Regression and Classification Trees

Nate Wells

Math 243: Stat Learning

November 4th, 2020

Outline

In today's class, we will. . .

- Investigate pruning algorithms for improving accuracy of regression trees
- Discuss classifcation trees for classification problems.

Section 1

Improving Regression Trees

Trees on Trees

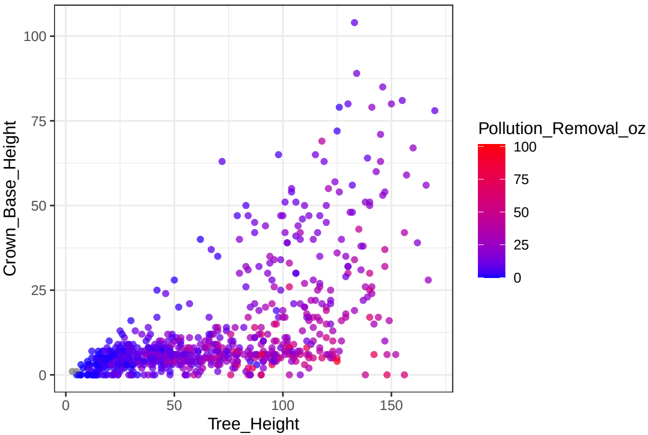
We use a subset of the `pdxTrees` dataset from the `pdxTrees` repo (maintined by K. McConville, I. Caldwell, and N. Horton)

Trees on Trees

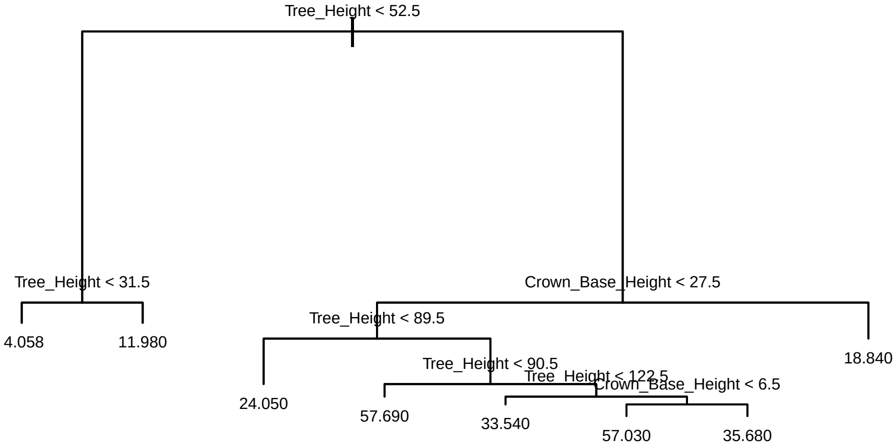
We use a subset of the pdxTrees dataset from the pdxTrees repo (maintined by K. McConville, I. Caldwell, and N. Horton)

```
## Rows: 1,000
## Columns: 10
## $ Species              <fct> PSME, CAJA, QUMU, CADE, PSME, CPSP, PRAV, PSME...
## $ Condition            <fct> Fair, Fair, Fair, Fair, Fair, Fair, Poor, Fair...
## $ Tree_Height          <int> 102, 23, 18, 78, 123, 85, 11, 145, 16, 72, 88,...
## $ Crown_Width_NS       <int> 52, 36, 6, 17, 52, 36, 9, 36, 10, 86, 25, 12, ...
## $ Crown_Width_EW       <int> 43, 40, 6, 18, 38, 52, 11, 35, 10, 86, 10, 16,...
## $ Crown_Base_Height    <int> 63, 5, 5, 6, 13, 5, 6, 9, 5, 8, 6, 4, 4, 3, 2,...
## $ Structural_Value     <dbl> 6694.04, 2444.75, 71.28, 4162.43, 6159.02, 113...
## $ Carbon_Storage_lb    <dbl> 1992.9, 917.5, 5.3, 1428.7, 1901.4, 11071.6, 2...
## $ Stormwater_ft        <dbl> 78.9, 43.9, 1.0, 19.8, 117.6, 52.0, 4.1, 80.1,...
## $ Pollution_Removal_oz <dbl> 21.2, 11.8, 0.3, 5.3, 31.6, 14.0, 1.1, 21.5, 1...
```

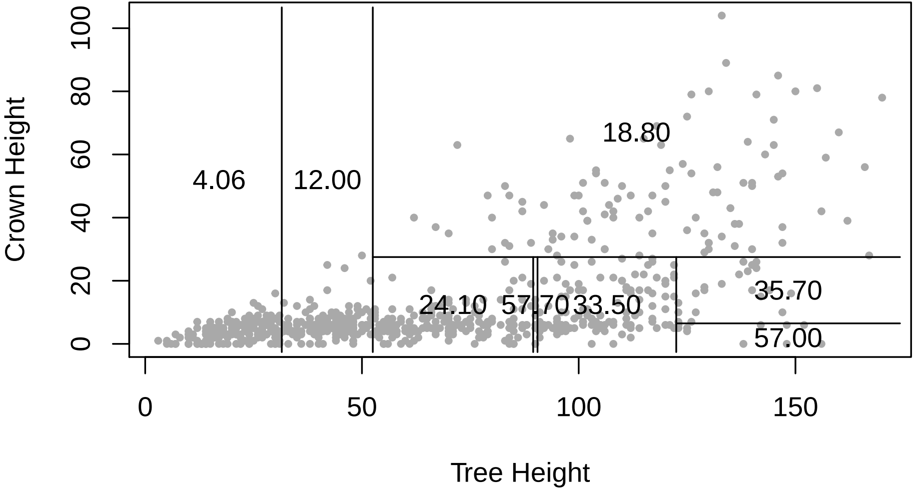
Pollution Removal



Regression Tree



Another Visualization



Tree Accuracy

Let's check MSE on a test set:

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

```
##   Tree_MSE
## 1 169.0145
```

Tree Accuracy

Let's check MSE on a test set:

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

```
##   Tree_MSE
## 1 169.0145
```

And compared to the linear model:

```
##     lm_MSE
## 1 412.3758
```

Tree Accuracy

Let's check MSE on a test set:

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

```
##   Tree_MSE
## 1 169.0145
```

And compared to the linear model:

```
##     lm_MSE
## 1 412.3758
```

Why did the tree model outperform the linear model?

Tree Accuracy

Let's check MSE on a test set:

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

```
##   Tree_MSE
## 1 169.0145
```

And compared to the linear model:

```
##     lm_MSE
## 1 412.3758
```

Why did the tree model outperform the linear model?

Nevertheless, what are some downsides to the tree model?

The general tree algorithm

1. Begin with the entire data set S and search every value of every predictor to cut S into two groups S_1 and S_2 that minimizes sum of squred error:

$$\text{SSE} = \sum_{i \in S_1} (y_i - \bar{y}_1)^2 + \sum_{i \in S_2} (y_i - \bar{y}_2)^2$$

The general tree algorithm

1. Begin with the entire data set S and search every value of every predictor to cut S into two groups S_1 and S_2 that minimizes sum of squred error:

$$\mathrm{SSE} = \sum_{i \in S_1} (y_i - \bar{y}_1)^2 + \sum_{i \in S_2} (y_i - \bar{y}_2)^2$$

2. Repeat step one on both S_1 and S_2.

The general tree algorithm

1. Begin with the entire data set S and search every value of every predictor to cut S into two groups S_1 and S_2 that minimizes sum of squred error:

$$\mathrm{SSE} = \sum_{i \in S_1} (y_i - \bar{y}_1)^2 + \sum_{i \in S_2} (y_i - \bar{y}_2)^2$$

2. Repeat step one on both S_1 and S_2.
3. Repeat on the new regions.

The general tree algorithm

1. Begin with the entire data set S and search every value of every predictor to cut S into two groups S_1 and S_2 that minimizes sum of squred error:

$$\mathrm{SSE} = \sum_{i \in S_1} (y_i - \bar{y}_1)^2 + \sum_{i \in S_2} (y_i - \bar{y}_2)^2$$

2. Repeat step one on both S_1 and S_2.
3. Repeat on the new regions.
4. ...

The general tree algorithm

1. Begin with the entire data set S and search every value of every predictor to cut S into two groups S_1 and S_2 that minimizes sum of squred error:

$$\mathrm{SSE} = \sum_{i \in S_1} (y_i - \bar{y}_1)^2 + \sum_{i \in S_2} (y_i - \bar{y}_2)^2$$

2. Repeat step one on both S_1 and S_2.
3. Repeat on the new regions.
4. ...
5. Stop?

The general tree algorithm

1. Begin with the entire data set S and search every value of every predictor to cut S into two groups S_1 and S_2 that minimizes sum of squred error:

$$\mathrm{SSE} = \sum_{i \in S_1} (y_i - \bar{y}_1)^2 + \sum_{i \in S_2} (y_i - \bar{y}_2)^2$$

2. Repeat step one on both S_1 and S_2.
3. Repeat on the new regions.
4. ...
5. Stop?

How do we decide when to abort algorithm?

The general tree algorithm

1. Begin with the entire data set S and search every value of every predictor to cut S into two groups S_1 and S_2 that minimizes sum of squred error:

$$\mathrm{SSE} = \sum_{i \in S_1} (y_i - \bar{y}_1)^2 + \sum_{i \in S_2} (y_i - \bar{y}_2)^2$$

2. Repeat step one on both S_1 and S_2.
3. Repeat on the new regions.
4. ...
5. Stop?

How do we decide when to abort algorithm?

Consider the RSS of a **big** tree. How might training and test RSS compare?

Subtrees

A **subtree** is a regression tree obtained by removing some of the branches and nodes from the full regression tree.

Subtrees

A **subtree** is a regression tree obtained by removing some of the branches and nodes from the full regression tree.

- Compare test and training RSS between full tree and a subtree.

Subtrees

A **subtree** is a regression tree obtained by removing some of the branches and nodes from the full regression tree.

- Compare test and training RSS between full tree and a subtree.

Like the best subset selection algorithm for linear models, we can improve test RSS by exhaustively searching all subtrees for the best performing model.

Subtrees

A **subtree** is a regression tree obtained by removing some of the branches and nodes from the full regression tree.

- Compare test and training RSS between full tree and a subtree.

Like the best subset selection algorithm for linear models, we can improve test RSS by exhaustively searching all subtrees for the best performing model.

- But this search is actually even more computationally expensive than best subset!

Subtrees

A **subtree** is a regression tree obtained by removing some of the branches and nodes from the full regression tree.

- Compare test and training RSS between full tree and a subtree.

Like the best subset selection algorithm for linear models, we can improve test RSS by exhaustively searching all subtrees for the best performing model.

- But this search is actually even more computationally expensive than best subset!
- So we instead restrict our attention to those subtrees most likely to improve RSS

Pruning Algorithm

Once a tree is fully grown, we *prune* it using *cost-complexity tuning*

Pruning Algorithm

Once a tree is fully grown, we *prune* it using *cost-complexity tuning*

- The goal is to find a tree of optimal size with the smallest error rate.

Pruning Algorithm

Once a tree is fully grown, we *prune* it using *cost-complexity tuning*

- The goal is to find a tree of optimal size with the smallest error rate.
- We consider a sequence of trees indexed by a tuning parameter α.

Pruning Algorithm

Once a tree is fully grown, we *prune* it using *cost-complexity tuning*

- The goal is to find a tree of optimal size with the smallest error rate.
- We consider a sequence of trees indexed by a tuning parameter α.

For each value of α, there exists a unique subtree T of the full tree T_0 that minimizes

$$\mathrm{RSS} + \alpha|T|$$

where $|T|$ is the number of terminal nodes of the tree T.

Pruning Algorithm

Once a tree is fully grown, we *prune* it using *cost-complexity tuning*

- The goal is to find a tree of optimal size with the smallest error rate.
- We consider a sequence of trees indexed by a tuning parameter α.

For each value of α, there exists a unique subtree T of the full tree T_0 that minimizes

$$\mathrm{RSS} + \alpha|T|$$

where $|T|$ is the number of terminal nodes of the tree T.

- That is, α penalizes a tree based on its number of terminal nodes.

Pruning Algorithm

Once a tree is fully grown, we *prune* it using *cost-complexity tuning*

- The goal is to find a tree of optimal size with the smallest error rate.
- We consider a sequence of trees indexed by a tuning parameter α.

For each value of α, there exists a unique subtree T of the full tree T_0 that minimizes

$$\text{RSS} + \alpha|T|$$

where $|T|$ is the number of terminal nodes of the tree T.

- That is, α penalizes a tree based on its number of terminal nodes.
- As α increases from 0 (i.e. the full tree), branches get pruned in a predictable way, making for relatively quick computation.

Pruning Algorithm

Once a tree is fully grown, we *prune* it using *cost-complexity tuning*

- The goal is to find a tree of optimal size with the smallest error rate.
- We consider a sequence of trees indexed by a tuning parameter α.

For each value of α, there exists a unique subtree T of the full tree T_0 that minimizes

$$\mathrm{RSS} + \alpha|T|$$

where $|T|$ is the number of terminal nodes of the tree T.

- That is, α penalizes a tree based on its number of terminal nodes.
- As α increases from 0 (i.e. the full tree), branches get pruned in a predictable way, making for relatively quick computation.
- We can find the optimal value of α using cross-validation

Pruning Algorithm

Once a tree is fully grown, we *prune* it using *cost-complexity tuning*

- The goal is to find a tree of optimal size with the smallest error rate.
- We consider a sequence of trees indexed by a tuning parameter α.

For each value of α, there exists a unique subtree T of the full tree T_0 that minimizes

$$\mathrm{RSS} + \alpha|T|$$

where $|T|$ is the number of terminal nodes of the tree T.

- That is, α penalizes a tree based on its number of terminal nodes.
- As α increases from 0 (i.e. the full tree), branches get pruned in a predictable way, making for relatively quick computation.
- We can find the optimal value of α using cross-validation

There are two ways to select the **best** subtree.

Pruning Algorithm

Once a tree is fully grown, we *prune* it using *cost-complexity tuning*

- The goal is to find a tree of optimal size with the smallest error rate.
- We consider a sequence of trees indexed by a tuning parameter α.

For each value of α, there exists a unique subtree T of the full tree T_0 that minimizes

$$\text{RSS} + \alpha|T|$$

where $|T|$ is the number of terminal nodes of the tree T.

- That is, α penalizes a tree based on its number of terminal nodes.
- As α increases from 0 (i.e. the full tree), branches get pruned in a predictable way, making for relatively quick computation.
- We can find the optimal value of α using cross-validation

There are two ways to select the **best** subtree.

1. Choose the tree with smallest MSE.

Pruning Algorithm

Once a tree is fully grown, we *prune* it using *cost-complexity tuning*

- The goal is to find a tree of optimal size with the smallest error rate.
- We consider a sequence of trees indexed by a tuning parameter α.

For each value of α, there exists a unique subtree T of the full tree T_0 that minimizes

$$\mathrm{RSS} + \alpha|T|$$

where $|T|$ is the number of terminal nodes of the tree T.

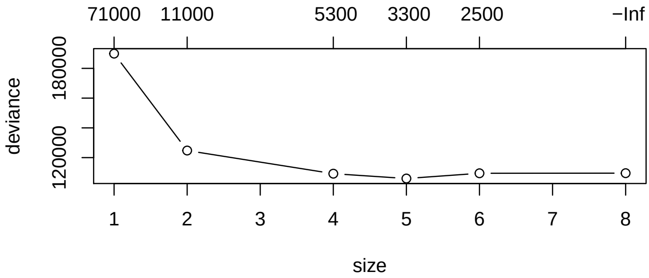
- That is, α penalizes a tree based on its number of terminal nodes.
- As α increases from 0 (i.e. the full tree), branches get pruned in a predictable way, making for relatively quick computation.
- We can find the optimal value of α using cross-validation

There are two ways to select the **best** subtree.

1. Choose the tree with smallest MSE.
2. Choose the *smallest* tree with MSE within 1 standard deviation of smallest MSE

Pruning Example

How does MSE vary as tree size changes?

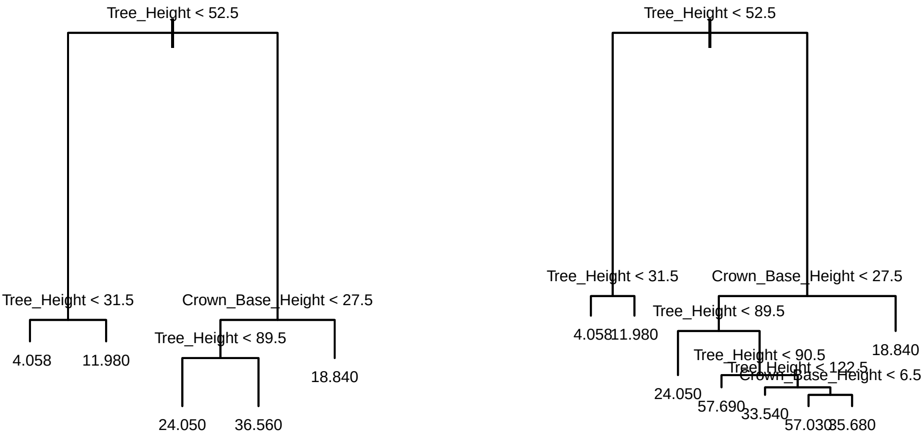


What are the test MSEs for the full tree and the subtree with 5 terminal nodes?

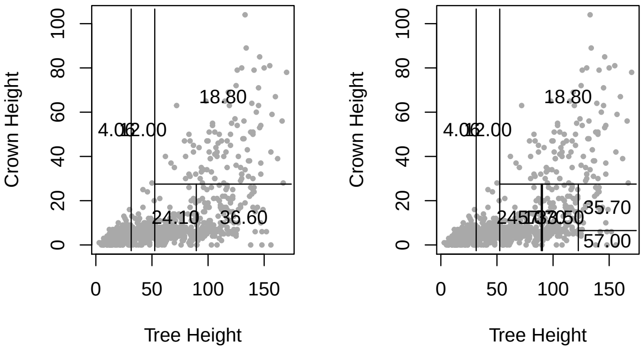
```
##   Full_Tree_MSE
## 1      169.0145
```

```
##   small_Tree_MSE
## 1       152.5175
```

Comparison


Tree_Height < 52.5
Tree_Height < 31.5
Crown_Base_Height < 27.5
Tree_Height < 89.5
4.058
11.980
18.840
24.050
36.560
Tree_Height < 52.5
Tree_Height < 31.5
Crown_Base_Height < 27.5
Tree_Height < 89.5
4.058
11.980
18.840
Tree_Height < 90.5
Tree_Height < 122.5
Crown_Base_Height < 6.5
24.050
57.690
33.540
57.030
35.680

Comparison 2



Creating Tree Models in R

There are two common packages for creating regression trees in R: `tree` and `rpart`.

Creating Tree Models in R

There are two common packages for creating regression trees in R: `tree` and `rpart`.

- The `tree` package is one of the oldest packages on CRAN. It is a (tiny) bit easier to use. But its plots are ugly. ISLR uses `tree`.

Creating Tree Models in R

There are two common packages for creating regression trees in R: `tree` and `rpart`.

- The `tree` package is one of the oldest packages on CRAN. It is a (tiny) bit easier to use. But its plots are ugly. ISLR uses `tree`.
- The `rpart` package is newer, computationally faster, and has more options. It also can be combined with the `partykit` and `ggparty` packages for **much** nicer plots. Applied Predictive Modeling uses `rpart` along with `caret` for cv.

Trees using tree

To fit a tree:

```
library(tree)
tree_model<-tree(Pollution_Removal_oz ~ ., data = small_pdxTrees)
```

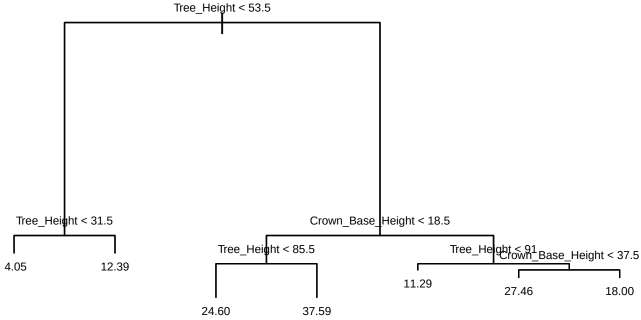
Trees using tree

To fit a tree:

```
library(tree)
tree_model<-tree(Pollution_Removal_oz ~ ., data = small_pdxTrees)
```

To view:

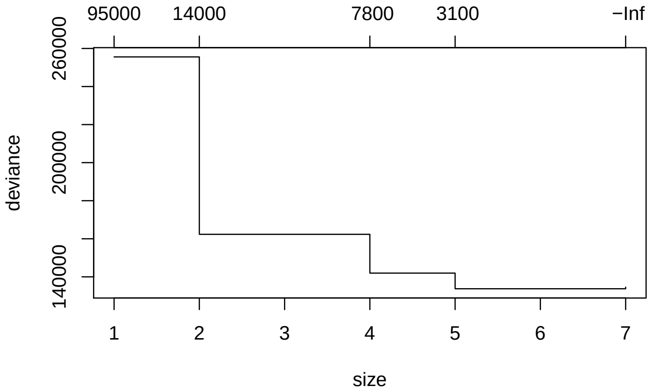
```
plot(tree_model)
text(tree_model, pretty = 0, cex = .5)
```



Trees in R via tree cont'd

To perform cost-complexity pruning:

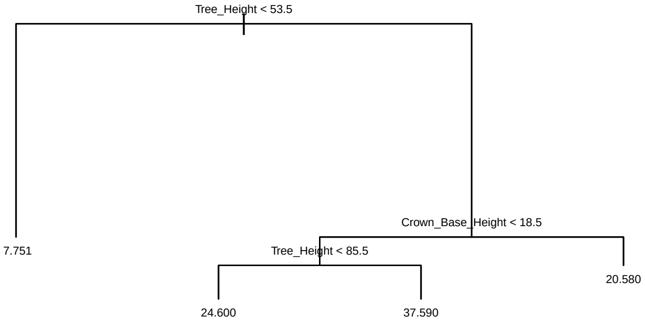
```
tree_model_cv<-cv.tree(tree_model)
plot(tree_model_cv)
```



Trees in R via tree cont'd

And to get a pruned tree:

```
pruned_tree<-prune.tree(tree_model, best = 4)
plot(pruned_tree)
text(pruned_tree, pretty = 0, cex = .5)
```



Section 2

Classification Trees

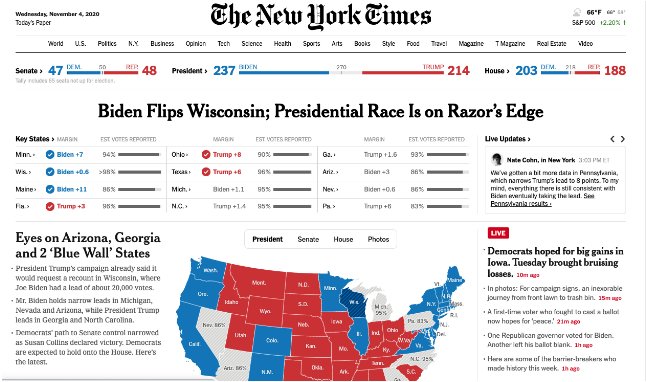
Trees for Classification Problems

Can we predict the winner of a presidential election based on demographics, state polling, economic conditions, and other features?

Trees for Classification Problems

Can we predict the winner of a presidential election based on demographics, state polling, economic conditions, and other features?

- No. Too stressful.


Wednesday, November 4, 2020
Today's Paper

The New York Times

66°F 66° 58°
S&P 500 +2.20% ↑

World U.S. Politics N.Y. Business Opinion Tech Science Health Sports Arts Books Style Food Travel Magazine T Magazine Real Estate Video

Senate › 47 DEM. 50 REP. 48
Tally includes 65 seats not up for election.
President › 237 BIDEN 270 TRUMP 214
House › 203 DEM. 218 REP. 188

Biden Flips Wisconsin; Presidential Race Is on Razor's Edge

Key States ›	Margin	Est. votes reported		Margin	Est. votes reported		Margin	Est. votes reported
Minn. ›	Biden +7	94%	Ohio ›	Trump +8	90%	Ga. ›	Trump +1.6	93%
Wis. ›	Biden +0.6	>98%	Texas ›	Trump +6	96%	Ariz. ›	Biden +3	86%
Maine ›	Biden +11	86%	Mich. ›	Biden +1.1	95%	Nev. ›	Biden +0.6	86%
Fla. ›	Trump +3	96%	N.C. ›	Trump +1.4	95%	Pa. ›	Trump +6	83%

Live Updates ›

Nate Cohn, in New York 3:03 PM ET
We've gotten a bit more data in Pennsylvania, which narrows Trump's lead to 8 points. To my mind, everything there is still consistent with Biden eventually taking the lead. See Pennsylvania results ›

Eyes on Arizona, Georgia and 2 'Blue Wall' States

- President Trump's campaign already said it would request a recount in Wisconsin, where Joe Biden had a lead of about 20,000 votes.
- Mr. Biden holds narrow leads in Michigan, Nevada and Arizona, while President Trump leads in Georgia and North Carolina.
- Democrats' path to Senate control narrowed as Susan Collins declared victory. Democrats are expected to hold onto the House. Here's the latest.

President Senate House Photos

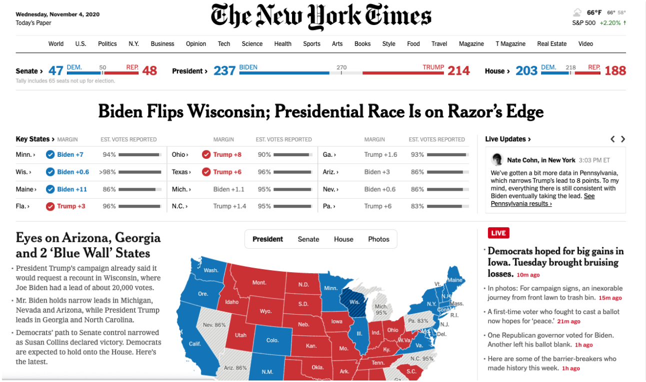
LIVE

- Democrats hoped for big gains in Iowa. Tuesday brought bruising losses. 10m ago
- In photos: For campaign signs, an inexorable journey from front lawn to trash bin. 15m ago
- A first-time voter who fought to cast a ballot now hopes for 'peace.' 21m ago
- One Republican governor voted for Biden. Another left his ballot blank. 1h ago
- Here are some of the barrier-breakers who made history this week. 1h ago

Trees for Classification Problems

Can we predict the winner of a presidential election based on demographics, state polling, economic conditions, and other features?

- No. Too stressful.



Wednesday, November 4, 2020
Today's Paper

The New York Times

66°F 66° 58°
S&P 500 +2.20% ↑

World U.S. Politics N.Y. Business Opinion Tech Science Health Sports Arts Books Style Food Travel Magazine T Magazine Real Estate Video

Senate › 47 DEM. 50 REP. 48
Tally includes 65 seats not up for election.
President › 237 BIDEN 270 TRUMP 214
House › 203 DEM. 218 REP. 188

Biden Flips Wisconsin; Presidential Race Is on Razor's Edge

Key States ›	MARGIN	EST. VOTES REPORTED		MARGIN	EST. VOTES REPORTED		MARGIN	EST. VOTES REPORTED
Minn. ›	Biden +7	94%	Ohio ›	Trump +8	90%	Ga. ›	Trump +1.6	93%
Wis. ›	Biden +0.6	>98%	Texas ›	Trump +6	96%	Ariz. ›	Biden +3	86%
Maine ›	Biden +11	86%	Mich. ›	Biden +1.1	95%	Nev. ›	Biden +0.6	86%
Fla. ›	Trump +3	96%	N.C. ›	Trump +1.4	95%	Pa. ›	Trump +6	83%

Live Updates ›

Nate Cohn, in New York 3:03 PM ET
We've gotten a bit more data in Pennsylvania, which narrows Trump's lead to 8 points. To my mind, everything there is still consistent with Biden eventually taking the lead. See Pennsylvania results ›

Eyes on Arizona, Georgia and 2 'Blue Wall' States

- President Trump's campaign already said it would request a recount in Wisconsin, where Joe Biden had a lead of about 20,000 votes.
- Mr. Biden holds narrow leads in Michigan, Nevada and Arizona, while President Trump leads in Georgia and North Carolina.
- Democrats' path to Senate control narrowed as Susan Collins declared victory. Democrats are expected to hold onto the House. Here's the latest.

President Senate House Photos

LIVE

- Democrats hoped for big gains in Iowa. Tuesday brought bruising losses. 10m ago
- In photos: For campaign signs, an inexorable journey from front lawn to trash bin. 15m ago
- A first-time voter who fought to cast a ballot now hopes for 'peace.' 21m ago
- One Republican governor voted for Biden. Another left his ballot blank. 1h ago
- Here are some of the barrier-breakers who made history this week. 1h ago

Trees for Classification Problems

Can we predict the species of a Portand tree based on its crown height and overall height?

Trees for Classification Problems

Can we predict the species of a Portand tree based on its crown height and overall height?

- YES!

Classifcation Trees

Classification trees are very similar to regression trees, except the terminal nodes predict levels of a categorical variable, rather than values of a quantitative variable

Classifcation Trees

Classification trees are very similar to regression trees, except the terminal nodes predict levels of a categorical variable, rather than values of a quantitative variable

- But to *grow* a classification tree, we need to make cuts based on a metric other than RSS (why?)

Classifcation Trees

Classification trees are very similar to regression trees, except the terminal nodes predict levels of a categorical variable, rather than values of a quantitative variable

- But to *grow* a classification tree, we need to make cuts based on a metric other than RSS (why?)

Some options for decision metric:

Classifcation Trees

Classification trees are very similar to regression trees, except the terminal nodes predict levels of a categorical variable, rather than values of a quantitative variable

- But to *grow* a classification tree, we need to make cuts based on a metric other than RSS (why?)

Some options for decision metric:

- *Classification error rate* (i.e. prop. obs. in region not in most common class)

Classifcation Trees

Classification trees are very similar to regression trees, except the terminal nodes predict levels of a categorical variable, rather than values of a quantitative variable

- But to *grow* a classification tree, we need to make cuts based on a metric other than RSS (why?)

Some options for decision metric:

- *Classification error rate* (i.e. prop. obs. in region not in most common class)
    - But because of the greedy algorithm used to split trees, CER tends to overfit to noise

Classifcation Trees

Classification trees are very similar to regression trees, except the terminal nodes predict levels of a categorical variable, rather than values of a quantitative variable

- But to *grow* a classification tree, we need to make cuts based on a metric other than RSS (why?)

Some options for decision metric:

- *Classification error rate* (i.e. prop. obs. in region not in most common class)
  - But because of the greedy algorithm used to split trees, CER tends to overfit to noise
- The *Gini index* as a measure of total variance across all K classes:

$$G = \sum_{i=1}^{K} \hat{p}_{mk}(1 - \hat{p}_{mk}) \quad \text{where } \hat{p}_{mk} = \text{prop. obs. in region m in class k}$$

Classifcation Trees

Classification trees are very similar to regression trees, except the terminal nodes predict levels of a categorical variable, rather than values of a quantitative variable

- But to *grow* a classification tree, we need to make cuts based on a metric other than RSS (why?)

Some options for decision metric:

- *Classification error rate* (i.e. prop. obs. in region not in most common class)
    - But because of the greedy algorithm used to split trees, CER tends to overfit to noise
- The *Gini index* as a measure of total variance across all K classes:

$$G = \sum_{i=1}^{K} \hat{p}_{mk}(1 - \hat{p}_{mk}) \quad \text{where } \hat{p}_{mk} = \text{prop. obs. in region m in class k}$$

- The Gini index is small if all $\hat{p}_{mk}$ are close to 0 or 1.