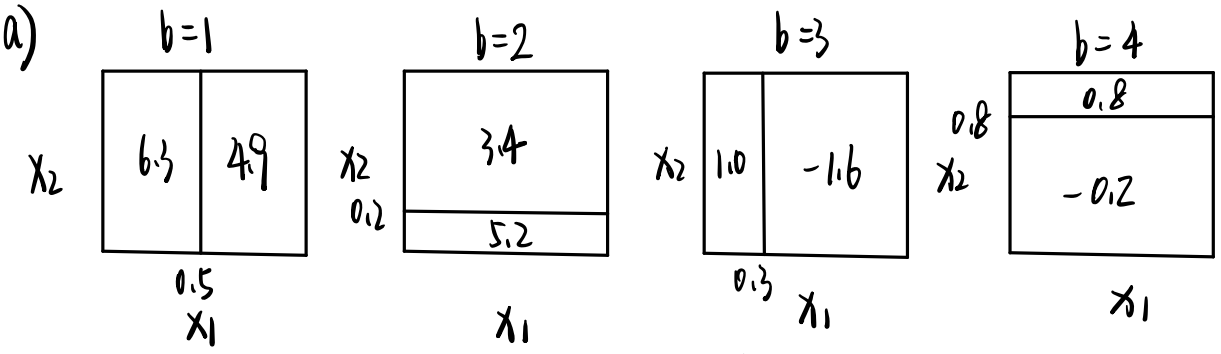
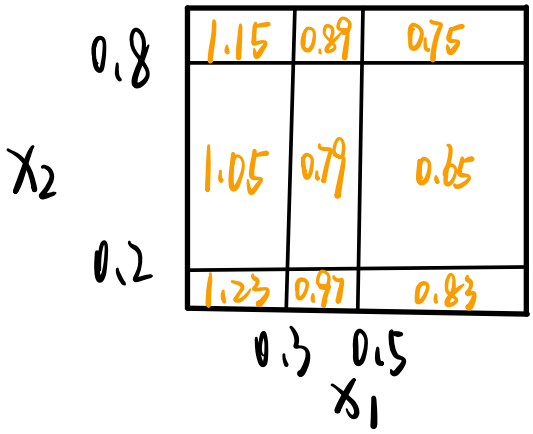
1. Let's investigate some interesting properties of the boosting predictor.

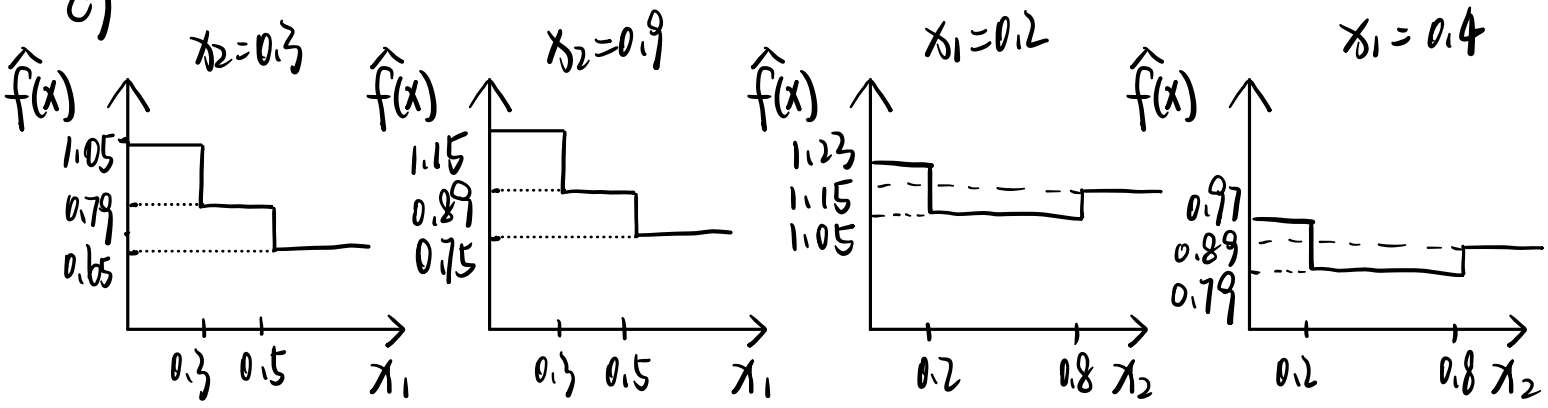
a)



b) The final model can be written as $\hat{f}(x) = \sum_{b=1}^{B} \lambda \hat{f}^b(x)$ where $\lambda = 0.1$



c)

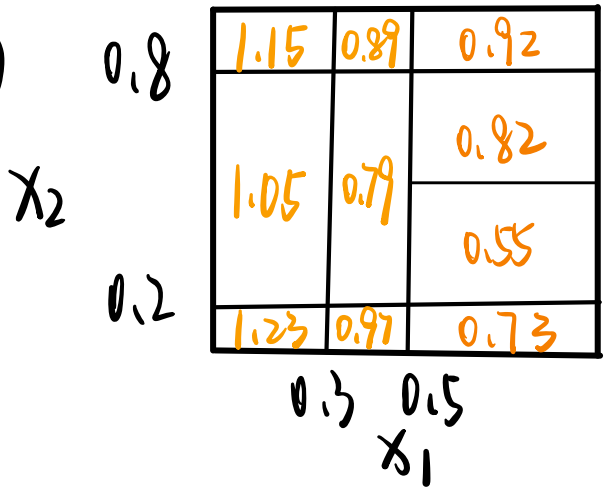


First of all, it has same slope for pairwise fuctions, and the value of the boosted predictor $\hat{f}$ changes when we change the values of x_1 and x_2, (but not always decrease or increase). Thus, when $x_1=0.2$ changes to $x_1=0.4$, the $\hat{f}$ derease and then increase for any value of x_2, but when $x_2=0.3$ changes to $x_2=0.9$, the $\hat{f}(x)$ decrease for any value of x_1.

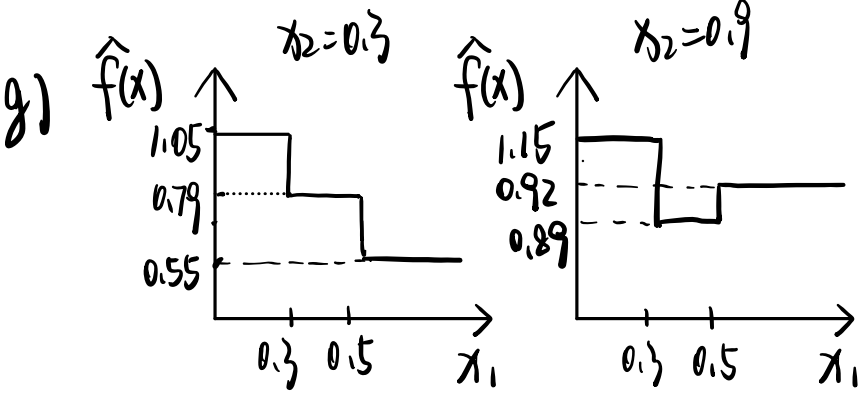
d) The boosted predictor f^ does not contain any interaction effects, which can be seen from the shape of plot that the slope of the predictor doesn't change for any changes in possible confounding x1 and x2 variables, indicating that the relationship between response variable and predictor will not change by another addition of variable. The additive property of the boosted predictor f allows for interpretability, as we interpret the effect of each predictor term individually. In addition, the depth d = 1 meaning that case each tree is a stump consisting of a single split and resulting in this additive model that is more interpretable.

e) Yes, as the learning rate λ is small for which it requires many trees B for better model fit and predict the date more precisely.

f)



g)



The predictor becomes less sensitive to the changes in x_2 values, as $x = 0.3$, it still contains a decreasing tendency but the lowest $\hat{f}$ value changes to 0.55 and others don't change. The $\hat{f}$ value decrease first and then increase for $x \leq 0.5$ and $0.5 < x \leq 1$ seperately, other 2 $\hat{f}$ values are the same. This time, the shape change for $\hat{f}$ as a function of x_1 for different values of x_2, the new boosted predictor does not enjoy the additive property in d), thus, it is even more to interpret with interaction depth of $d=2$ and the interaction

term is taken into consideration this time & the model is more complicated.

2. State whether each of the following statements are TRUE or FALSE. Briefly justify why in a couple of sentences.
   a) True. Alpha penalizes the number of terminal nodes to balance between the complexity of model and its fit on training data; when alpha = 0, it simply returns the pruning tree to the fully grown tree T0. And when alpha increase, the training error will increase with more terminal nodes similar as what lasso method did to control the complexity.
   b) False. Out-of-bag error estimation is meant to be useful for estimate the test error of a bagged model, without the need to perform cross-validation or the validation set approach. The bagged tree might only take two third of the whole data and left one third of training data as the OOB; however, an unpruned tree will take use of the full data set to fit model, and there are no available data points for OOB error estimation procedure.
   c) True. The misclassification error rate for classification tree is basically the fraction of the training observations in that region that do not belong to the most common class, which is an alternative for RSS and always decrease for more complex model on training data. Thus, it appears that Gini index or entropy can be a better alternative for misclassification error because they estimate the node purity while classification error might decrease without balancing the complexity of model.
   d) False. The bootstrapped tree repeatedly select data points from the same training data set and make its highly correlated. The bagged trees will also take on extremely powerful predictors on each top split resulting in similar trees.
   e) False. The smaller m will result in lower variance and high bias, and random forest will not run into overfitting for large number of predictors because it only consider for a small number of m predictors that “redecorates” the tree in terms of making it less correlated with each other on same top split.
   f) False. Because very small lambda can require using a very large value of B in order to achieve good performance.

3. Let’s dig deeper into the bootstrapping idea in bagging, where a “new” dataset is obtained by sampling the training data with replacement.
   a) 1/n
   b) 1- 1/n

c) The probability of each bootstrapped data point (x_i^*, y_i^*) that not equal to the first data point (x_1, y_1) is $(1 - \frac{1}{n})$.

There are n bootstraped data points with replacement that are chosen independly and so the joint probability for all of them that not equal to the first data point (x_1, y_1) is $(1 - \frac{1}{n})^n$

The probability of the (x_5, y_5) appearing is $1 - (1 - \frac{1}{5})^5$.

d) $P_j(n) = 1 - (1 - \frac{1}{n})^n$

when $n \to \infty$, $\lim_{n \to \infty} (1 - \frac{1}{n})^n = \frac{1}{e}$,

thus, $P_j(n)$ converges to $1 - \frac{1}{e} = \frac{2}{3}$.

e) when $n \to \infty$, only $2/3$ of data was used, $\frac{1}{3}$ of the data will be left.

f) The unused data are not wasted, because we can use the $1/3$ data to predict the response for the ith observations and get nearly B/n predictions for the ith observation ultimately, so we estimate the test error on a bagged model efficiently.