Clustering Modeling with Feature Selection

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.

```r
data <- read.csv("priv_mcare_f_pay_2022Oct18.csv")
hospital_data <- read.csv("Hospital_Master_Sheet.csv")
data2 <- read.csv("combined_features.csv")
```

Modeling with Cluster 50 number of claims

```r
data <- read.csv("priv_mcare_f_pay_2022Oct18.csv")
cluster_0 <- 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_1 <- c("ant_tls_fusion","hepat","intracranial_thromb","post_cerv_fusion",
               "post_tls_fusion")
cluster_2 <- 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(data2, {
  Cluster = NA
  Cluster[group %in% cluster_0] = 0
  Cluster[group %in% cluster_1] = 1
  Cluster[group %in% cluster_2] = 2
})

cluster_0 <- cluster_data %>%
  filter(Cluster == 0)
cluster_1 <- cluster_data %>%
  filter(Cluster == 1)
cluster_2 <- cluster_data %>%
  filter(Cluster == 2)
```

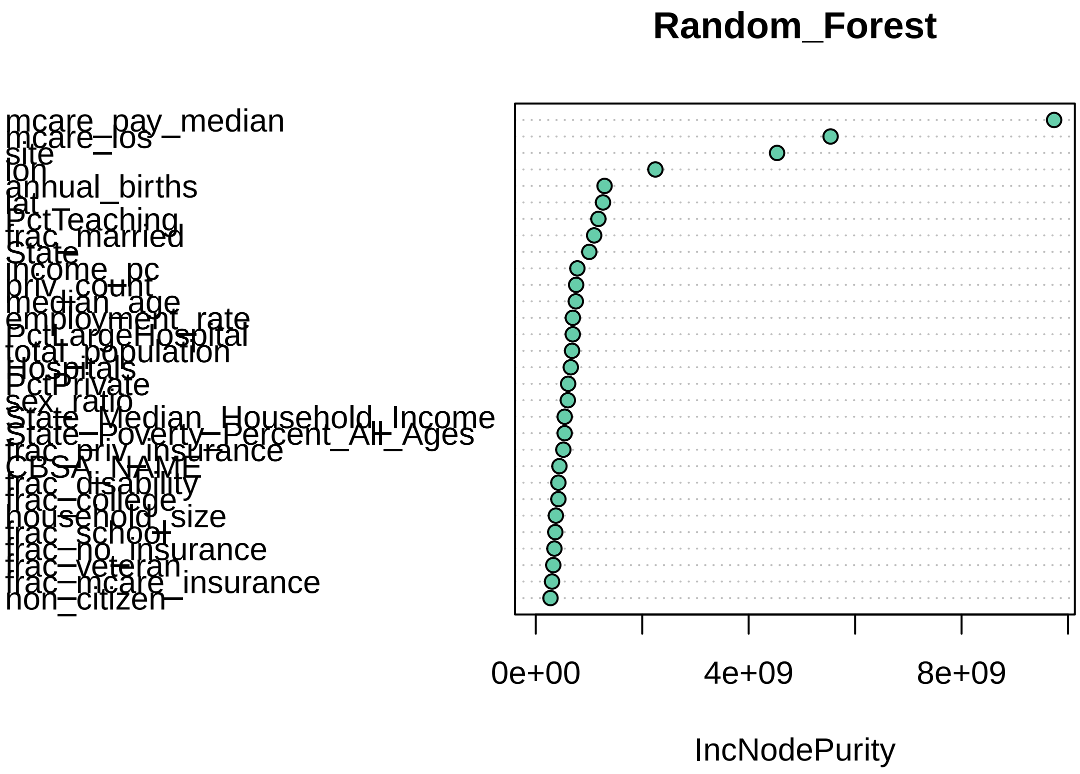
Cluster 0

```
# 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_0 %>% data_split(count_thresh = 49)
working_set <- split_dataset[[1]]
predict_set <- split_dataset[[2]]

model_data <- working_set  %>%
  select(-priv_pay_mean, -priv_pay_iqr, -mcare_pay_mean, -mcare_pay_sd, -msa, -Cluster,-group, -FIPS.Sta
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, -Cluster)
rm(predict_set)

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


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)
model = train(priv_pay_median ~ . , data=model_data, method = "rf", na.action = na.omit,trControl = tra

linear_model = train(priv_pay_median ~ . , data=model_data, method = "lm", na.action = na.omit,trControl

cv_mod <- model$pred

cv_mape = MAPE(cv_mod$pred, cv_mod$obs)

cv_lin_mod <- linear_model$pred

cv_lin_mape = MAPE(cv_lin_mod$pred, cv_lin_mod$obs)

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: 21.48 %
```

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

```
## Test MAPE: 18.33 %
```

```
cat("CV MAPE:" , round(100*cv_mape, 2), "%\n")
```

```
## CV MAPE: 26.56 %
```

```
cat("CV Lin MAPE:" , round(100*cv_lin_mape, 2), "%\n")
```

```
## CV Lin MAPE: 26.36 %
```

Cluster 1

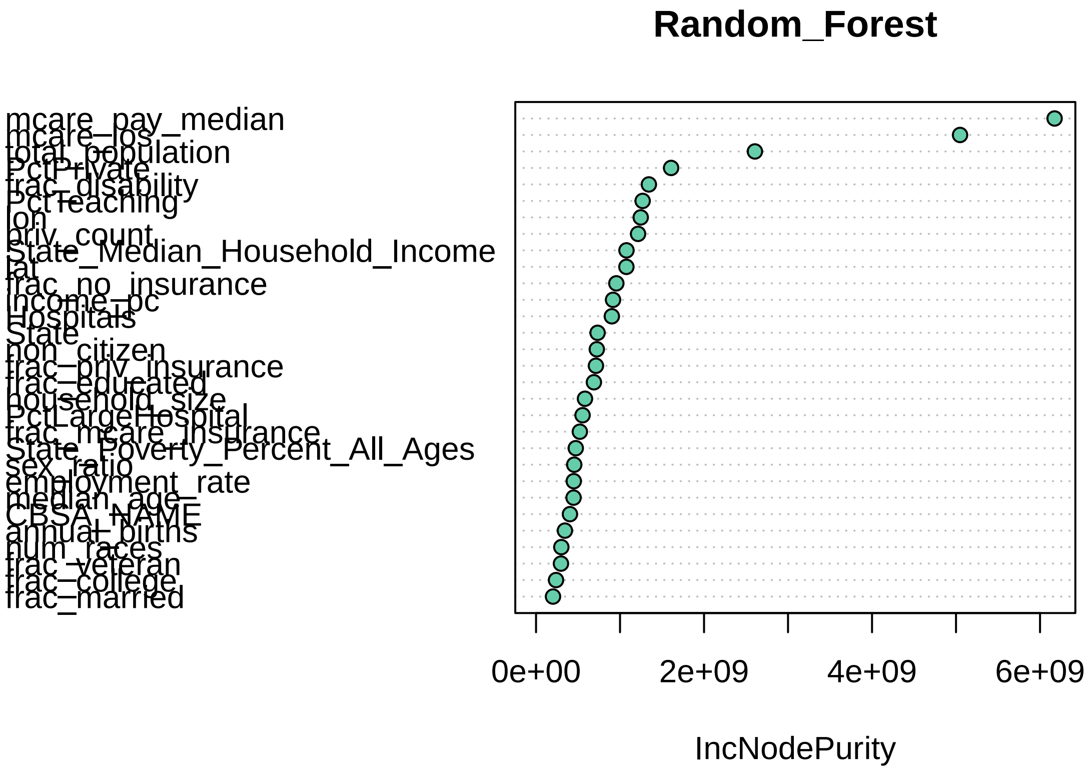
```
# Hospital data aggregation - validated for sameness
hospitals_msa <- hospital_data %>% aggregate_hospital_features()

# 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 <- working_set %>%
  select(-priv_pay_mean, -priv_pay_iqr, -mcare_pay_mean, -mcare_pay_sd, -msa,-Cluster, -group, -FIPS.Sta
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,-Cluster)
rm(predict_set)

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

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)

model_data <- model_data %>%
  select(-site)


model = train(priv_pay_median ~ . , data=model_data, method = "rf", na.action = na.omit,trControl = trai

linear_model = train(priv_pay_median ~ . , data=model_data, method = "lm", na.action = na.omit,trControl


cv_lin_mod <- linear_model$pred

cv_lin_mape = MAPE(cv_lin_mod$pred, cv_lin_mod$obs)

cv_mod <- model$pred

cv_mape = MAPE(cv_mod$pred, cv_mod$obs)

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: 11.82 %
```

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

```
## Test MAPE: 11.45 %
```

```
cat("CV MAPE:" , round(100*cv_mape, 2), "%\n")
```

```
## CV MAPE: 12.76 %
```

```
cat("CV Lin MAPE:" , round(100*cv_lin_mape, 2), "%\n")
```

```
## CV Lin MAPE: 14.35 %
```

Cluster 2

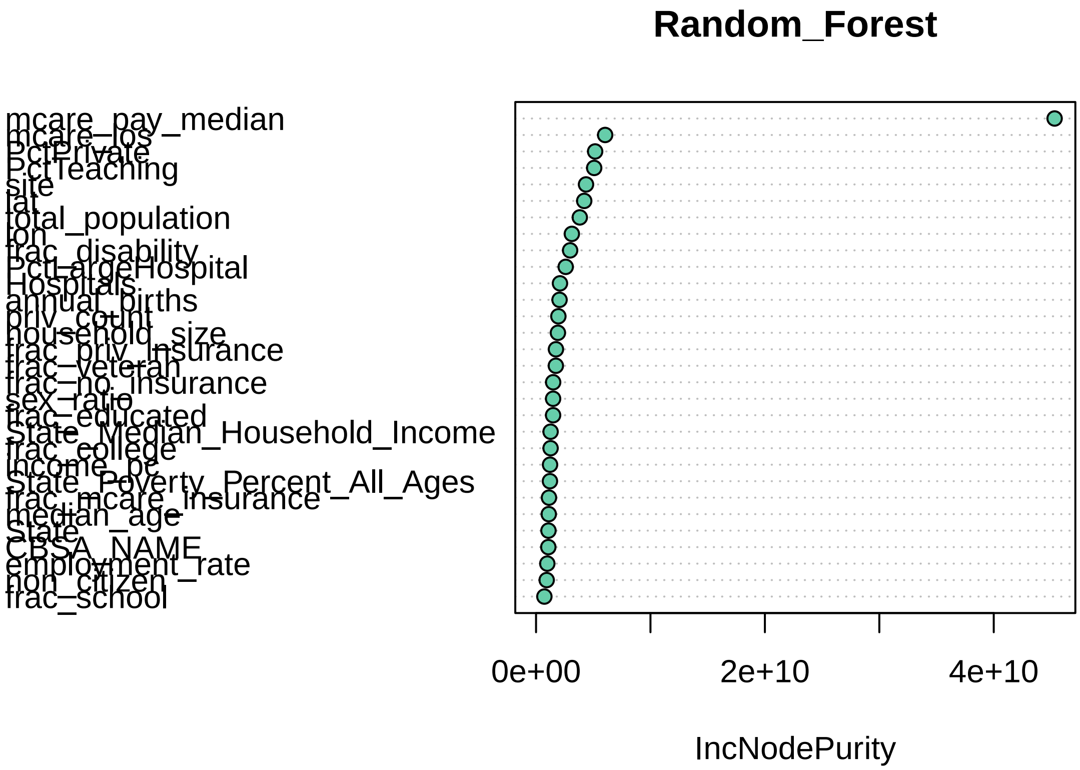
```
# Hospital data aggregation - validated for sameness
hospitals_msa <- hospital_data %>% aggregate_hospital_features()

# Data split into model data and predict - varies from original slightly
split_dataset <- cluster_2 %>% data_split(count_thresh = 49)
```

```r
working_set <- split_dataset[[1]]
predict_set <- split_dataset[[2]]

model_data <- working_set %>%
  select(-priv_pay_mean, -priv_pay_iqr, -mcare_pay_mean, -mcare_pay_sd, -msa,-Cluster, -group, -FIPS.Sta
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,-Cluster)
rm(predict_set)

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

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

Model Creation and Prediction are now compartmentalized

```r
# Random Forest model

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

model = train(priv_pay_median ~ . , data=model_data, method = "rf", na.action = na.omit,trControl = tra

linear_model = train(priv_pay_median ~ . , data=model_data, method = "lm", na.action = na.omit,trContro


cv_lin_mod <- linear_model$pred

cv_lin_mape = MAPE(cv_lin_mod$pred, cv_lin_mod$obs)

cv_mod <- model$pred

cv_mape = MAPE(cv_mod$pred, cv_mod$obs)

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

train_mape_percent = get_mape_percentage(train_predict)
```

```r
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: 16.88 %
```

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

```
## Test MAPE: 26.23 %
```

```
cat("CV MAPE:" , round(100*cv_mape, 2), "%\n")
```

```
## CV MAPE: 13.87 %
```

```
cat("CV Lin MAPE:" , round(100*cv_lin_mape, 2), "%\n")
```

```
## CV Lin MAPE: 25.21 %
```

Modeling with Cluster 35 number of claims

```
data <- read.csv("priv_mcare_f_pay_2022Oct18.csv")
cluster_0 <- 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_1 <- c("ant_tls_fusion","hepat","intracranial_thromb","post_cerv_fusion",
               "post_tls_fusion")
cluster_2 <- 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(data2, {
  Cluster = NA
  Cluster[group %in% cluster_0] = 0
  Cluster[group %in% cluster_1] = 1
  Cluster[group %in% cluster_2] = 2
})

cluster_0 <- cluster_data %>%
  filter(Cluster == 0)
cluster_1 <- cluster_data %>%
  filter(Cluster == 1)
cluster_2 <- cluster_data %>%
  filter(Cluster == 2)
```

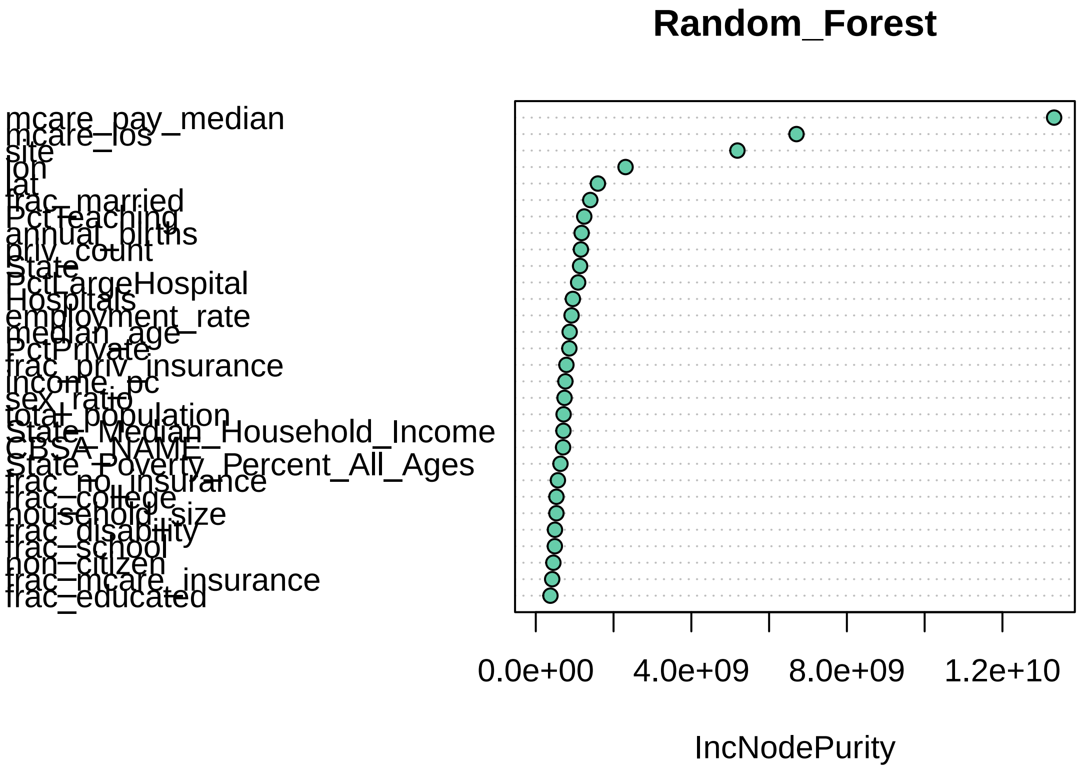
Cluster 0

```
# 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_0 %>% data_split(count_thresh = 34)
working_set <- split_dataset[[1]]
predict_set <- split_dataset[[2]]

model_data <- working_set  %>%
  select(-priv_pay_mean, -priv_pay_iqr, -mcare_pay_mean, -mcare_pay_sd, -msa, -Cluster, -group, -FIPS.S
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, -Cluster)
rm(predict_set)

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


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)
model = train(priv_pay_median ~ . , data=model_data, method = "rf", na.action = na.omit,trControl = tra

linear_model = train(priv_pay_median ~ . , data=model_data, method = "lm", na.action = na.omit,trControl

xgb_model = train(priv_pay_median ~ . , data=model_data, method = "xgbTree", na.action = na.omit,trCont

cv_mod <- model$pred

cv_mape = MAPE(cv_mod$pred, cv_mod$obs)

cv_lin_mod <- linear_model$pred

cv_lin_mape = MAPE(cv_lin_mod$pred, cv_lin_mod$obs)

cv_xgb_mod <- xgb_model$pred

cv_xgb_mape = MAPE(cv_xgb_mod$pred, cv_xgb_mod$obs)

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: 21.88 %
```

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

```
## Test MAPE: 19.46 %
```

```
cat("CV MAPE:" , round(100*cv_mape, 2), "%\n")
```

```
## CV MAPE: 24.35 %
```

```
cat("CV Lin MAPE:" , round(100*cv_lin_mape, 2), "%\n")
```

```
## CV Lin MAPE: 26.61 %
```

```
cat("CV XGB MAPE:" , round(100*cv_xgb_mape, 2), "%\n")
```

```
## CV XGB MAPE: 26.49 %
```

Cluster 1

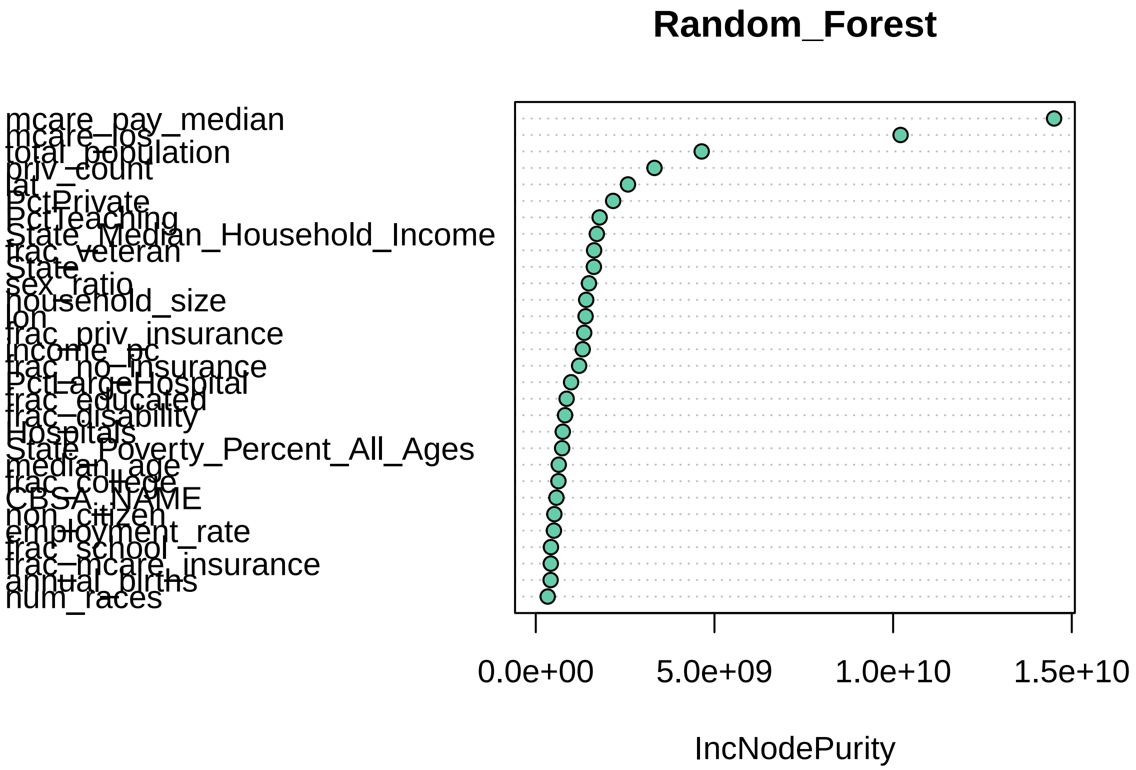
```
# Hospital data aggregation - validated for sameness
hospitals_msa <- hospital_data %>% aggregate_hospital_features()

# 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 <- working_set %>%
  select(-priv_pay_mean, -priv_pay_iqr, -mcare_pay_mean, -mcare_pay_sd, -msa,-Cluster, -group, -FIPS.Sta
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,-Cluster)
rm(predict_set)

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

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

Model Creation and Prediction are now compartmentalized

```
# Random Forest model
```

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

model_data <- model_data %>%
  select(-site)


model = train(priv_pay_median ~ . , data=model_data, method = "rf", na.action = na.omit,trControl = tra

linear_model = train(priv_pay_median ~ . , data=model_data, method = "lm", na.action = na.omit,trContro

xgb_model = train(priv_pay_median ~ . , data=model_data, method = "xgbTree", na.action = na.omit,trCont


cv_lin_mod <- linear_model$pred

cv_lin_mape = MAPE(cv_lin_mod$pred, cv_lin_mod$obs)

cv_mod <- model$pred

cv_mape = MAPE(cv_mod$pred, cv_mod$obs)

cv_xgb_mod <- xgb_model$pred

cv_xgb_mape = MAPE(cv_xgb_mod$pred, cv_xgb_mod$obs)

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

train_mape_percent = get_mape_percentage(train_predict)
```

```r
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: 12.12 %
```

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

```
## Test MAPE: 14.45 %
```

```
cat("CV MAPE:" , round(100*cv_mape, 2), "%\n")
```

```
## CV MAPE: 13.56 %
```

```
cat("CV Lin MAPE:" , round(100*cv_lin_mape, 2), "%\n")
```

```
## CV Lin MAPE: 15.87 %
```

```
cat("CV XGB MAPE:" , round(100*cv_xgb_mape, 2), "%\n")
```

```
## CV XGB MAPE: 13.4 %
```

Cluster 2

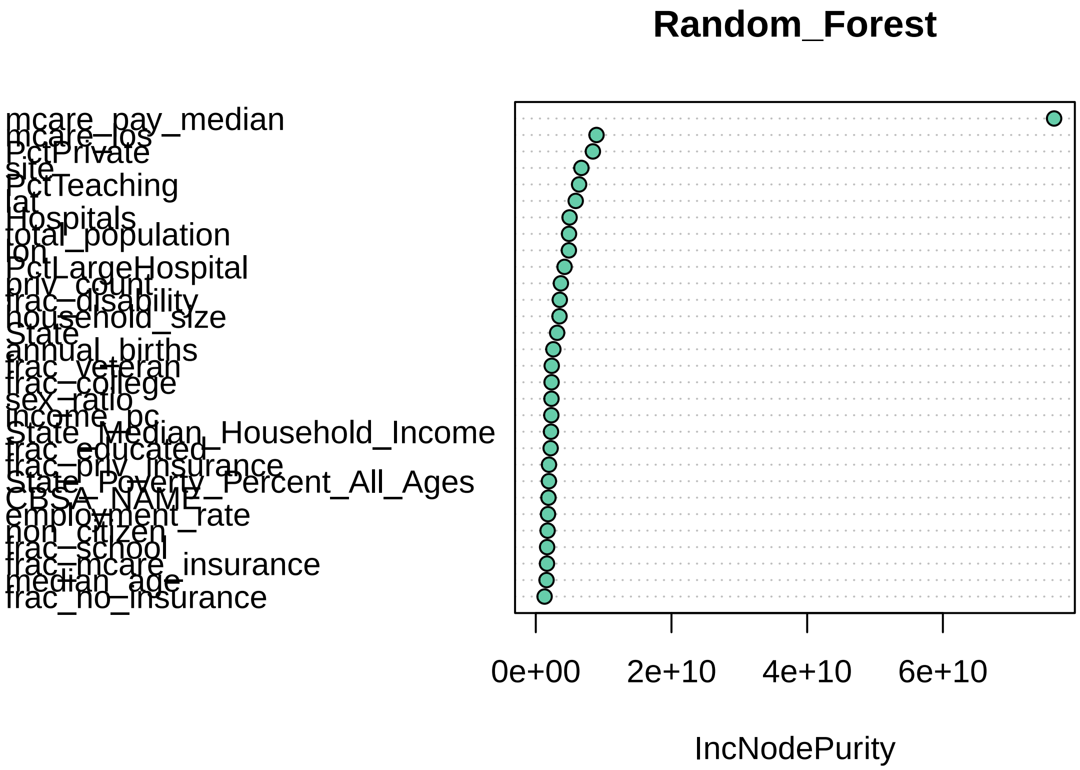
```
# Hospital data aggregation - validated for sameness
hospitals_msa <- hospital_data %>% aggregate_hospital_features()
```

```r
# 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 <- working_set %>%
  select(-priv_pay_mean, -priv_pay_iqr, -mcare_pay_mean, -mcare_pay_sd, -msa,-Cluster, -group, -FIPS.Sta
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,-Cluster)
rm(predict_set)

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

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

Model Creation and Prediction are now compartmentalized

```r
# Random Forest model

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

model = train(priv_pay_median ~ . , data=model_data, method = "rf", na.action = na.omit,trControl = tra

linear_model = train(priv_pay_median ~ . , data=model_data, method = "lm", na.action = na.omit,trContro

xgb_model = train(priv_pay_median ~ . , data=model_data, method = "xgbTree", na.action = na.omit,trCont


cv_lin_mod <- linear_model$pred

cv_lin_mape = MAPE(cv_lin_mod$pred, cv_lin_mod$obs)

cv_xgb_mod <- xgb_model$pred

cv_xgb_mape = MAPE(cv_xgb_mod$pred, cv_xgb_mod$obs)

cv_mod <- model$pred

cv_mape = MAPE(cv_mod$pred, cv_mod$obs)

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

train_mape_percent = get_mape_percentage(train_predict)
```

```r
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: 16.51 %
```

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

```
## Test MAPE: 24.21 %
```

```
cat("CV MAPE:" , round(100*cv_mape, 2), "%\n")
```

```
## CV MAPE: 14.69 %
```

```
cat("CV Lin MAPE:" , round(100*cv_lin_mape, 2), "%\n")
```

```
## CV Lin MAPE: 27.01 %
```

```
cat("CV XGB MAPE:" , round(100*cv_xgb_mape, 2), "%\n")
```

```
## CV XGB MAPE: 16.93 %
```

Modeling with Cluster 35 number of claims (With Procedure Group Included)

```
data <- read.csv("priv_mcare_f_pay_2022Oct18.csv")
cluster_0 <- 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_1 <- c("ant_tls_fusion","hepat","intracranial_thromb","post_cerv_fusion",
               "post_tls_fusion")
cluster_2 <- 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(data2, {
  Cluster = NA
  Cluster[group %in% cluster_0] = 0
  Cluster[group %in% cluster_1] = 1
  Cluster[group %in% cluster_2] = 2
})

cluster_0 <- cluster_data %>%
  filter(Cluster == 0)
cluster_1 <- cluster_data %>%
  filter(Cluster == 1)
cluster_2 <- cluster_data %>%
  filter(Cluster == 2)
```

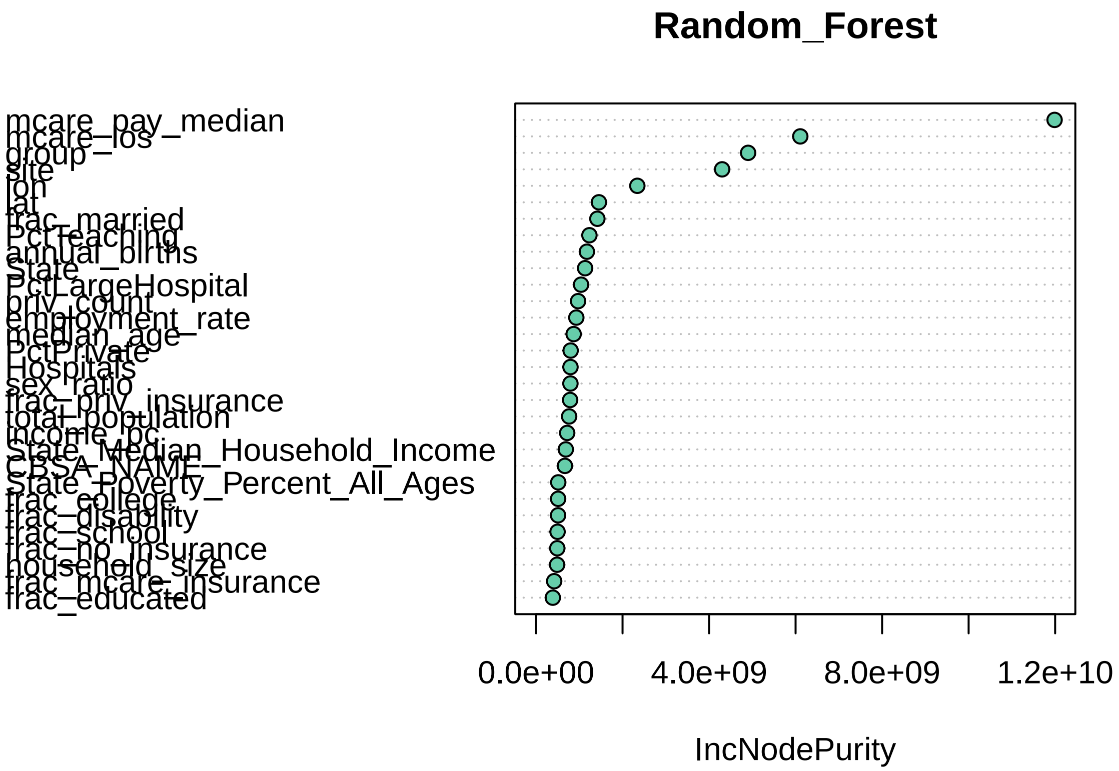
Cluster 0

```
# 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_0 %>% data_split(count_thresh = 34)
working_set <- split_dataset[[1]]
predict_set <- split_dataset[[2]]

model_data <- working_set  %>%
  select(-priv_pay_mean, -priv_pay_iqr, -mcare_pay_mean, -mcare_pay_sd, -msa, -Cluster, -FIPS.State.Code
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, -Cluster)
rm(predict_set)

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


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

Model Creation and Prediction are now compartmentalized

```r
# Random Forest model

# Fit Random Forest Model on training data
Random_Forest <- baseline_rdm_forest(data = train)
model = train(priv_pay_median ~ . , data=model_data, method = "rf", na.action = na.omit,trControl = trai

linear_model = train(priv_pay_median ~ . , data=model_data, method = "lm", na.action = na.omit,trControl

xgb_model = train(priv_pay_median ~ . , data=model_data, method = "xgbTree", na.action = na.omit,trContr

cv_mod <- model$pred

cv_mape = MAPE(cv_mod$pred, cv_mod$obs)

cv_lin_mod <- linear_model$pred

cv_lin_mape = MAPE(cv_lin_mod$pred, cv_lin_mod$obs)

cv_xgb_mod <- xgb_model$pred

cv_xgb_mape = MAPE(cv_xgb_mod$pred, cv_xgb_mod$obs)

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


train_mape_percent = get_mape_percentage(train_predict)
```

```r
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: 19.8 %
```

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

```
## Test MAPE: 18.41 %
```

```
cat("CV MAPE:" , round(100*cv_mape, 2), "%\n")
```

```
## CV MAPE: 20.79 %
```

```
cat("CV Lin MAPE:" , round(100*cv_lin_mape, 2), "%\n")
```

```
## CV Lin MAPE: 28.35 %
```

```
cat("CV XGB MAPE:" , round(100*cv_xgb_mape, 2), "%\n")
```

```
## CV XGB MAPE: 20.43 %
```

Cluster 1

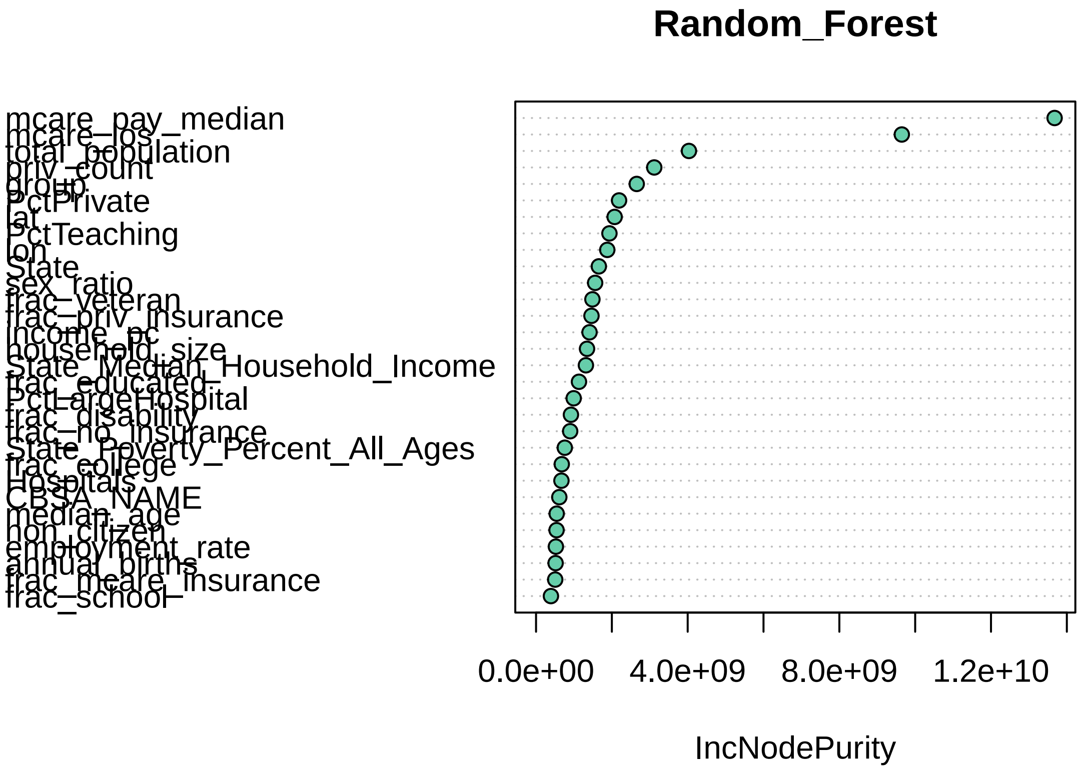
```
# Hospital data aggregation - validated for sameness
hospitals_msa <- hospital_data %>% aggregate_hospital_features()
```

```r
# 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 <- working_set %>%
  select(-priv_pay_mean, -priv_pay_iqr, -mcare_pay_mean, -mcare_pay_sd, -msa,-Cluster, -FIPS.State.Code
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,-Cluster)
rm(predict_set)

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

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

Model Creation and Prediction are now compartmentalized

```r
# Random Forest model

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

model_data <- model_data %>%
  select(-site)


model = train(priv_pay_median ~ . , data=model_data, method = "rf", na.action = na.omit,trControl = tra

linear_model = train(priv_pay_median ~ . , data=model_data, method = "lm", na.action = na.omit,trContro

xgb_model = train(priv_pay_median ~ . , data=model_data, method = "xgbTree", na.action = na.omit,trCont


cv_lin_mod <- linear_model$pred

cv_lin_mape = MAPE(cv_lin_mod$pred, cv_lin_mod$obs)

cv_mod <- model$pred

cv_mape = MAPE(cv_mod$pred, cv_mod$obs)

cv_xgb_mod <- xgb_model$pred

cv_xgb_mape = MAPE(cv_xgb_mod$pred, cv_xgb_mod$obs)

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

train_mape_percent = get_mape_percentage(train_predict)
```

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



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

test_mape_percent = get_mape_percentage(test_predict)
```

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

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

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

```
## Train MAPE: 11.97 %
```

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

```
## Test MAPE: 14.52 %
```

```r
cat("CV MAPE:" , round(100*cv_mape, 2), "%\n")
```

```
## CV MAPE: 13.61 %
```

```r
cat("CV Lin MAPE:" , round(100*cv_lin_mape, 2), "%\n")
```

```
## CV Lin MAPE: 14.44 %
```

```r
cat("CV XGB MAPE:" , round(100*cv_xgb_mape, 2), "%\n")
```

```
## CV XGB MAPE: 13.43 %
```

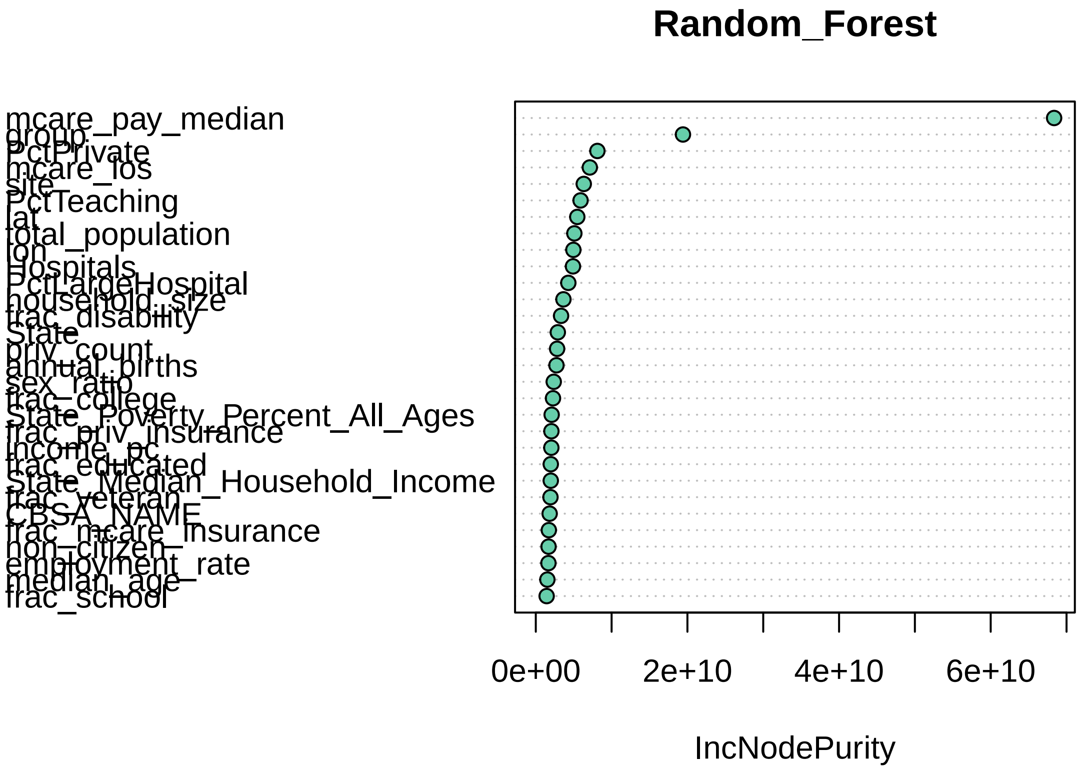
Cluster 2

```r
# 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 <- working_set %>%
  select(-priv_pay_mean, -priv_pay_iqr, -mcare_pay_mean, -mcare_pay_sd, -msa,-Cluster, -FIPS.State.Code
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,-Cluster)
rm(predict_set)

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

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

Model Creation and Prediction are now compartmentalized

```r
# Random Forest model

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

model = train(priv_pay_median ~ . , data=model_data, method = "rf", na.action = na.omit,trControl = tra

linear_model = train(priv_pay_median ~ . , data=model_data, method = "lm", na.action = na.omit,trContro

xgb_model = train(priv_pay_median ~ . , data=model_data, method = "xgbTree", na.action = na.omit,trCont


cv_lin_mod <- linear_model$pred

cv_lin_mape = MAPE(cv_lin_mod$pred, cv_lin_mod$obs)

cv_xgb_mod <- xgb_model$pred

cv_xgb_mape = MAPE(cv_xgb_mod$pred, cv_xgb_mod$obs)

cv_mod <- model$pred

cv_mape = MAPE(cv_mod$pred, cv_mod$obs)

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: 14.49 %
```

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

```
## Test MAPE: 21.39 %
```

```
cat("CV MAPE:" , round(100*cv_mape, 2), "%\n")
```

```
## CV MAPE: 13.6 %
```

```
cat("CV Lin MAPE:" , round(100*cv_lin_mape, 2), "%\n")
```

```
## CV Lin MAPE: 26.27 %
```

```
cat("CV XGB MAPE:" , round(100*cv_xgb_mape, 2), "%\n")
```

```
## CV XGB MAPE: 17.15 %
```