Compartmentalized Baseline Modeling

Ryan Rogers

10/28/2022

Library imports are left as-is. They'll be necessary in almost every version. New imports added for helper libraries

Modeling with threshold 50 number of claims

Will leave data import alone for now.

```
data <- read.csv("priv_mcare_f_pay_2022Oct18.csv")
hospital_data <- read.csv("Hospital_Master_Sheet.csv")
```

Hospital data aggregation, data split, and data filtering are now compartmentalized

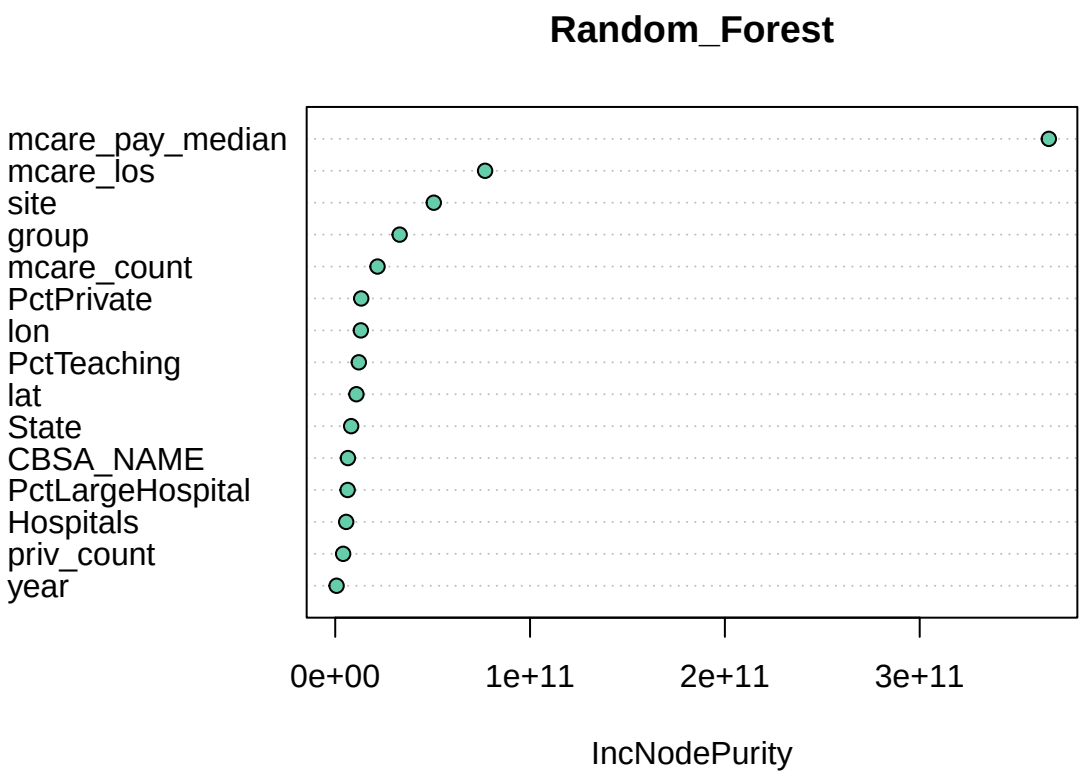
```
# Hospital data aggregation - validated for sameness
hospitals_msa <- hospital_data %>% aggregate_hospital_features()
rm(hospital_data)

# Data split into model data and predict - varies from original slightly
split_dataset <- data %>% data_split(count_thresh = 49)
working_set <- split_dataset[[1]]
predict_set <- split_dataset[[2]]

model_data <- left_join(working_set, hospitals_msa, by = "msa")  %>%
  select(-priv_pay_mean, -priv_pay_iqr, -mcare_pay_mean, -mcare_pay_sd, -Urban, -msa)
rm(working_set)

predict_data <- left_join(predict_set, hospitals_msa, by = "msa")  %>%
  select(-priv_pay_mean, -priv_pay_iqr, -mcare_pay_mean, -mcare_pay_sd, -Urban, -msa)
rm(predict_set)

# Train test split
train_test_data <- model_data %>% train_test_split(proportion_train = 0.8)
rm(model_data)

train <- train_test_data[[1]]
test <- train_test_data[[2]]
```

Model Creation and Prediction are now compartmentalized

```
# Random Forest model

# Fit Random Forest Model on training data
Random_Forest <- baseline_rdm_forest(data = train)

train_predict <- make_baseline_prediction(Random_Forest, train)
```

```
rm(train)

train_mape_percent = get_mape_percentage(train_predict)
```

```
varImpPlot(Random_Forest, bg = "aquamarine3")
```




```
test_predict <- make_baseline_prediction(Random_Forest, test)
rm(test)

test_mape_percent = get_mape_percentage(test_predict)
```

```
cat("With Threshold >50 claims for training set:\n")
```

```
## With Threshold >50 claims for training set:
```

```
cat("Train MAPE:" , round(train_mape_percent, 2), "%\n")
```

```
## Train MAPE: 14.06 %
```

```
cat("Test MAPE:" , round(test_mape_percent, 2), "%\n")
```

```
## Test MAPE: 31.58 %
```

Modeling with threshold 35 number of claims

Will leave data import alone for now.

```
data <- read.csv("priv_mcare_f_pay_2022Oct18.csv")
hospital_data <- read.csv("Hospital_Master_Sheet.csv")
```

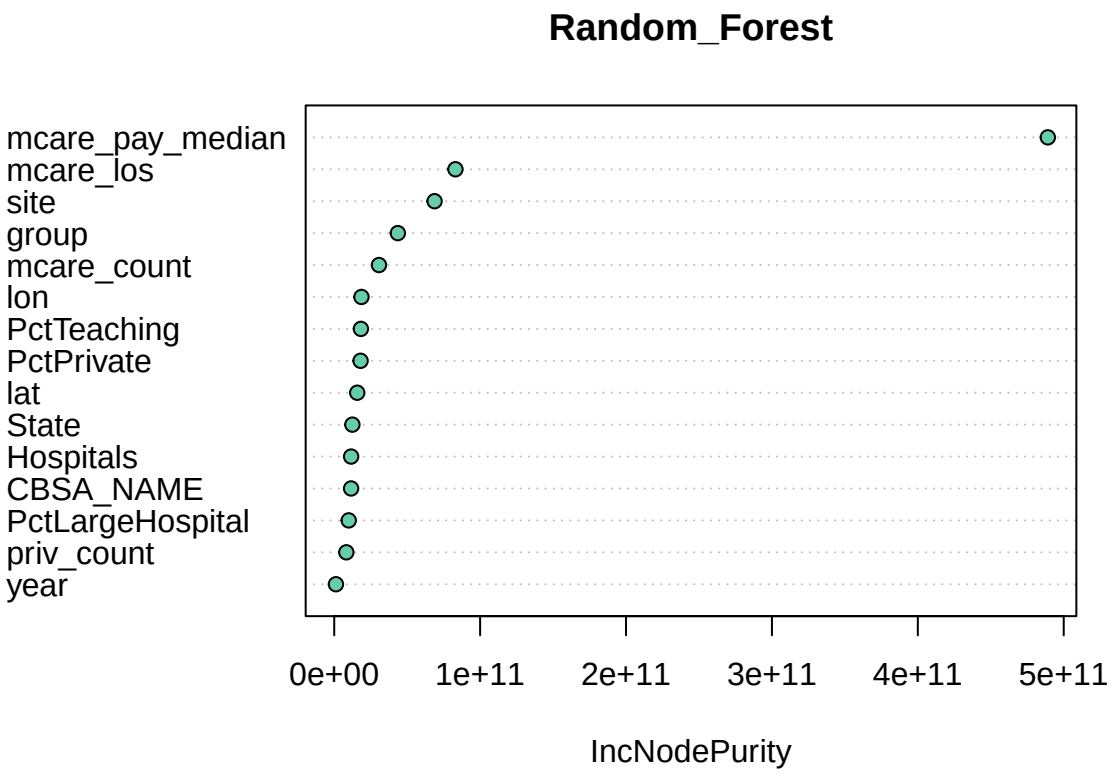
Hospital data aggregation, data split, and data filtering are now compartmentalized

```
# Hospital data aggregation - validated for sameness
hospitals_msa <- hospital_data %>% aggregate_hospital_features()
rm(hospital_data)

# Data split into model data and predict - varies from original slightly
split_dataset <- data %>% data_split(count_thresh = 34)
working_set <- split_dataset[[1]]
predict_set <- split_dataset[[2]]

model_data <- left_join(working_set, hospitals_msa, by = "msa")  %>%
  select(-priv_pay_mean, -priv_pay_iqr, -mcare_pay_mean, -mcare_pay_sd, -Urban, -msa)
rm(working_set)

predict_data <- left_join(predict_set, hospitals_msa, by = "msa")  %>%
  select(-priv_pay_mean, -priv_pay_iqr, -mcare_pay_mean, -mcare_pay_sd, -Urban, -msa)
rm(predict_set)

# Train test split
train_test_data <- model_data %>% train_test_split(proportion_train = 0.8)
rm(model_data)

train <- train_test_data[[1]]
test <- train_test_data[[2]]
```

Model Creation and Prediction are now compartmentalized

```
# Random Forest model

# Fit Random Forest Model on training data
Random_Forest <- baseline_rdm_forest(data = train)

train_predict <- make_baseline_prediction(Random_Forest, train)
rm(train)

train_mape_percent = get_mape_percentage(train_predict)
```

```
varImpPlot(Random_Forest, bg = "aquamarine3")
```




```
test_predict <- make_baseline_prediction(Random_Forest, test)
rm(test)

test_mape_percent = get_mape_percentage(test_predict)
```

```
cat("With Threshold >35 claims for training set:\n")
```

```
## With Threshold >35 claims for training set:
```

```
cat("Train MAPE:" , round(train_mape_percent, 2), "%\n")
```

```
## Train MAPE: 15.76 %
```

```
cat("Test MAPE:" , round(test_mape_percent, 2), "%\n")
```

```
## Test MAPE: 21.23 %
```