Announcements

Make sure to sign in on the google form (I send a list of which section questions are useful for which pset questions afterwards)

Pset 9 due Friday 11/17

Introductions

- One question or thought related to lecture last week (Decision trees, bagging, random forests)

Bagging and counting

In bagging, data points are bootstrapped and used to build overfit decision trees. By taking the average of these decision trees, we reduce the variance of predictions relative to a single tree built on all the data. The reason the trees are different is that they incorporate different data points; trees built from the same data points will be the same. Therefore, knowing how many data points overlap between bootstrap samples gives a rough measure of how different the trees will be.

1. Suppose we bootstrap n data points from our original n data points. Find the expected number of unique bootstrapped data points. Use the fact that $\left(1-\frac{1}{n}\right)^n \approx e^{-1}$ for large n to approximate the expected number. (Hint: Think of the number of unique data points as a sum of indicators for each data point being unique. Also, recall that for $r<1$, $\sum_{j=1}^n r^{j-1} = \frac{1-r^n}{1-r}$)

Let X be the number of unique bootstrapped data points and I_i be an indicator that draw i is unique.

$$\begin{aligned}
E(X) &= E\left(\sum_{i=1}^n I_i\right) \\
&= \sum_{i=1}^n P(\text{draw } i \text{ is unique}) \\
&= \sum_{i=1}^n \left(\frac{n-1}{n}\right)^{i-1} \\
&= \frac{1-\left(\frac{n-1}{n}\right)^n}{1-\left(\frac{n-1}{n}\right)} \\
&= n\left(1-\left(\frac{n-1}{n}\right)^n\right) \\
&\approx n(1-e^{-1})
\end{aligned}$$

where the second equality is by the fundamental bridge and the third is from the birthday problem.

2. What is the expected proportion of data points overlapping between two bootstrap samples of n items each (i.e., the proportion of data points in one bootstrap sample that are also in the other)?

Let X be the number of overlaps and I_i be an indicator of whether the i^{th} data point in the first bootstrap sample is in the second bootstrap sample.

$$\begin{aligned}
E(X) &= E\left(\sum_{i=1}^{n} I_i\right)\\
&= \sum_{i=1}^{n} P(\text{draw } i \text{ is in the other sample})\\
&= \sum_{i=1}^{n} \left(1 - \left(\frac{n-1}{n}\right)^n\right)\\
&= n\left(1 - \left(\frac{n-1}{n}\right)^n\right)\\
&\approx n(1 - e^{-1})
\end{aligned}$$

where the third equality is from the fact that the probability of draw i being in the other sample is the same as 1 minus the probability draw i is not in the other sample.

Categorical decision trees

In this class, we mainly look at decision trees as a non-parametric way of making a prediction about a continuous variable of interest. However, another (probably more common) use of decision trees is in predicting categorical variables. While looking at the sum of squared errors is a reasonable way to build decision trees for continuous predictions, other methods can be more useful for categorical predictions. This question will look at two such options.

1. A first method involves calculating the entropy of the parent and child nodes. Entropy is defined as:

$$E = -\sum_{i=1}^{k} p_i \log p_i$$

where p_i is the proportion of the items in the node from class i. (If $p_i = 0$, we treat its $p_i \log_2 p_i$ term as 0 since $p_i \log_2 p_i \to 0$ as $p_i \to 0$.) When splitting using this metric, the parent node's entropy E_{Parent} is calculated, and the weighted average of the children's entropies are calculated as

$$\frac{1}{n_{\text{Left}}} E_{\text{Left}} + \frac{1}{n_{\text{Right}}} E_{\text{Right}}$$

The split that yields the lowest entropy is chosen. Show that the entropy of a node is minimized when the node is "pure" (there are only items of one class in the node).

Because $0 \leq p_i \leq 1$, $\log p_i \leq 0$ with equality only when $p_i = 1$. With $p_i = 1$ for one entry and $p_i = 0$ for the rest, the entropy is 0. However, whenever there is a p_i such that $0 < p_i < 1$, there will be a term $p_i \log p_i < 0$, so the entropy will be positive since all the terms in the sum are at most 0. Thus, entropy is minimized when all items are in one class.

2. Show that the entropy of the node is maximized when there are equal proportions of items from each class in the node. Hint: Consider the function $g(p) = p \log p$. Also, you should use Jensen's inequality for sums:

$$\frac{1}{k}\sum_{i=1}^{k} g(p_i) \geq g\left(\frac{1}{k}\sum_{i=1}^{k} p_i\right)$$

when g is convex.

First, g is convex because $g'(p) = 1 + \log(p)$ and $g''(p) = \frac{1}{p} > 0$. Therefore, Jensen's inequality gives

$$\begin{aligned}\frac{1}{k}\sum_{i=1}^{k} g(p_i) \geq g\left(\frac{1}{k}\sum_{i=1}^{k} p_i\right) &\implies \frac{1}{k}\sum_{i=1}^{k} g(p_i) \geq g(1/k) \\ &\implies E/k \leq -1/k \log(1/k) \\ &\implies E \leq \log(k)\end{aligned}$$

where the second implication is from plugging in the formula for entropy and using the definition of g. If we let $p_i = 1/k$, we see that

$$\begin{aligned}E &= -\sum_{i=1}^{k} p_i \log p_i \\ &= \sum_{i=1}^{k} -\frac{1}{k}\log(1/k) \\ &= -\log(1/k) \\ &= \log(k)\end{aligned}$$

Thus, $p_i = 1/k$ for all i maximizes the entropy.

3. Explain why these results match our intuition of when a node should be split or not.

When a node is pure, there is no reason to split it, and its entropy is minimized, so we wouldn't split it. When a node contains an equal proportion of items from each category, its entropy is maximized, and we should split it.

4. Because calculating logs can be computationally expensive, another method is to maximize the Gini value:

$$G = \sum_{i=1}^{k} p_i^2$$

where p_i is the proportion of the items in the node from class i. Explain why the Gini value can be interpreted as the probability that a randomly chosen item in the node would be assigned its correct class when assigning classes randomly according to the proportions in the node.

We can interpret this as LOTP where we want $P(\text{Correct classification})$, so we use

$$P(\text{Correct classification}) = \sum_{i=1}^{k} P(\text{Correct classification}|\text{Item in class } i)P(\text{Item in class } i)$$

Then, $P(\text{Item in class } i) = p_i$ since we choose randomly according to the node proportions, and $P(\text{Correct classification}|\text{Item in class } i) = p_i$ since the probability of it being randomly assigned the (correct) class i is also p_i, so the sum comes out to G.

5. Find when the Gini value is maximized.

$$1 = \left(\sum_{i=1}^{k} p_i\right)^2 = \sum_{i=1}^{k} p_i^2 + 2\sum_{i<j} p_i p_j \implies \sum_{i=1}^{k} p_i^2 \leq 1$$

with equality when $p_i p_j = 0$ for all $i \neq j$. This only occurs when one $p_i = 1$ and the rest are 0.

6. Show that the Gini value is minimized when all classes have equal proportions in the node.

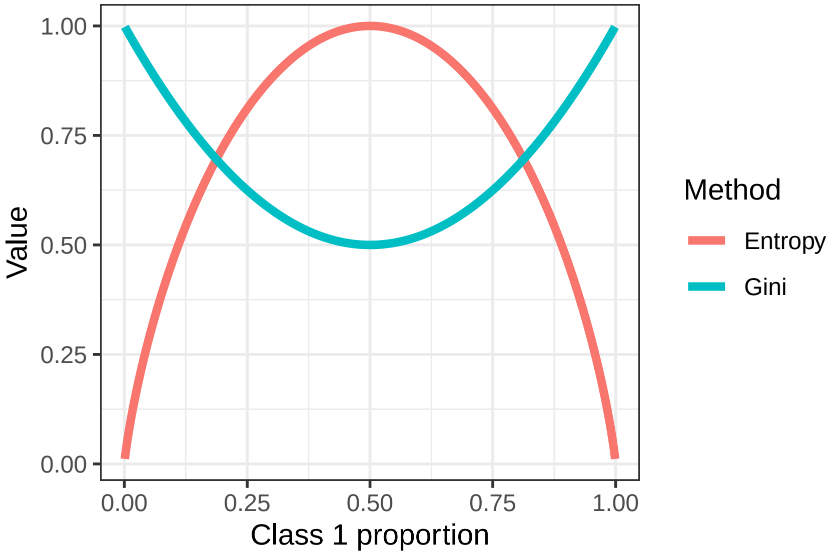
Since x^2 is a convex function, Jensen's inequality shows

$$\frac{1}{k}\sum_{i=1}^{k} p_i^2 \geq \left(\frac{1}{k}\sum_{i=1}^{k} p_i\right)^2 = 1/k^2$$

Multiplying both sides by k gives $\sum_{i=1}^{k} p_i^2 \geq 1/k$, and this has equality when $p_i = 1/k$.

7. Do we want to split a node with a high or low Gini value?

We want to split a node with a low Gini value since this is when the node has equal proportions from each category. When the Gini is maximized at 1, the node is pure, and we don't want to split it. The following plot shows that the Gini index is minimized when the entropy is maximized for a 2-class node, consistent with us splitting nodes with high entropy and low Gini indices.


1.00
0.75
0.50
0.25
0.00
Value
0.00
0.25
0.50
0.75
1.00
Class 1 proportion
Method
Entropy
Gini

The Entire Arboretum

The packages `rpart` and `randomForest` will contain most of the functions for decision trees and random forests. The most important functions for us will be:

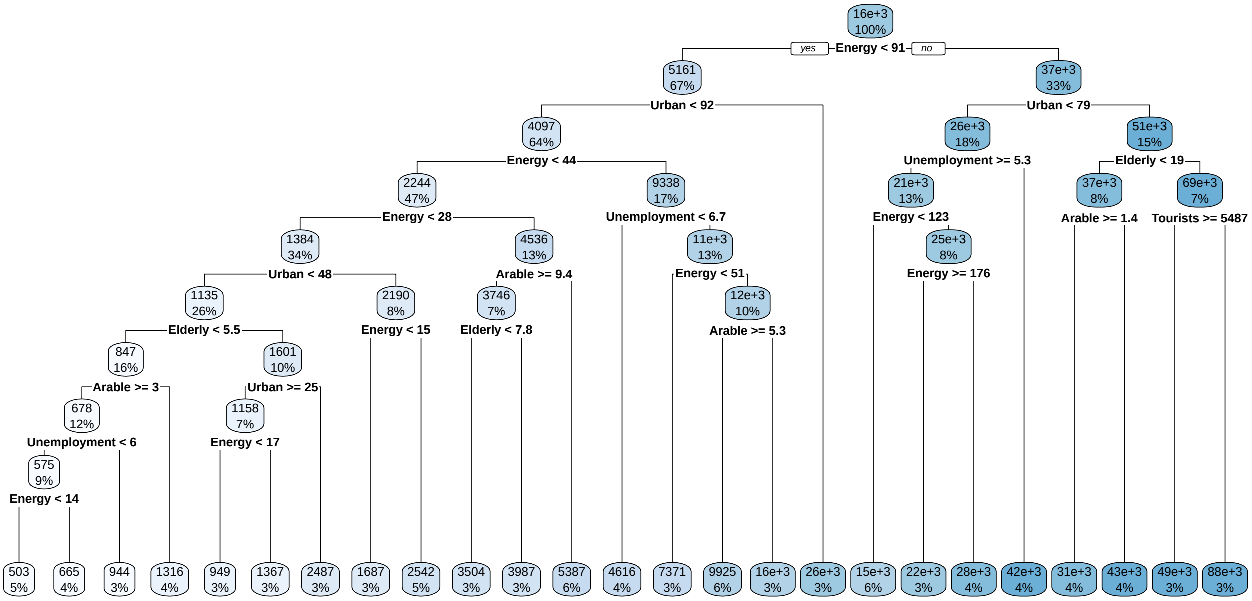
- `rpart(formula, data, control)` with `control` as `list(minsplit=1,cp=0,maxdepth=20)`, where `minsplit` says we want the node to be split if it has at least 1 observation; `cp` (complexity parameter) says we will accept any improvement in fit; and `maxdepth` is the maximum node depth.
- `prune(tree, cp)` where `cp` is the complexity parameter.
- `randomForest(formula, data, maxnodes, mtry, ntree)` where `mtry` is the number of variables randomly sampled as candidates at each split, and `ntree` is the number of trees to grow.

These problems will deal with a dataset of country-level statistics from UNdata, Varieties of Democracy, and the World Bank.

1. The following is a fit decision tree to predict GDP per capita along with RMSE and R^2 statistics for the tree's predictions relative to the true values. The decision tree uses `minsplit=1, cp=0, maxdepth=20` on the variables:

- `Urban`: Percent of people living in urban areas
- `Elderly`: Percent of people over the age of 60
- `Arable`: Percent of total land area that is farmable
- `Energy`: Gigajoules of energy produced per person
- `Unemployment`: Unemployment rate as a percent
- `Tourists`: Thousands of tourist/visitor arrivals

The 2010 US values for these variables are shown below. What is the US's estimated GDP per capita?

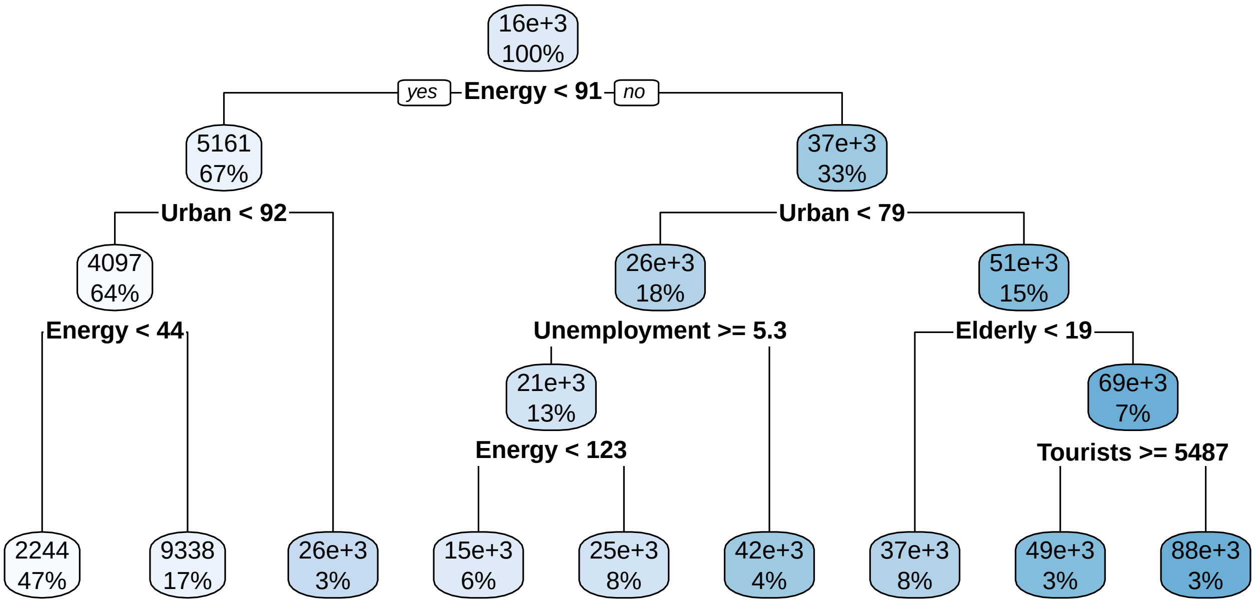


```
##      RMSE        R2
## 13262.239     0.689
```

Urban	Elderly	Arable	Energy	Unemployment	Tourists
80.8	18.3	17.2	301	9.6	60010

We step right (Energy >= 91), then right (Urban >= 79), then left (Elderly < 19), then left (Arable >= 1.4) to get $31,000 (the true value is $30678).

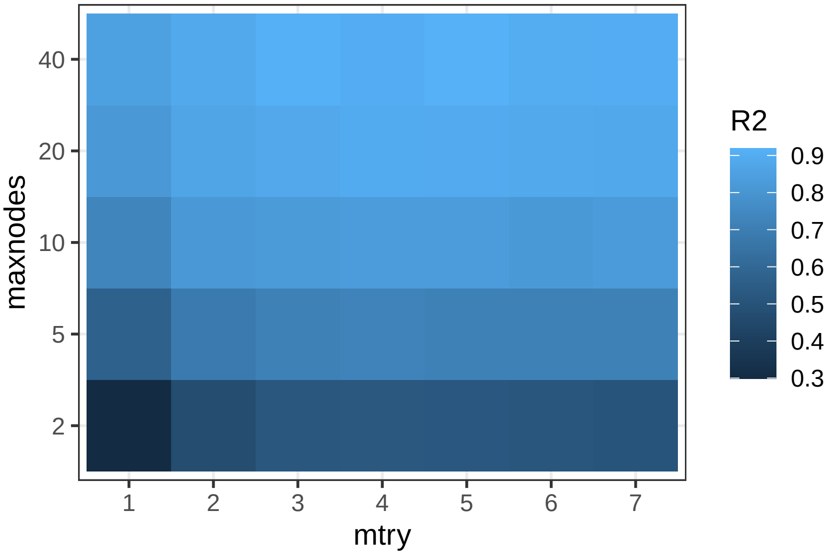
2. The following is a well-pruned regression tree to predict GDP per capita from the predictors above. The model starts with `minsplit=1, cp=0, maxdepth=20` and uses 10-fold cross-validation to prune based on the complexity parameter `cp`. Interpret the differences between the previous tree and the pruned one. What is the US's estimated GDP per capita?



```
##      RMSE        R2
## 15768.08      0.56
```

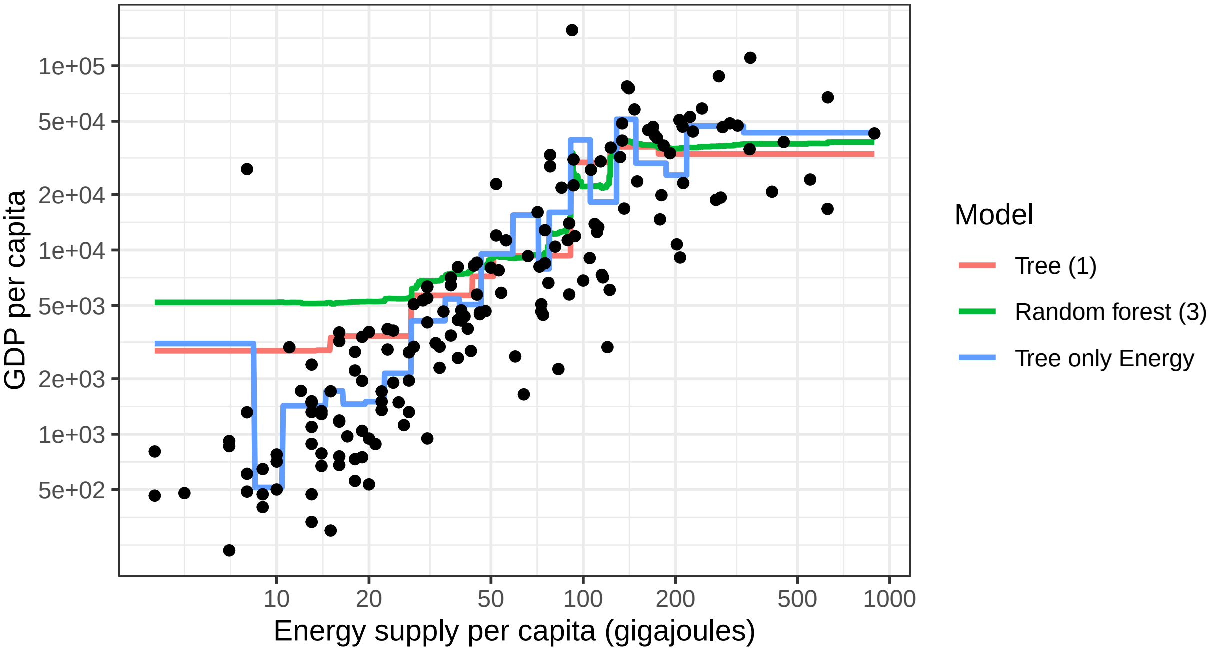
The RMSE and R^2 are worse than in the first tree, but this is likely because the first model is overfit. The predicted US GDP per capita is now \$37000 since the last arable split is gone.

3. The following is a set of random forest models to predict GDP per capita from the above predictors considering `mtry` 1 through 7 and `maxnodes` 2, 5, 10, 20, and 40 with `ntree=200`. The plot shows the R^2 of the out-of-bag predictions versus the true values for each parameter combination. Choose the best `mtry` and `maxnodes`.



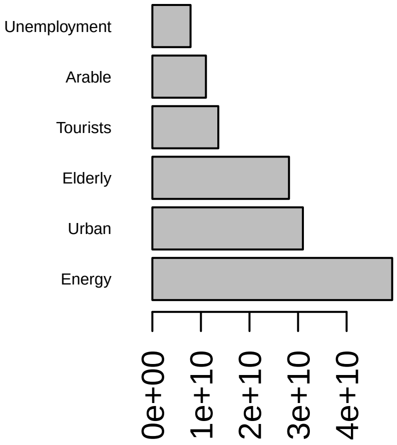
Any random forest with a maximum of 40 nodes and at least 2 included predictors sampled at each split would work.

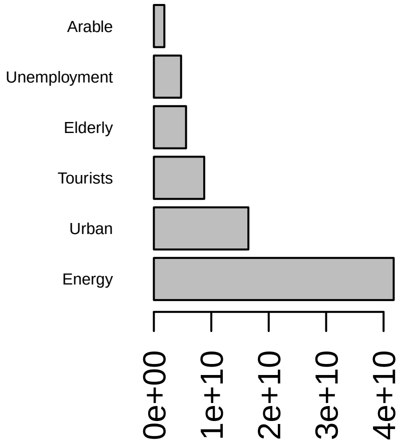
4. The following plot shows how GDP per capita changes with gigajoules of energy produced per person holding the other variables constant. We estimate just this effect by, for each country, replacing the `Energy` term with the range of energies in consideration and then averaging over the countries for each value in our range. We are comparing our decision tree from question 1, our best random forest from question 3, and a decision tree fit only on `Energy`. Describe what you see in a few sentences.



The tree using only `Energy` is most consistent with the observed data (and probably overfit) because the model can only use `Energy` as the splitting predictor. The other decision tree can split on other variables, so its reliance on the single variable `Energy` is not as strong. The random forest shows this even more clearly: because many splits don't even consider `Energy`, the trend is much less consistent with the energy supply alone, but the relationship is the smoothest.

5. Which predictors are most important in the first tree (left) and the best random forest (right)? How do they compare in relative importance?





The actual units on both of these are somewhat arbitrary (see the `rpart` and `randomForest` documentation for the details), but we see that energy consumption is the most important variable in both models; the urban and elderly populations are important; and the number of tourists, the unemployment rate, and proportion of arable land are the least important.