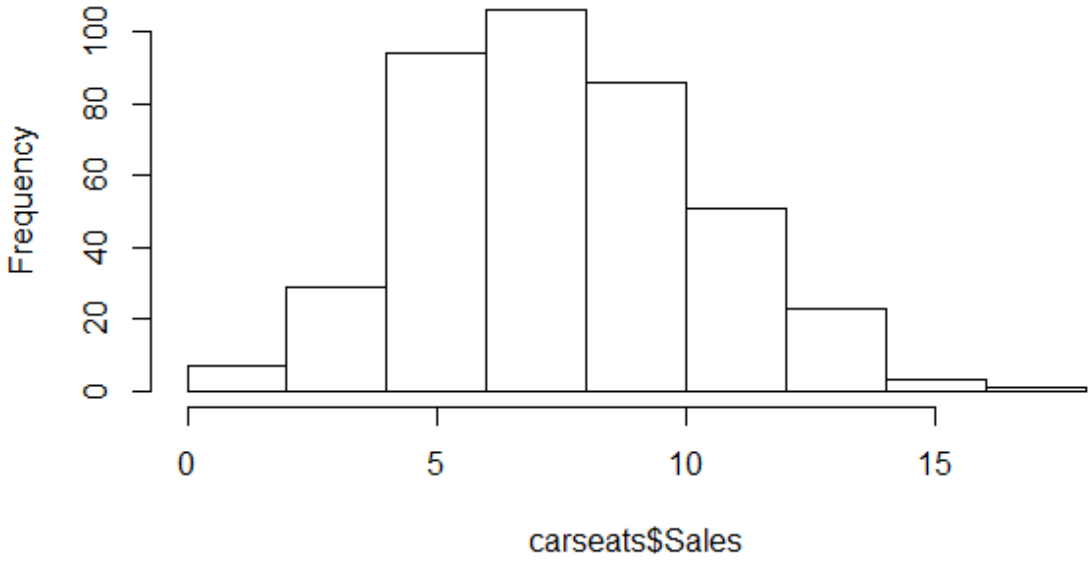
Técnicas de Aprendizado de Máquina Aplicadas a Negócios

Algoritmos de Classificação

Árvores de Classificação

Histograma de vendas de assentos para carros

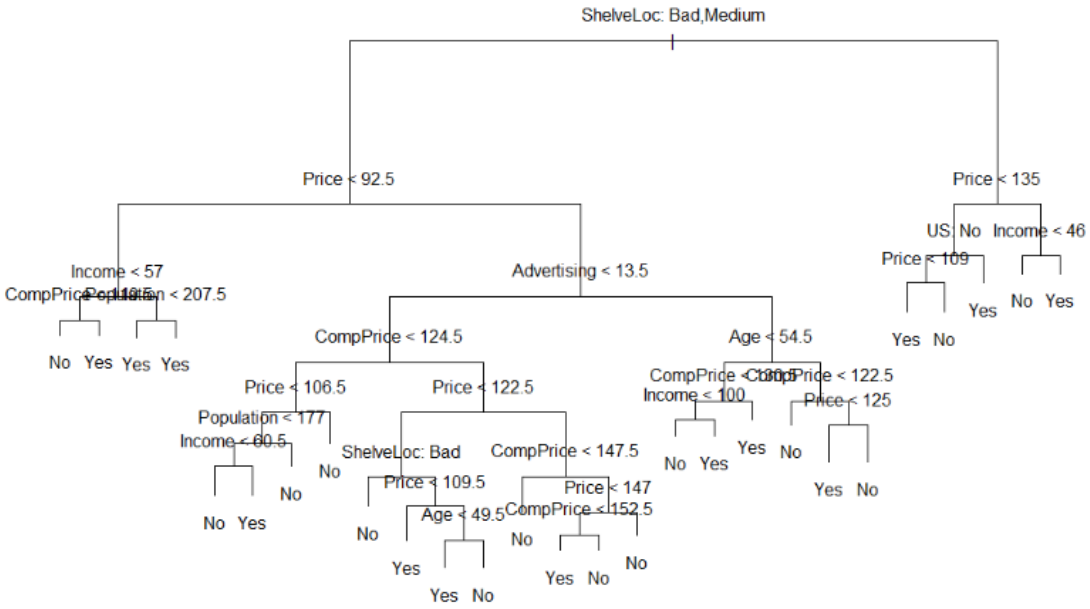


Primeiramente, a partir do histograma acima, nota-se que a dispersão dos dados varia de 0 a 17, aproximadamente, de modo que a maioria das lojas venderam entre 4 e 12,5 mil cadeiras e há um pico de lojas que venderam em torno de 7 mil cadeiras no período analisado. Além disso, há uma leve assimetria positiva dos dados, havendo uma maior concentração de dados na zona de valores menores da base.

Resumo da Árvore de Classificação

```
Classification tree:
tree(formula = High ~ . - Sales, data = carseats)
Variables actually used in tree construction:
[1] "ShelveLoc"   "Price"       "Income"      "CompPrice"   "Population"
[6] "Advertising" "Age"         "US"
Number of terminal nodes:  27
Residual mean deviance:  0.4575 = 170.7 / 373
Misclassification error rate: 0.09 = 36 / 400
```

Visualização da Árvore de Classificação

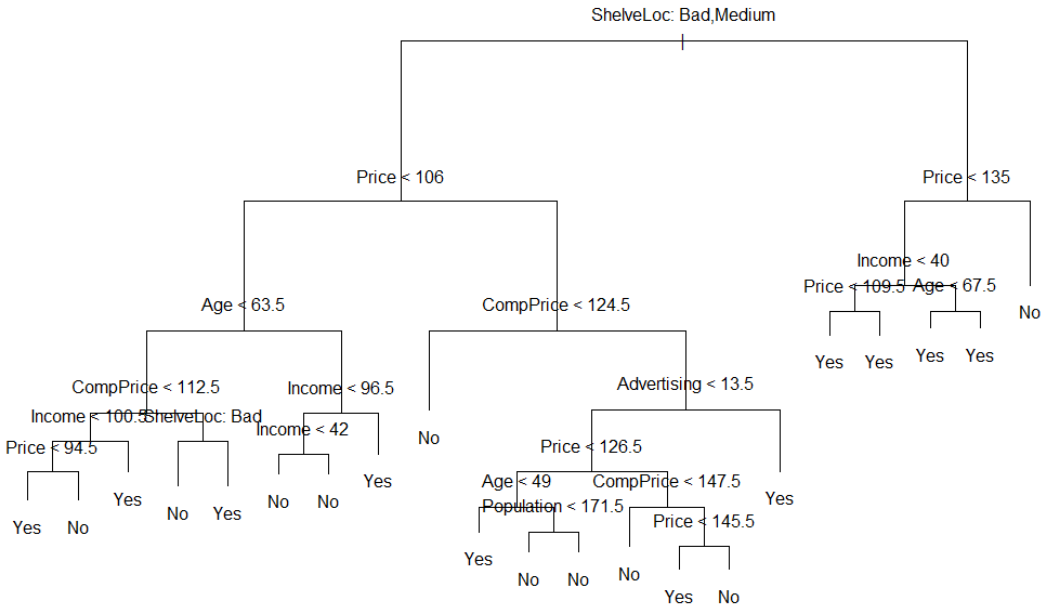


A partir do resumo e da visualização da árvore de classificação aplicada à base de dados Carseats, nota-se que, ao tentar classificar as vendas como altas (iguais ou maiores que 8 mil) ou baixas (abaixo de 8 mil), o algoritmo não utiliza as variáveis Education e Urban, bem como gera 27 nós terminais, indicando uma árvore relativamente complexa. Apresenta também baixos resultados de desvio médio residual (0,4575) e taxa de erro de classificação incorreta (0,09), indicando que o modelo apresenta uma precisão satisfatória. Além disso, é válido pontuar também que as primeiras variáveis analisadas pelo algoritmo são ShelveLoc e Price, como pode ser visto na visualização da árvore de classificação.

Resumo da árvore de classificação após a primeira poda

```
Classification tree:
tree(formula = High ~ . - Sales, data = carseats, subset = train)
Variables actually used in tree construction:
[1] "ShelveLoc"   "Price"       "Age"         "CompPrice"   "Income"      "Advertising" "Population"
Number of terminal nodes:  21
Residual mean deviance:  0.4442 = 101.7 / 229
Misclassification error rate: 0.112 = 28 / 250
```

Visualização da Árvore de Classificação após a primeira poda



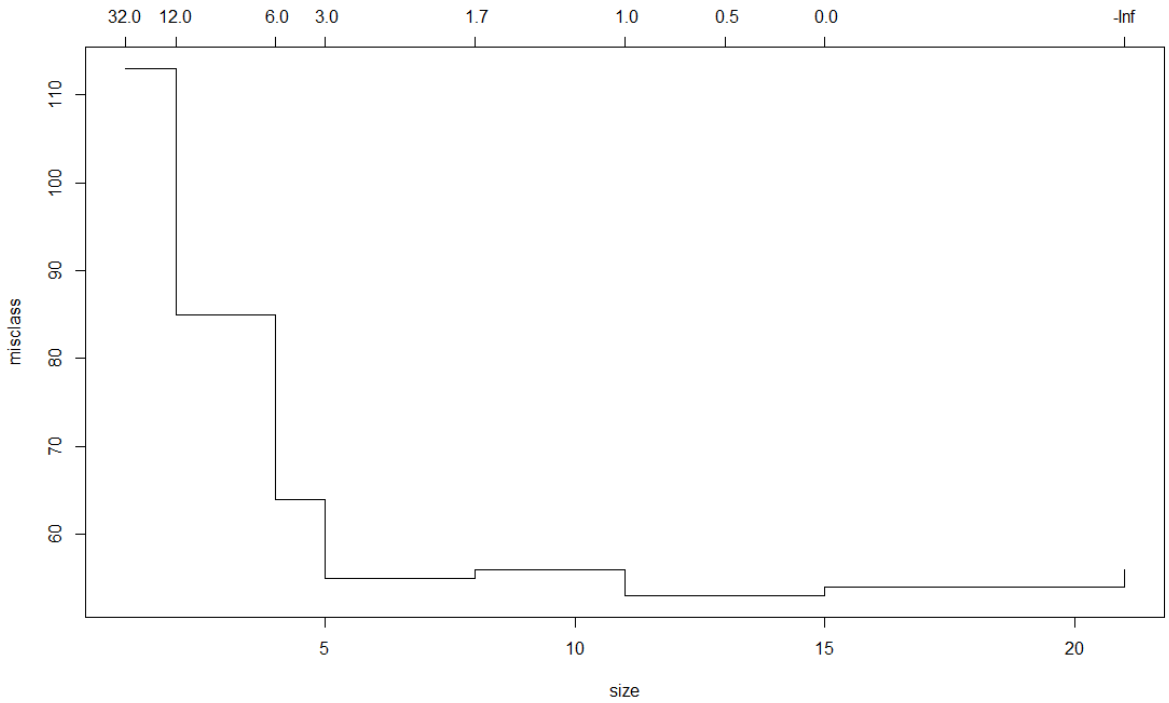
Após podar a árvore pela primeira vez, dividindo a base de dados em 250 observações para treinamento e 150 observações para amostras de teste, nota-se que houve uma redução no número de nós terminais, sendo 6 nós a menos que anteriormente, como também o número de variáveis utilizadas diminuiu, já que a variável US também não foi usada no modelo dessa vez. Além disso, o desvio médio residual diminuiu e a taxa de erro de classificação incorreta aumentou, mas foram variações pouco significativas para influenciar a precisão e a capacidade de generalização do modelo.

Tabela de confusão e acurácia

```
          High
tree.pred No Yes
      No  74  18
      Yes 19  39
> (74 + 39)/150
[1] 0.7533333
```

Analisando a tabela de confusão, nota-se que o algoritmo errou 18 observações que foram classificadas como altas, mas que eram baixas, e 19 observações que foram classificadas como baixas, mas eram altas. Sendo assim, a acurácia do modelo foi igual a 75,33%, aproximadamente.

Gráfico de validação cruzada

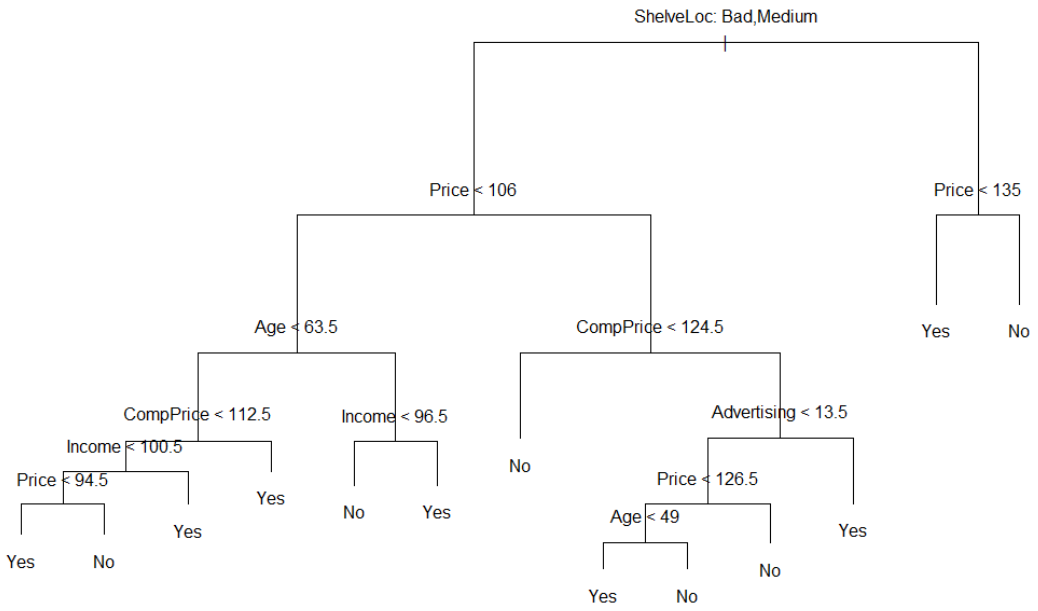


A partir do gráfico acima, é possível observar que não há variações significativas no erro de classificação para árvores de tamanho acima de 12, nesse caso, de modo que, a partir disso, a métrica torna-se instável. Assim, vamos podar a árvore novamente, agora para um tamanho igual a 12.

Resumo da árvore de classificação após a segunda poda

```
Classification tree:
snip.tree(tree = tree.carseats, nodes = c(6L, 17L, 18L, 89L,
45L))
Variables actually used in tree construction:
[1] "ShelveLoc"   "Price"       "Age"         "CompPrice"   "Income"      "Advertising"
Number of terminal nodes:  13
Residual mean deviance:  0.6447 = 152.8 / 237
Misclassification error rate: 0.116 = 29 / 250
```

Visualização da Árvore de Classificação após a segunda poda



Após podar a árvore novamente, é visível a redução na complexidade da árvore, a qual agora apresenta 13 nós terminais. A taxa de erro de classificação incorreta manteve-se praticamente constante em relação ao modelo anterior, já o desvio médio residual teve um aumento de 45.11%, indicando possivelmente uma menor qualidade de ajuste do modelo.

Tabela de confusão e acurácia

```
          High
tree.pred No Yes
      No  74  18
      Yes 19  39
> (74 + 39)/150
[1] 0.7533333
```

Ainda assim, analisando a tabela de confusão e a medida de acurácia do modelo, nota-se que a última redução no tamanho da árvore não comprometeu significativamente a precisão do modelo de classificação e reduziu o nível de complexidade da árvore.

Árvores de Regressão - Random Forest

Resumo da primeira floresta aleatória

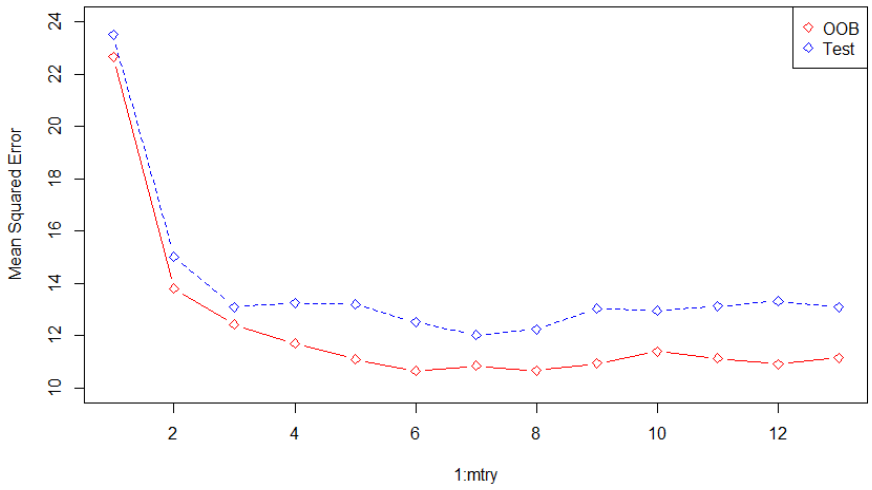
```
> rf.boston

Call:
 randomForest(formula = medv ~ ., data = boston, subset = train)
               Type of random forest: regression
                     Number of trees: 500
No. of variables tried at each split: 4

          Mean of squared residuals: 12.68651
                    % Var explained: 83.45
```

A partir do resumo acima, é possível verificar que, inicialmente, o número de árvores utilizadas no modelo é igual a 500, o número de variáveis que foram selecionadas em cada divisão de cada árvore foi igual a 4 de 13 variáveis, o erro médio quadrado foi aproximadamente igual a 12,69 e a porcentagem de variância explicada foi de 83,45%, relativamente alta.

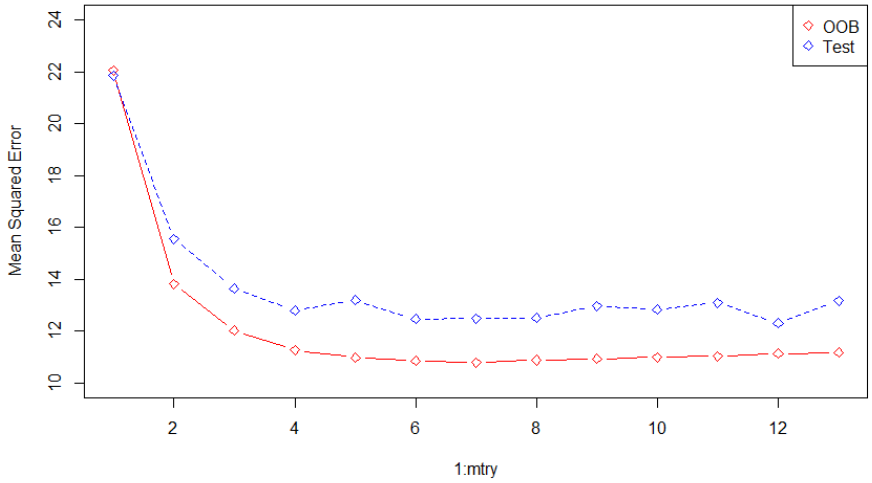
Erros OOB e Test com diferentes números de variáveis. 350 árvores



Depois de realizados uma série de ajustes na primeira floresta aleatória, incluindo restringir o seu tamanho para 35 árvores, foi plotado o gráfico acima, que mostra o erro médio quadrado no eixo vertical, e, no vertical, a quantidade de variáveis sendo levadas em conta. Nota-se, então, que o erro calculado a partir do conjunto de teste é um pouco maior que o erro calculado a partir da validação cruzada, como também é possível observar que a quantidade de variáveis ótima seria em torno de 4, tendo em vista que as métricas tornam-se instáveis a partir disso.

Abaixo, apresentamos os resultados obtidos considerando 35.000 árvores (15 min de processamento) - os demais parâmetros foram mantidos.

Erros OOB e Test com diferentes números de variáveis. 35.000 árvores

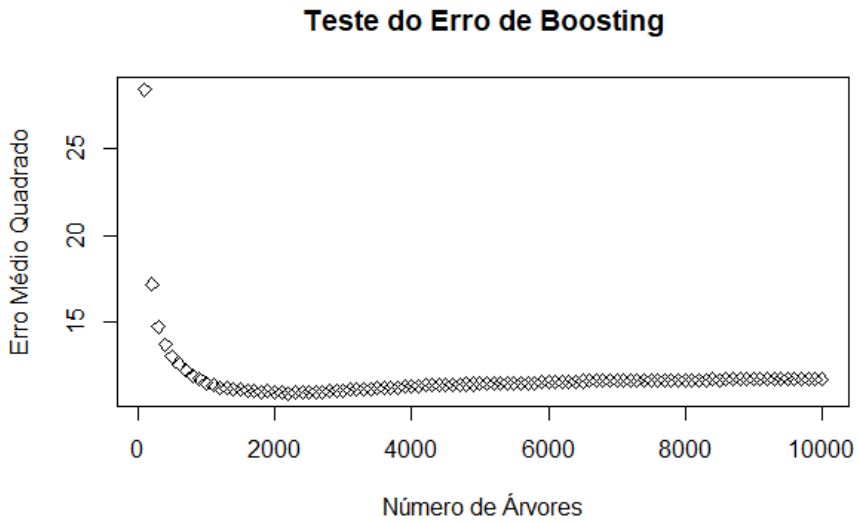


Comparando os resultados obtidos usando 350 e 35.000 (gráfico acima), pode-se notar, analisando o OOB, que:

- De maneira geral, houve uma redução do erro (como ao considerar 1, 3, 4 e 7 variáveis em cada divisão de cada árvore);
- No entanto, em alguns casos não houve alteração e, em outros, o erro aumentou (exemplos: 8 e 12 variáveis);
- A quantidade de variáveis ótima ainda seria em torno de 4, não havendo ganhos significativos quanto a precisão e capacidade de generalização do modelo ao aumentar a sua complexidade.

Boosting

Erro médio quadrado conforme quantidades de árvores



Ao aplicar a técnica de boosting ainda no mesmo conjunto de dados utilizados no modelo de florestas aleatórias e considerando os parâmetros determinados no direcionamento do laboratório, foi calculado o erro médio quadrado do modelo obtido ao utilizar diferentes quantidades de árvores, como pode ser visto na imagem acima. Pode-se observar que um modelo de boosting considerando entre 1.000 e 1.500 árvores parece ser a quantidade que fornece os melhores resultados, já que depois disso não há reduções significativas no erro médio quadrado e o modelo ainda não ficaria super ajustado.

Conclusão

Em discussão, o grupo concluiu que o principal aprendizado advindo do presente laboratório foi que, tanto no caso do algoritmo da Árvore de Decisão como no caso do Random Forest e do Boosting, é preciso haver um equilíbrio entre a precisão e a complexidade do modelo. Não faz sentido optar por um modelo altamente complexo e com alta precisão, pois ele pode não generalizar bem a previsão para novos dados em razão do overfitting, como também não faz sentido um modelo muito simples e com precisão insatisfatória de acordo com o problema que está sendo resolvido a partir dele, gerando um underfitting.

De modo geral, o ideal é chegar em um ponto em que, de acordo com métricas e técnicas de avaliação, como o erro quadrático médio e a validação cruzada, o modelo não é tão complexo (levando em conta que, quanto maior o tamanho da árvore, no caso do algoritmo da Árvore de Decisão, ou quanto maior o número de árvores utilizadas no modelo, no caso no Boosting e do Random Forest, maior a complexidade do modelo) e não há ganhos significativos em sua precisão.

Por fim, é importante pontuar também que, a partir da leitura e execução do laboratório, foi possível concluir que, em geral, algoritmos mais robustos, como o Random Forest e o Boosting, geralmente resultam em previsões mais precisas, conseguindo explicar a variância dos dados de forma melhor do que modelos mais simples, como o de Árvore de Decisão. Nesse sentido, o grupo sugere como ponto de melhoria que os diferentes algoritmos e técnicas sejam aplicados em um mesmo conjunto de dados, de modo a tornar os resultados da avaliação dos modelos mais comparáveis e constatar na prática esse aspecto mencionado anteriormente.