California Housing Prices

1. Overview

This is about the California median Housing Prices of 1990 of California districts. This data is available at https://www.kaggle.com/camnugent/california-housing-prices. There are different variables on which the housing prices depend upon, but here in this dataset we have nine variables. There are ten variables altogether including the outcome one. The outcome (dependent) variable is median_house_value (median price of house), and rest of the variables are the attributes of the outcome variable. The summary of the data gives us the datatypes of each variable and sheds some light into the dataset itself. The summary is as follows

```
## Loading required package: tidyverse
```

```
## -- Attaching packages ------------------------------------------------------------ tidyverse 1.3.0 --
```

```
## v ggplot2 3.3.0     v purrr   0.3.4
## v tibble  3.0.1     v dplyr   0.8.5
## v tidyr   1.0.3     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.5.0
```

```
## -- Conflicts --------------------------------------------------------------- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
## Loading required package: caret
```

```
## Loading required package: lattice
```

```
##
## Attaching package: 'caret'
```

```
## The following object is masked from 'package:purrr':
##
##     lift
```

```
##   longitude latitude housing_median_age total_rooms total_bedrooms population
## 1   -122.23    37.88                 41         880            129        322
## 2   -122.22    37.86                 21        7099           1106       2401
## 3   -122.24    37.85                 52        1467            190        496
## 4   -122.25    37.85                 52        1274            235        558
## 5   -122.25    37.85                 52        1627            280        565
## 6   -122.25    37.85                 52         919            213        413
##   households median_income median_house_value ocean_proximity
## 1        126        8.3252             452600        NEAR BAY
## 2       1138        8.3014             358500        NEAR BAY
## 3        177        7.2574             352100        NEAR BAY
## 4        219        5.6431             341300        NEAR BAY
## 5        259        3.8462             342200        NEAR BAY
## 6        193        4.0368             269700        NEAR BAY
## [1] 20640
```

```
##    longitude         latitude     housing_median_age  total_rooms
##  Min.   :-124.3   Min.   :32.54   Min.   : 1.00      Min.   :    2
##  1st Qu.:-121.8   1st Qu.:33.93   1st Qu.:18.00      1st Qu.: 1448
##  Median :-118.5   Median :34.26   Median :29.00      Median : 2127
##  Mean   :-119.6   Mean   :35.63   Mean   :28.64      Mean   : 2636
##  3rd Qu.:-118.0   3rd Qu.:37.71   3rd Qu.:37.00      3rd Qu.: 3148
##  Max.   :-114.3   Max.   :41.95   Max.   :52.00      Max.   :39320
##
##  total_bedrooms     population      households     median_income
##  Min.   :   1.0   Min.   :    3   Min.   :   1.0   Min.   : 0.4999
##  1st Qu.: 296.0   1st Qu.:  787   1st Qu.: 280.0   1st Qu.: 2.5634
##  Median : 435.0   Median : 1166   Median : 409.0   Median : 3.5348
##  Mean   : 537.9   Mean   : 1425   Mean   : 499.5   Mean   : 3.8707
##  3rd Qu.: 647.0   3rd Qu.: 1725   3rd Qu.: 605.0   3rd Qu.: 4.7432
##  Max.   :6445.0   Max.   :35682   Max.   :6082.0   Max.   :15.0001
##  NA’s   :207
##  median_house_value   ocean_proximity
##  Min.   : 14999     <1H OCEAN :9136
##  1st Qu.:119600     INLAND    :6551
##  Median :179700     ISLAND    :   5
##  Mean   :206856     NEAR BAY  :2290
##  3rd Qu.:264725     NEAR OCEAN:2658
##  Max.   :500001
##
```

```
## ’data.frame’:    20640 obs. of  10 variables:
##  $ longitude         : num  -122 -122 -122 -122 -122 ...
##  $ latitude          : num  37.9 37.9 37.9 37.9 37.9 ...
##  $ housing_median_age: num  41 21 52 52 52 52 52 52 42 52 ...
##  $ total_rooms       : num  880 7099 1467 1274 1627 ...
##  $ total_bedrooms    : num  129 1106 190 235 280 ...
##  $ population        : num  322 2401 496 558 565 ...
##  $ households        : num  126 1138 177 219 259 ...
##  $ median_income     : num  8.33 8.3 7.26 5.64 3.85 ...
##  $ median_house_value: num  452600 358500 352100 341300 342200 ...
##  $ ocean_proximity   : Factor w/ 5 levels "<1H OCEAN","INLAND",..: 4 4 4 4 4 4 4 4 4 4 ...
```

It is the data of 20640 blocks in California, where each row belongs to a block. We have longitude, latitude, housing median age (housing_median_age), total rooms (total_rooms), population, households, median income (median_income), median house value (median_house_value), and ocean proximity (ocean_proximity) of each block of housing. All variables are predictors except ‘median_house_value’, where the median_house_value is the outcome (dependent) variable. All variables are numeric, except for ocean_proximity, which is a categorical variable describing each block how far it is from the ocean. It has five categories values, where each of the values are obvious by its label, and it shows some blocks are in island, as ocean_proximity’s label is ‘Island’. The label ‘<1H OCEAN’ signifies that the given block in the dataset is less than an hour drive from the ocean. The summary of the dataset you saw above gives you the feel of the data. There are 207 NA’s in the variable ‘total_bedrooms’. Except ocean_proximity, all other variables are numeric. Longitude is in negative numbers since the area lies in the West of Prime Meridian.

The goal of this project is to find a best model which can predict the median house value in best way possible. In other words, it is finding a model which has least RMSE (Root Mean Square Error), which I have defined in the following section. At first, I will be using linear regression then different methods available in train function to estimate the RMSE. For each model I will estimate RMSE because we know that the smaller the RMSE, the better the model is in predicting the median house value. I will finally use advanced machine learning techniques to estimate the RMSE.

2. Methods

There are 207 NAs in total_bedrooms column, and I will remove those corresponding rows. As we have good number of observations, removing them will not make much difference in our estimates/results.

Next thing, I want to do is introducing dummy variables for the categorical variable, ocean_proximity so that I could use this variable for calculating the correlation coefficients with the outcome variable, and with other independent variables. Dummy variable is set to 1 for the category we are setting dummy for, and 0 otherwise. The five dummy variables are Inland, Near_Ocean, Near_Bay, Island and Less_One_Hr. They are the dummies for those housings which are at inland, near to ocean, near to bay, at Island, and in less than an hour drive from ocean, respectively. Let's observe the correlations among different variables.

```
## Warning in set.seed(1, sample.kind = "Rounding"): non-uniform 'Rounding' sampler
## used
```

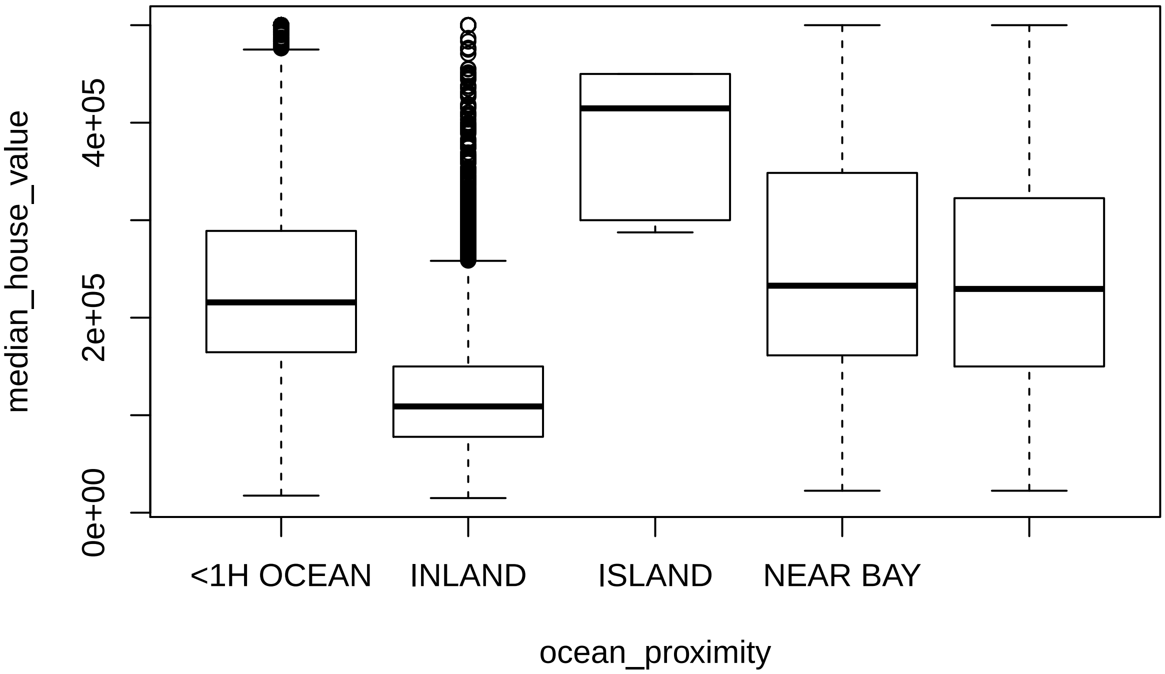
```
##                      longitude    latitude housing_median_age  total_rooms
## longitude           1.00000000 -0.92389539        -0.10920514  0.045828766
## latitude           -0.92389539  1.00000000         0.01126427 -0.037298778
## housing_median_age -0.10920514  0.01126427         1.00000000 -0.358887428
## total_rooms         0.04582877 -0.03729878        -0.35888743  1.000000000
## total_bedrooms      0.06886319 -0.06673022        -0.31942542  0.931063755
## population          0.10011504 -0.10916866        -0.29511054  0.857946342
## households          0.05623293 -0.07210604        -0.30208751  0.919461478
## median_income      -0.01250650 -0.08237451        -0.11466967  0.197016736
## median_house_value -0.04452841 -0.14549713         0.10874197  0.134607670
## Inland             -0.05524042  0.35219348        -0.23711672  0.024583054
## Near_Ocean          0.04575514 -0.16126026         0.02258718 -0.006868114
## Near_Bay           -0.47647682  0.35982561         0.25400725 -0.023721478
## Island              0.01002218 -0.01758136         0.01795885 -0.007972647
## Less_One_Hr         0.32316309 -0.44987110         0.04549401 -0.003129523
##                    total_bedrooms   population   households median_income
## longitude             0.068863189  0.100115045  0.056232929  -0.012506502
## latitude             -0.066730219 -0.109168665 -0.072106037  -0.082374507
## housing_median_age   -0.319425415 -0.295110541 -0.302087507  -0.114669675
## total_rooms           0.931063755  0.857946342  0.919461478   0.197016736
## total_bedrooms        1.000000000  0.877058566  0.979805642  -0.006209130
## population            0.877058566  1.000000000  0.906391131   0.008418537
## households            0.979805642  0.906391131  1.000000000   0.015788671
## median_income        -0.006209130  0.008418537  0.015788671   1.000000000
## median_house_value    0.054323386 -0.022146534  0.069763071   0.685076749
## Inland               -0.009056671 -0.021349439 -0.041243026  -0.236952794
## Near_Ocean            0.002370114 -0.022461192  0.004286969   0.025025106
## Near_Bay             -0.019685472 -0.061121194 -0.011076466   0.057981179
## Island               -0.004590809 -0.010937776 -0.009583451  -0.009771729
## Less_One_Hr           0.019551594  0.074286485  0.043165698   0.169039074
##                    median_house_value       Inland   Near_Ocean     Near_Bay
## longitude                 -0.04452841 -0.055240419  0.045755140 -0.476476815
## latitude                  -0.14549713  0.352193479 -0.161260264  0.359825610
## housing_median_age         0.10874197 -0.237116723  0.022587183  0.254007252
## total_rooms              0.13460767  0.024583054 -0.006868114 -0.023721478
## total_bedrooms           0.05432339 -0.009056671  0.002370114 -0.019685472
## population              -0.02214653 -0.021349439 -0.022461192 -0.061121194
## households               0.06976307 -0.041243026  0.004286969 -0.011076466
## median_income            0.68507675 -0.236952794  0.025025106  0.057981179
## median_house_value       1.00000000 -0.484879331  0.139723504  0.160881028
## Inland                  -0.48487933  1.000000000 -0.261659997 -0.242658881
## Near_Ocean               0.13972350 -0.261659997  1.000000000 -0.135786406
## Near_Bay                 0.16088103 -0.242658881 -0.135786406  1.000000000
## Island                   0.02481008 -0.011276340 -0.006309984 -0.005851769
## Less_One_Hr              0.25811789 -0.608046798 -0.340249196 -0.315541123
##                          Island  Less_One_Hr
## longitude           0.010022180  0.323163092
## latitude           -0.017581357 -0.449871103
## housing_median_age  0.017958852  0.045494013
## total_rooms        -0.007972647 -0.003129523
## total_bedrooms     -0.004590809  0.019551594
## population         -0.010937776  0.074286485
## households         -0.009583451  0.043165698
## median_income      -0.009771729  0.169039074
## median_house_value  0.024810077  0.258117889
## Inland             -0.011276340 -0.608046798
## Near_Ocean         -0.006309984 -0.340249196
## Near_Bay           -0.005851769 -0.315541123
## Island              1.000000000 -0.014663173
## Less_One_Hr        -0.014663173  1.000000000
```

The dependent variable, median_house_value is positively correlated with housing_median_age, total_rooms, total_bedrooms, households, median_income, Near_Ocean, Near_Bay, Island and Less_One_Hr. The positive relationship means the direction of relationship between two variables are in the same direction, i.e. if one increases then the next also increases and vice-versa. For example, the more the total_rooms, the higher the median house value in the block, and so on. The negative correlation exists between the outcome and these variables: longitude, latitude, population and Inland, i.e. the relationship is in negative direction. For example, if the house is at inland, its value is lower.

The correlations between four different predictors are stronger than among the others. The correlation between total_rooms, total_bedrooms, population and households are really high, and these variables will be combined into some principal components for the analysis if time permits. This positive correlations make sense since more total_rooms means more total_bedrooms, more people and more households.

I create additional column in the dataset by dividing data in a hundred different categories one for each independent variable: longitude, latitude, housing_median_age, total_rooms, total_bedrooms, population, households and median_income. These additional columns will help us in using advanced machine learning .

Let us see the distribution of median house value (median_house_value) as per the ocean proximity (ocean_proximity



The above boxplot shows the median value of houses on island are highest among all categories, whereas the median value of houses at inland is lowest, and other categories fall in-between.

From the dataset we know we can predict the median housing value using the predictors given there. To do so, first, I will run linear regression and estimate RMSE. We know the lower the RMSE, the better is the model in predicting the outcome variable. Then I will use other machine learning techniques like glm (generalized linear model), knn (k-nearest neighbors), rpart (in short, regression trees) available in train function. Each model is for predicting the outcome variable, and based on that prediction we can calculate the RMSE which is defined as

```
RMSE  <- function(predicted, true_value) {

  sqrt(mean((predicted - true_value)^2))

}
```

, where the true_value is the median_house_value in the test_set, since we have two sets of datasets after partitioning the given dataset: one for training our models, the train_set, and the next set is the test_set to evaluate the performance of our model. The predicted values are values predicted from the model in the test dataset, and RMSE is the square root of the mean of squared difference between the predicted and true values.

Then I will use advanced machine learning technique like I learned in machine learning course for movie rating system. The crux of machine learning is to train models using train dataset, and use the trained model to predict the outcome variable in the test dataset. First, I will assume the predicted median house value is the average of all median house values present in the train dataset. I will calculate RMSE based on that predicted value comparing that value with the median house value present in the test dataset. I will build on this algorithm by adding an additional independent variable at a time. The first independent variable I will add on the algorithm is median_income. To make use of median_income variable easier in this machine learning technique, I have its category variable, per_md_incm. Such categorization is justified since each such category impacts the prediction of median house value in different amounts and helps to improve the prediction. The same logic applies for the categorization of other independent variables down the road. I will use per_md_incm variable in estimating its values for different category of it (from train dataset) by taking the mean of the difference between median_house_value and the average (average of median_house_value). For detail, you can see the code in the results section below. Now, using the average and these different category values of median_income, I predict the median_house_values using the test dataset. Again, I calculate RMSE from those predicted values and its corresponding observed values in the test dataset. We keep on adding one independent variable after other in the algorithm since each one is correlated with the outcome variable. Then, I compare RMSE of the different algorithms, and the one with the least RMSE is the best for predicting the median house value.

3. Results

As we know we have to predict the median_house_value, and other variables present in the dataset are its attributes. Let us run a simple linear regression. The RMSE calculated from linear regression model is 67007. The RMSEs calculated using the train functions using different methods are all higher than what we got from simple linear regression model. The highest RMSE we got is from Knn method. Thus far, the simple linear regression model has the lowest RMSE, 67007. See the code for how different techniques are used for the results.

```
##
## Call:
## lm(formula = median_house_value ~ total_rooms + total_bedrooms +
##     housing_median_age + population + households + median_income +
##     longitude + latitude + ocean_proximity, data = train_set)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -556447  -42807  -10625   28887  778796
##
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)              -2.227e+06  9.245e+04 -24.086  < 2e-16 ***
## total_rooms              -6.512e+00  8.319e-01  -7.828 5.22e-15 ***
## total_bedrooms            1.039e+02  7.261e+00  14.312  < 2e-16 ***
## housing_median_age        1.074e+03  4.624e+01  23.232  < 2e-16 ***
## population               -3.808e+01  1.127e+00 -33.785  < 2e-16 ***
## households                4.885e+01  7.849e+00   6.225 4.93e-10 ***
## median_income             3.922e+04  3.563e+02 110.058  < 2e-16 ***
## longitude                -2.625e+04  1.071e+03 -24.521  < 2e-16 ***
## latitude                 -2.481e+04  1.055e+03 -23.517  < 2e-16 ***
## ocean_proximityINLAND     -4.003e+04  1.843e+03 -21.724  < 2e-16 ***
## ocean_proximityISLAND      1.531e+05  3.083e+04   4.966 6.91e-07 ***
## ocean_proximityNEAR BAY   -4.443e+03  2.018e+03  -2.201  0.02771 *
## ocean_proximityNEAR OCEAN  4.472e+03  1.664e+03   2.688  0.00719 **
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 68840 on 18379 degrees of freedom
## Multiple R-squared:  0.6441, Adjusted R-squared:  0.6439
## F-statistic:  2772 on 12 and 18379 DF,  p-value: < 2.2e-16
```

```
## Warning: ‘data_frame()‘ is deprecated as of tibble 1.1.0.
## Please use ‘tibble()‘ instead.
## This warning is displayed once every 8 hours.
## Call ‘lifecycle::last_warnings()‘ to see where this warning was generated.
## Warning in nominalTrainWorkflow(x = x, y = y, wts = weights, info = trainInfo, :
## There were missing values in resampled performance measures.
```

All RMSEs calculated by far are listed below

```
## # A tibble: 6 x 2
##   method                RMSEs
##   <chr>                 <dbl>
## 1 RMSE_lm              67007.
## 2 RMSE from train_lm   68936.
## 3 RMSE_glmm            69149.
## 4 RMSE from Knn       100710.
## 5 RMSE from Knn_cv     95263.
## 6 RMSE from Rpart      85233.
```

Now, let us design the algorithm on our own and see how RMSE fares. First of all, assuming the predicted median_house_value is the average of all median_house_value present in the train dataset. Now let us see how RMSE does in this case. The RMSE is 116,141 in this case. Let us keep on adding one independent variable after other and see how RMSE changes. The RMSE kept on decreasing as we kept on adding additional independent variable each time.

```
# Using mu as the predicted median_house_value

mu <- mean(train_set$median_house_value)
RMSE1 <- RMSE(mu, test_set$median_house_value)

rmse_results <- bind_rows(rmse_results,
                          data_frame(method="Mean only",
                                     RMSEs = RMSE1 ))
rmse_results
```

```
## # A tibble: 7 x 2
##   method                RMSEs
##   <chr>                 <dbl>
## 1 RMSE_lm              67007.
## 2 RMSE from train_lm   68936.
## 3 RMSE_glmm            69149.
## 4 RMSE from Knn       100710.
## 5 RMSE from Knn_cv     95263.
## 6 RMSE from Rpart      85233.
## 7 Mean only           116141.
```

```
# RMSE1 is 116140.7

# Including median_income's category variable in the process

# Estimating incm_i for each category per_md_incm, a category variable for median_income variable
md_incm_avg <- train_set %>% group_by(per_md_incm)  %>%
  summarise(incm_i= mean( median_house_value - mu))

# Predicting from the test_set, based on the above estimation, from the train_set
predicted <- mu + test_set %>%
  left_join(md_incm_avg, by=’per_md_incm’) %>%
  .$incm_i


RMSE2 <- RMSE(predicted, test_set$median_house_value)

rmse_results <- bind_rows(rmse_results,
                          data_frame(method="Mean plus median income avg",
                                     RMSEs = RMSE2 ))
rmse_results
```

```
## # A tibble: 8 x 2
##   method                           RMSEs
##   <chr>                            <dbl>
## 1 RMSE_lm                         67007.
## 2 RMSE from train_lm              68936.
## 3 RMSE_glmm                       69149.
## 4 RMSE from Knn                  100710.
## 5 RMSE from Knn_cv                95263.
## 6 RMSE from Rpart                 85233.
## 7 Mean only                      116141.
## 8 Mean plus median income avg     80047.
```

```
# RMSE2 is 80046.91

 # Adding ocean_proximity in the process
ocn_avgs <- train_set %>%
  left_join(md_incm_avg, by=’per_md_incm’) %>%
  group_by(ocean_proximity) %>%
  summarize(b_ocn = mean(median_house_value - mu - incm_i))


predicted<- test_set %>%
  left_join(md_incm_avg, by=’per_md_incm’) %>%
  left_join(ocn_avgs, by=’ocean_proximity’) %>%
  mutate(pred = mu + incm_i + b_ocn) %>%
  .$pred

RMSE3 <- RMSE(predicted, test_set$median_house_value)

rmse_results <- bind_rows(rmse_results,
                          data_frame(method="Mean, median income avg and ocn_proximity",
                                     RMSEs = RMSE3 ))
rmse_results
```

```
## # A tibble: 9 x 2
##   method                                         RMSEs
##   <chr>                                          <dbl>
## 1 RMSE_lm                                       67007.
## 2 RMSE from train_lm                            68936.
## 3 RMSE_glmm                                     69149.
## 4 RMSE from Knn                                100710.
## 5 RMSE from Knn_cv                              95263.
## 6 RMSE from Rpart                            85233.
## 7 Mean only                                 116141.
## 8 Mean plus median income avg                80047.
## 9 Mean, median income avg and ocn_proximity  70855.
```

```
# RMSE3 is 70854.93

# Adding latitude’s category variable in the process
lat_avgs <- train_set %>%
  left_join(md_incm_avg, by=’per_md_incm’) %>%
  left_join(ocn_avgs, by = ’ocean_proximity’) %>%
  group_by(per_lat) %>%
  summarize(b_lat = mean(median_house_value - mu - incm_i- b_ocn))


predicted <- test_set %>%
  left_join(md_incm_avg, by=’per_md_incm’) %>%
  left_join(ocn_avgs, by=’ocean_proximity’) %>%
  left_join(lat_avgs, by = ’per_lat’) %>%
  mutate(pred = mu  + incm_i + b_ocn +  b_lat) %>%
  .$pred

RMSE4 <- RMSE(predicted, test_set$median_house_value)
rmse_results <- bind_rows(rmse_results,
                          data_frame(method="Mean, median income avg, ocn_proximity and latitude",
                                     RMSEs = RMSE4))
rmse_results
```

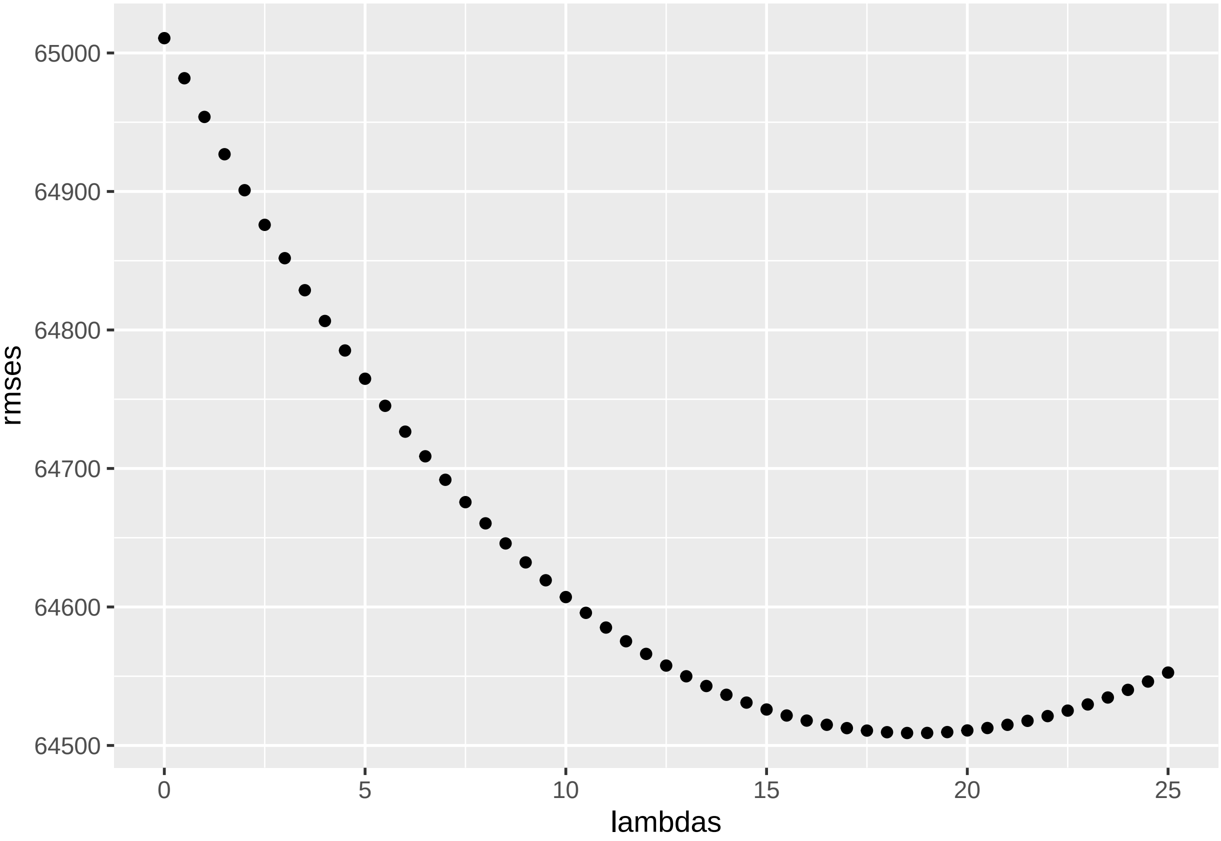
```
## # A tibble: 10 x 2
##    method                                                RMSEs
##    <chr>                                                 <dbl>
##  1 RMSE_lm                                              67007.
##  2 RMSE from train_lm                                   68936.
##  3 RMSE_glmm                                            69149.
##  4 RMSE from Knn                                       100710.
##  5 RMSE from Knn_cv                                     95263.
##  6 RMSE from Rpart                                      85233.
##  7 Mean only                                           116141.
##  8 Mean plus median income avg                          80047.
##  9 Mean, median income avg and ocn_proximity            70855.
## 10 Mean, median income avg, ocn_proximity and latitude  66492.
```

```
# RMSE4 is 66491.92


# Adding household median age in the process
md_age_avgs <- train_set %>%
  left_join(md_incm_avg, by=’per_md_incm’) %>%
  left_join(ocn_avgs, by = ’ocean_proximity’) %>%
  left_join(lat_avgs, by = ’per_lat’) %>%
  group_by(per_hs_md_age) %>%
  summarize(b_md_age = mean(median_house_value - mu - incm_i- b_ocn- b_lat))
```

```
predicted <- test_set %>%
  left_join(md_incm_avg, by='per_md_incm') %>%
  left_join(ocn_avgs, by='ocean_proximity') %>%
  left_join(lat_avgs, by = 'per_lat') %>%
  left_join(md_age_avgs, by ='per_hs_md_age') %>%
  mutate(pred = mu  + incm_i + b_ocn +  b_lat + b_md_age) %>%
  .$pred

RMSE5 <- RMSE(predicted, test_set$median_house_value)
rmse_results <- bind_rows(rmse_results,
                          data_frame(method="Mean, median incone avg, ocn_proximity, latitue and housin
                                     RMSEs = RMSE5 ))
rmse_results
```

```
## # A tibble: 11 x 2
##    method                                                                RMSEs
##    <chr>                                                                 <dbl>
##  1 RMSE_lm                                                              6.70e4
##  2 RMSE from train_lm                                                   6.89e4
##  3 RMSE_glmm                                                            6.91e4
##  4 RMSE from Knn                                                        1.01e5
##  5 RMSE from Knn_cv                                                     9.53e4
##  6 RMSE from Rpart                                                      8.52e4
##  7 Mean only                                                            1.16e5
##  8 Mean plus median income avg                                          8.00e4
##  9 Mean, median income avg and ocn_proximity                            7.09e4
## 10 Mean, median income avg, ocn_proximity and latitude                  6.65e4
## 11 Mean, median incone avg, ocn_proximity, latitue and housing median age 6.50e4
```

```
# RMSE5 is 65010.69

# Using regularization

lambdas <- seq(0, 25, 0.5) # setting lambda's sequence value
rmses <- sapply(lambdas, function(x){
  incm_i <- train_set %>%
    group_by(per_md_incm) %>%
    summarize(incm_i = sum(median_house_value - mu)/(n()+x))
  b_ocn <- train_set %>%
    left_join(incm_i, by="per_md_incm") %>%
    group_by(ocean_proximity) %>%
    summarize(b_ocn = sum(median_house_value - incm_i - mu)/(n()+x))

  b_lat <- train_set %>%
    left_join(md_incm_avg, by = 'per_md_incm') %>%
    left_join(ocn_avgs, by = 'ocean_proximity') %>%
    group_by(per_lat) %>%
    summarize(b_lat = sum(median_house_value - mu - incm_i- b_ocn)/(n()+x))

b_md_age <- train_set %>%
  left_join(md_incm_avg, by='per_md_incm') %>%
  left_join(ocn_avgs, by = 'ocean_proximity') %>%
  left_join(lat_avgs, by = 'per_lat') %>%
  group_by(per_hs_md_age) %>%
  summarize(b_md_age = sum(median_house_value - mu - incm_i- b_ocn- b_lat)/(n()+x))

  predicted <- test_set %>%
    left_join(incm_i, by = 'per_md_incm') %>%
    left_join(b_ocn, by = 'ocean_proximity') %>%
    left_join(b_lat, by = 'per_lat') %>%
    left_join(b_md_age, by ='per_hs_md_age') %>%
    mutate(pred = mu + incm_i + b_ocn + b_lat + b_md_age) %>%
    .$pred
  return(RMSE(predicted, test_set$median_house_value))
})

qplot(lambdas, rmses)  # Q-Plot
```



```
lambda <- lambdas[which.min(rmses)] # Optimal lambda
lambda # Printing out Optimal lambda, and it is 18.5
```

```
## [1] 18.5
```

```
RMSE_reg <- rmses[which.min(rmses)] # minimum rmses, and it is 64509.01
rmse_results <- bind_rows(rmse_results,
                          data_frame(method=" Reg_RMSE",
                                     RMSEs = RMSE_reg ))
rmse_results
```

```
## # A tibble: 12 x 2
##    method                                                                    RMSEs
##    <chr>                                                                     <dbl>
##  1 "RMSE_lm"                                                                6.70e4
##  2 "RMSE from train_lm"                                                     6.89e4
##  3 "RMSE_glmm"                                                              6.91e4
##  4 "RMSE from Knn"                                                          1.01e5
##  5 "RMSE from Knn_cv"                                                       9.53e4
##  6 "RMSE from Rpart"                                                        8.52e4
##  7 "Mean only"                                                              1.16e5
##  8 "Mean plus median income avg"                                            8.00e4
##  9 "Mean, median income avg and ocn_proximity"                              7.09e4
## 10 "Mean, median income avg, ocn_proximity and latitude"                    6.65e4
## 11 "Mean, median incone avg, ocn_proximity, latitue and housing median a~ 6.50e4
## 12 " Reg_RMSE"                                                              6.45e4
```

```
# Displaying all stored RMSEs by far in its descending order
rmse_results %>% arrange(desc(RMSEs))
```

```
## # A tibble: 12 x 2
##    method                                                                    RMSEs
##    <chr>                                                                     <dbl>
##  1 "Mean only"                                                              1.16e5
##  2 "RMSE from Knn"                                                          1.01e5
##  3 "RMSE from Knn_cv"                                                       9.53e4
##  4 "RMSE from Rpart"                                                        8.52e4
##  5 "Mean plus median income avg"                                            8.00e4
##  6 "Mean, median income avg and ocn_proximity"                              7.09e4
##  7 "RMSE_glmm"                                                              6.91e4
##  8 "RMSE from train_lm"                                                     6.89e4
##  9 "RMSE_lm"                                                                6.70e4
## 10 "Mean, median income avg, ocn_proximity and latitude"                    6.65e4
## 11 "Mean, median incone avg, ocn_proximity, latitue and housing median a~ 6.50e4
## 12 " Reg_RMSE"                                                              6.45e4
```

We saw the RMSE calculated using independent variables median_income, ocean_proximity, latitude, housing_median_age did best in terms of minimizing the RMSE. To make the model further better I used regularization to penalize the volatile estimates. So, the model using those four independent variables plus the regularization is the best predicting model for median_house_value among all the models we tried. The lowest RMSE we got from this model is 64,509.

4. Conclusion

Hence, I used different models to predict the median house value in California. I used linear regression, glm, knn, rpart methods to estimate RMSE, and I found linear regression model performed best among these. After that I wrote different algorithms adding additional independent variable at a time, and the more independent variable I added, the better the model performed in terms of minimizing the RMSE. Using regularization made the model even better and brought the RMSE down to 64,509. I have plan to include all other independent variables from the dataset in the best model I have so far.

Using this algorithm, we can predict the median house value in California, and using this algorithm we can predict the housing prices in other regions, too. Same technique used here can be used in other dataset where regression analysis could be used for prediction. This is not the perfect model, since there are yet other independent variables to be included from the dataset in the algorithm.

In the future I will include all the predictors in the algorithms so that the algorithm will be better. I will collect additional variables' data if possible, since the housing value also depends on crime rates, quality of school in the neighborhood and so on. As you have seen these predictors in our dataset have explained 64% percent of variation in the outcome variable as it is shown by R^2 in the linear regression model. Last but not the least, I will borrow cloud computation from the internet to run 'Random Forest' and 'Rborist' methods present in train function, since these methods took forever for my computer to run upon, and I had to abort the executions in the middle of processings.

In addition to that I will use Principal Component Analysis (PCA) to reduce the number of predictors and use other relevant algorithms to reduce the RMSE further down. I will look into the data to find out why the older houses likely to have more median house values than the new ones.