QA1. What is the key idea behind bagging? Can bagging deal both with high variance (overfitting) and high bias (underfitting)?

The key idea behind bagging is to create a more accurate and stable model by selecting base models randomly and with replacement from the sample. Bagging can deal with high variance (overfitting), but it cannot deal with bias (underfitting).

QA2. Why bagging models are computationally more efficient when compared to boosting

models with the same number of weak learners?

Bagging models are computationally more efficient, because boosting models work sequentially meaning that each layer builds upon the last. In a bagging model, each model is trained independently and in parallel. This means that bagging models don’t take as long to create and can be easier to troubleshoot should there be any issues.

QA3. James is thinking of creating an ensemble mode to predict whether a given stock will go up or down in the next week. He has trained several decision tree models, but each model is not performing any better than a random model. The models are also very similar to each other. Do you think creating an ensemble model by combining these tree models can boost the

performance? Discuss your answer.

I think that creating an ensemble model would not help to boost performance. Since all of James’ models are similar, combining them would only create a similar model. There is not enough diversity in his models for performance to be boosted. This suggests that James’ model may be using an m that is too large and too close to p. James’ should tune his parameters and run more decision tree models before trying to create an ensemble model.

QA4. Consider the following Table that classifies some objects into two classes of edible (+) and

non- edible (-), based on some characteristics such as the object color, size and shape. What

would be the Information gain for splitting the dataset based on the “Size” attribute?  
Parent

Yellow Small Round ++ (P)  
Yellow Small Round – (N)  
Green Small Irregular ++ (P2)  
Green Large Irregular – (N2)  
Yellow Large Round ++ (P3)  
Yellow Small Round ++ (P4)  
Yellow Small Round ++ (P5)  
Yellow Small Round ++ (P6)  
Green Small Round – - (N3)  
Yellow Large Round – (N4)  
Yellow Large Round ++ (P7)  
Yellow Large Round – - (N5)  
Yellow Large Round – - (N6)  
Yellow Large Round – - (N7)  
Yellow Small Irregular ++ (P8)  
Yellow Large Irregular ++ (P9)

Child

Yellow Small Round ++ (P1S)  
Yellow Small Round – (N1S)  
Green Small Irregular ++ (P2S)  
Green Large Irregular – (N1L)  
Yellow Large Round ++ (P1L)  
Yellow Small Round ++ (P3S)  
Yellow Small Round ++ (P4S)  
Yellow Small Round ++ (P5S)  
Green Small Round – - (N2S)  
Yellow Large Round – (N2L)  
Yellow Large Round ++ (P2L)  
Yellow Large Round – - (N3L)  
Yellow Large Round – - (N4L)  
Yellow Large Round – - (N5L)  
Yellow Small Irregular ++ (P6S)  
Yellow Large Irregular ++ (P3L)

P = positive

N = negative

Parent entropy: -(9/16\*log2(9/16)) – (7/16 \* log2(7/16)) = 0.9887

Child entropy:

Small: -(6/8 \* log2(6/8)) – (2/8 \* log2(2/8)) = 0.8113

Large: -(3/8 \* log2(3/8)) – (5/8 \*log2(5/8)) = 0.9544

Average: (8/16 \* 0.8113) + (8/16 \*0.9544) =.88285

0.9887-.88285 = 0.10585 = Information gain

QA5. Why is it important that the m parameter (number of attributes available at each split) to

be optimally set in random forest models? Discuss the implications of setting this parameter too

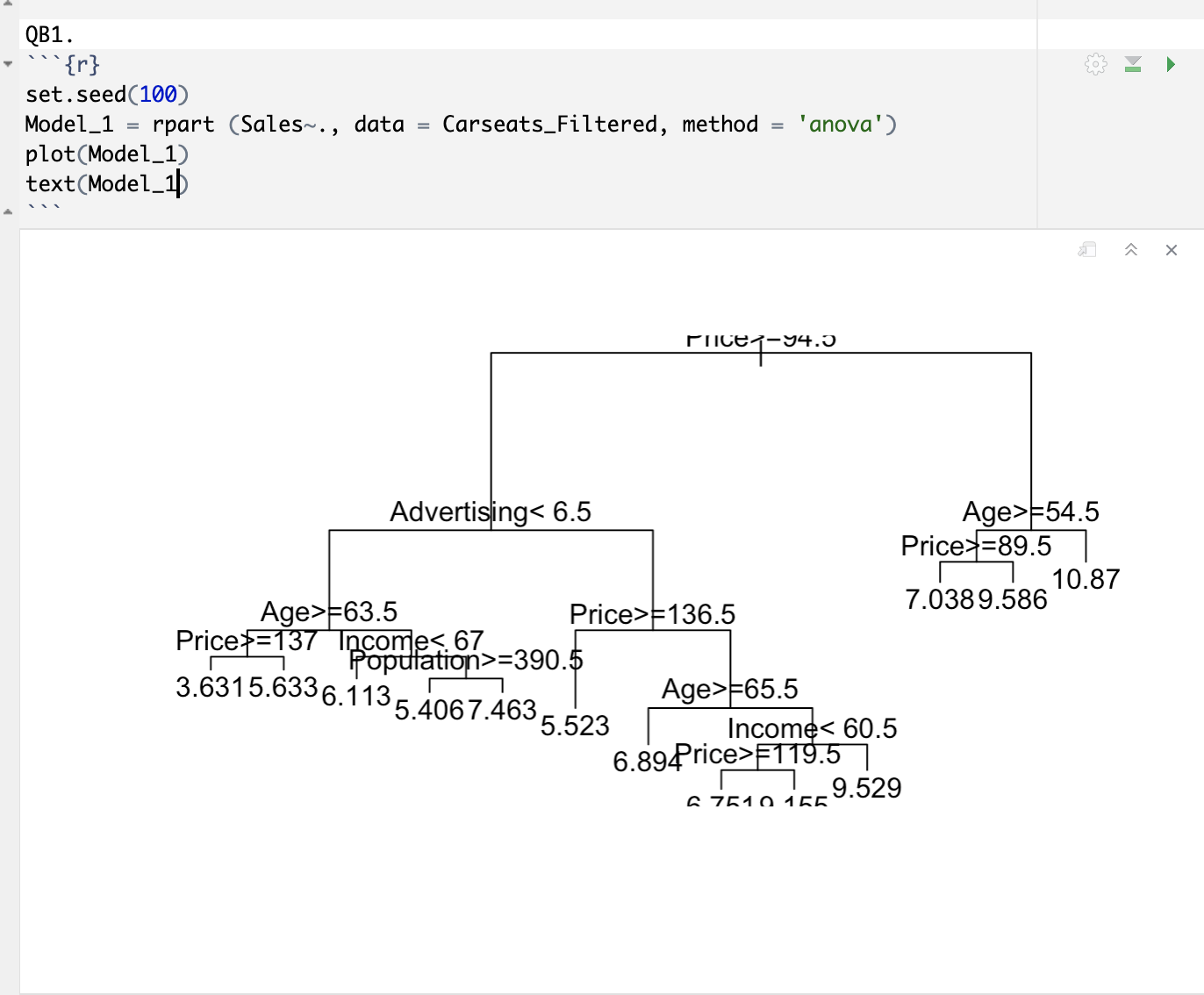
small or too large.

It is important that the m parameter be set optimally, because if m is too large then each model is not diverse enough. If each model is too similar, we lack diversity, and it wouldn’t make sense to use an ensemble model such as a random forest. If m=p, then that is just bagging and not a random forest model. If m is too low, then we run the risk of each individual tree not being representative of its data. If we create an ensemble model of trees that do not predict the data well, our ‘boosted’ model will also perform poorly.

LINK TO GITHUB WHERE R CODE IS STORED: https://github.com/hcroninkent/hcronin/tree/main/MIS64037

QB1.

The price attribute is used at the root node for splitting.



QB2.

Sales = 9

• Price=6.54

• Population=124

• Advertising=0

• Age=76

• Income= 110

• Education=10

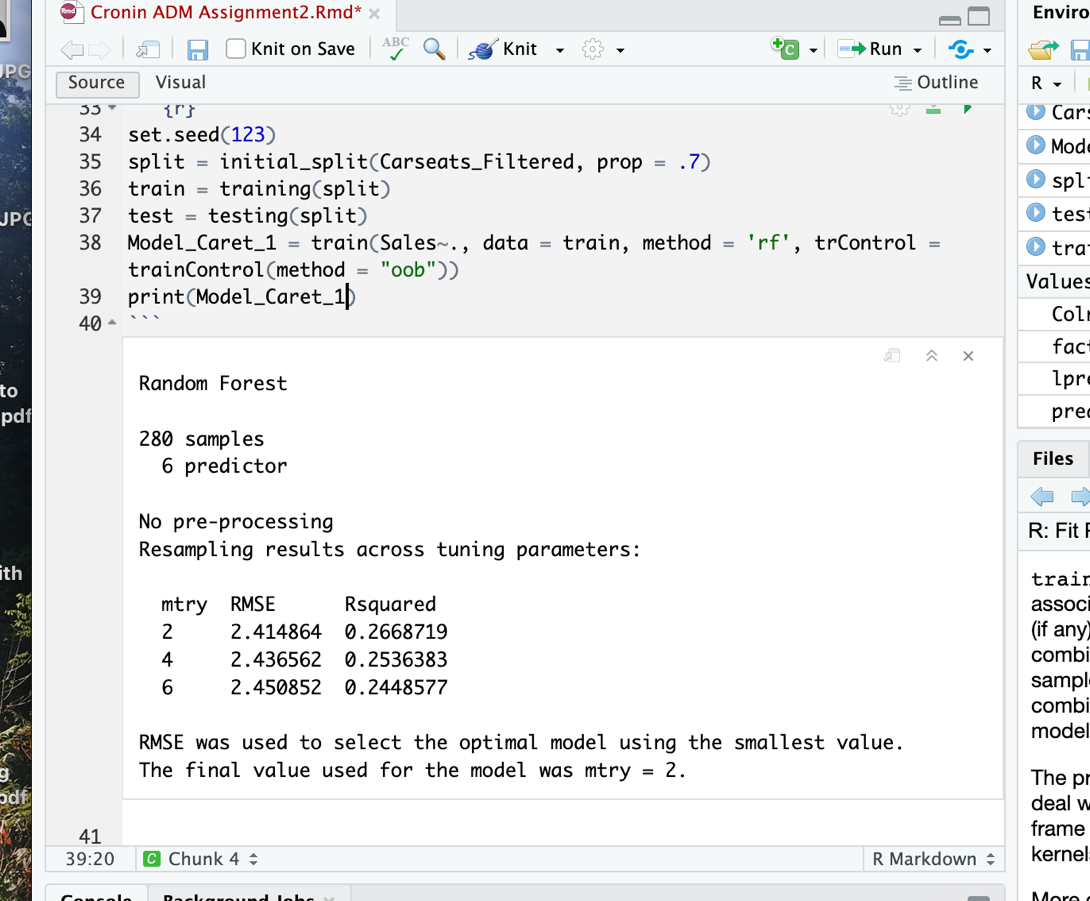
Under this model, the estimated sales are: 8.737, because price is less than 94.5 and age is greater than 54.5.

Diagram

Description automatically generated

QB3.

The optimal mtry is 2.



QB4.

The optimal mtry is 3.

Graphical user interface, text, application, email

Description automatically generated