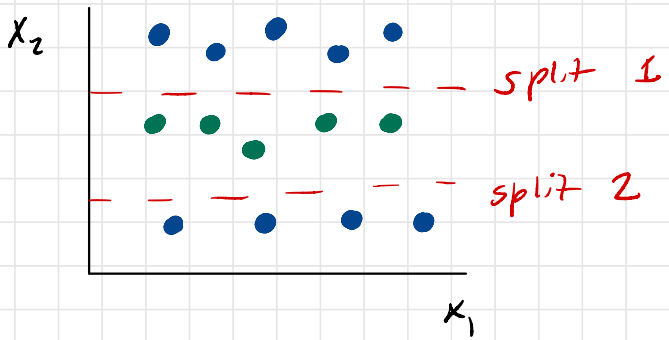
Pset 4

Bennett Hellman

I worked with Kyle Mandelbaum, Anvitha Krishnan, Issy Siegel & Dev Das.

1. True, there is no constraint that prohibits consecutive splits on the same variable. This could occur if there are multiple distinct "bins" in one variable as shown below:



2. True, misclassification error is $1 - \max(p, 1-p)$ while gini is $2p(1-p)$ and these are obviously not always equivalent, e.g. $p = .25$. If one tree had $p = .3$ and the other $p = .7$ they would have the same misclassification, .3, but different gini.

3. False, from p. 108: misclassification criterion do not always work well with greedy algorithms because they don't reward splits that are more desirable and thus stunt the growth of the tree.

4. False, if they are imbalanced classes than the classification error will not be .5

5. False, classification trees are not particularly sensitive to outliers since data points are equally weighted in the impurity measures. However, regression trees are since their impurity measures (e.g. MSE) account for the distance a point is away.

6. False, the randomness comes from a subset of variables as well.

7. True, logistic regression is a linear classifier even though its underlying sigmoid function is nonlinear. The decision boundary is linear, making it a linear prediction method.

8. True, standardizing / normalizing points does not effect the proportional distance between points and thus has no effect on trees' performance.

9. True, since the second tree trains on in-sample error XGBoost can only improve. This is not necessarily the case for out-of-sample.

10. False, while XGBoost is a weak learner random forest could be high variance trees and not necessarily weak learners.

In [1]:
```
using StatsBase, Random, LinearAlgebra, Plots, DecisionTree, DataFrames, ML
```

In [2]:
```
data = CSV.read("stevens.csv", DataFrame, header=1, pool=true);
```

In [4]:
```
y = data[:, 7]
X = data[:, 1:6]
(train_X, train_y), (test_X, test_y) = IAI.split_data(:classification, X, y
```

a)

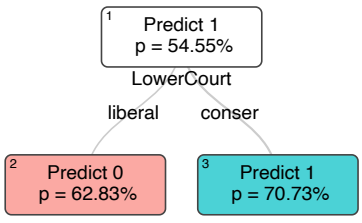
All models had identical training accuracy and AUC, with the exception of the entropy based model having slightly improved AUC. Additionally, all models had effectively the same out-of-sample accuracy. Again, the entropy based model had slightly higher AUC as opposed to the other models. These nearly identical when rounded performance metrics are a rarity and we would not expect this to happen often. This is a result of the misclassification and gini measures creating identical trees.

In [5]:
```
lnr = IAI.OptimalTreeClassifier(random_seed=15095, criterion=:misclassifica
grid = IAI.GridSearch(lnr)
IAI.fit!(grid, train_X, train_y, validation_criterion= :auc)
lnr = IAI.get_learner(grid)
```

```
┌ Warning: ProgressMeter by default refresh meters with additional inform
ation in IJulia via `IJulia.clear_output`, which clears all outputs in th
e cell.
│   - To prevent this behaviour, do `ProgressMeter.ijulia_behavior(:appen
d)`.
│   - To disable this warning message, do `ProgressMeter.ijulia_behavior(:
clear)`.
└ @ ProgressMeter /Users/iai/builds/InterpretableAI/SystemImage/SysImgBui
lder/.julia/packages/ProgressMeter/Vf8un/src/ProgressMeter.jl:620
Refitting with best parameters...  100%|████████████████| Time: 0:00:00
4m  Parameters:  cp=>0.08069
```

Out[5]:

Collapse Expand Save PNG



In [6]:
```
println("In-Sample Accuracy = ", round(IAI.score(lnr,train_X, train_y,crite
println("In-Sample AUC = ", round(IAI.score(lnr,train_X, train_y,criterion=
println("Out-of-Sample Accuracy = ", round(IAI.score(lnr,test_X, test_y,cri
println("Out-of-Sample AUC = ", round(IAI.score(lnr,test_X, test_y,criterio
```

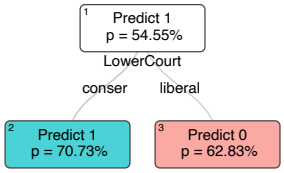
```
In-Sample Accuracy = 0.6692
In-Sample AUC = 0.669
Out-of-Sample Accuracy = 0.6647
Out-of-Sample AUC = 0.6645
```

In [7]:
```
lnr = IAI.OptimalTreeClassifier(random_seed=15095, criterion=:gini, max_dep
grid = IAI.GridSearch(lnr)
IAI.fit!(grid, train_X, train_y, validation_criterion= :auc)
lnr = IAI.get_learner(grid)
```

```
┌ Warning: ProgressMeter by default refresh meters with additional inform
ation in IJulia via `IJulia.clear_output`, which clears all outputs in th
e cell.
│   - To prevent this behaviour, do `ProgressMeter.ijulia_behavior(:appen
d)`.
│   - To disable this warning message, do `ProgressMeter.ijulia_behavior(:
clear)`.
└ @ ProgressMeter /Users/iai/builds/InterpretableAI/SystemImage/SysImgBui
lder/.julia/packages/ProgressMeter/Vf8un/src/ProgressMeter.jl:620
Refitting with best parameters...  100%|████████████████| Time: 0:00:00
4m  Parameters:  cp=>0.06777
```

Out[7]:

Collapse Expand Save PNG



In [8]:
```
println("In-Sample Accuracy = ", round(IAI.score(lnr,train_X, train_y,crite
println("In-Sample AUC = ", round(IAI.score(lnr,train_X, train_y,criterion=
println("Out-of-Sample Accuracy = ", round(IAI.score(lnr,test_X, test_y,cri
println("Out-of-Sample AUC = ", round(IAI.score(lnr,test_X, test_y,criterio
```

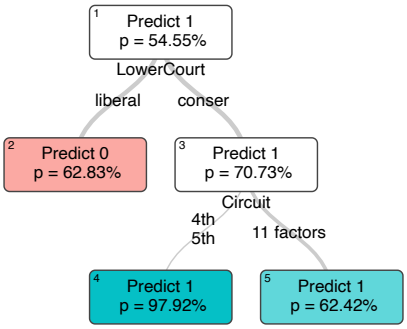
```
In-Sample Accuracy = 0.6692
In-Sample AUC = 0.669
Out-of-Sample Accuracy = 0.6647
Out-of-Sample AUC = 0.6645
```

In [9]:
```
lnr = IAI.OptimalTreeClassifier(random_seed=15095, criterion=:entropy, max_
grid = IAI.GridSearch(lnr)
IAI.fit!(grid, train_X, train_y, validation_criterion= :auc)
lnr = IAI.get_learner(grid)
```

```
┌ Warning: ProgressMeter by default refresh meters with additional inform
ation in IJulia via `IJulia.clear_output`, which clears all outputs in th
e cell.
│   - To prevent this behaviour, do `ProgressMeter.ijulia_behavior(:appen
d)`.
│   - To disable this warning message, do `ProgressMeter.ijulia_behavior(:
clear)`.
└ @ ProgressMeter /Users/iai/builds/InterpretableAI/SystemImage/SysImgBui
lder/.julia/packages/ProgressMeter/Vf8un/src/ProgressMeter.jl:620
Refitting with best parameters...  100%|████████████████████| Time: 0:00:00
4m  Parameters:  cp=>0.03886
```

Out[9]:

Collapse Expand Save PNG



In [10]:
```
println("In-Sample Accuracy = ", round(IAI.score(lnr, train_X, train_y, cri
println("In-Sample AUC = ", round(IAI.score(lnr, train_X, train_y, criterio
println("Out-of-Sample Accuracy = ", round(IAI.score(lnr, test_X, test_y, c
println("Out-of-Sample AUC = ", round(IAI.score(lnr, test_X, test_y, criter
```

```
In-Sample Accuracy = 0.6692
In-Sample AUC = 0.7034
Out-of-Sample Accuracy = 0.6647
Out-of-Sample AUC = 0.6676
```

b)

Neither accuracy nor AUC converged on a stable optimal value or provided smooth functions. However, both metrics had local optimal values at a depth of twenty-five. Based on only that

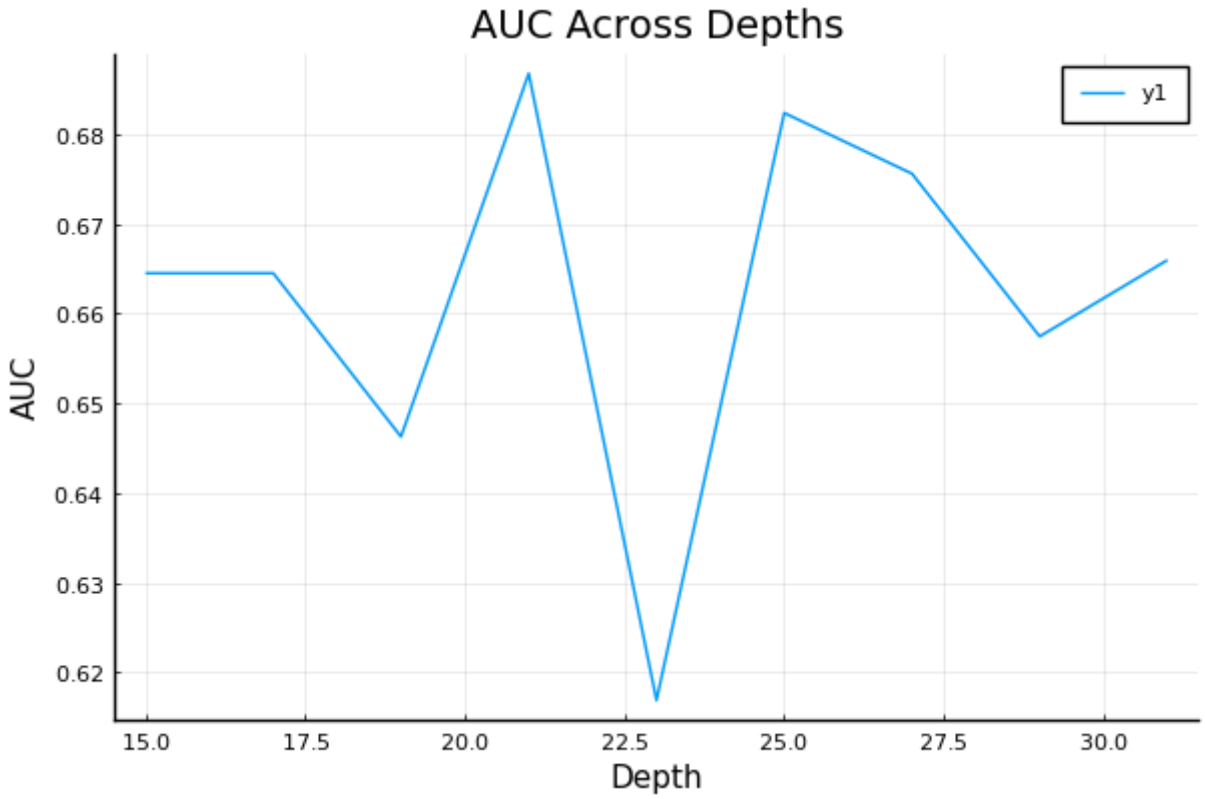
information, I would likely select a depth of twenty-five for the tree. On the other hand, the complexity parameter converged to a very small value once it reached a depth of 19.

In [62]:
```
auc = []
accuracy = []
cp = []
dep =[15,17,19,21,23,25,27,29,31]

for i=1:length(dep)
    lnr = IAI.OptimalTreeClassifier(random_seed=15095, criterion=:gini, max
    grid = IAI.GridSearch(lnr)
    IAI.fit_cv!(grid, train_X, train_y, n_folds=5)
    lnr = IAI.get_learner(grid)

    append!(auc,IAI.score(lnr,test_X, test_y,criterion=:auc))
    append!(accuracy,IAI.score(lnr,test_X, test_y,criterion=:misclassificat
    append!(cp,lnr.cp)
end
```

```
┌ Warning: ProgressMeter by default refresh meters with additional inform
ation in IJulia via `IJulia.clear_output`, which clears all outputs in th
e cell.
│   - To prevent this behaviour, do `ProgressMeter.ijulia_behavior(:appen
d)`.
│   - To disable this warning message, do `ProgressMeter.ijulia_behavior(:
clear)`.
└ @ ProgressMeter /Users/iai/builds/InterpretableAI/SystemImage/SysImgBui
lder/.julia/packages/ProgressMeter/Vf8un/src/ProgressMeter.jl:620
Refitting with best parameters...  100%|████████████████| Time: 0:00:02
4m  Parameters:  cp=>0.0009074
```

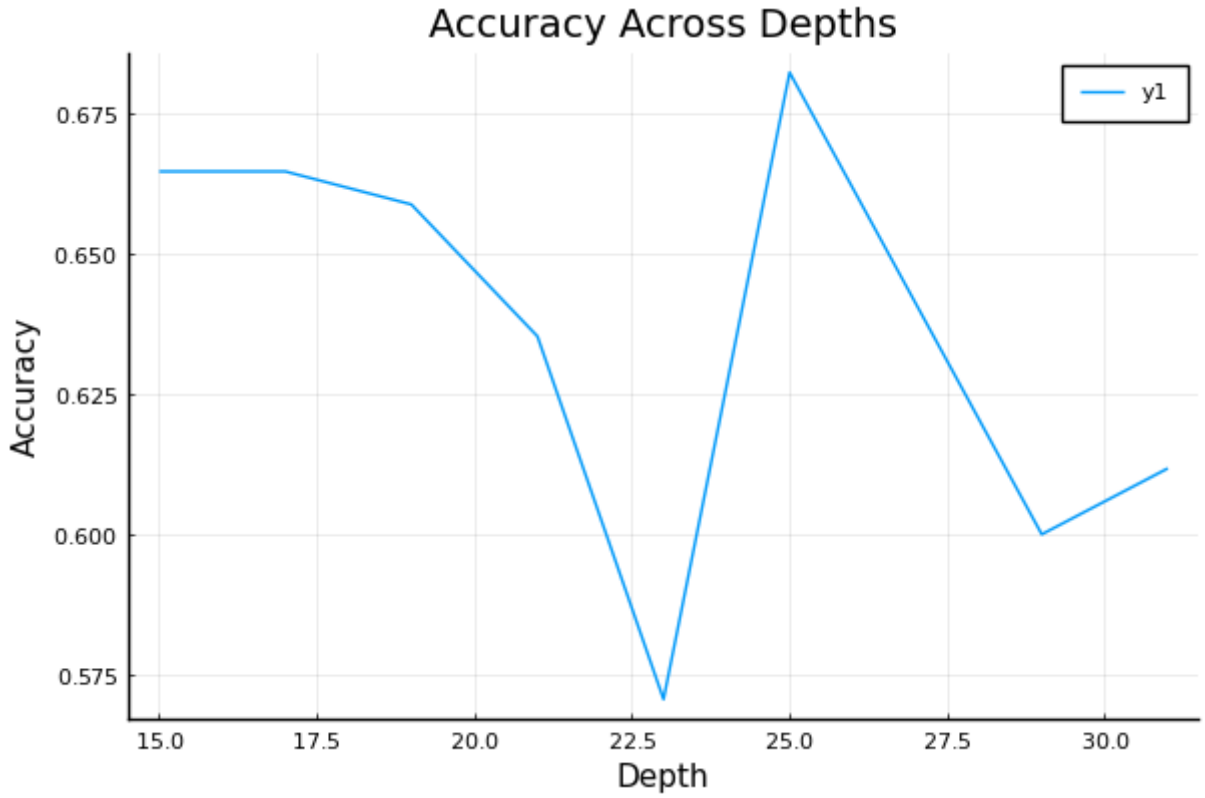
In [63]:
```
pyplot()
plot(dep, auc, title = "AUC Across Depths", ylabel = "AUC", xlabel = "Depth
```

Out[63]:



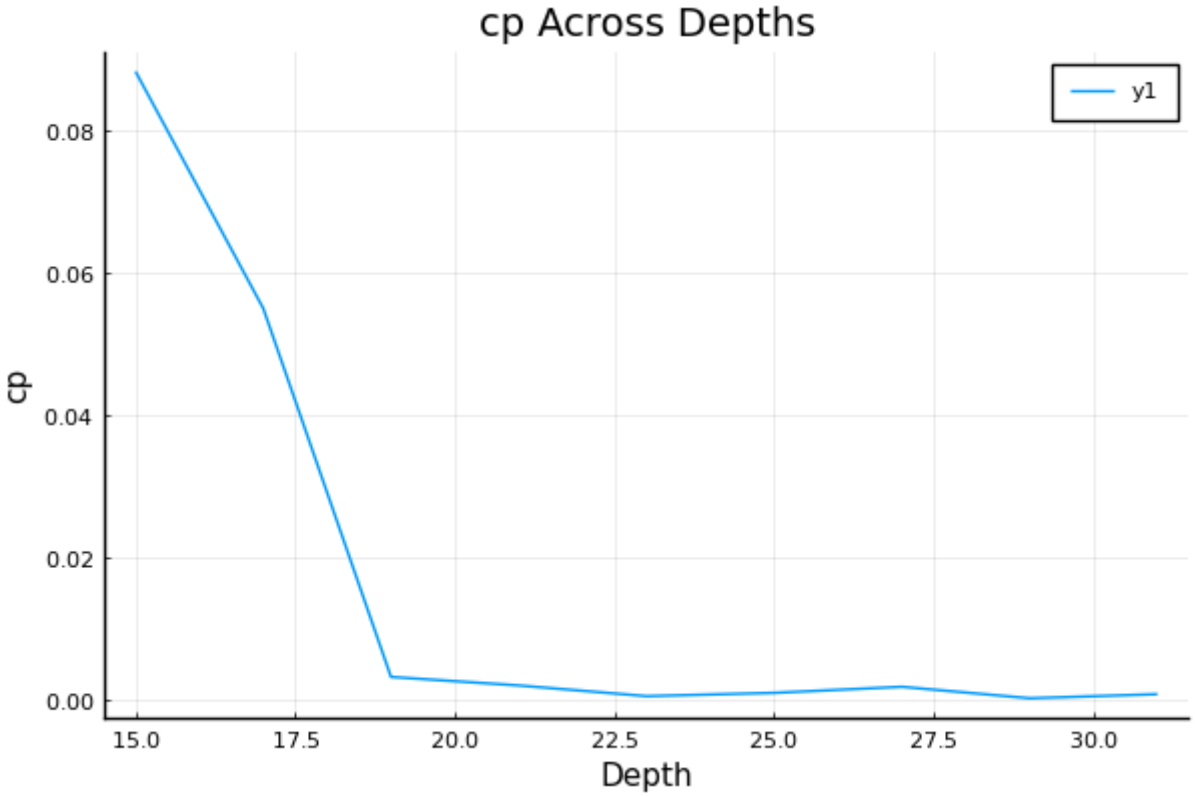
In [64]:
```
pyplot()
plot(dep, accuracy, title = "Accuracy Across Depths", ylabel = "Accuracy",
```

Out[64]:



In [65]:
```
plot(dep, cp, title = "cp Across Depths", ylabel = "cp", xlabel = "Depth")
```

Out[65]:



c)

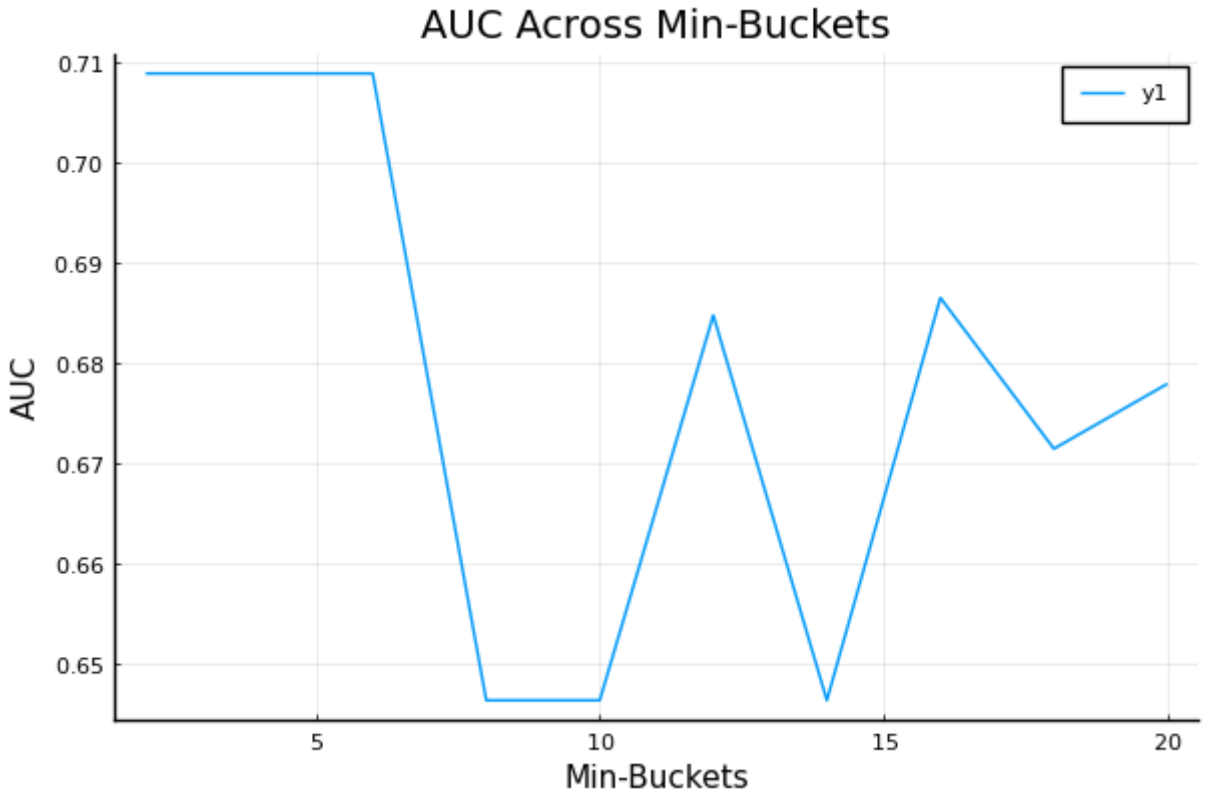
Both out-of-sample accuracy and AUC have their highest values with the lowest threshold for minimum number of observations per bucket. In an effort to prevent overfitting, I would select the highest value of min-bucket that still optimizes accuracy and AUC. In this case, I would select 6 for min-buckets. In this case, the complexity parameter converged to a very small value for low values of min-bucket.

In [15]:
```
auc = []
accuracy = []
cp = []
min_buck =[2, 4, 6, 8, 10, 12, 14, 16, 18, 20]

for i=1:length(dep)
    lnr = IAI.OptimalTreeClassifier(random_seed=15095, criterion=:gini, max
    grid = IAI.GridSearch(lnr)
    IAI.fit_cv!(grid, train_X, train_y, n_folds=5)
    lnr = IAI.get_learner(grid)

    append!(auc,IAI.score(lnr,test_X, test_y,criterion=:auc))
    append!(accuracy,IAI.score(lnr,test_X, test_y,criterion=:misclassificat
    append!(cp,lnr.cp)
end
```

```
┌ Warning: ProgressMeter by default refresh meters with additional inform
ation in IJulia via `IJulia.clear_output`, which clears all outputs in th
e cell.
│  - To prevent this behaviour, do `ProgressMeter.ijulia_behavior(:appen
d)`.
│  - To disable this warning message, do `ProgressMeter.ijulia_behavior(:
clear)`.
└ @ ProgressMeter /Users/iai/builds/InterpretableAI/SystemImage/SysImgBui
lder/.julia/packages/ProgressMeter/Vf8un/src/ProgressMeter.jl:620
Refitting with best parameters...  100%|████████████████| Time: 0:00:00
4m  Parameters:  cp=>0.006245
```

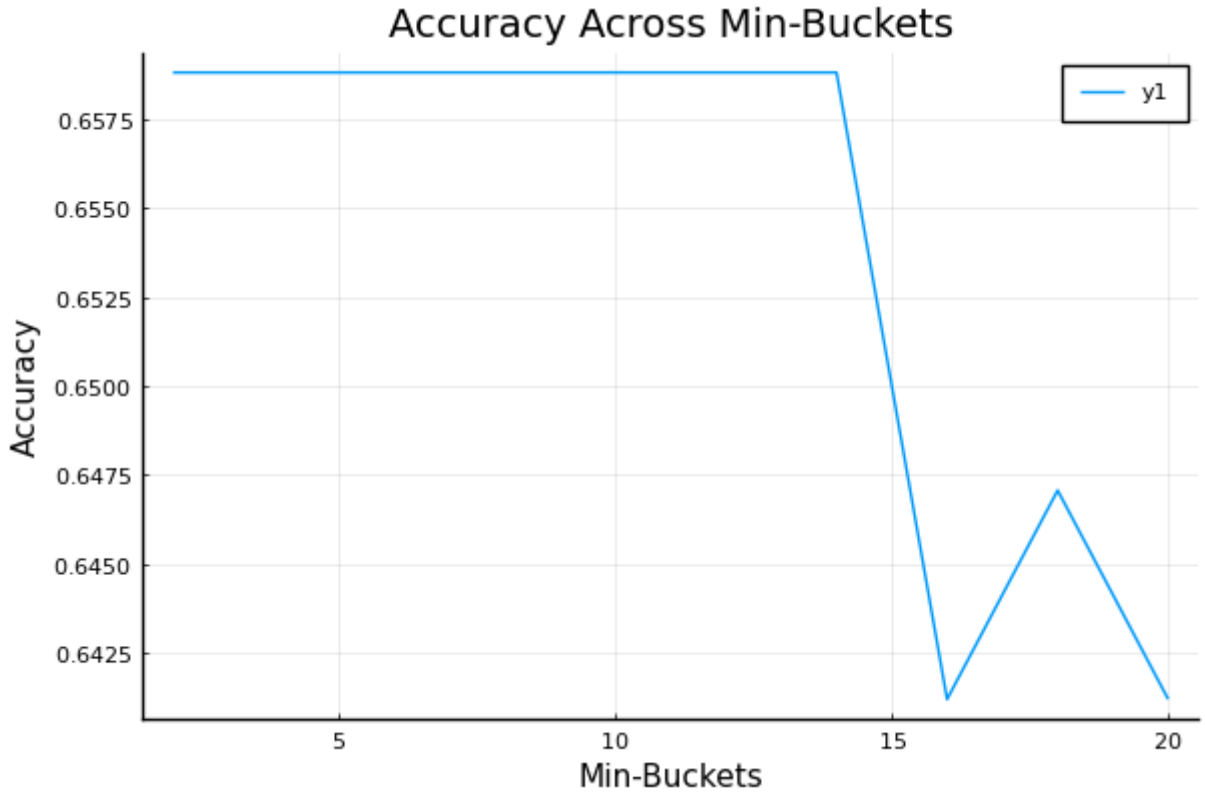
In [16]:
```
pyplot()
plot(min_buck, auc, title = "AUC Across Min-Buckets", ylabel = "AUC", xlabe
```

Out[16]:



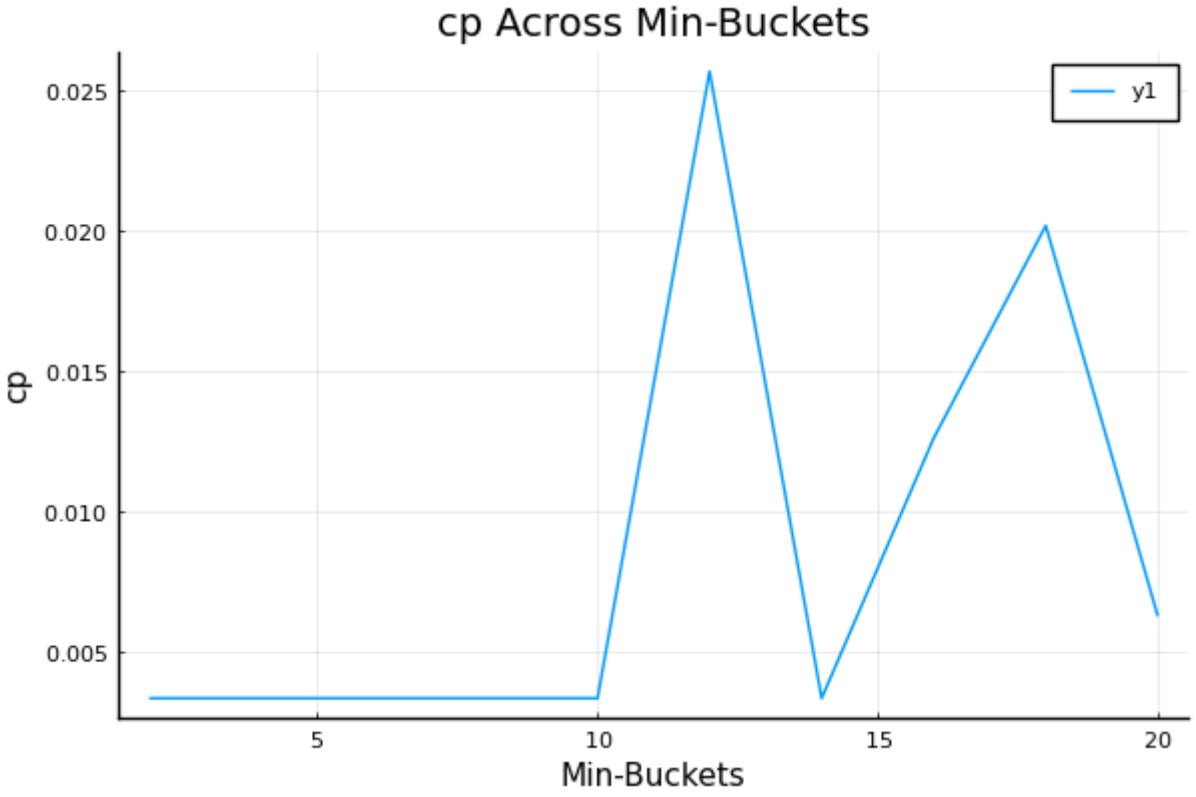
In [17]:
```
pyplot()
plot(min_buck, accuracy, title = "Accuracy Across Min-Buckets", ylabel = "A
```

Out[17]:



In [18]:
```
plot(min_buck, cp, title = "cp Across Min-Buckets", ylabel = "cp", xlabel =
```

Out[18]:



d)

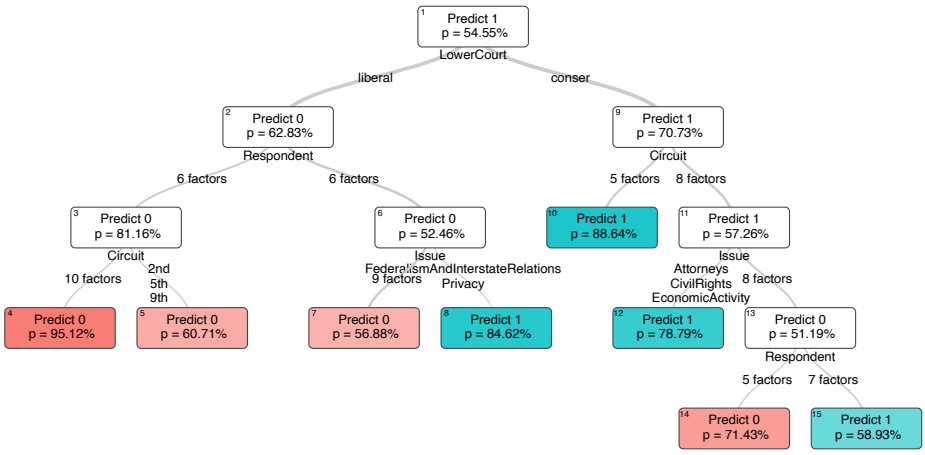
CART

In [19]:
```
lnr = IAI.OptimalTreeClassifier(random_seed=15095, localsearch=false, crite
grid = IAI.GridSearch(lnr, max_depth=[2,4], minbucket=5:10)
IAI.fit_cv!(grid, train_X, train_y, n_folds=5)
lnr = IAI.get_learner(grid)
```

```
┌ Warning: ProgressMeter by default refresh meters with additional inform
ation in IJulia via `IJulia.clear_output`, which clears all outputs in th
e cell.
│   - To prevent this behaviour, do `ProgressMeter.ijulia_behavior(:appen
d)`.
│   - To disable this warning message, do `ProgressMeter.ijulia_behavior(:
clear)`.
└ @ ProgressMeter /Users/iai/builds/InterpretableAI/SystemImage/SysImgBui
lder/.julia/packages/ProgressMeter/Vf8un/src/ProgressMeter.jl:620
Refitting with best parameters...  100%|████████████████████| Time: 0:00:00
4m  Parameters:  minbucket=>7 cp=>0.01398 max_depth=>4
```

Out[19]:

Collapse Expand Save PNG



In [20]:
```
println("Out-of-Sample Accuracy = ", round(IAI.score(lnr, test_X, test_y, c
println("Out-of-Sample AUC = ", round(IAI.score(lnr, test_X, test_y, criter
```

```
Out-of-Sample Accuracy = 0.5941
Out-of-Sample AUC = 0.6115
```

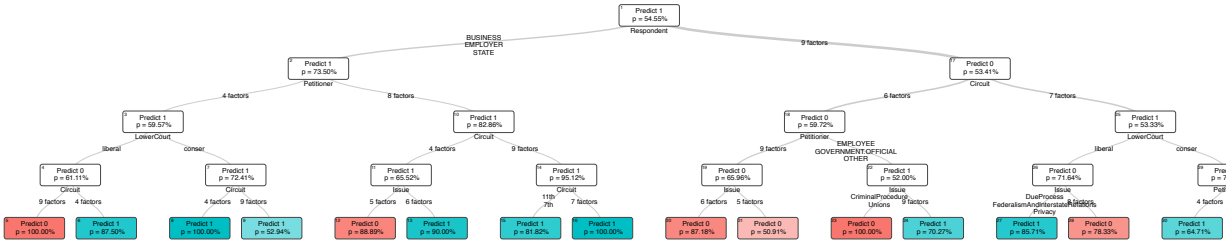
OCT

In [27]:
```
lnr = IAI.OptimalTreeClassifier(random_seed=15095, criterion=:gini)
grid = IAI.GridSearch(lnr, max_depth=[2,4], minbucket=5:10)
IAI.fit_cv!(grid, train_X, train_y, n_folds=5)
lnr = IAI.get_learner(grid)
```

```
┌ Warning: ProgressMeter by default refresh meters with additional inform
ation in IJulia via `IJulia.clear_output`, which clears all outputs in th
e cell.
│   - To prevent this behaviour, do `ProgressMeter.ijulia_behavior(:appen
d)`.
│   - To disable this warning message, do `ProgressMeter.ijulia_behavior(:
clear)`.
└ @ ProgressMeter /Users/iai/builds/InterpretableAI/SystemImage/SysImgBui
lder/.julia/packages/ProgressMeter/Vf8un/src/ProgressMeter.jl:620
Refitting with best parameters...  100%|████████████████████| Time: 0:00:00
4m  Parameters:  minbucket=>7 cp=>0.001371 max_depth=>4
```

Out[27]:

Collapse Expand Save PNG



In [28]:
```
println("Out-of-Sample Accuracy = ", round(IAI.score(lnr, test_X, test_y, c
println("Out-of-Sample AUC = ", round(IAI.score(lnr, test_X, test_y, criter
```

```
Out-of-Sample Accuracy = 0.6412
Out-of-Sample AUC = 0.6533
```

Random Forest

In [23]:
```
lnr = IAI.RandomForestClassifier(random_seed=15095, criterion=:gini)
grid = IAI.GridSearch(lnr, max_depth=[2,4], minbucket=5:10, num_trees=[50,1
IAI.fit_cv!(grid, train_X, train_y, n_folds=5)
lnr = IAI.get_learner(grid)
```

```
┌ Warning: ProgressMeter by default refresh meters with additional inform
ation in IJulia via `IJulia.clear_output`, which clears all outputs in th
e cell.
│   - To prevent this behaviour, do `ProgressMeter.ijulia_behavior(:appen
d)`.
│   - To disable this warning message, do `ProgressMeter.ijulia_behavior(:
clear)`.
└ @ ProgressMeter /Users/iai/builds/InterpretableAI/SystemImage/SysImgBui
lder/.julia/packages/ProgressMeter/Vf8un/src/ProgressMeter.jl:620
Refitting with best parameters...  100%|████████████████| Time: 0:00:00
4m  Parameters:  minbucket=>8 num_trees=>50 max_depth=>4
```

Out[23]:
```
Fitted RandomForestClassifier
```

In [24]:
```
println("Out-of-Sample Accuracy = ", round(IAI.score(lnr, test_X, test_y, c
println("Out-of-Sample AUC = ", round(IAI.score(lnr, test_X, test_y, criter
```

```
Out-of-Sample Accuracy = 0.6882
Out-of-Sample AUC = 0.735
```

Boosted Tree

In [31]:
```
lnr = IAI.XGBoostClassifier(random_seed=15095, criterion=:entropy)
grid = IAI.GridSearch(lnr, max_depth=[2,4,6], minbucket=5:10, num_estimator
IAI.fit_cv!(grid, train_X, train_y, n_folds=5)
lnr = IAI.get_learner(grid)
```

```
┌ Warning: ProgressMeter by default refresh meters with additional inform
ation in IJulia via `IJulia.clear_output`, which clears all outputs in th
e cell.
│   - To prevent this behaviour, do `ProgressMeter.ijulia_behavior(:appen
d)`.
│   - To disable this warning message, do `ProgressMeter.ijulia_behavior(:
clear)`.
└ @ ProgressMeter /Users/iai/builds/InterpretableAI/SystemImage/SysImgBui
lder/.julia/packages/ProgressMeter/Vf8un/src/ProgressMeter.jl:620
Refitting with best parameters...  100%|████████████████| Time: 0:00:00
4m  Parameters:  minbucket=>5 num_estimators=>50 max_depth=>2
```

Out[31]:
```
Fitted XGBoostClassifier
```

In [33]:
```
println("Out-of-Sample Accuracy = ", round(IAI.score(lnr, test_X, test_y, c
println("Out-of-Sample AUC = ", round(IAI.score(lnr, test_X, test_y, criter
```

```
Out-of-Sample Accuracy = 0.6412
Out-of-Sample AUC = 0.682
```

Sparse Logistic Regression

In [53]:
```
lnr = IAI.OptimalFeatureSelectionClassifier(random_seed=15095, criterion=:e
grid = IAI.GridSearch(lnr, sparsity=1:10,gamma=[.001,0.01,0.1,0.25,.5,1,2,5
IAI.fit_cv!(grid, train_X, train_y, n_folds=5)
lnr = IAI.get_learner(grid)
```

```
┌ Warning: ProgressMeter by default refresh meters with additional inform
ation in IJulia via `IJulia.clear_output`, which clears all outputs in th
e cell.
│   - To prevent this behaviour, do `ProgressMeter.ijulia_behavior(:appen
d)`.
│   - To disable this warning message, do `ProgressMeter.ijulia_behavior(:
clear)`.
└ @ ProgressMeter /Users/iai/builds/InterpretableAI/SystemImage/SysImgBui
lder/.julia/packages/ProgressMeter/Vf8un/src/ProgressMeter.jl:620
Refitting with best parameters...  100%|████████████████| Time: 0:00:00
4m  Parameters:  gamma=>2 sparsity=>1
```

Out[53]:
```
Fitted OptimalFeatureSelectionClassifier:
  Constant: 0.882576
  Weights:
    LowerCourt==liberal: -1.38754
  (Higher score indicates stronger prediction for class `1`)
```

In [54]:
```
println("Out-of-Sample Accuracy = ", round(IAI.score(lnr, test_X, test_y, c
println("Out-of-Sample AUC = ", round(IAI.score(lnr, test_X, test_y, criter
```

```
Out-of-Sample Accuracy = 0.6647
Out-of-Sample AUC = 0.6645
```

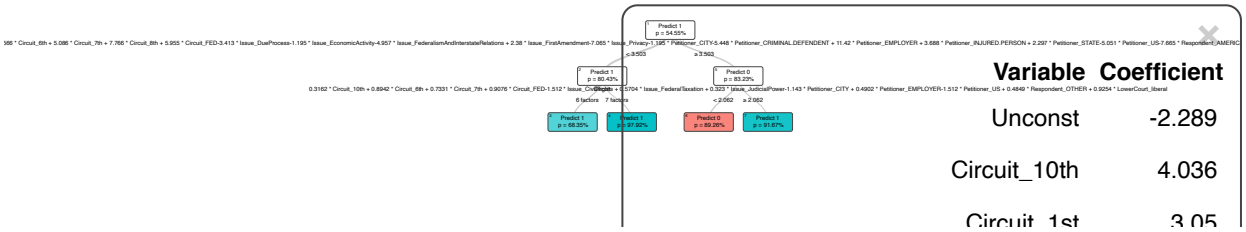
Additional Model: OCT-H

In [55]:
```
lnr = IAI.OptimalTreeClassifier(random_seed=15095, criterion=:gini, hyperpl
grid = IAI.IAI.GridSearch(lnr, max_depth=1:2, cp = cp = [0.001, 0.005] )
IAI.fit_cv!(grid, train_X, train_y, n_folds=5)
lnr = IAI.get_learner(grid)
```

```
┌ Warning: ProgressMeter by default refresh meters with additional inform
ation in IJulia via `IJulia.clear_output`, which clears all outputs in th
e cell.
│   - To prevent this behaviour, do `ProgressMeter.ijulia_behavior(:appen
d)`.
│   - To disable this warning message, do `ProgressMeter.ijulia_behavior(:
clear)`.
└ @ ProgressMeter /Users/iai/builds/InterpretableAI/SystemImage/SysImgBui
lder/.julia/packages/ProgressMeter/Vf8un/src/ProgressMeter.jl:620
Refitting with best parameters...  100%|████████████████████| Time: 0:00:29
4m  Parameters:  cp=>0.005 max_depth=>2
```

Out[55]:

Collapse Expand Save PNG



In [57]:
```
println("Out-of-Sample Accuracy = ", round(IAI.score(lnr, test_X, test_y, c
println("Out-of-Sample AUC = ", round(IAI.score(lnr, test_X, test_y, criter
```

```
Out-of-Sample Accuracy = 0.6706
Out-of-Sample AUC = 0.6593
```

15.095 Homework #4 Executive Summary

Hyper-parameter tuning methodology: In general, hyperparameters were first tuned for a few points over a large range. Once a general region of optimal values was found, a more granular search would be applied in that region. This was an effort to find a globally optimal value without impractical computation time.

Model 1 *(CART):*

Out-of-sample Accuracy: 0.5941, Out-of-sample AUC: 0.6115

Variables Used: The first three variables split on in the tree was the political leaning of the lower court, which district court it presided in, and the respondent type.

Model 2 *(OCT):*

Out-of-sample Accuracy: 0.6412, Out-of-sample AUC: 0.6533

Classification Trees Interpretability: Typically, CART & OCTs are viewed as some of the most interpretable models. However, given the data set is only multi-factor categorical variables, the typical interpretability suffers. No longer can you easily convey a simple tree to a decision maker.

Variables Used: The first three variables split on in the tree was the respondent type, petitioner type, and which circuit court the case was in previously.

Model 3 *(Random Forest):*

Out-of-sample Accuracy: 0.6882, Out-of-sample AUC: 0.735

Random Forest Interpretability: This method provides no easy visual to a decision-maker and thus is not very interpretable. However, one could relay the concept in layman's terms as "the average of many decision trees", with mixed success.

Variables Used: The random forest inevitably used every variable because it randomly selects a subset of variables for each tree in the forest.

Model 4 *(XGBoost):*

Out-of-sample Accuracy: 0.6412, Out-of-sample AUC: 0.6820

XGBoost Interpretability: This model is even less interpretable than random forests because there is no easy way to explain the prediction of errors to a non-data scientist.

Variables Used: Due to the large number of trees XGBoost leverages, it almost inevitably used all the variables. However, one could extract the most important features.

Model 5 *(Sparse Logistic Regression):*

Out-of-sample Accuracy: 0.6647, Out-of-sample AUC: 0.6645

Sparse Logistic Regression Interpretability: This model has slightly better interpretability compared to random forests because sparsity allows one to communicate to a decision maker

the important variables. The regression easily conveys information through coefficient's respective signs, but not interpretable values.

Variables Used: The model was tuned to only have one variable so the only one used was whether or not the lower court was liberal.

Model 6 *(OCT-H):*

Out-of-sample Accuracy: 0.6706, Out-of-sample AUC: 0.6593

Additional Model Selection: The final model selected was optimal classification trees with hyperplanes because it can achieve similar results to XGBoost with a tree of small depth, and thus increased interpretability.

OCT-H Interpretability: OCT-H achieve slightly less explainability compared to CART, which places it near the top of models in terms of interpretability.

Variables Used: OCT-H issued a combination of numerous dummy variables for levels of different categorical variables in this particularly unsparse OCT-H.

Model Selection Process: Since the most interpretable models, trees, are hardly interpretable because of the multi-class categorical variables, I decided the most important aspect in the model was out-of-sample performance. Thus, I would select the random forest model due to its accuracy and the fact that whichever model I select in this scenario, there will be substantial effort in explaining it. The biggest downside of a random forest as opposed to a difficulty interpretable CART is the lack of a visual. However, in this case I think performance trumps a confusing visual.