Benemérita Universidad Autónoma de Puebla
Facultad de Ciencias de la Computación

Máquinas de aprendizaje

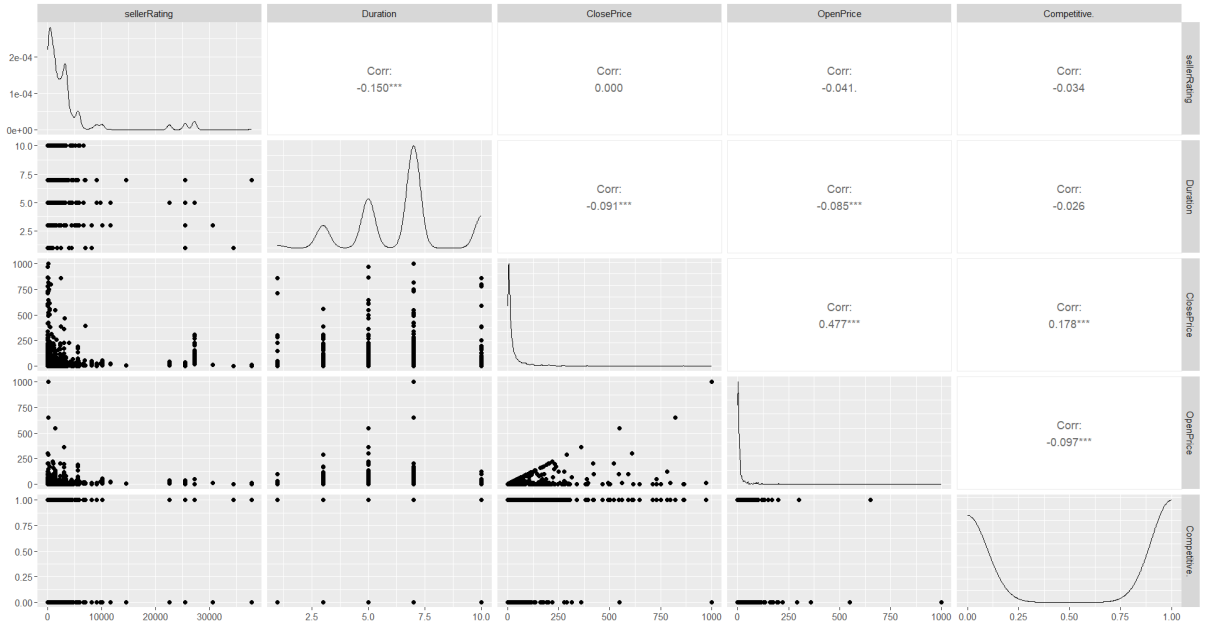
Reporte: Tarea de árboles de decisión

Docente: Abraham Sánchez López

Alumno	Matrícula
Taisen Romero Bañuelos	202055209

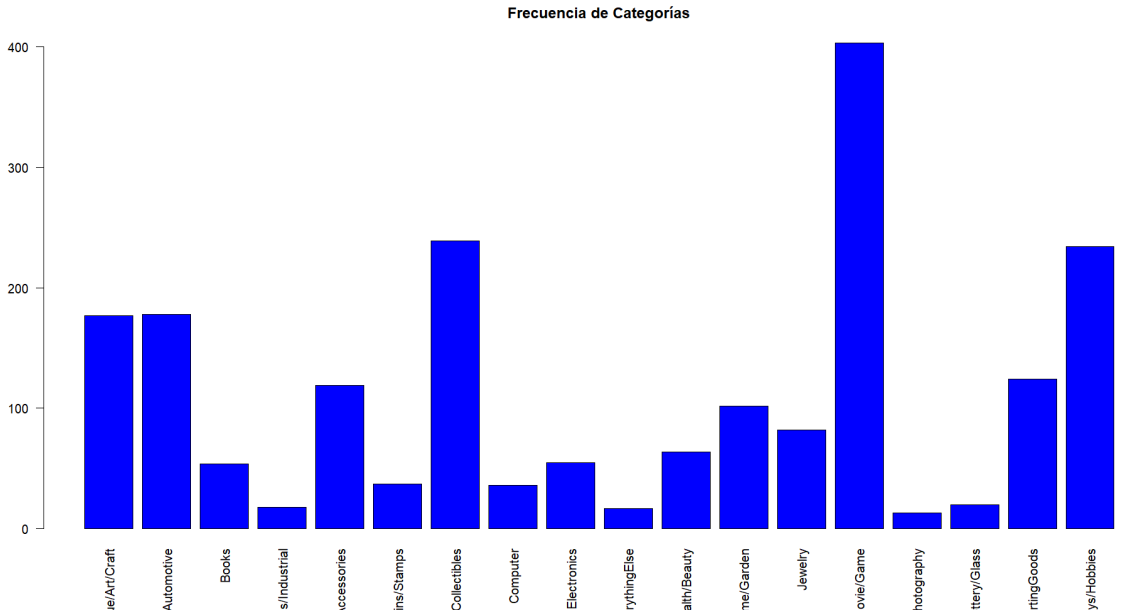
Tarea - Árboles de decisión

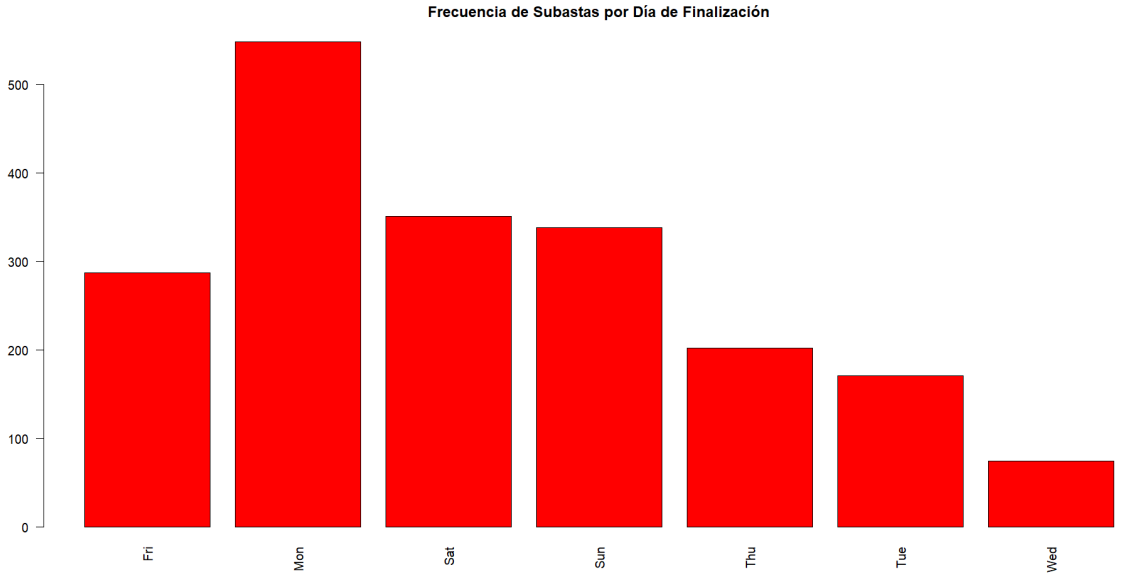
Lo primero es el análisis exploratorio de datos. Para agilizar el proceso usé la función ggpairs() para que me de la correlación, distribución e histograma de los datos numéricos en un único plot.



La correlación entre variables no es muy grande, la más notable es entre el precio de apertura y el de cierre (corr 0.47), lo que significa que cuanto mayor sea el precio de apertura el precio de cierre tiende a ser mayor. De ahí en fuera las demás correlaciones no me parecen significativas pero se podría decir también que las subastas más competitivas pueden llevar a un precio de cierre más alto (ClosePrice vs Competitive - Corr: 0.178). Sobre los gráficos de dispersión hay uno que llama mi atención, nuevamente se trata de OpenPrice vs ClosePrice. Este gráfico muestra una tendencia creciente, aunque hay muchos valores cerca de 0. Y sobre los histogramas, el de Competitive tiene una forma de “U”, lo que podría sugerir que los extremos (subastas muy competitivas o poco competitivas) se comportan diferente.

Para las variables categóricas quise saber cuál era la categoría más popular y también en qué día se cierran más subastas. Estos fueron los resultados.





Bueno, el primer plot no mostró por completo los nombres, pero en ambos gráficos los resultados hablan por sí mismos.

A) Bueno, aquí se nos pide hacer el árbol con minbucket=50 y maxdepth=7. Una vez hecho veamos qué nos dice la función summary().

```
          CP nsplit rel error    xerror       xstd
1 0.23897059      0 1.0000000 1.0000000 0.03152207
2 0.11213235      1 0.7610294 0.7720588 0.03026193
3 0.08823529      3 0.5367647 0.5808824 0.02797877
4 0.05330882      4 0.4485294 0.4632353 0.02589008
5 0.03676471      6 0.3419118 0.4007353 0.02451557
6 0.00000000      7 0.3051471 0.3455882 0.02311713

Variable importance
  ClosePrice    OpenPrice     Category sellerRating     Duration       endDay     currency
          40           39            9            6            3            2            1
```

La tabla de complejidad nos indica que el árbol ya está optimizado (por su CP=0.0, lo que significa que no habrá cambios si elegimos el mejor CP de forma automática y no manual). También, es evidente que las variables más importantes a considerar son el precio de apertura y cierre de la subasta, y tal vez la categoría también, pero considerablemente menos que las otras 2 mencionadas.

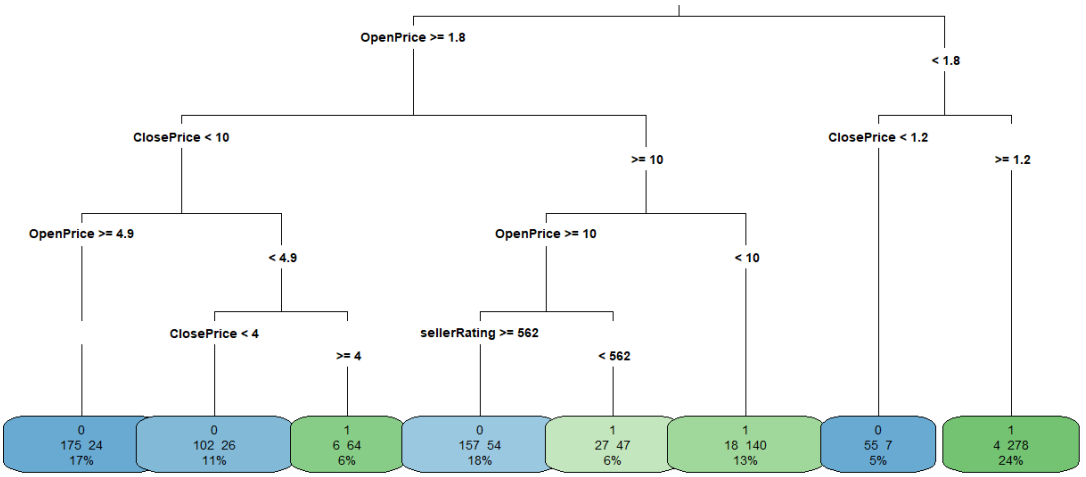
Un dato a considerar es que si imprimimos manualmente la tabla de complejidad del árbol nos damos cuenta de que la variable Category no fue usada pese a ser más importante que SellerRating. Esto me hace pensar que quizá podría establecer las variables que se usan en la construcción del árbol para evitar estas aparentes fallas, sin embargo, puede que haya otro motivo por el cual el modelo consideró mejor una variable sobre la otra.

```
> printcp(tree) #Tabla de complejidad del árbol no podado

Classification tree:
rpart(formula = Competitive. ~ ., data = train, method = "class",
    control = rpart.control(minbucket = 50, maxdepth = , cp = 0))

Variables actually used in tree construction:
[1] ClosePrice   OpenPrice    sellerRating

Root node error: 544/1184 = 0.45946

n= 1184

        CP nsplit rel error  xerror     xstd
1 0.238971      0   1.00000 1.00000 0.031522
2 0.112132      1   0.76103 0.77206 0.030262
3 0.088235      3   0.53676 0.58088 0.027979
4 0.053309      4   0.44853 0.46324 0.025890
5 0.036765      6   0.34191 0.40074 0.024516
6 0.000000      7   0.30515 0.34559 0.023117
>
```

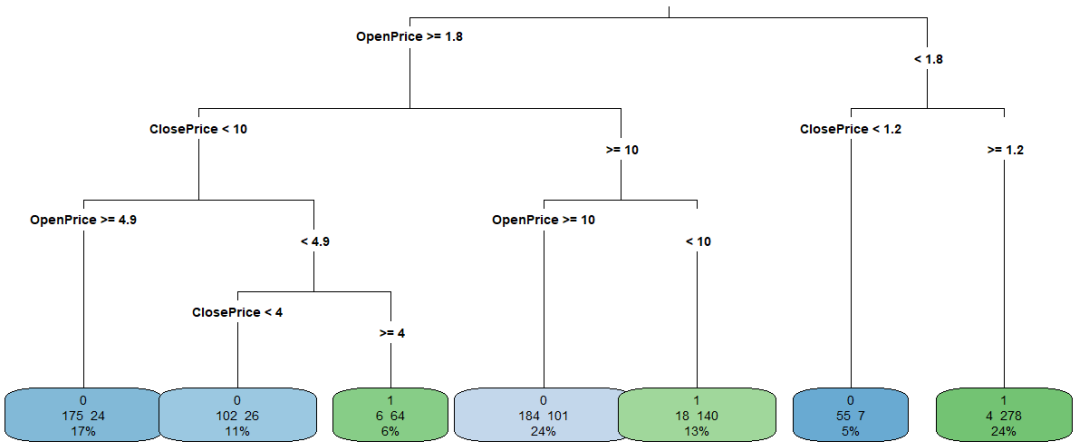
Como ya se dijo, el árbol ya estaba optimizado, sin embargo usaré un CP mayor para comparar. Usaré el que genera 6 splits (0.036).

```
> #El mejor CP era de 0.000, pero veamos que tal el segundo mejor
> pruned_tree1 <- prune(tree, cp = optimal_cp) #El CP óptimo (sin cambios = 0)
> pruned_tree2 <- prune(tree, cp = 0.036765) #Probar con un CP mayor
>
```

Árbol con CP óptimo (pruned_tree1):



Árbol con un CP mayor (pruned_tree2):



Como observamos, son casi idénticos pero con leves diferencias (hay una ramificación menos). Ahora comparemos sus reglas, aquí se nota más la diferencia del CP.

```
> rpart.rules(pruned_tree1)
 Competitive.
         0.11 when OpenPrice <  1.8         & ClosePrice <   1.2
         0.12 when OpenPrice >=         4.9 & ClosePrice <  10.0
         0.20 when OpenPrice is 1.8 to  4.9 & ClosePrice <   4.0
         0.26 when OpenPrice >=        10.3 & ClosePrice >=         10.0 & sellerRating >= 562
         0.64 when OpenPrice >=        10.3 & ClosePrice >=         10.0 & sellerRating <  562
         0.89 when OpenPrice is 1.8 to 10.3 & ClosePrice >=         10.0
         0.91 when OpenPrice is 1.8 to  4.9 & ClosePrice is  4.0 to 10.0
         0.99 when OpenPrice <  1.8         & ClosePrice >=          1.2
>
```

```
> rpart.rules(pruned_tree2)
 Competitive.
         0.11 when OpenPrice <  1.8          & ClosePrice <   1.2
         0.12 when OpenPrice >=         4.9 & ClosePrice <  10.0
         0.20 when OpenPrice is 1.8 to  4.9 & ClosePrice <   4.0
         0.35 when OpenPrice >=        10.3 & ClosePrice >=         10.0
         0.89 when OpenPrice is 1.8 to 10.3 & ClosePrice >=         10.0
         0.91 when OpenPrice is 1.8 to  4.9 & ClosePrice is  4.0 to 10.0
         0.99 when OpenPrice <  1.8          & ClosePrice >=          1.2
```

Además de que hay menos reglas, las que hay son levemente menos complejas para el árbol podado con un CP mayor. Si analizamos con detalle podemos notar que además de haber reglas que no aparecen en el modelo podado, el modelo podado generaliza más las reglas, esto se puede notar cuando todas las subastas con OpenPrice >= 10.3 y ClosePrice >= 10.0 se agruparon bajo la regla 0.35, en lugar de dividirse en dos grupos (antes 0.26 y 0.64). Si bien esto puede evitar el sobreajuste puede ser perjudicial si lo que queremos es diferenciar más casos. Lo bueno es que eliminó las condiciones sobre SellerRating (la variable que consideré cambiar por Category). De hecho, diría que si tuviera que eliminar un predictor elegiría esa variable. No es tan importante como los precios de apertura y cierre (y es mucha la diferencia).

B) Si hablamos de practicidad considero que este modelo cumple su función. Nos confirma que las variables más importantes son los precios de apertura y cierre, ahorrándonos así varios ensayos para llegar a la misma conclusión pero de manera empírica. Obviamente podría mejorarse para que considere otros factores además de esas dos variables, pero adoptar un enfoque así podría significar adoptar otro modelo.

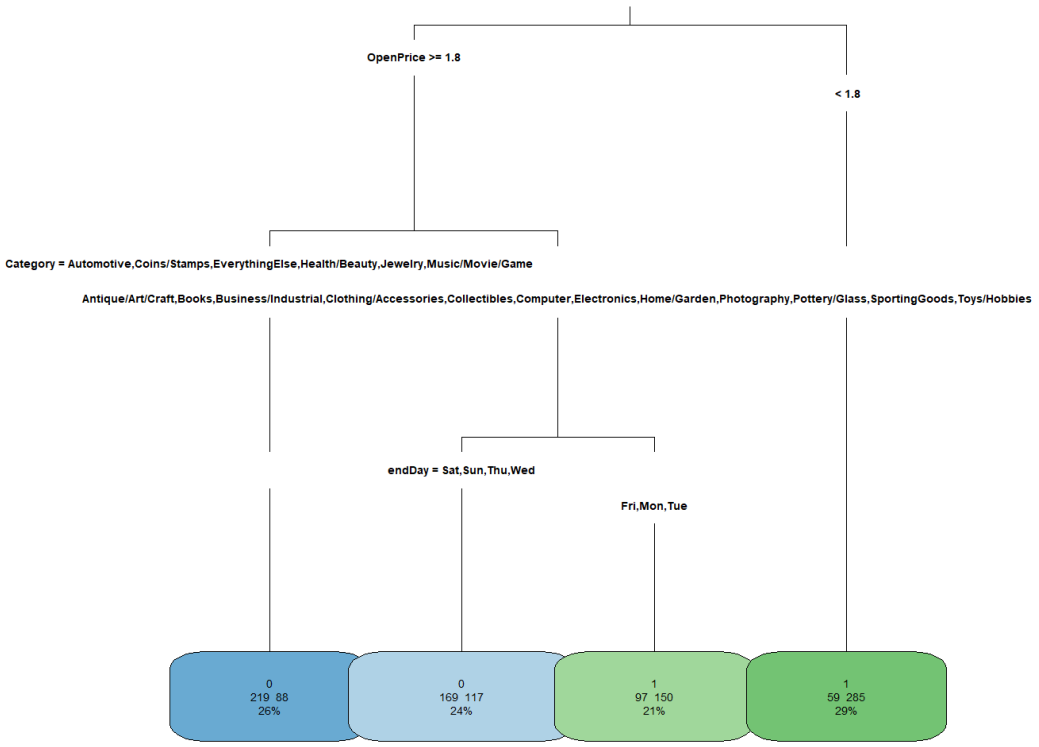
C) La información que considero más llamativa sobre las reglas es que si queremos predecir el éxito de una subasta en curso podemos basarnos en el precio de apertura y en el último precio propuesto (ya que sería el más cercano a ser el precio de cierre). Y si queremos predecir el resultado de una subasta que ya terminó podemos basarnos en el precio de cierre y este será un mejor predictor ya que un precio de apertura bajo no impide que el precio de cierre sea también bajo, lo que también se traduce como que **un precio de apertura alto no es garantía de competitividad**. También me pareció curioso que la observación que hice sobre la variable SellerRating fuera corroborada por la poda al eliminar dicho predictor.

Ahora bien, las cosas que me parecieron de poca utilidad fueron las reglas de competitividad media que consideran la variable SellerRating (se que parece que ya me ensañe con la variable pero tengo mis motivos), pues, realmente no aportan mucho más allá de lo que ya aportan las variables de precio de apertura y cierre.

D) Bueno, lo que hice para este nuevo árbol fue no usar las variables sellerRating, ClosePrice y Duration. Después, tuve que hacer variables tontas (dummy) para segmentar las categorías que estaban dentro de la variable Category, ya que si no hacía este paso no reconocía bien las diferencias entre categorías y además el árbol quedaba muy raro. Algo así se ve el nuevo dataset (por razones de espacio no puse en la captura todas las variables tontas).

```
> head(data_dummy)
  currency sellerRating Duration endDay ClosePrice OpenPrice Competitive. CategoryAntique/Art/Craft CategoryAutomotive
1       US         3249        5    Mon       0.01      0.01            0                       0                   0
```

Entonces, una vez hecho ese ajuste para la variable Category hice el árbol y después le apliqué la mejor poda posible.

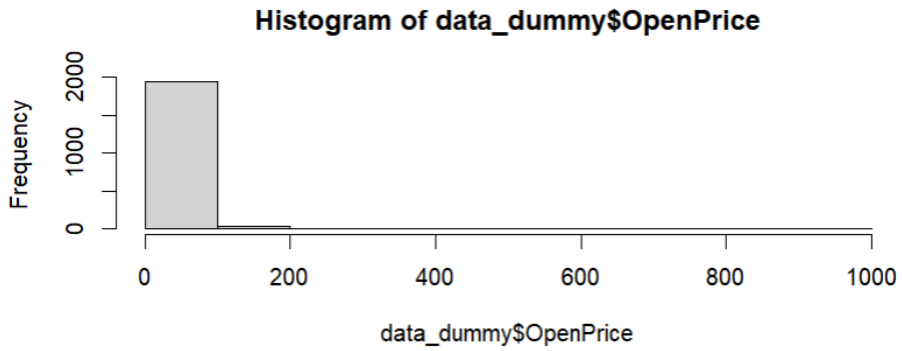


El plot no es tan bonito como me gustaría pero más adelante trataré ese detalle. Aquí podemos leer el árbol como “Si la subasta tiene un precio de apertura menor a 1.8 entonces casi siempre es competitiva”, más adelante hablaré de esto. Además, “Si el precio de apertura es mayor o igual a 1.8, entonces la categoría del producto y el día de cierre determinan la competitividad”. También, los productos en las categorías Automotive, Coins/Stamps, EveryhingElse, Health/Beauty, Jewerly, Music/Movie/Game tienen una menor competitividad independientemente del día, pero las subastas que terminan en Lunes, Martes o Viernes suelen ser más competitivas (esto corrobora una de las gráficas del inicio).

Veamos las reglas para corroborar esta lectura del árbol

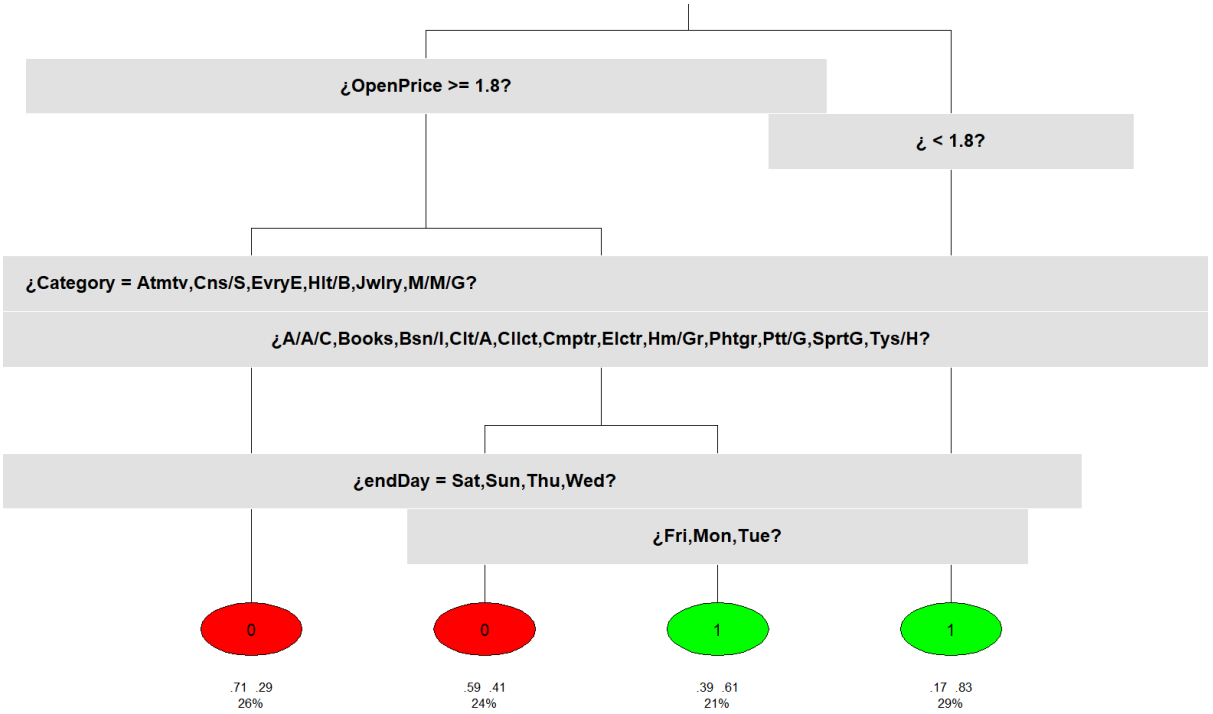
```
> rpart.rules(pruned_new_tree)
 Competitive.
        0.29 when OpenPrice >= 1.8 & Category is Automotive or Coins/Stamps or EverythingElse or Health/Beauty or Jewelry or Music/Movie/Game
        0.41 when OpenPrice >= 1.8 & Category is Antique/Art/Craft or Books or Business/Industrial or Clothing/Accessories or Collectibles or Computer or
        0.61 when OpenPrice >= 1.8 & Category is Antique/Art/Craft or Books or Business/Industrial or Clothing/Accessories or Collectibles or Computer or
        0.83 when OpenPrice <  1.8



 or Home/Garden or Photography or Pottery/Glass or SportingGoods or Toys/Hobbies & endDay is Sat or Sun or Thu or Wed
 or Home/Garden or Photography or Pottery/Glass or SportingGoods or Toys/Hobbies & endDay is  Fri or Mon or Tue
```

Ahora hablemos del elefante en la habitación. Si hay una relación directamente proporcional entre la competitividad y el precio de inicio ¿Por qué un precio menor a 1.8 tiene mejor competitividad según el árbol? Bueno, esto se puede deber a que hay un mayor porcentaje de observaciones en este nodo y por lo tanto sea más probable que hayan subastas competitivas, pero personalmente considero más probable que se deba a otros factores. Primero, que como se mencionó, el precio de cierre es más importante y que por lo tanto un precio bajo de apertura no implica una mala competitividad (correlación no implica causalidad). Y la segunda opción es que nuestra profundidad (maxdepth=7) esté impidiendo que el árbol descubra las condiciones bajo las cuales una subasta con un precio de apertura menor a 1.8 sea competitiva. Incluso podría ser que hay pocas subastas con un precio de apertura menor a 1.8 y que los datos atípicos están alterando la detección de patrones. O también podría ser una señal de que la distribución de datos para la variable OpenPrice esté sesgada.



Bueno, al parecer si está sesgada la distribución, por lo que es algo a considerar.

Antes de pasar al siguiente inciso quiero dejar una versión mejorada del plot del segundo árbol (el árbol de este inciso).



Bueno, regresé después de escribir el inciso E). Creo que más bien la mayoría de las subastas con un precio de apertura bajo tienden a ser más competitivas porque en términos generales el comportamiento de una subasta es empezar de ofertas más "modestas" a ofertas cada vez mayores. Es decir, lo más común es empezar todas las subastas con un precio relativamente bajo, por lo que en efecto, la distribución sesgada a la izquierda de OpenPrice se ve reflejada en el árbol.

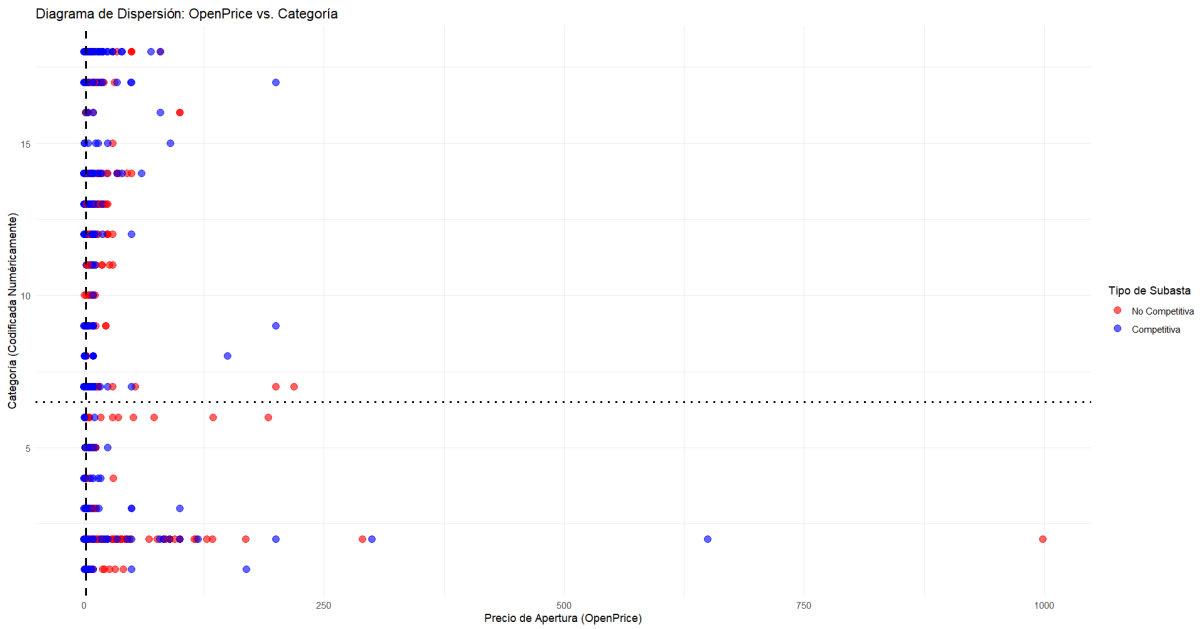
E) Según las instrucciones hay que usar variables cuantitativas para este ploteo, así que decidí convertir en una variable numérica la variable Category, ya que según la función summary(), es una variable muy importante, y considero que es fundamental considerar a las variables sean más relevantes para este análisis.

```
> summary(pruned_new_tree)
Call:
rpart(formula = Competitive. ~ . - sellerRating - Clos
    Duration, data = train_dummy, method = "class", co
    maxdepth = 7, cp = 0.03676471))
  n= 1184

          CP nsplit rel error    xerror       xstd
1 0.23897059      0 1.0000000 1.0000000 0.03152207
2 0.04871324      1 0.7610294 0.7720588 0.03026193
3 0.03676471      3 0.6636029 0.7408088 0.02997114

Variable importance
OpenPrice  Category    endDay  currency
       69        21         9         1
```

Para hacer esto hice un dataframe temporal y usé la función as.numeric() y después inserté una columna nueva en el nuevo DF que contiene el valor numérico de la categoría.

```
> head(new_df)
           Category currency sellerRating Duration endDay ClosePrice OpenPrice Competitive. CategoryNum
1  Music/Movie/Game       US         3249        5    Mon       0.01      0.01            0          14
2  Music/Movie/Game       US         3249        5    Mon       0.01      0.01            0          14
4  Music/Movie/Game       US         3249        5    Mon       0.01      0.01            0          14
10 Music/Movie/Game       US         3249        5    Mon       0.01      0.01            0          14
11 Music/Movie/Game       US         3249        5    Mon       0.01      0.01            0          14
12 Music/Movie/Game       US         3249        5    Mon       0.01      0.01            0          14
```

Entonces, así quedaría el plot solicitado.

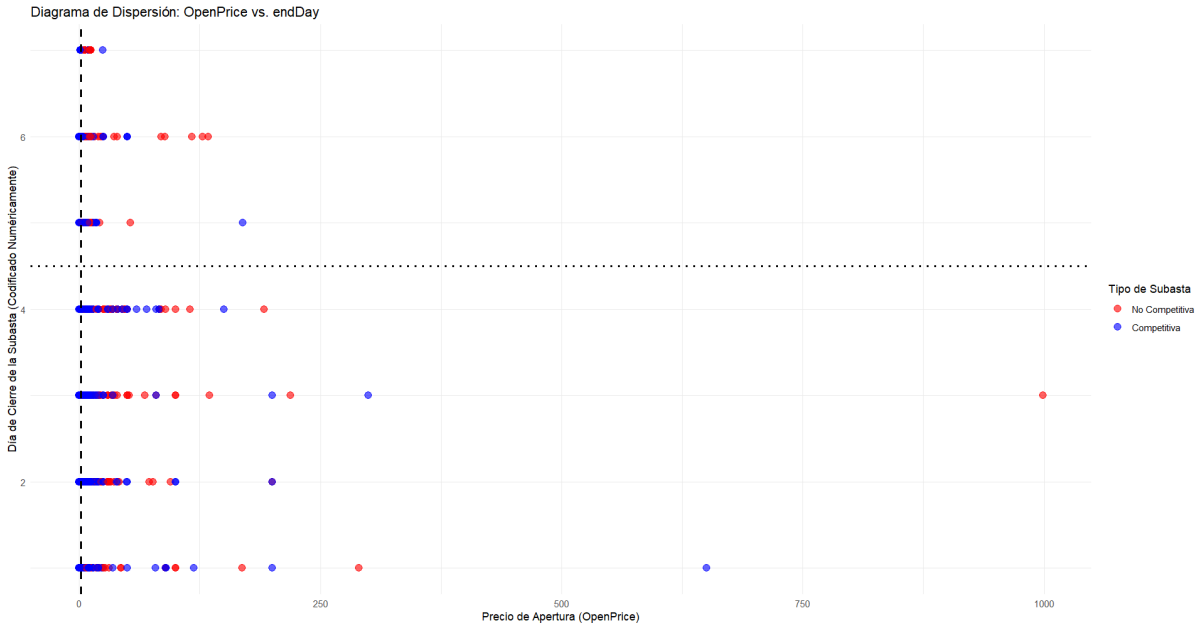


A primera vista, me parece que las líneas divisorias (punteadas) representan bien el valor de ambas variables. Podemos observar que se representa fielmente que la mayoría de subastas competitivas (puntos azules) tienden a ser bajas. Además, hay categorías que tienen más puntos rojos que azules, lo que ilustra la competitividad inferior de ciertas categorías.

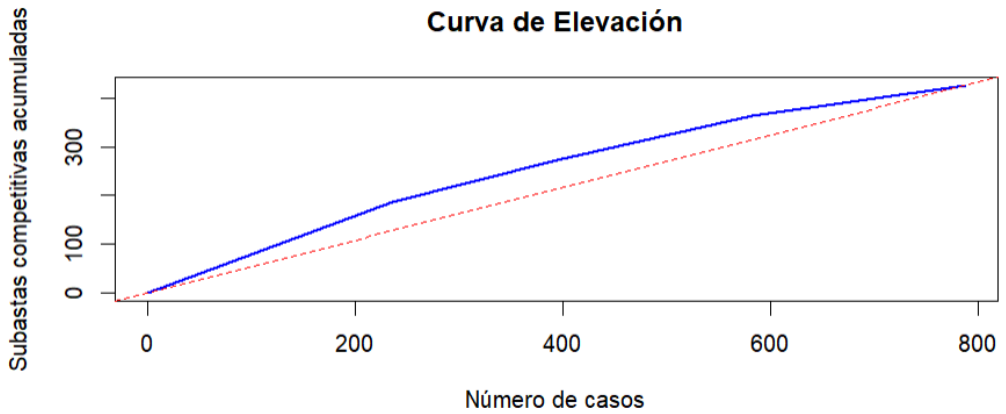
Sobre el trabajo de separar las clases, pareciera que la variable OpenPrice distingue adecuadamente las subastas competitivas de las que no, sin embargo, no puedo decir lo mismo sobre las categorías. La línea del eje x no ayuda a distinguir bien entre las subastas competitivas de las no competitivas, y se puede notar en que las subastas rojas y azules están muy mezcladas en las mismas regiones. Esto me hace pensar que la categoría, pese a ser considerada una variable importante no es un predictor tan fuerte como OpenPrice. Veamos qué pasa si usamos la variable endDay en lugar de las categorías.

Recordemos que el modelo clasificó los días Fri, Mon, Tue como más competitivos (numéricamente tienen el valor 1,2 y 6, respectivamente. Debido a que los días están ordenados la línea divisoria no hace mucho, quizá podría hacer mejor su trabajo si agrupamos los días competitivos arriba y los no competitivos abajo, pero aún así,

debido a que la mayoría de las subastas están a la izquierda es difícil determinar si la representación de las subastas competitivas está bien hecha ya que perfectamente podrían haber muchas subastas superpuestas la una sobre la otra pero nosotros sólo veríamos un único punto, en cambio, para el día 4 (Sun) pareciera que hay muchas subastas competitivas, pero perfectamente podría ser que las pocas que tiene sólo están mejor distribuidas, ya que si recordamos, ese día fue clasificado como no competitivo.



F) La precisión del modelo predictivo a primera vista no es tan buena, pero si consideramos la curva de elevación y que tenemos 788 observaciones en el dataframe de entrenamiento, creo que tampoco está tan mal el resultado obtenido. Aunque ello no significa que no haya mejoras que hacer.



La línea roja vendría representando un modelo aleatorio, lo que significa que nuestro modelo (línea azul) es mejor que el azar, pero aún así, pareciera que el modelo aún no separa bien las clases. Lo digo porque la distancia entre la línea roja y la azul no es mucha, porque la precisión podría mejorar (ya que hay varios falsos positivos y falsos negativos) y también por el diagrama de dispersión del inciso anterior.

```
> confusionMatrix(pred, test_dummy$Competitive.) #Matríz de confusión
Confusion Matrix and Statistics

          Reference
Prediction   0   1
         0 240 153
         1 122 273

               Accuracy : 0.651
                 95% CI : (0.6166, 0.6843)
    No Information Rate : 0.5406
    P-Value [Acc > NIR] : 2.11e-10

                  Kappa : 0.3019

 Mcnemar's Test P-Value : 0.07044

            Sensitivity : 0.6630
            Specificity : 0.6408
         Pos Pred Value : 0.6107
         Neg Pred Value : 0.6911
             Prevalence : 0.4594
         Detection Rate : 0.3046
   Detection Prevalence : 0.4987
      Balanced Accuracy : 0.6519

       'Positive' Class : 0
```

G) Con base en los resultados, el precio de apertura es un factor clave para que una subasta tenga al menos dos ofertas, esto tiene sentido porque un precio bajo incentiva a que uno quiera extender la mano con mayor facilidad. También el día de la semana demostró ser importante así como la categoría, pero aún así el precio de apertura les gana en importancia, por lo que si uno tiene que sacar adelante una subasta de una mala categoría podría compensarlo poniendo un precio bajo de apertura y cerrando en un día estratégico, o si se tiene un buen precio de salida uno podría arriesgarse un poco más con respecto a alguna de las otras dos variables si no es que ambas (en la dosis está el veneno). La moneda no tuvo mención alguna a lo largo del reporte debido a que no parece ser un predictor fuerte, en cambio, el precio de cierre si.

Por lo tanto, mi recomendación para un vendedor sería que siempre empiece con un precio bajo (menor a 1.8) y que procure cerrar la subasta en un día de alta actividad (como Lunes, Martes o Viernes). Si su producto es de una de las siguientes categorías: Jewelry, Health/Beauty, Music/Movie/Game. Entonces deberá tomarse las recomendaciones anteriores con seriedad para aumentar sus posibilidades. En cambio, si se tiene un buen precio de apertura y una buena categoría podría arriesgarse a programar el día de cierre para un día inactivo.