Final Project

Brendan Gubbins

5/25/2021

Loading the data

```
rm(list=setdiff(ls(), c("lat", "lon")))

pacman::p_load(tidyverse, magrittr, data.table, R.utils)
housing = read.csv("C:\\Users\\Brendan Gubbins\\Desktop\\QC_MATH_342W_Spring_2021\\writing_assignments\
housing = as_tibble(housing)
```

Feature Selection

```
housing_data = housing %>%
  select(approx_year_built, cats_allowed, common_charges, coop_condo, date_of_sale, dining_room_type,
         dogs_allowed, fuel_type, full_address_or_zip_code, garage_exists, kitchen_type,
         maintenance_cost, num_bedrooms, num_floors_in_building, num_full_bathrooms, num_half_bathrooms
         parking_charges, sq_footage, total_taxes, sale_price, pct_tax_deductibl)

pacman::p_load(skimr)
skim(housing_data)
```

Table 1: Data summary

Name	housing_data
Number of rows	2230
Number of columns	21
Column type frequency:	
character	14
numeric	7
Group variables	None

Variable type: character

skim_variable	n_missing	complete_rate	min	max	empty	n_unique	whitespace
cats_allowed	0	1.00	1	3	0	3	0
common_charges	1684	0.24	4	7	0	258	0
coop_condo	0	1.00	5	5	0	2	0
date_of_sale	1702	0.24	8	10	0	222	0
dining_room_type	448	0.80	4	11	0	5	0
dogs_allowed	0	1.00	2	5	0	3	0
fuel_type	112	0.95	3	8	0	6	0
full_address_or_zip_code	0	1.00	5	59	0	1177	0
garage_exists	1826	0.18	1	11	0	6	0
kitchen_type	16	0.99	4	19	0	13	0
maintenance_cost	623	0.72	5	7	0	609	0
parking_charges	1671	0.25	3	5	0	89	0
total_taxes	1646	0.26	4	7	0	293	0
sale_price	1702	0.24	8	9	0	315	0

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
approx_year_built	40	0.98	1962.71	21.08	1893	1950	1958	1970	2017	
num_bedrooms	115	0.95	1.65	0.74	0	1	2	2	6	
num_floors_in_building	650	0.71	7.79	7.52	1	3	6	7	34	
num_full_bathrooms	0	1.00	1.23	0.44	1	1	1	1	3	
num_half_bathrooms	2058	0.08	0.95	0.30	0	1	1	1	2	
sq_footage	1210	0.46	955.36	380.86	100	743	881	1100	6215	
pct_tax_deductibl	1754	0.21	45.40	6.95	20	40	50	50	75	

Loading new data

```r
nyc_zip_codes = read.csv("C:\\Users\\Brendan Gubbins\\Downloads\\nyc-zip-codes.csv")

nyc_zip_codes %<>%
  select(ZipCode, Neighborhood) %>%
  rename(zip_code = ZipCode, neighborhood = Neighborhood)

# dropping an incomplete address
housing_data %<>%
  filter(full_address_or_zip_code != "11364")

housing_data %<>%
  mutate(zip_code = as.numeric(str_extract(substr(full_address_or_zip_code, 5, length(full_address_or_zi

housing_data = left_join(housing_data, nyc_zip_codes, by = "zip_code")
```

Converting date_of_sale into day_of_sale, month_of_sale. Also approx_year_built into decade_built

```
pacman::p_load(lubridate)

housing_data %<>%
  mutate(day_of_sale = day(mdy(date_of_sale)),
         month_of_sale = month(mdy(date_of_sale))) %>%
  select(-date_of_sale)
```

Converting cats_allowed, dogs_allowed into pets_allowed (binary), and garage_exists into binary

```
housing_data %<>%
  mutate(cats_allowed = if_else(cats_allowed == "no", 0, 1)) %>%
  mutate(dogs_allowed = if_else(dogs_allowed == "no", 0, 1)) %>%
  mutate(garage_exists = if_else(is.na(garage_exists), 0, 1)) %>%
  mutate(pets_allowed = if_else(cats_allowed == 1 | dogs_allowed == 1, 1, 0)) %>%
  select(-cats_allowed, -dogs_allowed)
```

Converting coop_condo, dining_room_type, fuel_type, kitchen_type into categorical features

```
housing_data %<>%
  mutate(coop_condo = as.factor(coop_condo))

housing_data %<>%
  mutate(dining_room_type = if_else(dining_room_type == "none" | dining_room_type == "dining area", "oth
  mutate(dining_room_type = as.factor(dining_room_type))

tabulate = sort(table(housing_data$fuel_type))

housing_data %<>%
  mutate(fuel_type = if_else(fuel_type %in% names(tabulate[tabulate < 62]), "other", fuel_type))

housing_data %<>%
  mutate(fuel_type = as.factor(fuel_type))

housing_data %<>%
  mutate(kitchen_type = if_else(kitchen_type == "eat in" | kitchen_type == "Eat in" | kitchen_type == "
  mutate(kitchen_type = if_else(kitchen_type == "combo" | kitchen_type == "Combo", "combo", kitchen_type
  mutate(kitchen_type = if_else(kitchen_type == "efficiency" | kitchen_type == "efficiemcy" | kitchen_ty
                                | kitchen_type == "efficiency kitchene" | kitchen_type == "efficiency ki
housing_data %<>%
  mutate(kitchen_type = as.factor(kitchen_type))

housing_data = housing_data[housing_data$kitchen_type != "1955",]
```

Cleaning `parking_charges`, `total_taxes`, `common_charges`, `sale_price`, `maintenance_cost`

```
housing_data %<>%
  mutate(parking_charges = as.numeric(gsub("[\\$,]", "", parking_charges)),
         parking_charges = if_else(is.na(parking_charges), 0, parking_charges))

housing_data %<>%
  mutate(total_taxes = as.numeric(gsub("[\\$,]", "", total_taxes)))

# condos pay charges

housing_data %<>%
  mutate(common_charges = as.numeric(gsub("[\\$,]", "", common_charges)),
         common_charges = if_else(is.na(common_charges) & coop_condo == "co-op", 0, common_charges))

housing_data %<>%
  mutate(sale_price = as.numeric(gsub("[\\$,]", "", sale_price)))

housing_data = housing_data[!is.na(housing_data$sale_price),]

# co-ops pay maintenance

housing_data %<>%
  mutate(maintenance_cost = as.numeric(gsub("[\\$,]", "", maintenance_cost)),
         maintenance_cost = if_else(is.na(maintenance_cost) & coop_condo == "condo", 0, maintenance_cos
```

`pct_tax_deductibl` applies to co-op only

```
housing_data %<>%
  mutate(pct_tax_deductibl = if_else(housing_data$coop_condo == "condo", 0, as.numeric(pct_tax_deductib
```

Converting `NA` half bathrooms into `0`

```
housing_data %<>%
  mutate(num_half_bathrooms = if_else(is.na(num_half_bathrooms), 0, as.numeric(num_half_bathrooms)))
```

Converting `sq_footage` to percentiles

```
# correcting an error
housing_data %<>%
  mutate(sq_footage = if_else(sq_footage > 6000, 1200, as.numeric(sq_footage)))

housing_data %<>%
  mutate(sq_footage = ecdf(sq_footage)(sq_footage))
```

Geocoding

```r
latlon = geocode(housing_data$full_address_or_zip_code, output = "latlon")
lat = latlon$lat
lon = latlon$lon

housing_data %<>%
  mutate(latitude = lat,
         longitude = lon)

# grand central terminal
gc_coords = c(40.7527, 73.9772)
grand_central = array(NA, nrow(housing_data))

for (i in 1 : nrow(housing_data)) {
  grand_central[i] = distm(gc_coords, c(abs(housing_data$latitude[i]), abs(housing_data$longitude[i])),
}

housing_data %<>%
  mutate(grand_central = grand_central)
```

Missingness Dummy Variables

```r
M = tbl_df(apply(is.na(housing_data), 2, as.numeric))
```

```
## Warning: `tbl_df()` was deprecated in dplyr 1.0.0.
## Please use `tibble::as_tibble()` instead.
```

```r
colnames(M) = paste("is_missing_", colnames(housing_data), sep = "")
M %<>%
  select_if(function(x){sum(x) > 0})

M = tbl_df(t(unique(t(M))))

housing_data %<>%
  relocate(sale_price)

housing_data = cbind(housing_data, M)
```

Train-Test Split

```r
K = 5
test_prop = 1 / K
train_indices = sample(1 : nrow(housing_data), round((1 - test_prop) * nrow(housing_data)))
housing_train = housing_data[train_indices, ]
y_train = housing_train$sale_price
X_train = housing_train
X_train$sale_price = NULL
```

```
test_indices = setdiff(1 : nrow(housing_data), train_indices)
housing_test = housing_data[test_indices, ]
y_test = housing_test$sale_price
X_test = housing_test
X_test$sale_price = NULL
```

Imputation with missForest

```
pacman::p_load(missForest)

train_bind = cbind(X_train, sale_price = y_train)
test_bind = cbind(X_test, sale_price = NA)
X_bind = rbind(train_bind, test_bind)

X_bind$full_address_or_zip_code = NULL
neighborhood = X_bind$neighborhood
X_bind$neighborhood = NULL

Ximp = missForest(as.data.frame(X_bind))$ximp
```

```
##   missForest iteration 1 in progress...done!
##   missForest iteration 2 in progress...done!
##   missForest iteration 3 in progress...done!
##   missForest iteration 4 in progress...done!
##   missForest iteration 5 in progress...done!
##   missForest iteration 6 in progress...done!
```

```
# if coop, discount
Ximp %<>%
  mutate(monthly_cost = if_else(coop_condo == "co-op",
                                (maintenance_cost + common_charges) * (100 - pct_tax_deductibl) / 100,
                                maintenance_cost + common_charges))
Ximp %<>%
  select(-sale_price, -maintenance_cost, -common_charges, -zip_code) # -pct_tax_deductibl

Ximp = cbind(Ximp, neighborhood = as.factor(neighborhood), sale_price = X_bind$sale_price)

Ximp %<>%
  select(-pct_tax_deductibl, -is_missing_pct_tax_deductibl)


X_train = Ximp[!is.na(Ximp$sale_price),]
X_train$sale_price = NULL

X_test = Ximp[is.na(Ximp$sale_price),]
X_test$sale_price = NULL
```

Regression Tree

```
pacman::p_load(YARF)
```

```
## YARF can now make use of 11 cores.
```

```
tree_mod = YARFCART(X_train, y_train)
```

```
## YARF initializing with a fixed 1 trees...
## YARF factors created...
## YARF after data preprocessed... 42 total features...
## Beginning YARF regression model construction...done.
## Calculating OOB error...done.
```

```
illustrate_trees(tree_mod, max_depth = 5, length_in_px_per_half_split = 40, open_file = TRUE)

tree_mod
```

```
## YARF v1.1 for regression
## Missing data feature ON.
## 1 trees, training data n = 416 and p = 42
## Model construction completed within 0 minutes.
## No OOB results to show (no trees have been fit as of yet).
```

Linear Regression

```
mod = lm(y_train ~ ., X_train)

summary(mod)$r.squared
```

```
## [1] 0.8578943
```

```
sqrt(mean(mod$residuals^2))
```

```
## [1] 68342.96
```

```
y_hat = predict(mod, X_test)

residuals = y_test - y_hat
sqrt(mean(residuals^2))
```

```
## [1] 85623.27
```

Random Forest

```r
rf_mod = YARF(X_train, y_train)
```

```
## YARF initializing with a fixed 500 trees...
## YARF factors created...
## YARF after data preprocessed... 42 total features...
## Beginning YARF regression model construction...done.
## Calculating OOB error...done.
```

```r
rf_mod
```

```
## YARF v1.1 for regression
## Missing data feature ON.
## 500 trees, training data n = 416 and p = 42
## Model construction completed within 0.02 minutes.
## OOB results on all observations:
##   R^2: 0.82713
##   RMSE: 75379.16
##   MAE: 53512.71
##   L2: 2.36372e+12
##   L1: 22261287
```

```r
y_hat = predict(rf_mod, X_test)

residuals = y_test - y_hat
oos_rmse = sqrt(mean(residuals^2))
oos_rmse
```

```
## [1] 83359.95
```