Assignment 3 Question 1

Nguyen Phuc Thai

2023-04-26

1 Loading data, packages and cleaning data

```
##loading packages
pacman::p_load(tidyverse, tidymodels)
```

```
## loading data
setwd("G:/My Drive/Adelaide uni/Stats7022/Assignment 3")
jobs<-readRDS('jobs.rds')

## separating the dataset containing only variable of interest
## also removing jobs with no annual salary as we cannot train
##model using data with no response
jobs1=jobs%>%select(state,job_type,sector,annual_salary,salary)%>%
  filter(!is.na(annual_salary))
```

Jobs with undisclosed salary will be removed as salary is the response for the model. Although this means removing most of the observations from the dataset (close to 90%), there are still more than 2000 observations for the model, which should be sufficient.

There are 8 jobs with annual salary that is 1.5 or lower. These values seem so erroneous that they are no more informative than missing values, hence will be removed from the data.

There are 64 jobs with annual salary disclosed to be between $9 and $105, these are treated mistyped hourly payments and hence will be re-cleaned accordingly. Similarly, there are 2 jobs with annual salary 624 and 775, respectively, which seems like weekly payment and will be fixed accordingly.

Jobs with unusually high salary will be kept (56 have salary of more than $1 million per year, a few even exceed $100 million (damnnn! sorry but can't help myself)). It is more likely to have exceedingly high-paid jobs than extremely low ones because of minimum wage laws. Hence, these values will be kept despite their may be error in them, as they may need to be inspected one by one to to verify whether they are wrong or not.

```r
## filter out all seemingly correct jobs
jobs_final=jobs1%>%
  filter(annual_salary>1000)

## jobs with hourly payment disclosed
jobs_hourly=jobs1%>%
  filter(annual_salary<=105 & annual_salary>2)

## jobs with weekly payment disclosed
jobs_weekly=jobs1%>%
  filter(annual_salary>200 & annual_salary<1000)

## for jobs with hourly pay wrongly disclosed, multiply the pay by
#1800 hours to get annualy payment, for weekly, multiply by 52
jobs_hourly$annual_salary=jobs_hourly$annual_salary*1800
jobs_weekly$annual_salary=jobs_weekly$annual_salary*52


## re-combine to a single dataset
jobs_final=rbind(jobs_hourly,jobs_weekly,jobs_final)
```

Some jobs has missing value in the sector and state columns, they will be replaced as 'Other' for this value combines all infrequent types of sector and state.

In addition, together with the column job_type, all factor levels that appear less than 1% of the time in the dataset will be lumped together as the level 'Other' as well. Although this has been done before, but with different combining criteria, there are still levels that appears so few times in the data that they cannot be informative enough.

```r
## converting the data to factor type and replacing missing values
jobs_final=jobs_final%>%
  select(-salary)%>%
  mutate(sector=as.factor(replace_na(sector,'Other')),
         state=as.factor(replace_na(state,'Other')),
         job_type=as.factor(job_type))

## lumping infrequent factors level into "Other"
jobs_final=jobs_final%>%
  mutate(sector=fct_lump_prop(sector,0.01),
         state=fct_lump_prop(state,0.01),
         job_type=fct_lump_prop(job_type,0.01))
```

2 Splitting data

The split is done with stratification based on the response such that the response in the train and test set are similar. Also, the default ratio is used, so the train set contains about 75% of the data

```
## setting seed for reproducibility
set.seed(1)

## data splitting
jobs_split <- initial_split(jobs_final, strata = annual_salary)
jobs_train <- training(jobs_split)
jobs_test <- testing(jobs_split)
```

3 Model tuning

3.1 Dataset for tuning, and tuning parameter

Since the training set has 2210 observations, cross validation methods with 5 folds should have enough data to be used for tuning. The parameter to be tuned is the number of predictors to be used to build a tree and minimum number of observation in a node for that node to continue to be split (min n).

```
# cross validation folds
folds <- vfold_cv(jobs_train, v = 5,  strata = annual_salary)

#tuning grid
rf_grid <- grid_regular(mtry(c(1,ncol(jobs_final)-1)),
                        min_n(),
                        levels = 5)
```

3.2 model recipe, workflow and tuning code

```
## the recipe
recipe_rf<- recipe(annual_salary~. ,data=jobs_train)

## the model
rf_model <- rand_forest(mtry = tune(),
                        min_n = tune(),
                        trees = 1000) %>%
  set_mode("regression") %>%
```

```
  set_engine("ranger")


### the workflow
wf_model=workflow()%>%
  add_model(rf_model)%>%
  add_recipe(recipe_rf)
```

```
## parameter tunnnig
doParallel::registerDoParallel()
rf_tune <- tune_grid(wf_model,
                         resamples = folds,
                         grid = rf_grid)
```

3.3 Tuning result

```
rf_tune%>%autoplot()
```

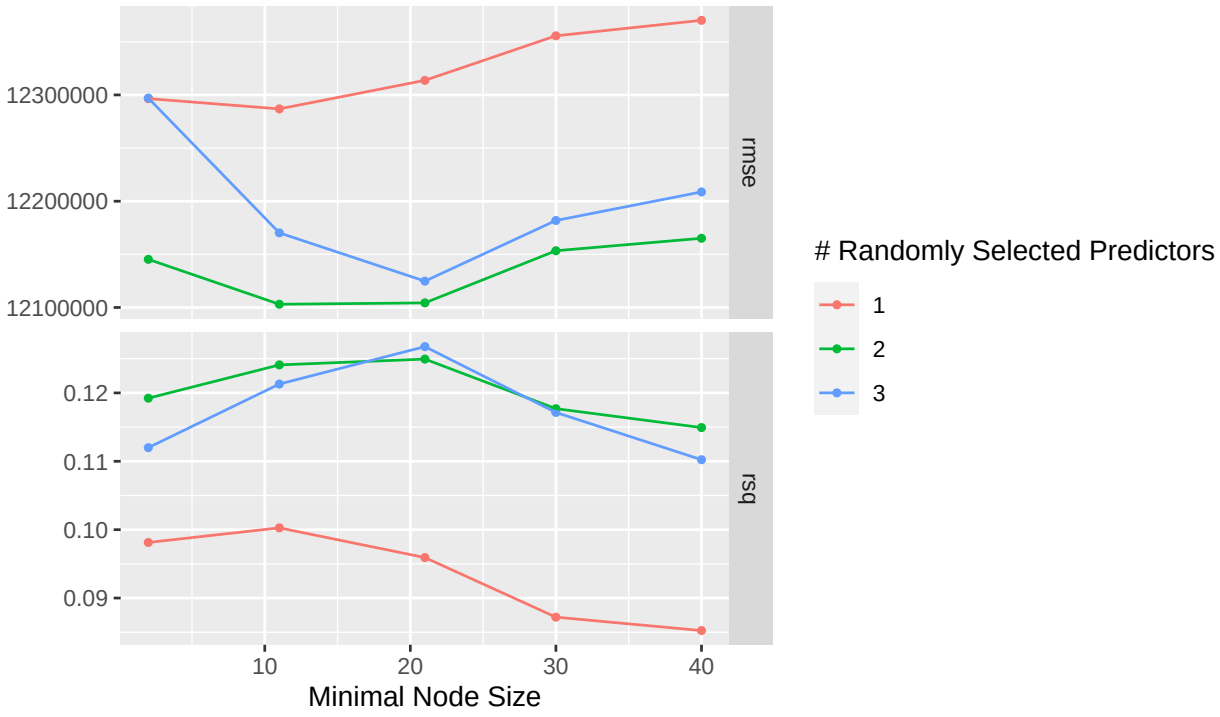

Figure 1: Tuning plots

Figure 1 shows that with the available predictors, the model is not fitting to the data well enough. The best model based on R squared criteria use 3 predictors to build trees and stop

splitting if a node have 21 observations, but only obtain R squared of less than 0.13. Models using 2 predictors obtain the best RMSE consistently, but the errror value is relatively high. The best model here will be chosen based on R square result, which is the model use 3 predictors to build tree and min n 21.

4 Fitting to the test data

```
## finalize the workflow with the best model from tuning
wf_model <- wf_model %>%
  finalize_workflow(select_best(rf_tune, metric = "rsq"))
## fitting the model again on all training data and use the model on
#the test data
rf_fit <- wf_model %>% last_fit(split = jobs_split)
```

```
## final model performance
rf_fit %>% collect_metrics()
```

```
## # A tibble: 2 x 4
##   .metric .estimator     .estimate .config
##   <chr>   <chr>              <dbl> <chr>
## 1 rmse    standard   17529939.     Preprocessor1_Model1
## 2 rsq     standard          0.0849 Preprocessor1_Model1
```

As expected from the tuning process, the model is not informative on the test data.

The model's performance is quite bad, which can partially be explained by the outliers (exceedingly high-paid jobs). However, the outliers still only make up of a very small percentages (56 jobs with more than $1 million annual pay out of more than 2000) of the data that simply removing them would not improve the model enough. For this problem, having better predictors will be more likely to have significant impact on the model.