Week 9 Lab Report

Nguyen Tien Anh Quach

2024-04-04

Please turn in a knitted PDF of your code, but this time SHOW YOUR CODE CHUNKS. Answer each question in the area provided.

#Setup Remember that residential carbon data from week 5? Lets revisit it and try and predict state level carbon using tree based methods.

We are interested in modeling state-level residential carbon production per capita, so we need to calculate a per capita value. Our units are now metric tons carbon per capita.

We want to whether the following variables help predict residential carbon per capita. We will use the same explanatory variables from week 5:

temp: Average annual temperature (degrees F) trumpvote: Percent of state that voted for President Trump climatechange: standardized climate change google search share bachelorsdegree: % of state population with Bachelor's Degree Incomepercapitaus: Median per capita income (USD) rps: Renewable portfolio standard (0/1) = Renewable Portfolio Standards mandates a certain percentage of energy production from renewable sources. States that have an RPS have been coded a 1 and states without a zero (0). urban_percent: Percent of state population that is urban West: the state is located in the west

Create a dataframe with only the variables you are interested in modeling. This is so that we can directly compare it to the linear models you all ran a few weeks ago. In practice, when doing machine learning, you would let R pick out the most important variables.

Decision tree

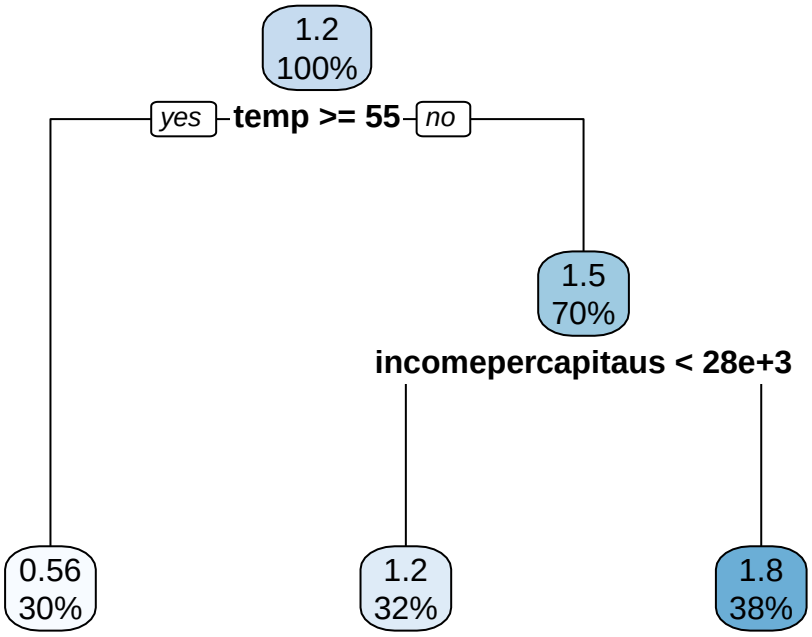
Build a decision tree using tidymodels with res.carbon.pc.mt as the response variable, and the rest as the predictor variables. Train the tree on 3/4 of the data. Print the tree (7 points)

Code:

```
mod1 <- set_engine(decision_tree(mode = "regression"), "rpart")

dtree_tflow <-
  # Plug the data
  carbon %>%
  # Begin the tidyflow
  tidyflow(seed = 23152) %>%
  # Separate the data into training/testing (we are keeping 3/4 of the data for training)
  plug_split(initial_split, prop = 3/4) %>%
  # Plug the formula
  plug_formula(res.carbon.pc.mt ~ temp + trumpvote + climatechange + bachelorsdegree + incomepercapitaus
  # Plug the model
  plug_model(mod1)

carbon_dtree <- fit(dtree_tflow)
carbon_tree <- pull_tflow_fit(carbon_dtree)$fit
rpart.plot(carbon_tree)
```



Question: Interpret the tree in words. What is the most important variable for predicting residential carbon? (3 points)

Answer:

The top box (root node) is temp, which demonstrated that temperature is the most important variable predicting residential carbon per capita. For the entire sample, the average residential carbon per capita is 1.2 metric tons carbon per capita.
30% of the sample has temperature equal to or higher than 55 degrees F and its average is 0.56 metric tons carbon per capita. This is a leaf node.
Within the rest of the sample (70%), the internal node is represented by the variable incomepercapitaus, leading to two leaf nodes (with 1.2 and 1.8 as their averages).

Bagged tree

Build a bagged tree using tidymodels on the same data. Train the tree on 3/4 of the data. Bootstrap 100 times. (7 points)

Code:

```
btree <- bag_tree(mode = "regression") %>% set_engine("rpart", times = 100)

bag_tflow <-
  carbon %>%
  tidyflow(seed = 56651) %>%
  plug_split(initial_split, prop = 3/4) %>%
  plug_formula(res.carbon.pc.mt ~ temp + trumpvote + climatechange + bachelorsdegree + incomepercapitaus
```

```
  plug_model(btree)

carbon_btree <- fit(bag_tflow)
```

Random forest

Build a random forest using tidymodels on the same data. Train the forest on 3/4 of the data. Set mtry to 1/3 of the predictor variables. Don't worry about tuning any of the other parameters. Print the variable importance (7 points)

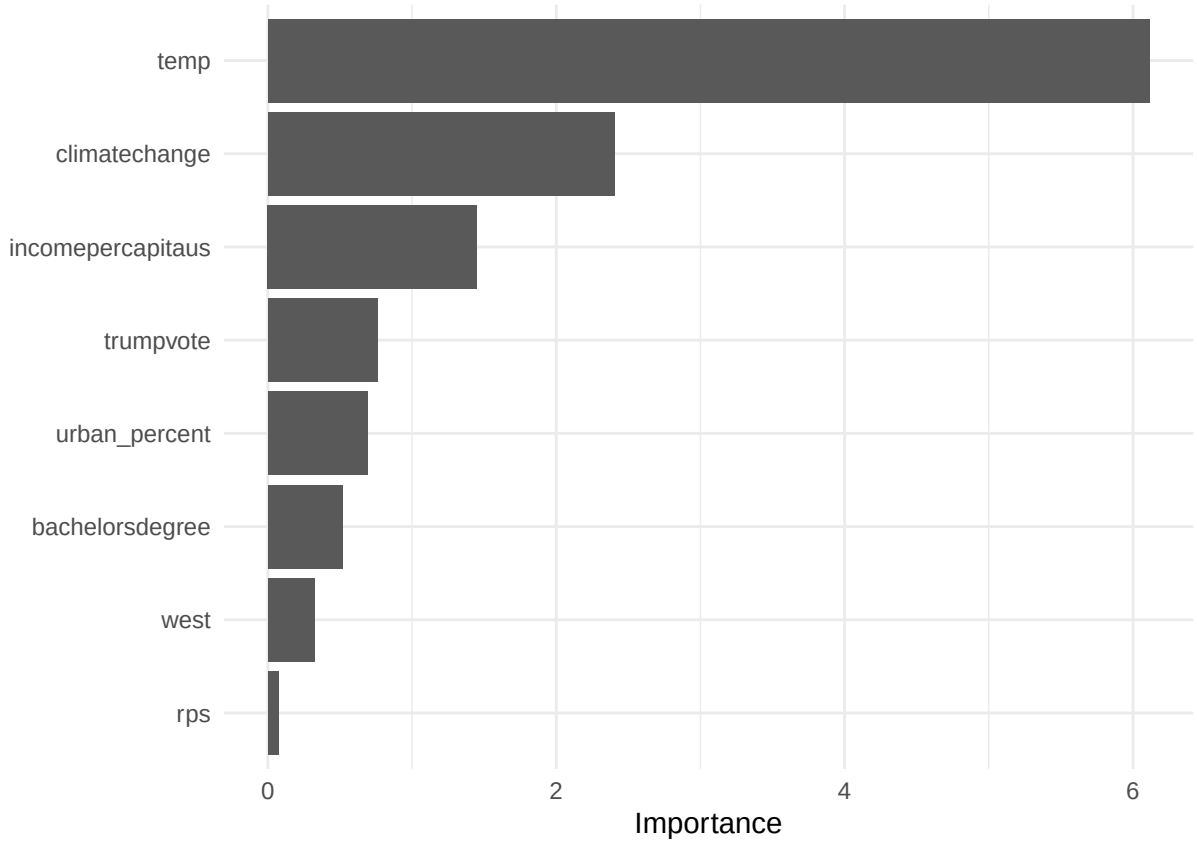
Code:

```
# Define the random forest
rf_mod <-
  rand_forest(mode = "regression", mtry = 5) %>%
  set_engine("ranger", importance = "impurity")

# Define the `tidyflow` with the random forest model
# and include all variables (including scie_score and read_score)
rf_tflow <-
  carbon %>%
  tidyflow(seed = 230051) %>%
  plug_formula(res.carbon.pc.mt ~ temp + trumpvote + climatechange + bachelorsdegree + incomepercapitaus
  plug_split(initial_split, prop = 3/4) %>%
  plug_model(rf_mod)

carbon_rf <- rf_tflow %>% fit()

carbon_rf %>%
  pull_tflow_fit() %>%
  .[['fit']] %>%
  vip() +
  theme_minimal()
```



Question: How does random forest rank variable importance? (3 points)

Answer:

Random forest ranks variable importance based on the decrease in node impurity, measured by the calculation of the Gini impurity. The mean decrease in Gini impurity (overall decrease/number of trees) will then be used to calculate the variable importance.
In the figure above, temp is still the most important predictor and is ca. four to five times more important than climate change.

Boosted regression tree

Build a boosted regression tree using tidymodels on the same data. Train on 3/4 of the data. Set ntree to 500. Don't worry about tuning any of the other parameters (7 points)

Code:

```
boost_mod <-
  boost_tree(mode = "regression", trees = 500) %>%
  set_engine("xgboost")

boost_tflow <-
  carbon %>%
  tidyflow(seed = 512131) %>%
  plug_formula(res.carbon.pc.mt ~ temp + trumpvote + climatechange + bachelorsdegree + incomepercapitaus
  plug_split(initial_split, prop = 3/4) %>%
  plug_model(boost_mod)
```

```
carbon_boost <- fit(boost_tflow)
```

Multiple linear regression

Build a multiple linear regression using tidymodels and the same data and formula as above (hint - google parsnip linear models). Train on 3/4 of the data. (7 points)

Code:

```
carbon_mlr_mod <- linear_reg() %>% set_engine("lm")

mlr_tflow <-
  carbon %>%
  tidyflow(seed = 51231) %>%
  plug_formula(res.carbon.pc.mt ~ temp + trumpvote + climatechange + bachelorsdegree + incomepercapitaus
  plug_split(initial_split,prop = 3/4) %>%
  plug_model(carbon_mlr_mod)

carbon_mlr <- fit(mlr_tflow)
```

Model comparison

Calculate the RMSE for each of the 5 modeling techniques above on the in-sample (training) data and the out-of-sample (testing data). Provide a table of all 10 values. Which modelling technique performed best in and out of sample? (9 points)

Code:

```
##decision tree
carbon_dtree %>% predict_training() %>%
  rmse(res.carbon.pc.mt, .pred) %>% select(.estimate)

carbon_dtree %>% predict_testing() %>%
  rmse(res.carbon.pc.mt, .pred) %>% select(.estimate)

##bagged tree

carbon_btree %>% predict_training() %>%
  rmse(res.carbon.pc.mt, .pred) %>% select(.estimate)

carbon_btree %>% predict_testing() %>%
  rmse(res.carbon.pc.mt, .pred) %>% select(.estimate)

##random forest

carbon_rf %>% predict_training() %>%
  rmse(res.carbon.pc.mt, .pred) %>% select(.estimate)

carbon_rf %>% predict_testing() %>%
  rmse(res.carbon.pc.mt, .pred) %>% select(.estimate)

##boosted regression
carbon_boost %>% predict_training() %>%
  rmse(res.carbon.pc.mt, .pred) %>% select(.estimate)
```

```
carbon_boost %>% predict_testing() %>%
  rmse(res.carbon.pc.mt, .pred) %>% select(.estimate)

##mlr

carbon_mlr %>% predict_training() %>%
  rmse(res.carbon.pc.mt, .pred) %>% select(.estimate)

carbon_mlr %>% predict_testing() %>%
  rmse(res.carbon.pc.mt, .pred) %>% select(.estimate)
```

Answer:

According to the table below, boosted regression performed best for in-sample prediction, whereas bagged tree performed best for out-of-sample prediction.
Overall, bagged tree seems to be the best modelling technique with low in and out of sample predictions.

```
###make table
library(kableExtra)

# Create a dataframe with your data and row names
table <- data.frame(
  "Decision tree" = c("0.34", "0.40"),
  "Bagged tree" = c("0.16", "0.32"),
  "Random forest" = c("0.16", "0.38"),
  "Boosted regression" = c("0.0006", "0.52"),
  "Multiple linear regression" = c("0.22", "0.38"),
  row.names = c("Training RMSE", "Testing RMSE")
)

# Print the dataframe as a kable
kable(table, format = "markdown")
```

	Decision.tree	Bagged.tree	Random.forest	Boosted.regression	Multiple.linear.regression
Training RMSE	0.34	0.16	0.16	0.0006	0.22
Testing RMSE	0.40	0.32	0.38	0.52	0.38