Training and Testing Hazard Parent Nodes: Wind and Rain

```
# Environment: Clear working environment
rm(list = ls())

# load libraries
library(rpart)
library(here)
library(rpart.plot)
library(caret)
library(ggplot2)
```

Reusable functions

```
rmse <- function(actual, predicted) {
  sqrt(mean((actual - predicted)^2))
}
```

Inputs

```
# inputs
base_train <- read.csv(here("data", "base_train.csv"))
base_validation <- read.csv(here("data", "base_validation.csv") )
base_test <- read.csv(here("data", "base_validation.csv") )
```

```
# Combining train and validation datasets to one
# Because we are going to use CV to train the models later
# naming it df_base_train2 to remain consistent with df naming
df_base_train2  <- rbind(base_train, base_validation)

cat("number of rows in combined train data:", nrow(df_base_train2), sep = " ")
```

```
## number of rows in combined train data: 7184
```

Wind Model Training & Testing

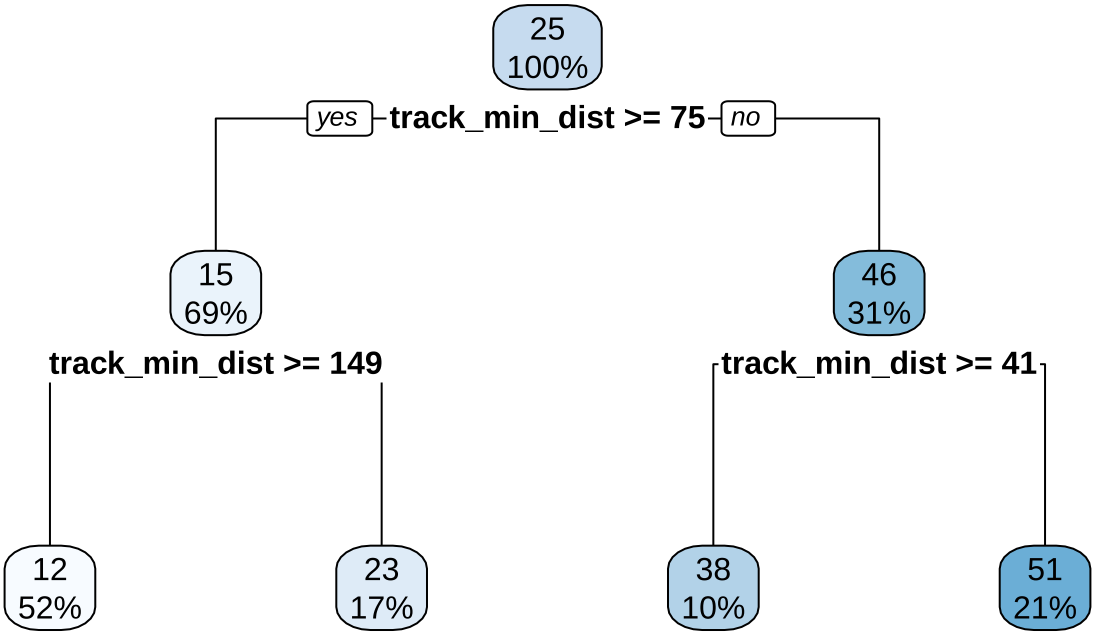
Decision trees

```
base_wind_model <- rpart(wind_max ~ track_min_dist + island_groups,
                         data = df_base_train2,
                         method = "anova")
```

```
dec_wind_pred <- predict(base_wind_model, base_test)

rmse_dec_wind_pred <- rmse(actual = df_base_train2$wind_max,
                           predicted = dec_wind_pred)

cat("rmse of decision tree of wind model", rmse_dec_wind_pred, sep = " ")
```

```
## rmse of decision tree of wind model 20.9515
```

```
rpart.plot(base_wind_model)
```



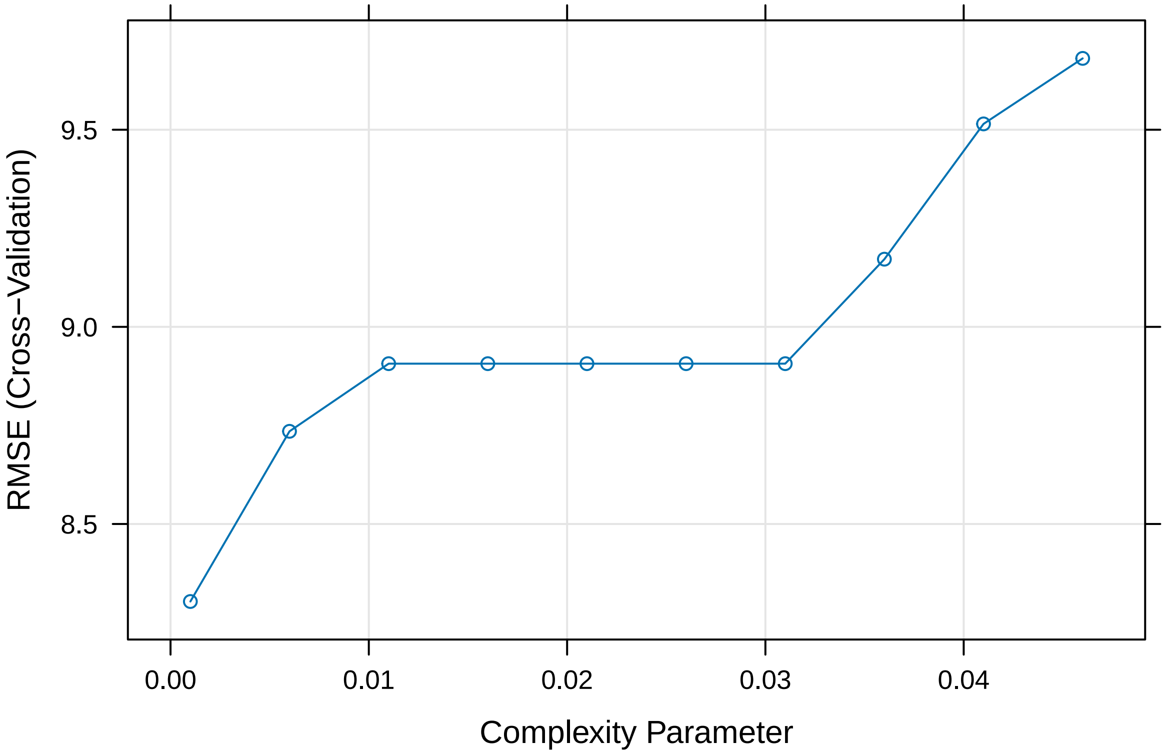
Optimizing For Best parameters

```
set.seed(123)
# Define training control
control <- trainControl(method = "cv", number = 5)

# Set tuning grid
```

```r
grid <- expand.grid(
  cp = seq(0.001, 0.05, by = 0.005)  # try several cp values
)

# Train model
dec_base_wind_model_tuned <- train(
  wind_max ~ track_min_dist + island_groups, data = df_base_train2,
  method = "rpart",
  trControl = control,
  tuneGrid = grid
)

print(dec_base_wind_model_tuned)
```

```
## CART
##
## 7184 samples
##    2 predictor
##
## No pre-processing
## Resampling: Cross-Validated (5 fold)
## Summary of sample sizes: 5748, 5746, 5747, 5747, 5748
## Resampling results across tuning parameters:
##
##   cp     RMSE      Rsquared   MAE
##   0.001  8.303410  0.7865472  5.728446
##   0.006  8.735074  0.7637364  6.015736
##   0.011  8.906656  0.7543552  6.312006
##   0.016  8.906656  0.7543552  6.312006
##   0.021  8.906656  0.7543552  6.312006
##   0.026  8.906656  0.7543552  6.312006
##   0.031  8.906656  0.7543552  6.312006
##   0.036  9.171699  0.7393462  6.460167
##   0.041  9.514772  0.7200191  6.761375
##   0.046  9.681026  0.7097058  6.872794
##
## RMSE was used to select the optimal model using the smallest value.
## The final value used for the model was cp = 0.001.
```

```r
plot(dec_base_wind_model_tuned)
```



```r
png(here("adjusted SCM/outputs", "dec_wind_base_model.png"),
       width = 6,
       units = "in",
       height = 4,
    res = 300)
dec_base_wind_plot <- rpart.plot(dec_base_wind_model_tuned$finalModel)
dev.off()
```

```
## pdf
##   2
```

```r
dec_wind_pred_tuned <- predict(dec_base_wind_model_tuned, base_test)

rmse_dec_wind_pred_tuned <- rmse(actual = df_base_train2$wind_max,
                           predicted = dec_wind_pred_tuned)

cat("rmse of tuned decision tree of wind model", rmse_dec_wind_pred_tuned, sep = " ")
```

```
## rmse of tuned decision tree of wind model 21.13228
```

Linear Regression

```
reg_wind_model <- lm(wind_max ~ track_min_dist + island_groups,
                     data = df_base_train2)

reg_wind_pred <- predict(reg_wind_model, base_test)

rmse_reg_wind_pred <- rmse(actual = df_base_train2$wind_max,
                           predicted = dec_wind_pred)

cat("rmse of decision tree of wind model", rmse_reg_wind_pred, sep = " ")
```

```
## rmse of decision tree of wind model 20.9515
```

Output

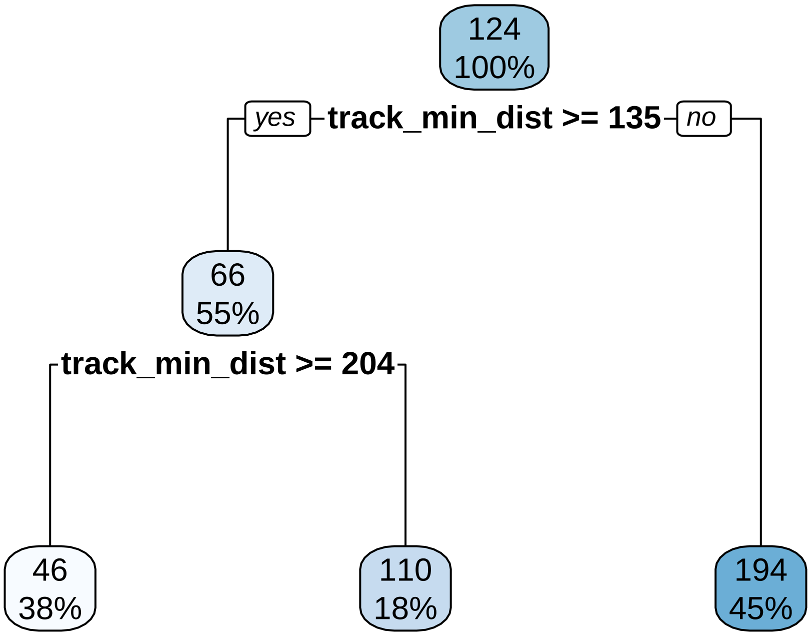
We output the caret tuned decision tree model

```
full_path <- here("adjusted SCM/new base models")
saveRDS(dec_base_wind_model_tuned,
        file = file.path(full_path, paste0("dec_base_wind_model_tuned", ".rds")))
```

Rain Model Training & Testing

```
base_rain_model <- rpart(rain_total ~ track_min_dist + island_groups,
                      data = df_base_train2,
                      method = "anova")
```

```
rpart.plot(base_rain_model)
```



```r
dec_rain_pred <- predict(base_rain_model, base_test)

rmse_dec_rain_pred <- rmse(actual = df_base_train2$rain_total,
                           predicted = dec_rain_pred)

cat("rmse of decision tree of rain model", rmse_dec_rain_pred, sep = " ")
```

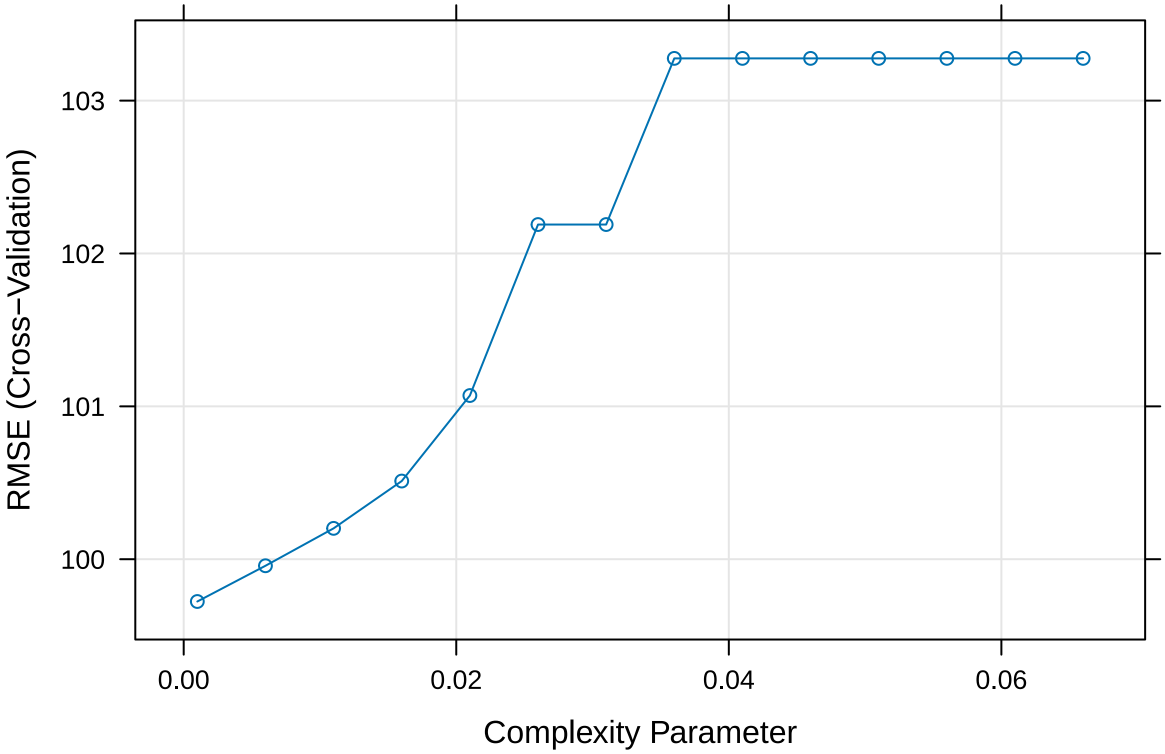
```
## rmse of decision tree of rain model 129.4887
```

Optimizing For Best parameters

```r
set.seed(123)
# Define training control
control <- trainControl(method = "cv", number = 5)

# Set tuning grid
grid <- expand.grid(
  cp = seq(0.001, 0.07, by = 0.005)  # try several cp values
)

# Train model
dec_base_rain_model_tuned <- train(
  rain_total ~ track_min_dist + island_groups, data = df_base_train2,
  method = "rpart",
  trControl = control,
  tuneGrid = grid
)

print(dec_base_rain_model_tuned)
```

```
## CART
##
## 7184 samples
##    2 predictor
##
## No pre-processing
## Resampling: Cross-Validated (5 fold)
## Summary of sample sizes: 5748, 5747, 5746, 5747, 5748
## Resampling results across tuning parameters:
##
##   cp     RMSE       Rsquared   MAE
##   0.001   99.72339  0.3194382  67.47874
##   0.006   99.95717  0.3147058  67.79174
##   0.011  100.20194  0.3113975  67.87942
##   0.016  100.51143  0.3070774  68.11902
##   0.021  101.07087  0.2992208  68.40240
##   0.026  102.18927  0.2832561  69.84635
##   0.031  102.18927  0.2832561  69.84635
##   0.036  103.27638  0.2684042  71.32285
##   0.041  103.27638  0.2684042  71.32285
##   0.046  103.27638  0.2684042  71.32285
##   0.051  103.27638  0.2684042  71.32285
##   0.056  103.27638  0.2684042  71.32285
##   0.061  103.27638  0.2684042  71.32285
##   0.066  103.27638  0.2684042  71.32285
##
## RMSE was used to select the optimal model using the smallest value.
## The final value used for the model was cp = 0.001.
```

```
plot(dec_base_rain_model_tuned)
```



```
png(here("adjusted SCM/outputs", "dec_rain_base_model.png"),
       width = 6,
       units = "in",
       height = 4,
    res = 300)
rpart.plot(dec_base_rain_model_tuned$finalModel)
dev.off()
```

```
## pdf
##   2
```

```
dec_rain_pred_tuned <- predict(dec_base_rain_model_tuned, base_test)

rmse_dec_rain_pred_tuned <- rmse(actual = df_base_train2$rain_total,
                           predicted = dec_rain_pred_tuned)

cat("rmse of tuned decision tree of rain model", rmse_dec_rain_pred_tuned, sep = " ")
```

```
## rmse of tuned decision tree of rain model 130.3024
```

Saving Tuned Rain_Total Decision Tree Model

```r
full_path <- here("adjusted SCM/new base models")
saveRDS(dec_base_rain_model_tuned,
        file = file.path(full_path, paste0("dec_base_rain_model_tuned", ".rds")))
```