Decision Trees with Spatial Data

Mariam Walaa

Outline

- Decision Trees
  - What are they?
  - Why are they used?
  - What are the benefits of using them?
- How Decision Trees Work
  - Interpreting a Tree
  - Splitting Metrics
  - Variable Importance
- Land Use Classification
  - Background of the data
  - R code walk-through
    - Prepare the data
    - Create training labels and train model
    - Evaluate the model

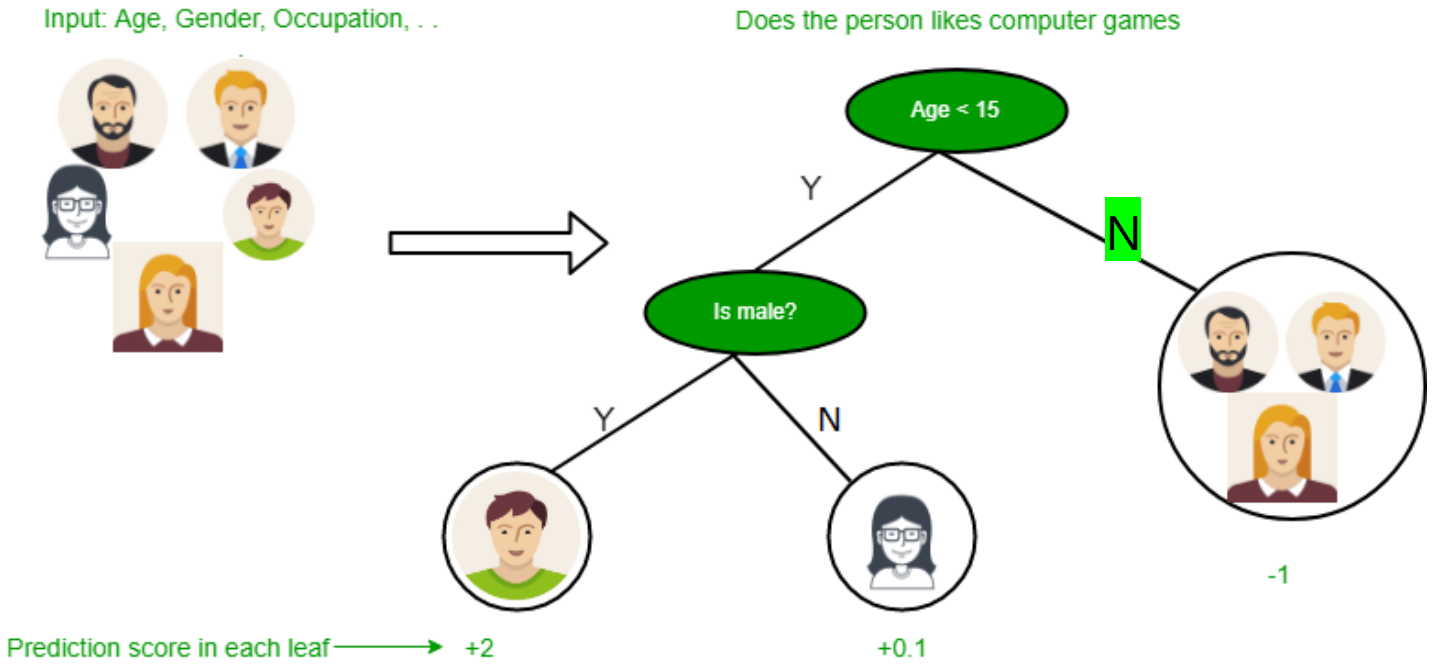
Decision Trees

What are they?

- A model for predicting categorical or continuous target variables.

How do they work?

- Each green node represents an independent variable
- Each branch represents an outcome for that independent variable
- Each value at a final (i.e. terminal) node represents the "score" or "percentage likelihood" of the observation in hand satisfying the target variable category



Source: https://www.geeksforgeeks.org/decision-tree-introduction-example/

Decision Trees

Why are they used?

- Because we want to make a model that predicts some target variable
- We have several independent variables that may be predictive of this target variable
- We want to predict a target variable that is categorical or continuous

What are the benefits of using them?

- They are easy to visualize
- They are easy to understand

Land Use Classification

Data

- Landsat 8 satellite imagery created with 11 raster bands for Calgary, Alberta, Canada
- Data made available by urbanSpatial: https://github.com/urbanSpatial/classifying_satellite_imagery_in_R/tree/master/data

Task

- Predict **Land Use** based on the spectral profiles of point locations in Calgary
- Exercise based on tutorial by urbanSpatial provided here: https://urbanspatial.github.io/classifying_satellite_imagery_in_R/

Data Background

Satellite Imagery and Spectral Bands

- Satellite imagery are images of Earth captured by satellites
- Satellite Imagery captures different surfaces differently
- We can account for these differences through compositions of multiple spectral bands of satellite imagery
- Here is a list of the spectral bands we'll be using as **independent variables** in the model:

Band	Description
Band 1	Coastal aersol
Band 2	Blue
Band 3	Green
Band 4	Red
Band 5	Near Infrared (NIR)
Band 6	Shortwave Infrared (SWIR) 1
Band 7	Shortwave Infrared (SWIR) 2
Band 8	Panchromatic
Band 9	Cirrus
Band 10	Thermal Infrared (TIRS) 1
Band 11	Thermal Infrared (TIRS) 2

Source: https://urbanspatial.github.io/classifying_satellite_imagery_in_R/

Data Background

Satellite Imagery and Spectral Bands

- We can use these spectral bands to perform various analyses, but we need to simplify these spectral profiles further
- We can do this simplification by classifying the spectral bands into categories called **Land Use** categories
- We can use a **Decision Tree** to perform this classification

Area of Study: Calgary

- This is a map of Calgary using the 11 raster bands
- We will train and build a decision tree that classifies this area into 3 **Land Use** categories:
  - Developed Land
  - Undeveloped Land
  - Water

R code walk-through

Step 1: Prepare The Data

```r
# Load 11 raster bands
band1 <- raster("./data/band1.tif")
band2 <- raster("./data/band2.tif")
band3 <- raster("./data/band3.tif")
band4 <- raster("./data/band4.tif")
band5 <- raster("./data/band5.tif")
band6 <- raster("./data/band6.tif")
band7 <- raster("./data/band7.tif")
band8 <- raster("./data/band8.tif")
band9 <- raster("./data/band9.tif")
band10 <- raster("./data/band10.tif")
band11 <- raster("./data/band11.tif")

# Aggregate the cell size of band8
band8 <- aggregate(band8, fact = 2)

# Combine all raster bands into one image
image <- stack(band1, band2, band3, band4, band5, band6, band7, band8, band9, band10, band11)
```

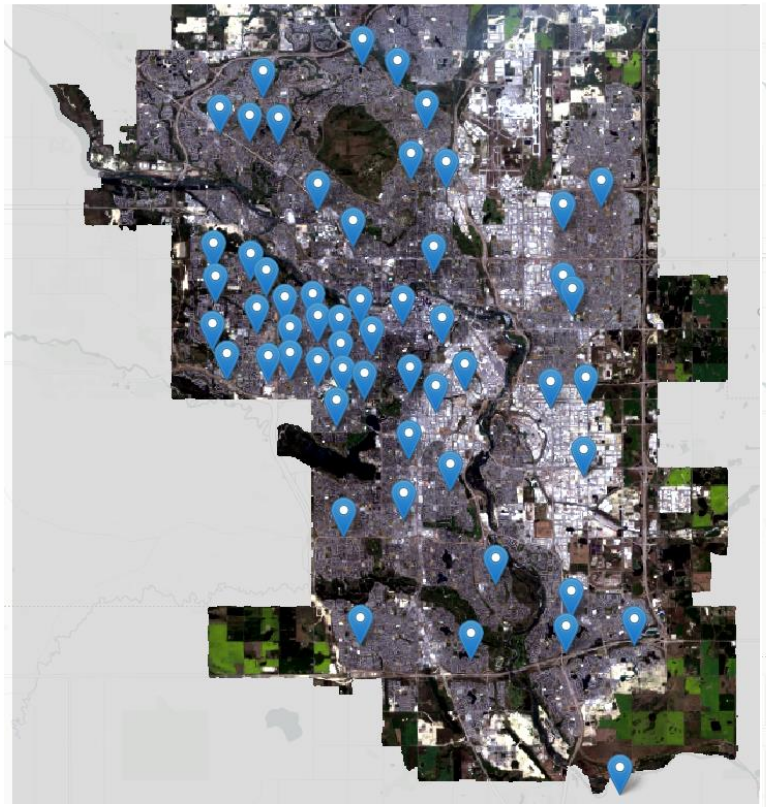
R code walk-through

Step 2: Create Training Labels

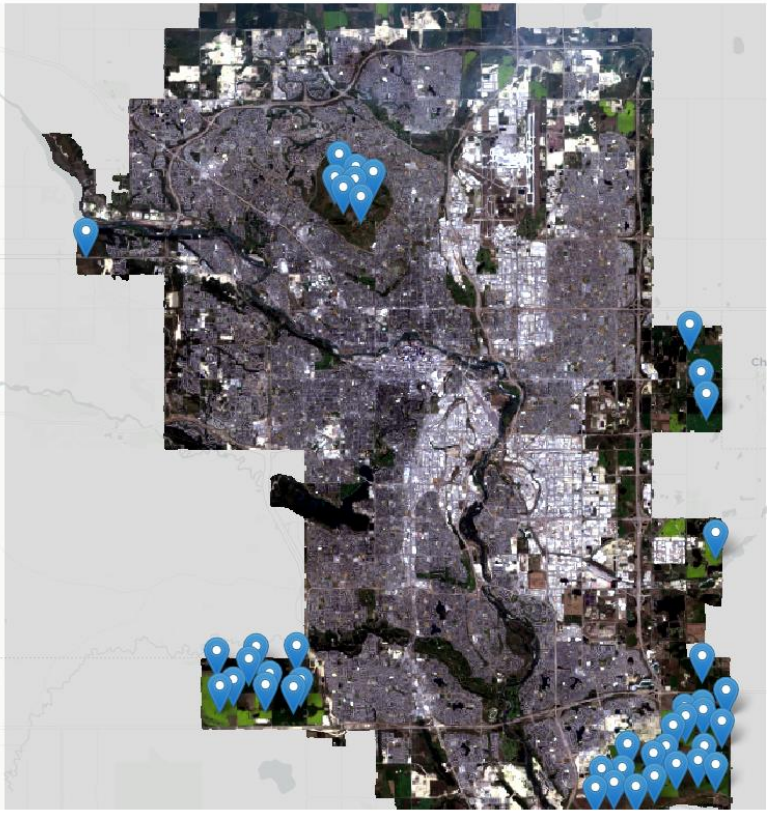
```r
## 1. Developed Land Training Labels
# Launch MapView to manually create training points for Developed Land
points_l <- viewRGB(image, r = 4, g = 3, b = 2) %>% editMap()
# Store training points as Developed Land in a sf data frame
developed <- points_l$finished$geometry %>% st_sf() %>% mutate(class = "developed", id = 1)


## 2. Undeveloped Land Training Labels
# Launch MapView to manually create training points for Undeveloped Land
points_u <- viewRGB(image, r = 4, g = 3, b = 2) %>% editMap()
# Store training points as Undeveloped Land in a sf data frame
undeveloped <- points_u$finished$geometry %>% st_sf() %>% mutate(class = "undeveloped", id = 2)


## 3. Water Training Labels
# Launch MapView to manually create training points for Water
points_w <- viewRGB(image, r = 4, g = 3, b = 2) %>% editMap()
# Store training points as Water in a sf data frame
water <- points_w$finished$geometry %>% st_sf() %>% mutate(class = "water", id = 3)
```

Creating Training Points: Side-by-Side Comparison

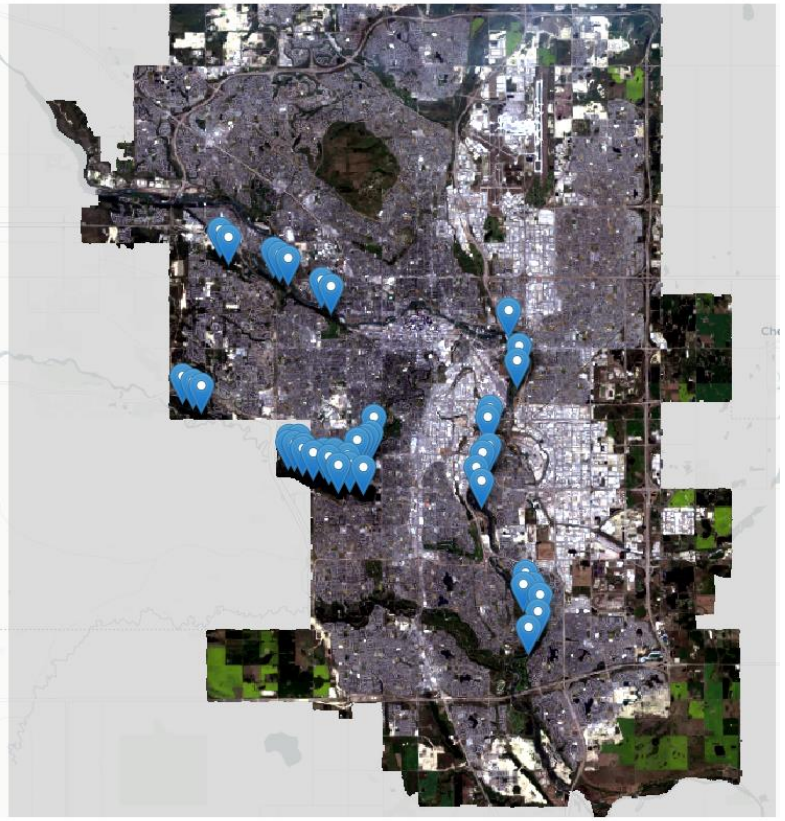
Developed Land Labels



Undeveloped Land Labels



Water Labels



Dataset Summary

```
> head(df)
  training_points.class band1 band2 band3 band4 band5 band6 band7 band8 band9 band10 band11
1             developed 10044  9315  8813  8264 15990 12662  9294  8638  5078  29698  27075
2             developed 11567 11141 11368 11943 14467 15099 14078 11048  5126  30305  27538
3             developed  9783  8922  8467  7472 19966 11933  8541  7813  5077  28111  25656
4             developed 10799 10261  9753  9622 17439 15709 11539  9530  5107  30695  27741
5             developed 11707 11168 10420 10596 11571 11960 11566 10399  5079  31852  28759
6             developed 10927 10228  9824  9988 14632 12732 10932  9968  5079  30281  27571
> dim(df)
[1] 150  12
> df %>% group_by(training_points.class) %>% summarise(count = n())
`summarise()` ungrouping output (override with `.groups` argument)
# A tibble: 3 x 2
  training_points.class count
  <chr>                 <int>
1 developed                56
2 undeveloped              45
3 water                    49
```

R code walk-through

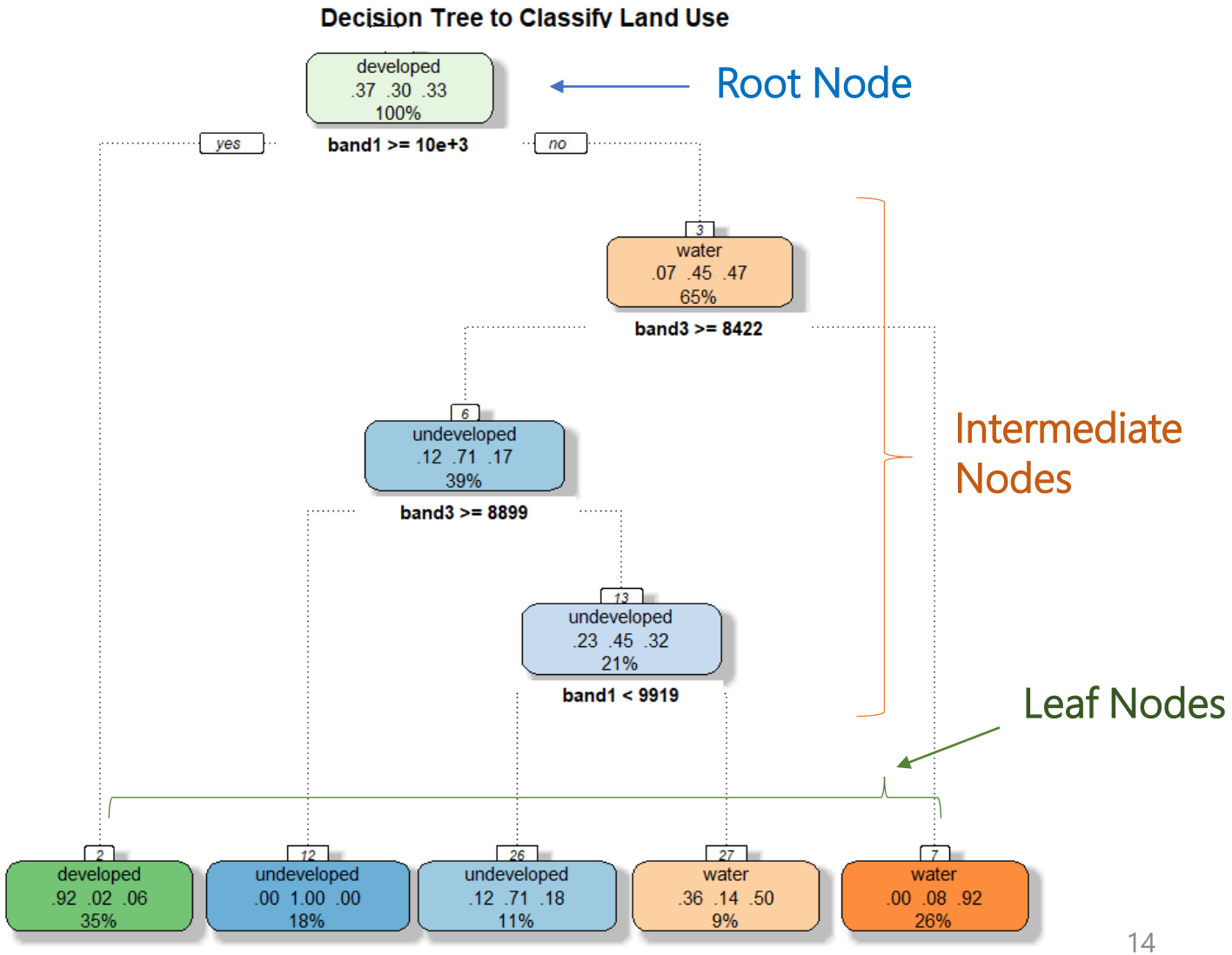
Step 3: Build The Decision Tree

```r
# Create the decision tree
model.class <- rpart(as.factor(training_points.class) ~., # Input labels
                     data = df, # Input data
                     method = 'class') # Method is classification

# Plot the decision tree
fancyRpartPlot(model = model.class, main = "Decision Tree to Classify Land Use")

# Predict the land use for the study area using the trained model
predictions <- predict(object = image, model.class, type = 'class') %>% ratify()

# Plot the predictions on the study area
levels(predictions) <- levels(predictions)[[1]] %>%
  mutate(legend = c("Developed Land", "Undeveloped Land", "Water"))
levelplot(predictions, maxpixels = 1e6, col.regions = c('burlywood', 'darkgreen', 'blue'),
          scales = list(draw = FALSE), main = "Predicted Classification of Land Use on Study Area")

## Create a confusion matrix to evaluate the model predictions
test <- raster::extract(predictions, training_points) %>%
  as.data.frame() %>% rename(id = ".")
testProbs <- data.frame(obs = as.factor(training_points$id),
  pred = as.factor(test$id)) %>%
  mutate(correct = ifelse(obs == pred, 1, 0))
confusionMatrix(testProbs$obs, testProbs$pred)
```

How Decision Tree Works

How can we interpret the tree?

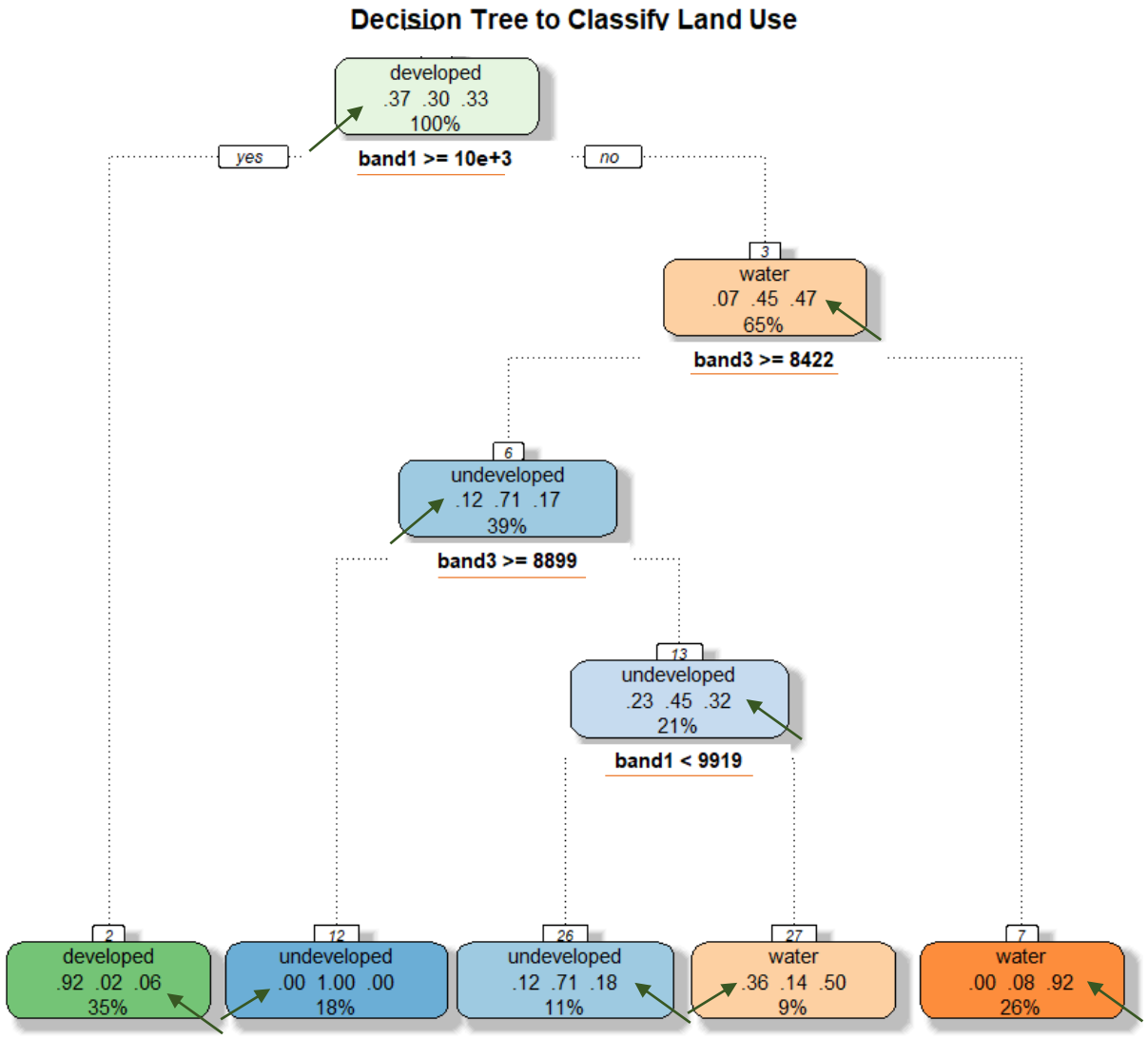
- We can go through this decision tree from top to bottom asking the same questions for each individual observation.
- There are three types of nodes here:
  1. Root Node: The top node of the tree. This node contains the most important variable.
  2. Intermediate Nodes: The middle nodes of the tree. These nodes contain other variables ordered by importance.
  3. Leaf Nodes: These are the bottom nodes of the tree. These determine what category the observation belongs to.



How Decision Tree Works

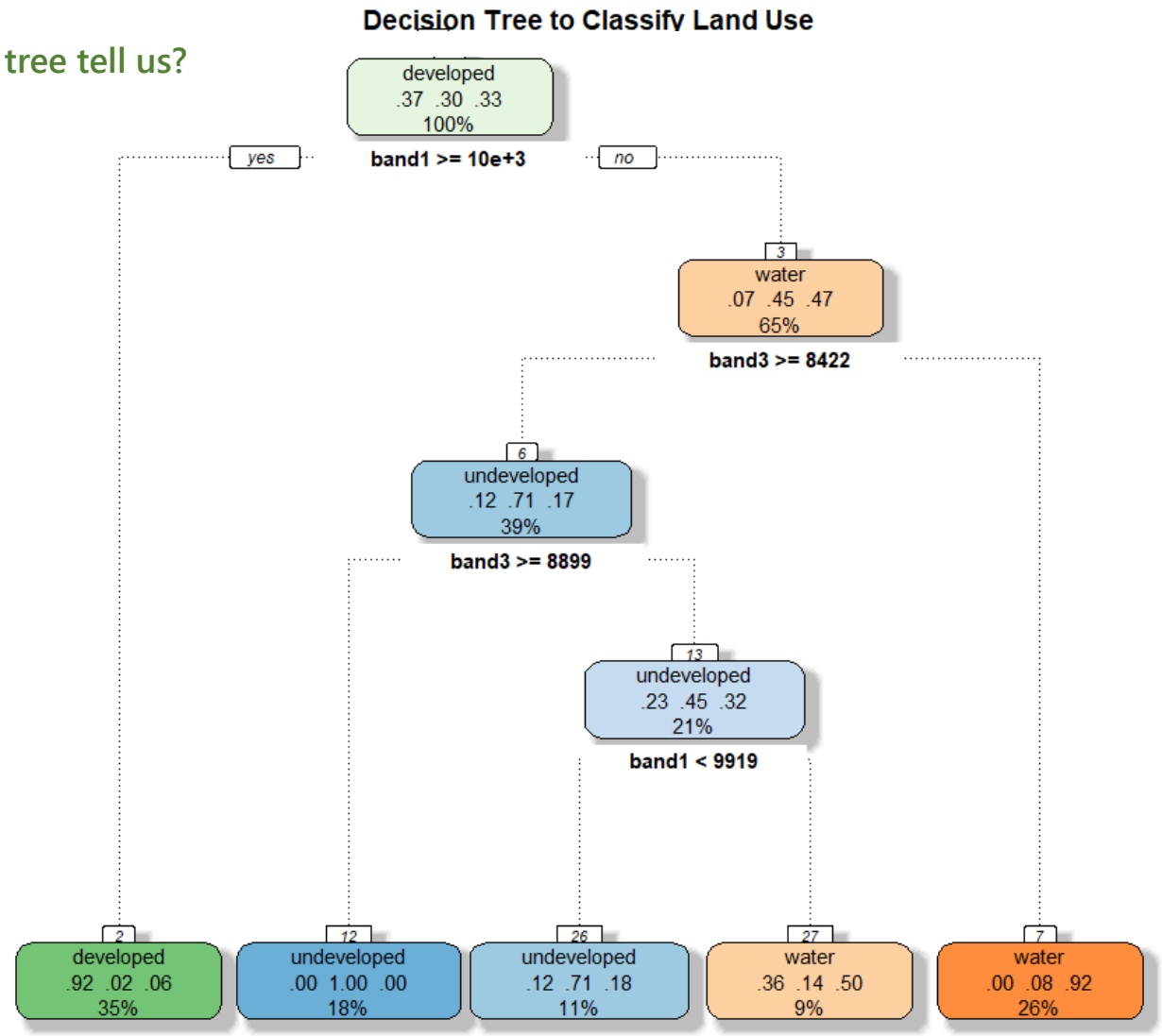
How can we interpret the tree?

- At each node (except for the leaf nodes), there is a condition on the variable that the node is associated with.
  - Each condition has a binary outcome: 'yes' or 'no'
- Each node contains the following information:
  - Name of the variable it represents (stated within the condition statement)
  - Name of the class it predicts at that stage
  - Predicted probabilities for the observation to belong to each of the 3 classes
  - Overall probability that the observation belongs to a specific class



How Decision Tree Works

What does this decision tree tell us?



How Decision Tree Works

How are the splits being made?

- The best splits of a variable are determined by splitting a variable multiple times, computing a measure of information, and determining which split provides the most information from the variable. There are metrics for evaluating splits:
  - Gini Impurity
  - Information Gain
- In summary: The more balanced the probability of each class in a node, the more impure that node is.
  - **Example:** If there is a 33% chance of being Developed Land, 33% chance of being Undeveloped Land, and 33% chance of being Water at a certain node, it is an impure node

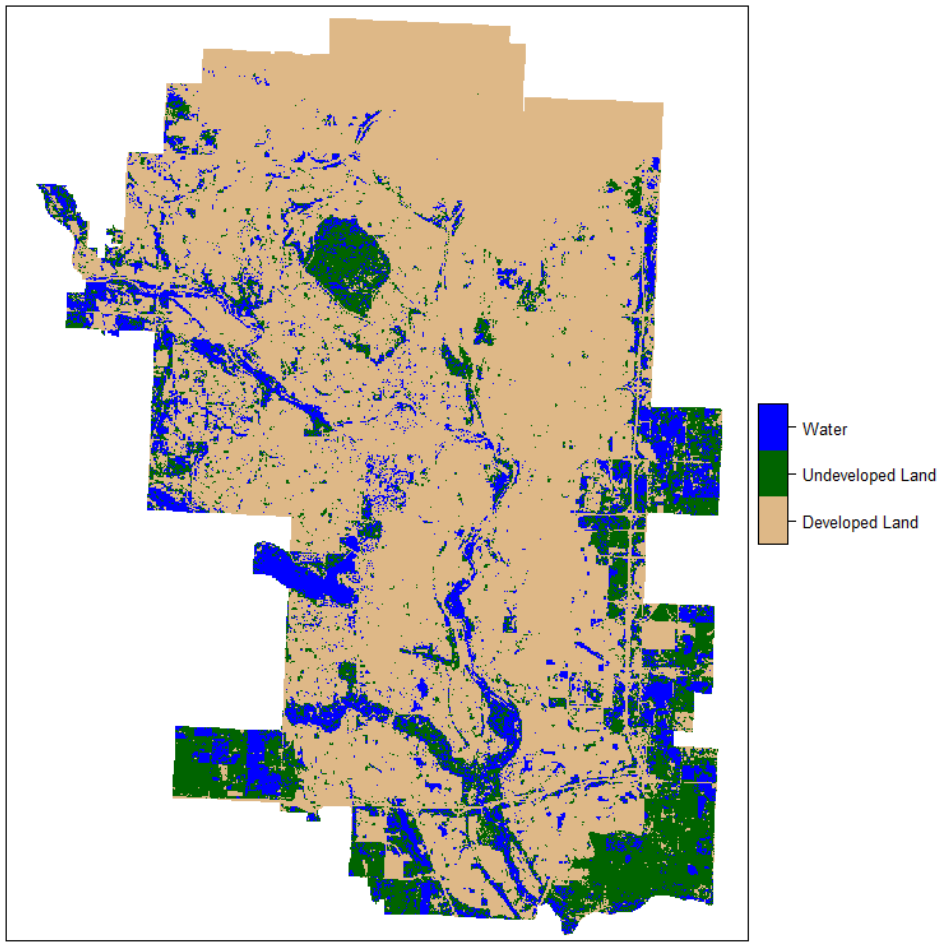
How is the order of independent variables determined?

- The model ranks independent variables by a measure of variable importance to the target variable
- The higher the independent variable appears in the tree, the more important the variable is to the model

Decision Tree Predictions: Side-By-Side Comparison

True Color Composite

Predicted Classification of Land Use on Study Area



Decision Tree Evaluation

Evaluation Summary:

- A **confusion matrix** with positive values only on the diagonal entries indicates 100% accuracy for each class.

  That is not the case here, but we can observe the following:

  - 56 observations belong to class 1, but only 49 points were predicted to be of class 1 (developed).
  - 45 observations belong to class 2, but only 39 points were predicted to be of class 2 (undeveloped).
  - 49 observations belong to class 3, but only 43 points were predicted to be of class 3 (water).
- A **Mcnemar's Test P-value** of 0.7212 indicates there is not enough evidence to reject the null hypothesis that there is no difference between the predictions and the true observations.

```
Confusion Matrix and Statistics

          Reference
Prediction  1  2  3
         1 49  2  5
         2  1 39  5
         3  3  3 43

Overall Statistics

               Accuracy : 0.8733
                 95% CI : (0.8093, 0.922)
    No Information Rate : 0.3533
    P-Value [Acc > NIR] : <0.0000000000000002

                  Kappa : 0.8094

 Mcnemar's Test P-Value : 0.7212

Statistics by Class:

                     Class: 1 Class: 2 Class: 3
Sensitivity            0.9245   0.8864   0.8113
Specificity            0.9278   0.9434   0.9381
Pos Pred Value         0.8750   0.8667   0.8776
Neg Pred Value         0.9574   0.9524   0.9010
Prevalence             0.3533   0.2933   0.3533
Detection Rate         0.3267   0.2600   0.2867
Detection Prevalence   0.3733   0.3000   0.3267
Balanced Accuracy      0.9262   0.9149   0.8747
```

Model Limitations and Considerations

- The accuracy of the model in comparison with the true composite **depends on how well the training points were created**. A high model accuracy of 87% may not mean that the "true accuracy" of the model is 87% as we can see from the side-by-side comparison.
- **The number of training data points** we have may affect how good the model is at making predictions. Also**, the number of training data points for each class** may affect how good the model is at making predictions (i.e. class imbalance).

References

- https://urbanspatial.github.io/classifying_satellite_imagery_in_R/
- https://en.wikipedia.org/wiki/Decision_tree_learning
- https://en.wikipedia.org/wiki/Satellite_imagery
- https://en.wikipedia.org/wiki/McNemar's_test
- https://www.geeksforgeeks.org/decision-tree-introduction-example/
- https://towardsdatascience.com/decision-trees-explained-3ec41632ceb6