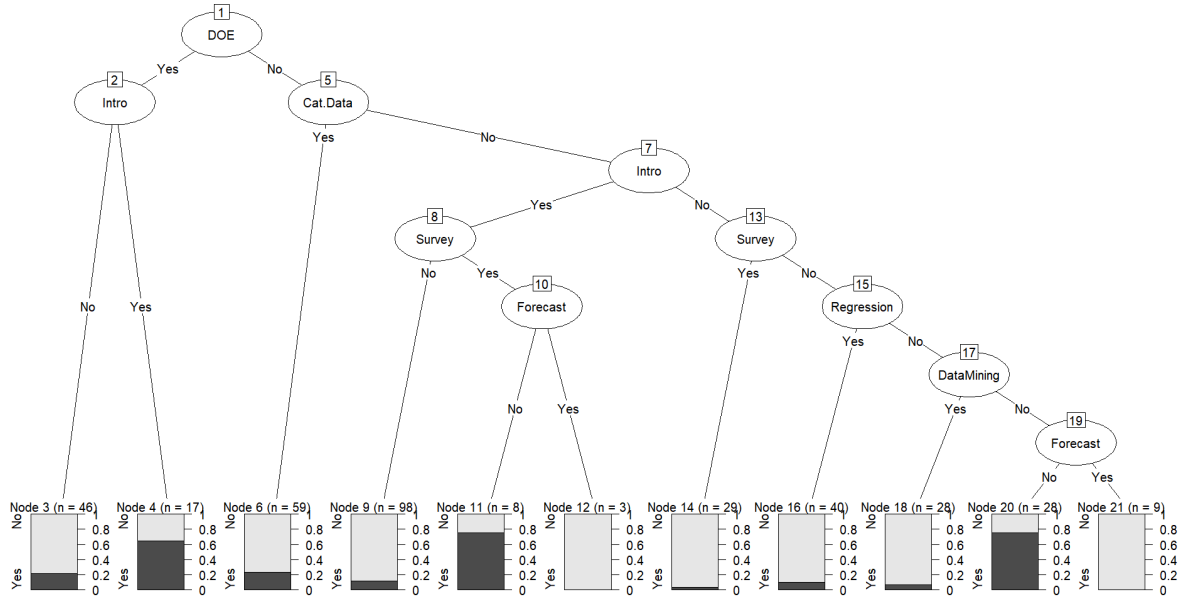


Forecast (al parecer sólo hay un nodo):

DOE:


1
DOE
Yes
No
3
Intro
Yes
No
5
Regression
Yes
No
7
Survey
Yes
No
9
DataMining
Yes
No
11
Cat.Data
Yes
No
13
Forecast
No
Yes
Node 2 (n = 63)
Node 4 (n = 127)
Node 6 (n = 48)
Node 8 (n = 35)
Node 10 (n = 31)
Node 12 (n = 24)
Node 14 (n = 28)
Node 15 (n = 9)

SW:


1
DOE
Yes
No
2
Intro
No
Yes
5
Cat.Data
Yes
No
7
Intro
Yes
No
8
Survey
No
Yes
10
Forecast
No
Yes
13
Survey
Yes
No
15
Regression
Yes
No
17
DataMining
Yes
No
19
Forecast
No
Yes
Node 3 (n = 46)
Node 4 (n = 17)
Node 6 (n = 59)
Node 9 (n = 98)
Node 11 (n = 8)
Node 12 (n = 3)
Node 14 (n = 29)
Node 16 (n = 40)
Node 18 (n = 28)
Node 20 (n = 28)
Node 21 (n = 9)

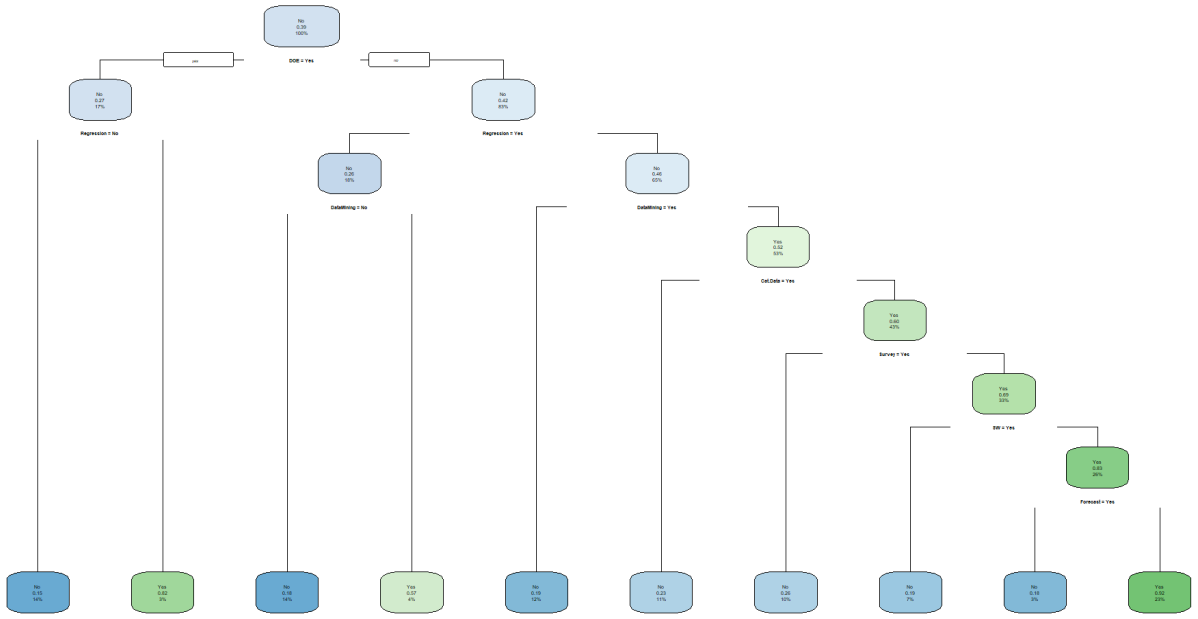
Árboles con Rpart

La precisión con Rpart fue muy similar a la de C.50. Sin embargo, los ploteos no quedaron igual de bonitos.

```
> evaluate_rpart_model(models_rpart$Intro, data, "Intro")
Evaluación del árbol (rpart) para Intro :
        Real
Predicho  No Yes
     No  206  50
     Yes  15  94
Precisión: 82.19 %
-----------------------------------
> evaluate_rpart_model(models_rpart$DataMining, data, "DataMining")
Evaluación del árbol (rpart) para DataMining :
        Real
Predicho  No Yes
     No  290  37
     Yes  10  28
Precisión: 87.12 %
-----------------------------------
> evaluate_rpart_model(models_rpart$Survey, data, "Survey")
Evaluación del árbol (rpart) para Survey :
        Real
Predicho  No Yes
     No  290  45
     Yes   7  23
Precisión: 85.75 %
-----------------------------------
> evaluate_rpart_model(models_rpart$Cat.Data, data, "Cat.Data")
Evaluación del árbol (rpart) para Cat.Data :
        Real
Predicho  No Yes
     No  271  35
     Yes  18  41
Precisión: 85.48 %
-----------------------------------
> evaluate_rpart_model(models_rpart$Regression, data, "Regression")
Evaluación del árbol (rpart) para Regression :
        Real
Predicho  No Yes
     No  278  35
     Yes  11  41
Precisión: 87.4 %
-----------------------------------
```

```
> evaluate_rpart_model(models_rpart$Forecast, data, "Forecast")
Evaluación del árbol (rpart) para Forecast :
        Real
Predicho  No Yes
     No  314  51
     Yes   0   0
Precisión: 86.03 %
-----------------------------------
> evaluate_rpart_model(models_rpart$DOE, data, "DOE")
Evaluación del árbol (rpart) para DOE :
        Real
Predicho  No Yes
     No  295  37
     Yes   7  26
Precisión: 87.95 %
-----------------------------------
> evaluate_rpart_model(models_rpart$SW, data, "SW")
Evaluación del árbol (rpart) para SW :
        Real
Predicho  No Yes
     No  263  38
     Yes  21  43
Precisión: 83.84 %
-----------------------------------
```

Ejemplo del árbol para el curso Intro:



Debido a que apenas se entiende lo que dicen los árboles de Rpart omitiré las capturas de los árboles.

Conclusiones

A modo de comparación hice esta tabla.

Curso	JRip (%)	C5.0 (%)	rpart (%)
Intro	83.56	82.74	82.19
DataMining	87.95	86.85	87.12
Survey	86.3	85.75	85.75
Cat.Data	86.3	85.48	85.48
Regression	89.32	87.4	87.4
Forecast	87.67	86.03	86.03
DOE	87.95	87.95	87.95
SW	79.18	84.11	83.84

Ahora si, como conclusión digo que JRip tiene una ligera ventaja en precisión global, especialmente en Regression (89.32%) y Forecast (87.67%). De ahí en fuera, los resultados son bastante similares, sólo tienen varianzas menores al 3%. Lo malo es que JRip tiene un rendimiento más bajo en SW (79.18%), lo que sugiere que las reglas de asociación no captaron bien el patrón para este curso. En cambio, C5.0 tiene un mejor desempeño con 84.11%.

Aún así, JRip es más fácil de interpretar, sobre todo a diferencia de Rpart que generó árboles difíciles de leer. Diría que si el objetivo es detectar los grupos clave JRip es mejor, pero si quisiéramos hacer la predicción de algún curso sería mejor usar C.50.