(d) [harder] Assume the y values are unique in $\mathbb{D}$. Imagine if $N_0 = 1$ so that each leaf gets one observation and its $\hat{y} = y_i$ (where i denotes the number of the observation that lands in the leaf) and thus it's very overfit and needs to be "regularized". Write up an algorithm that finds the optimal tree by pruning one node at a time iteratively. "Prune" means to identify an inner node whose daughter nodes are both leaves and deleting both daughter nodes and converting the inner node into a leaf whose $\hat{y}$ becomes the average of the responses in the observations that were in the deleted daughter nodes. This is an example of a "backwards stepwise procedure" i.e. the iterations transition from more complex to less complex models.

(e) [difficult] Provide an example of an $f(x)$ relationship with medium noise δ where vanilla OLS would beat regression trees in oos predictive accuracy. Hint: this is a trick question.

(f) [easy] Write down the step-by-step $\mathcal{A}$ for classification trees. This should be short because you can reference the steps you wrote for the regression trees in (a).

It's the same process for (a) just instead of damn regression trees, we use classification trees.

(g) [difficult] Think of another objective function that makes sense besides the Gini that can be used to compare the "quality" of splits within inner nodes of a classification tree.

Problem 5

These are some questions related to the CART algorithms.

(a) [easy] Write down the step-by-step $\mathcal{A}$ for regression trees.

1) Select the best split by dividing the data to break up the data points

2) Partition the data by using the indicator fc to determine whether it's greater or less than the determined split value

3) Repeat the process for each subgroup

4) Stop the recursion

(b) [difficult] Describe $\mathcal{H}$ for regression trees. This is very difficult but doable. If you can't get it in mathematical form, describe it as best as you can in English.

$\mathcal{H}$ for regression trees can include any tree that can be formed by selecting splits, and determining the values of these features at which the splits occur

(c) [harder] Think of another "leaf assignment" rule besides the average of the responses in the node that makes sense.

You can use the median of the responses, it's less sensitive to extreme outliers too

(f) [easy] Describe how g_{final} is constructed when using nested resampling on three splits of the data. The data is first split into 3 parts, then for the outer loop, 2 parts are combined as a training set, and the third is a test set. It iterates until each set serves as a test set. Then for the inner loop, there's model selection and hyper parameter tuning across each fold. Then we assess each model and choose the best one which is g final

(g) [easy] Describe how you would use this model selection procedure to find hyperparameter values in algorithms that require hyperparameters.
I would just go through this process a bunch of times, each time using a different hyper parameter, until I find the most effective one

(h) [difficult] Given raw features $x_1, \ldots, x_{p_{raw}}$, produce the most expansive set of transformed p features you can think of so that $p \gg n$.

(i) [easy] Describe the methodology from class that can create a linear model on a subset of the transformed featuers (from the previous problem) that will not overfit.

(b) [easy] Using two splits of the data, how would you select a model?

I would train a bunch of models on the training data, then apply each trained model to the validation set to see which one performs best. I would then choose the best performing model

(c) [easy] Discuss the main limitation with using two splits to select a model.

It can lead to biased evaluations bc the characteristics of the our dataset are not properly represented in the training and validation set.

(d) [easy] Using three splits of the data, how would you perform model selection?

I would train a bunch of models in Dtest, then use them in Dvalid to tweak them, then choose the best performing model on Dtest

(e) [easy] How does using both inner and outer folds in a double cross-validation nested resampling procedure improve the model selection procedure?

the inner loop focuses on model selection and hyper parameter tuning by evaluating different models across multiple datasets. the outer loop then evaluates the performance of the chosen model, providing an unbiased estimate of it's generalizibility to new data

Problem 3

These are some questions related to extrapolation.

(a) [easy] Define extrapolation and describe why it is a net-negative during prediction.

It's the process of estimating, beyond the original observation range, the value of a variable based on it's relationship w/ another variable.

It's net negative bc it often relies on the assumption that existing trends and patterns will be the same outside the observed data range, which in 4 dimensions, can become extremely inaccurate

(b) [easy] Do models extrapolate differently? Explain.

Yes, for example, linear models can extend straight line trends, and could then misjudge non linear data, while non linear models like polynomial regression can adapt to more complex patterns, but risk overfitting

(c) [easy] Why do polynomial regression models suffer terribly from extrapolation?

bc there's major overfitting. You're fitting complex curves that are unpredictable outside the observed range

Problem 4

These are some questions related to the model selection procedure discussed in lecture.

(a) [easy] Define the fundamental problem of "model selection".

overfitting vs underfitting. If you choose a model that is basic, you risk not capturing everything and underfitting. If you choose a model that is too complex, you risk capturing the noise, which would be overfitting.

(d) [difficult] We fit the following model: $\hat{y} = b_0 + b_1 x_1 + b_2 \ln(x_2)$. We spoke about in class that b_1 represents loosely the predicted change in response for a proportional movement in x_2. So e.g. if x_2 increases by 10%, the response is predicted to increase by $0.1b_2$. Prove this approximation from first principles.

Let x_2' be the new x_2 after 10% increase: $x_2' = 1.1 x_2$

Now we can substitute it into the model: $\hat{y}' = b_0 + b_1 x_1 + b_2 \ln(x_2')$

$\rightarrow \hat{y}' = b_0 + b_1 x_1 + b_2 \ln(1.1 x_2) \rightarrow \hat{y}' = b_0 + b_1 x_1 + b_2 \ln(1.1) + \ln(x_2)$

Now if we back substitute: $\hat{y}' = b_0 + b_1 x_1 + b_2 (\ln(1.1) + \ln(x_2)) \rightarrow \hat{y}' = b_0 + b_1 x_1 + b_2 \ln(x_2) + b_2 \ln(1.1)$

$\rightarrow \hat{y} = b_2 \ln(1.1)$ ($\ln(1.1)$ is $\approx .095$)

$\rightarrow \Delta y = \hat{y}' - \hat{y} = .1 b_2$. $\therefore$ a 10% increase in x_2 leads to an appropriate increase in $\hat{y}$

(e) [easy] When does the approximation from the previous question work? When do you expect the approximation from the previous question not to work?

It works when the percentage increase in x_2 is small enough that the approximation of $\ln(1.1) \approx .1$ remains reasonable. It would fail when this isn't the case

(f) [harder] We fit the following model: $\ln(\hat{y}) = b_0 + b_1 x_1 + b_2 \ln(x_2)$. What is the interpretation of b_1? What is the *approximate* interpretation of b_2? Although we didn't yet discuss the "true" interpretation of OLS coefficients, do your best with this.

b_1 represents the expected percentage change in y

b_2 indicates that a 1% increase in x_2 leads to approximately b_2% change in y

(g) [easy] Show that the model from the previous question is equal to $\hat{y} = m_0 m_1^{x_1} x_2^{b_2}$ and interpret m_1.

Problem 2

These are some questions related to polynomial-derived features and logarithm-derived features in use in OLS regression.

(a) [harder] What was the overarching problem we were trying to solve when we started to introduce polynomial terms into $\mathcal{H}$? What was the mathematical theory that justified this solution? Did this turn out to be a good solution? Why / why not?

we couldn't capture the non linear relationship between data points, it was greatly underfit. the theory was to just create a curve that goes through the data points. It was not worth it bc now we just overfit a lot

(b) [harder] We fit the following model: $\hat{y} = b_0 + b_1 x + b_2 x^2$. What is the interpretation of b_1? What is the interpretation of b_2? Although we didn't yet discuss the "true" interpretation of OLS coefficients, do your best with this.

b_1 represents the linear effect of the variable x on $\hat{y}$, showing how much $\hat{y}$ is changing when x changes that same amt

b_2 represents the quadratic effect x has on $\hat{y}$, it shows the change in the rate of increase or decrease of $\hat{y}$ as x increases

(c) [difficult] Assuming the model from the previous question, if $x \in \mathcal{X} = [10.0, 10.1]$, do you expect to "trust" the estimates b_1 and b_2? Why or why not?

I don't trust it bc there's very limited variability in x, the restricted domain does not provide enough data to properly estimate how $\hat{y}$ changes w/ x.