Baseline Modeling Clustering

Ryan Rogers/Tyler Marshall

10/31/2022

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")
```

Modeling with Cluster 50 number of claims

```
data <- read.csv("priv_mcare_f_pay_2022Oct18.csv")
cluster_1 <- c("bariatric", "breast reconstruction","bsp", "bunionectomy", "clavicle fixation",
               "fess", "hysterect","kidney ablation", "lap appendectomy","liver ablation",
               "mastectomy","navigation","orthovisc_monovisc","partial shoulder arthroplasty",
               "pka","pnn","prostatectomy","radius/ulna internal fixation",
               "robotic_assisted_surgery", "rtc_slap_bank", "septoplasty")
cluster_2 <- c("ant_tls_fusion","hepat","intracranial_thromb","post_cerv_fusion",
               "post_TLS_fusion", "post_tls_fusion")
cluster_3 <- c("ankle_fix","ant_cerv_fusion","cardiac ablation",
               "cardiac ablation_additional_discrete","cardiac ablation_linear_focal",
               "cardiac_ablaton_anesthesia","cardiac_ablaton_ice", "colorect",
               "femoral shaft fixation","hernia","hip_fracture_fixation","laac",
               "lung ablation","prox_tibia_fixation","proximal humerus","revision_tha",
               "revision_tka","tavr", "tha", "thoracic", "tka", "tpa", "tsa")

cluster_data <- within(data, {
  Cluster = NA
  Cluster[group %in% cluster_1] = 1
  Cluster[group %in% cluster_2] = 2
  Cluster[group %in% cluster_3] = 3
})

cluster_1 <- cluster_data %>%
  filter(Cluster == 1)
cluster_2 <- cluster_data %>%
  filter(Cluster == 2)
cluster_3 <- cluster_data %>%
  filter(Cluster == 3)
```

Cluster 1

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

# Data split into model data and predict - varies from original slightly
split_dataset <- cluster_1 %>% 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 = 0.8)
rm(model_data)

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

Model Creation and Prediction are now compartmentalized

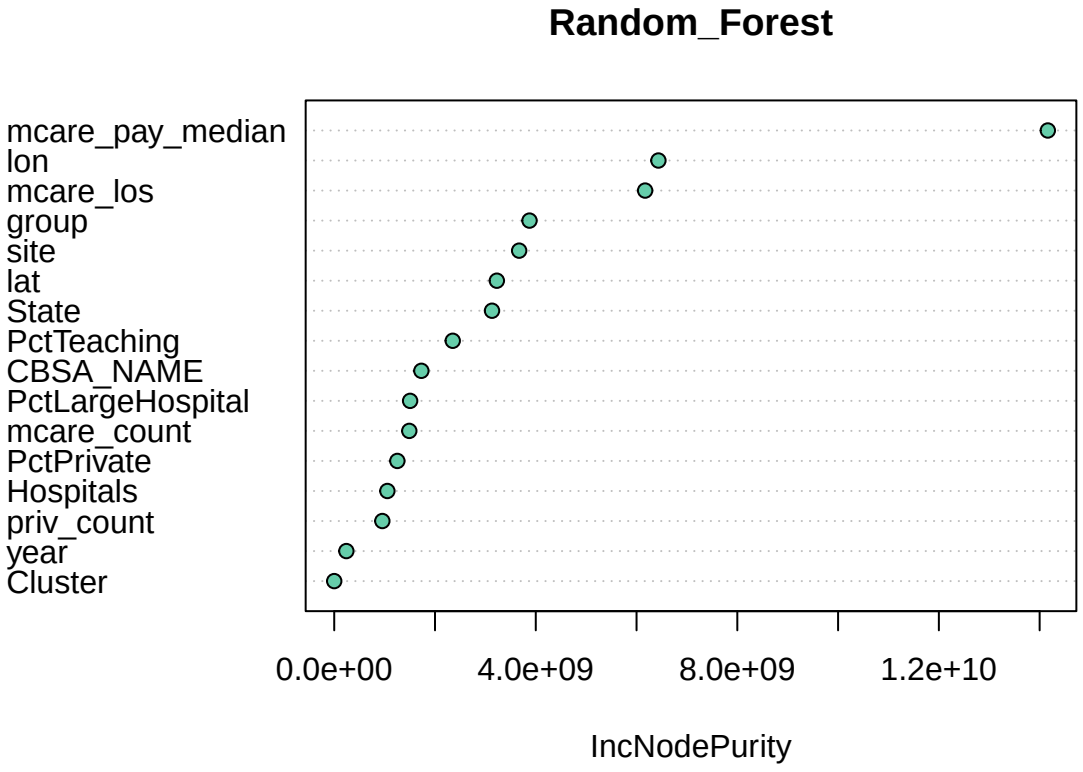
```
# Random Forest model

# 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: 17.89 %
```

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

```
## Test MAPE: 33.87 %
```

Cluster 2

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

# Data split into model data and predict - varies from original slightly
split_dataset <- cluster_2 %>% 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 = 0.8)
rm(model_data)

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

Model Creation and Prediction are now compartmentalized

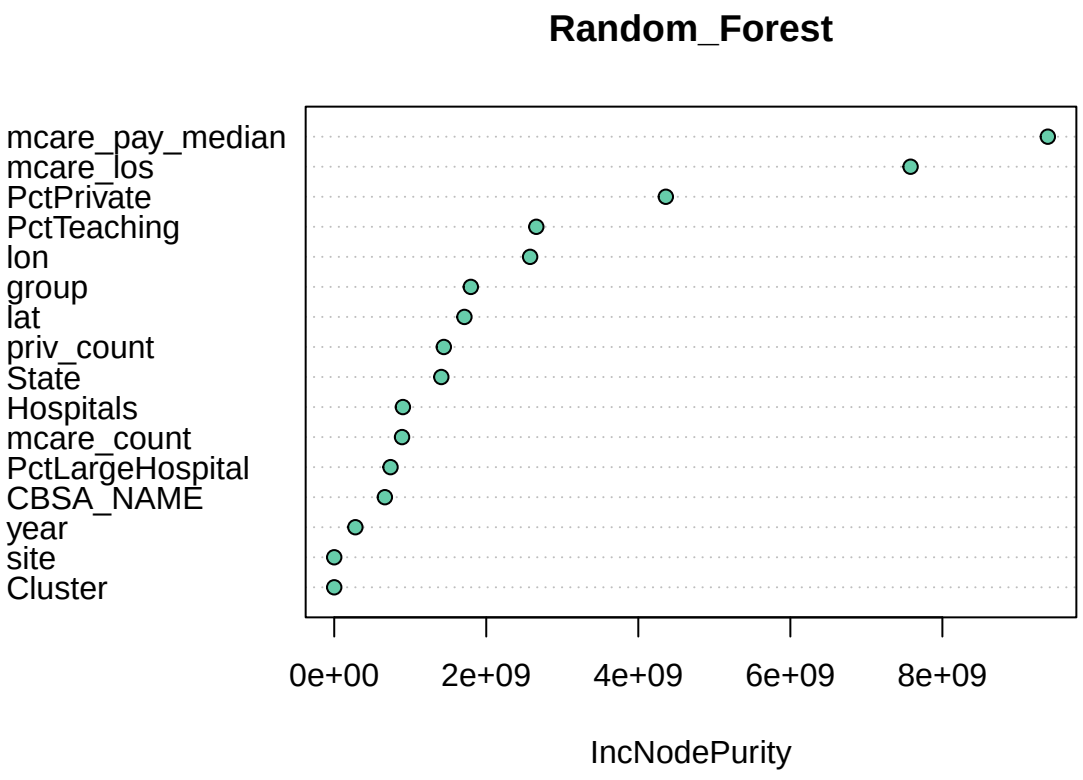
```
# Random Forest model

# 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: 10.91 %
```

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

```
## Test MAPE: 18.04 %
```

Cluster 3

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

# Data split into model data and predict - varies from original slightly
split_dataset <- cluster_3 %>% 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 = 0.8)
rm(model_data)

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

Model Creation and Prediction are now compartmentalized

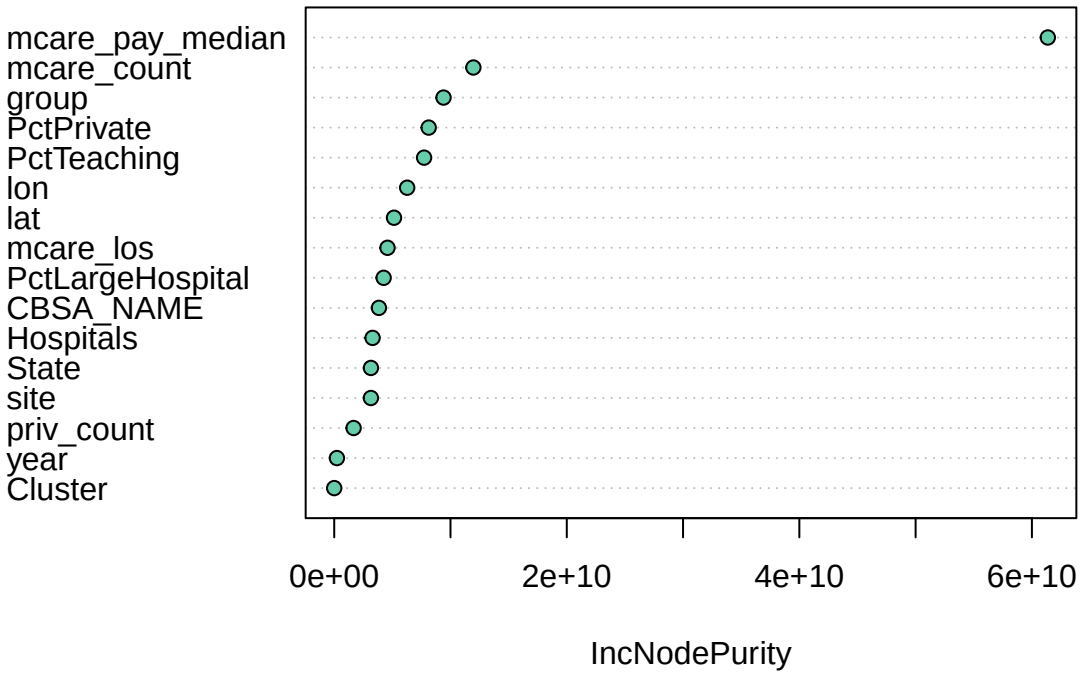
```
# Random Forest model

# 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: 12.07 %
```

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

```
## Test MAPE: 20.19 %
```

Modeling with threshold 35 number of claims

Cluster 1

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

# Data split into model data and predict - varies from original slightly
split_dataset <- cluster_1 %>% 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 = 0.8)
rm(model_data)

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

Model Creation and Prediction are now compartmentalized

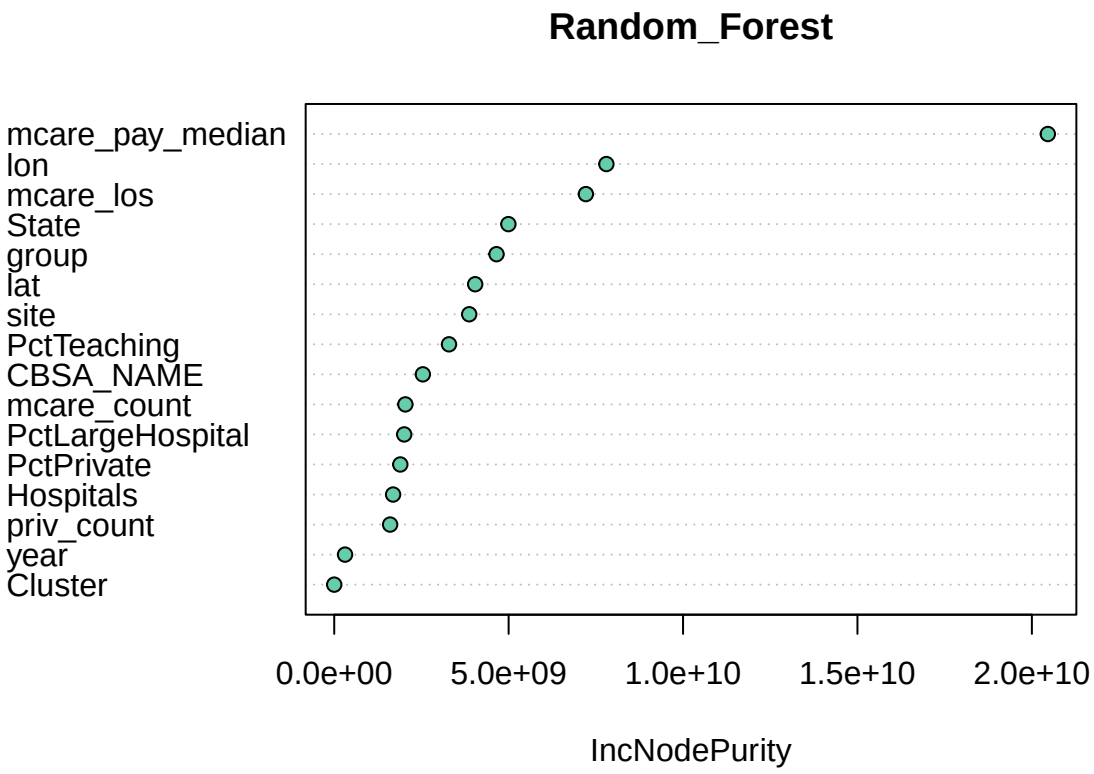
```
# Random Forest model

# 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: 16.06 %
```

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

```
## Test MAPE: 34.57 %
```

Cluster 2

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

# Data split into model data and predict - varies from original slightly
split_dataset <- cluster_2 %>% 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 = 0.8)
rm(model_data)

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

Model Creation and Prediction are now compartmentalized

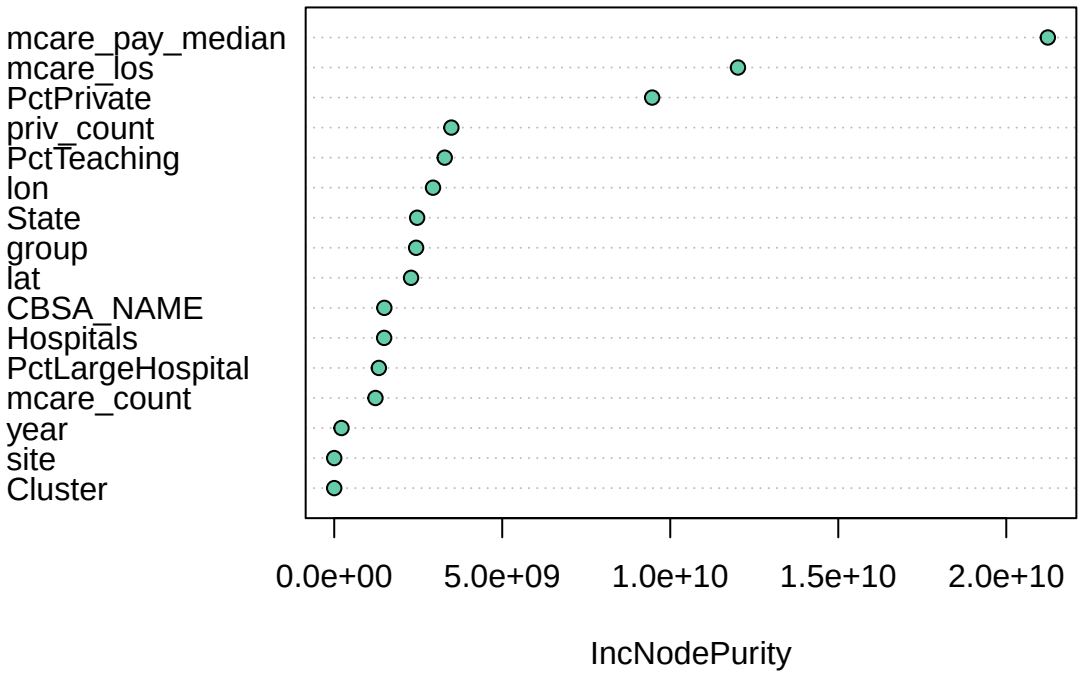
```
# Random Forest model

# 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: 10.71 %
```

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

```
## Test MAPE: 16.04 %
```

Cluster 3

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

# Data split into model data and predict - varies from original slightly
split_dataset <- cluster_3 %>% 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 = 0.8)
rm(model_data)

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

Model Creation and Prediction are now compartmentalized

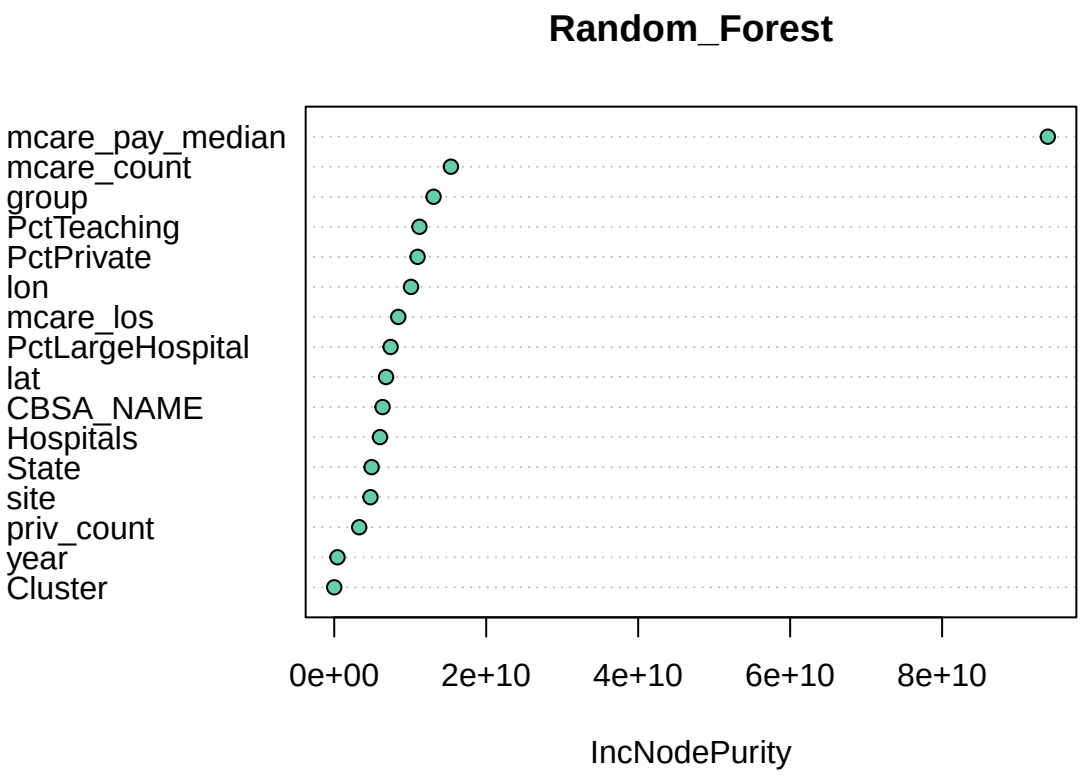
```
# Random Forest model

# 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)
```

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

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

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

```
## Train MAPE: 12.39 %
```

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

```
## Test MAPE: 19.58 %
```