DAT102x: Predicting Evictions

Jan. 2019,

Chia-Ta Liu

Executive Summary

Over the past decades, compare to incomes, housing costs have risen dramatically. Most poor families spend more than half of their salary on housing costs but only one in four families get help in the affordable housing program. Under these conditions, the number of people becomes harder to afford the rental price or other reasons. As a result, they are easily evicted by the landlord. This dataset provided poverty, income, health, ethnicity, and other sociodemographic factors to build a model for predicting eviction levels in counties across the United States.

Here are the steps of research following:

1. Exploratory Data Analysis
2. Data cleaning
3. Feature selection
4. Model building
5. Conclusion

First of all, exploring date by calculating summary. It is important to overview the dataset. This step can help us to understand the shape of it. Second, doing data cleaning to clean the dataset let us get useful data. In this part, fill null values should be focused. Third, according to the correlation between features to do feature selecting. After that, building a regression model and adjustment parameters to get the best model and test performance with R-squared.

Exploratory Data Analysis

Data structure

In the beginning, it is important to describe dataset. It can help us to understand data structure and overview the data.

Describe dataset

```
> str(train)
Classes 'data.table' and 'data.frame':  2546 obs. of  49 variables:
 $ row_id                                   : int  0 1 2 3 4 5 6 7 8 9 ...
 $ county_code                              : chr  "a4e2211" "583e0c7" "4776bfd" "97fb48d" ...
 $ year                                     : chr  "b" "a" "b" "a" ...
 $ state                                    : chr  "d725a95" "533155c" "d725a95" "d725a95" ...
 $ population                               : num  45009 9872 17625 134136 6936 ...
 $ renter_occupied_households               : num  6944 1224 1725 18180 551 ...
 $ pct_renter_occupied                      : num  37.2 31.8 22 36.8 17.6 ...
 $ median_gross_rent                        : num  643 517 671 603 668 ...
 $ median_household_income                  : num  33315 43724 37777 30607 44237 ...
 $ median_property_value                    : num  98494 85444 136162 70062 187066 ...
 $ rent_burden                              : num  33.4 26.5 32.5 32 29.3 ...
 $ pct_white                                : num  0.412 0.839 0.874 0.264 0.925 ...
 $ pct_af_am                                : num  0.49346 0.01559 0.04104 0.24084 0.00515 ...
 $ pct_hispanic                             : num  0.0702 0.0374 0.0469 0.0811 0.0358 ...
 $ pct_am_ind                               : num  0.00259 0.07349 0.0045 0.37799 0.01404 ...
 $ pct_asian                                : num  0.004575 0.005771 0.004873 0.007244 0.000707 ...
 $ pct_nh_pi                                : num  0.000201 0.000803 0 0.000399 0.003175 ...
 $ pct_multiple                             : num  0.0159 0.0282 0.0287 0.0256 0.0167 ...
 $ pct_other                                : num  0.000993 0 0.0002 0.002804 0 ...
 $ poverty_rate                             : num  18.5 11.9 11.9 26 10.7 ...
 $ rucc                                     : chr  "Nonmetro - Urban population of 20,000 or more, adjacent to a
metro area" "Nonmetro - Urban population of 2,500 to 19,999, adjacent to a metro area" "Nonmetro - Completely rura
l or less than 2,500 urban population, adjacent to a metro area" "Nonmetro - Urban population of 20,000 or more, a
djacent to a metro area" ...
 $ urban_influence                          : chr  "Micropolitan adjacent to a large metro area" "Noncore adjacen
t to a small metro with town of at least 2,500 residents" "Noncore adjacent to a small metro and does not contain
a town of at least 2,500 residents" "Micropolitan adjacent to a small metro area" ...
 $ economic_typology                        : chr  "Nonspecialized" "Nonspecialized" "Recreation" "Nonspecialized
" ...
 $ pct_civilian_labor                       : num  0.407 0.547 0.43 0.379 0.433 0.514 0.419 0.534 0.503 0.392 ...
 $ pct_unemployment                         : num  0.093 0.04 0.065 0.084 0.064 0.071 0.062 0.035 0.056 0.065 ...
 $ pct_uninsured_adults                     : num  0.239 0.204 0.3 0.335 0.23 0.273 0.26 0.087 0.229 0.266 ...
 $ pct_uninsured_children                   : num  0.068 0.092 0.108 0.101 0.119 0.096 0.094 0.035 0.104 0.09 ...
 $ pct_adult_obesity                        : num  0.332 0.315 0.291 0.398 0.242 0.231 0.356 0.264 0.314 0.358 ..

 $ pct_adult_smoking                        : num  0.277 0.208 0.245 0.254 0.204 0.132 0.238 0.122 0.21 0.233 ...
 $ pct_diabetes                             : num  0.145 0.129 0.106 0.157 0.099 0.079 0.132 0.087 0.102 0.136 ..
.
 $ pct_low_birthweight                      : num  0.12 0.06 0.08 0.11 0.079 0.11 0.079 0.07 0.07 0.129 ...
 $ pct_excessive_drinking                   : num  0.077 0.094 NA 0.084 0.193 0.18 NA 0.2 0.112 0.078 ...
 $ pct_physical_inactivity                  : num  0.313 0.31 0.244 0.349 0.215 0.196 0.31 0.217 0.293 0.348 ...
 $ air_pollution_particulate_matter_value   : num  12.17 8.29 13.13 12.23 8.91 ...
 $ homicides_per_100k                       : num  14 NA NA 22.7 NA ...
 $ motor_vehicle_crash_deaths_per_100k      : num  18.2 28.6 11.1 34.7 23.3 ...
 $ heart_disease_mortality_per_100k         : int  318 306 266 325 187 256 370 192 260 401 ...
 $ pop_per_dentist                          : num  2420 3330 6699 4810 NA ...
 $ pop_per_primary_care_physician           : num  1960 890 3509 2219 3410 ...
 $ pct_female                               : num  0.532 0.509 0.451 0.519 0.487 0.513 0.511 0.501 0.506 0.516 ..
.
 $ pct_below_18_years_of_age                : num  0.252 0.252 0.166 0.263 0.196 0.233 0.251 0.279 0.259 0.259 ..
.
 $ pct_aged_65_years_and_older              : num  0.153 0.188 0.189 0.125 0.203 0.097 0.188 0.111 0.155 0.148 ..
.
 $ pct_adults_less_than_a_high_school_diploma: num  0.233 0.0733 0.2066 0.2483 0.0586 ...
 $ pct_adults_with_high_school_diploma      : num  0.375 0.398 0.303 0.335 0.276 ...
 $ pct_adults_with_some_college             : num  0.278 0.331 0.301 0.29 0.414 ...
 $ pct_adults_bachelors_or_higher           : num  0.114 0.198 0.189 0.127 0.251 ...
 $ birth_rate_per_1k                        : num  12.92 11.05 7.9 13.14 6.08 ...
 $ death_rate_per_1k                        : num  11.21 12.28 10.16 10.2 5.94 ...
 $ evictions                                : int  681 0 29 841 2 4191 24 225 93 6 ...
```

Figure 1-1: data structure

From figure 1-1, it is easy to notice that there are 6 categorical variables and 43 numerical variables, consists of 49 variables and total are 2546 rows.

```
> summary(train)
     row_id       county_code            year               state             population       renter_occupied_households pct_renter_occupied median_gross_rent median_household_income
 Min.   :   0.0   Length:2546        Length:2546        Length:2546        Min.   :    116   Min.   :    14              Min.   : 7.305      Min.   : 336.0    Min.   : 19328
 1st Qu.: 636.2   Class :character   Class :character   Class :character   1st Qu.:  10294   1st Qu.:  1052              1st Qu.:22.884      1st Qu.: 577.2    1st Qu.: 38496
 Median :1272.5   Mode  :character   Mode  :character   Mode  :character   Median :  23863   Median :  2580              Median :26.866      Median : 642.0    Median : 44480
 Mean   :1272.5                                                            Mean   : 106246   Mean   : 15008              Mean   :28.147      Mean   : 688.8    Mean   : 46051
 3rd Qu.:1908.8                                                            3rd Qu.:  67969   3rd Qu.:  8099              3rd Qu.:32.093      3rd Qu.: 750.0    3rd Qu.: 51526
 Max.   :2545.0                                                            Max.   :5279852   Max.   :882101              Max.   :70.610      Max.   :1728.0    Max.   :123452
                                                                                                                                                                 NA's   :2
 median_property_value  rent_burden      pct_white         pct_af_am          pct_hispanic       pct_am_ind           pct_asian           pct_nh_pi            pct_multiple
 Min.   : 32287        Min.   : 9.986   Min.   :0.05093   Min.   :0.000000   Min.   :0.00000   Min.   :0.0000000   Min.   :0.000000   Min.   :0.0000000   Min.   :0.000000
 1st Qu.: 85288        1st Qu.:26.047   1st Qu.:0.65522   1st Qu.:0.005669   1st Qu.:0.01818   1st Qu.:0.0009991   1st Qu.:0.002081   1st Qu.:0.0000000   1st Qu.:0.009623
 Median :108844        Median :28.780   Median :0.85548   Median :0.021864   Median :0.03606   Median :0.0023870   Median :0.004961   Median :0.0000000   Median :0.014561
 Mean   :129610        Mean   :28.521   Mean   :0.77627   Mean   :0.089774   Mean   :0.09060   Mean   :0.0124670   Mean   :0.011653   Mean   :0.0006449   Mean   :0.017698
 3rd Qu.:151696        3rd Qu.:31.160   3rd Qu.:0.93533   3rd Qu.:0.094011   3rd Qu.:0.08989   3rd Qu.:0.0052790   3rd Qu.:0.010626   3rd Qu.:0.0004018   3rd Qu.:0.020696
 Max.   :904937        Max.   :49.535   Max.   :0.99511   Max.   :0.858997   Max.   :0.93620   Max.   :0.8013643   Max.   :0.337672   Max.   :0.0965272   Max.   :0.208475
 NA's   :2
   pct_other          poverty_rate        rucc            urban_influence    economic_typology  pct_civilian_labor pct_unemployment  pct_uninsured_adults pct_uninsured_children
 Min.   :0.0000000   Min.   : 0.000   Length:2546        Length:2546        Length:2546        Min.   :0.2130     Min.   :0.01900   Min.   :0.0510       Min.   :0.01400
 1st Qu.:0.0000000   1st Qu.: 8.386   Class :character   Class :character   Class :character   1st Qu.:0.4203     1st Qu.:0.04400   1st Qu.:0.1680       1st Qu.:0.05700
 Median :0.0002017   Median :11.543   Mode  :character   Mode  :character   Mode  :character   Median :0.4690     Median :0.05700   Median :0.2140       Median :0.07700
 Mean   :0.0008863   Mean   :12.370                                                            Mean   :0.4677     Mean   :0.05942   Mean   :0.2159       Mean   :0.08638
 3rd Qu.:0.0011023   3rd Qu.:15.291                                                            3rd Qu.:0.5150     3rd Qu.:0.07100   3rd Qu.:0.2597       3rd Qu.:0.10600
 Max.   :0.0198221   Max.   :44.732                                                            Max.   :1.0000     Max.   :0.18200   Max.   :0.4950       Max.   :0.28300

 pct_adult_obesity pct_adult_smoking  pct_diabetes    pct_low_birthweight pct_excessive_drinking pct_physical_inactivity air_pollution_particulate_matter_value homicides_per_100k
 Min.   :0.1510    Min.   :0.0460    Min.   :0.0410   Min.   :0.04000     Min.   :0.0420         Min.   :0.1200          Min.   : 7.543                         Min.   :-0.400
 1st Qu.:0.2850    1st Qu.:0.1740    1st Qu.:0.0940   1st Qu.:0.07000     1st Qu.:0.1270         1st Qu.:0.2430          1st Qu.:10.502                         1st Qu.: 2.598
 Median :0.3080    Median :0.2110    Median :0.1090   Median :0.08000     Median :0.1630         Median :0.2780          Median :12.016                         Median : 4.500
 Mean   :0.3067    Mean   :0.2146    Mean   :0.1096   Mean   :0.08407     Mean   :0.1633         Mean   :0.2762          Mean   :11.703                         Mean   : 5.848
 3rd Qu.:0.3330    3rd Qu.:0.2500    3rd Qu.:0.1230   3rd Qu.:0.09100     3rd Qu.:0.1960         3rd Qu.:0.3100          3rd Qu.:12.971                         3rd Qu.: 7.900
 Max.   :0.4710    Max.   :0.5110    Max.   :0.1980   Max.   :0.23100     Max.   :0.3090         Max.   :0.4410          Max.   :14.881                         Max.   :50.490
                   NA's   :408                        NA's   :126         NA's   :810                                    NA's   :1                             NA's   :1598
 motor_vehicle_crash_deaths_per_100k heart_disease_mortality_per_100k pop_per_dentist pop_per_primary_care_physician   pct_female      pct_below_18_years_of_age
 Min.   : 3.09                       Min.   :109.0                    Min.   :  490   Min.   :  189                   Min.   :0.2850   Min.   :0.0880
 1st Qu.:13.44                       1st Qu.:239.0                    1st Qu.: 1819   1st Qu.: 1409                   1st Qu.:0.4950   1st Qu.:0.2060
 Median :19.50                       Median :276.0                    Median : 2694   Median : 1980                   Median :0.5040   Median :0.2250
 Mean   :20.92                       Mean   :279.7                    Mean   : 3504   Mean   : 2588                   Mean   :0.4991   Mean   :0.2262
 3rd Qu.:26.38                       3rd Qu.:316.0                    3rd Qu.: 4220   3rd Qu.: 2864                   3rd Qu.:0.5110   3rd Qu.:0.2437
 Max.   :76.05                       Max.   :482.0                    Max.   :28130   Max.   :23399                   Max.   :0.5720   Max.   :0.3590
 NA's   :308                                                          NA's   :190     NA's   :175
 pct_aged_65_years_and_older pct_adults_less_than_a_high_school_diploma pct_adults_with_high_school_diploma pct_adults_with_some_college pct_adults_bachelors_or_higher
 Min.   :0.0630              Min.   :0.01603                            Min.   :0.1271                      Min.   :0.1370               Min.   :0.01887
 1st Qu.:0.1440              1st Qu.:0.09700                            1st Qu.:0.3087                      1st Qu.:0.2657               1st Qu.:0.13815
 Median :0.1680              Median :0.13087                            Median :0.3566                      Median :0.3013               Median :0.17668
 Mean   :0.1716              Mean   :0.14789                            Mean   :0.3532                      Mean   :0.3009               Mean   :0.19800
 3rd Qu.:0.1948              3rd Qu.:0.19441                            3rd Qu.:0.4014                      3rd Qu.:0.3360               3rd Qu.:0.23291
 Max.   :0.3450              Max.   :0.46593                            Max.   :0.5503                      Max.   :0.4487               Max.   :0.58408

 birth_rate_per_1k death_rate_per_1k   evictions
 Min.   : 3.612    Min.   : 0.000    Min.   :    0.0
 1st Qu.: 9.915    1st Qu.: 8.558    1st Qu.:    4.0
 Median :11.306    Median :10.478    Median :   29.0
 Mean   :11.482    Mean   :10.407    Mean   :  378.0
 3rd Qu.:12.836    3rd Qu.:12.160    3rd Qu.:  160.8
 Max.   :28.923    Max.   :27.397    Max.   :29251.0
```

Figure 1-2: data summary

From figure 1-2, this figure shows more detail information about each column, according to that, we can find Max, Min, Mean and Median. The most important is that include the number of missing values.

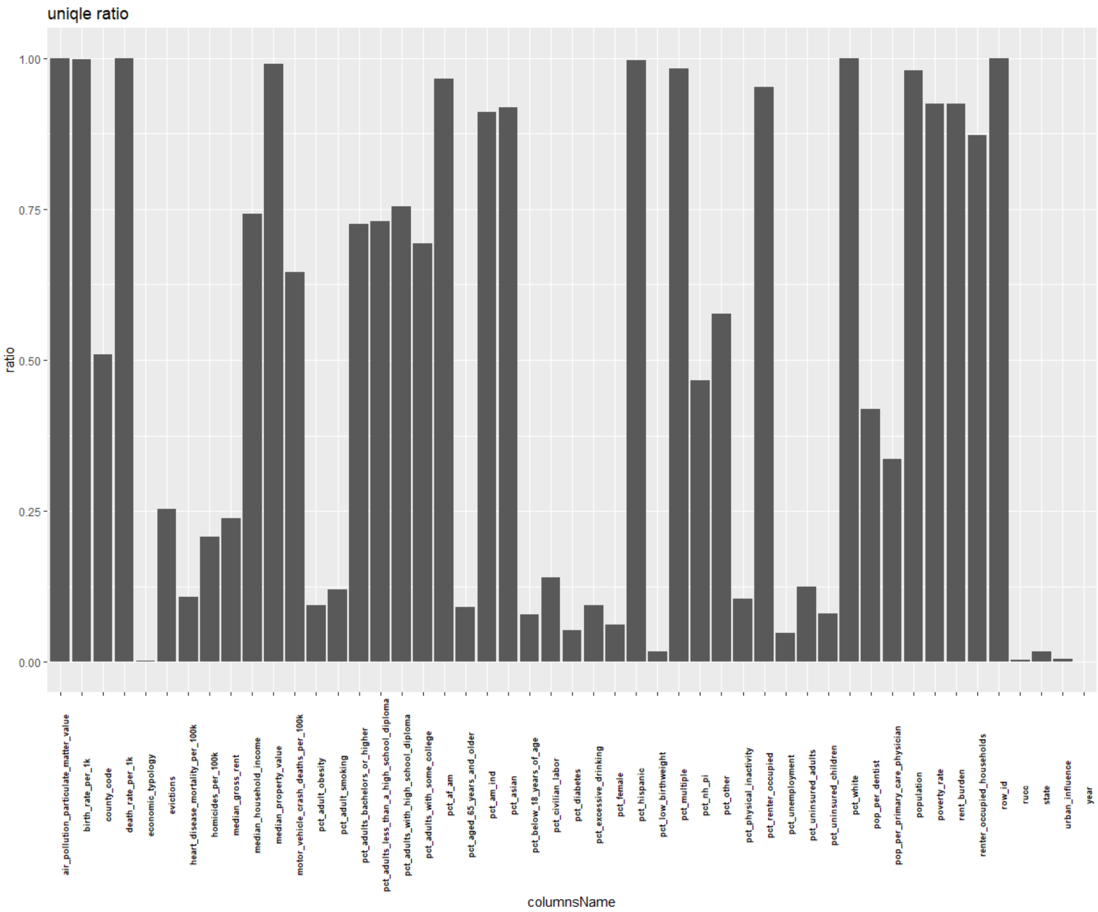

Figure 2: unique ratio

Data Cleaning

This step is important to train model, especially fill missing values. By using visualization, it helps us to understand the distribution range. If the range is too wide, to drop it out is considerable.

1. Missing values

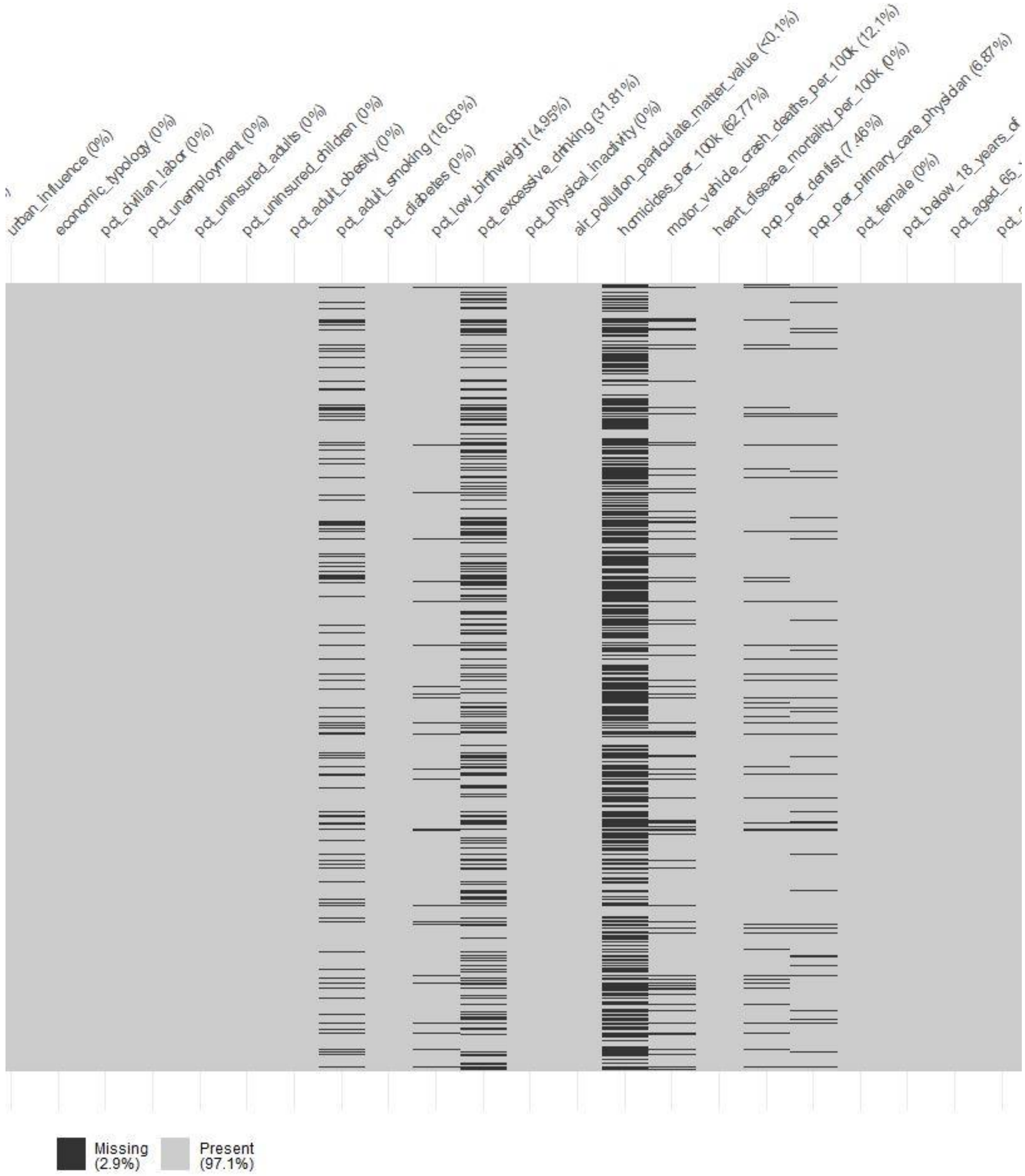

Figure 3: missing value

From figure 3, this figure shows the distribution of missing values. Total have 2.9% missing values. It concentrates in homicides per 100k which is 62.77% and per excessive drinking which is 31.81%. It is clearer than the figure 2.

2. Imputed method

There are many ways to imputed the missing values, for example, the median is often used to impute it. But there is a better way to instead it, R mice package. This powerful package can choose many different methods to impute the NA value such as KNN, CART or even random forest. But the negative is maybe cost too much time, especially if data have a categorical variable.

3. Relationship

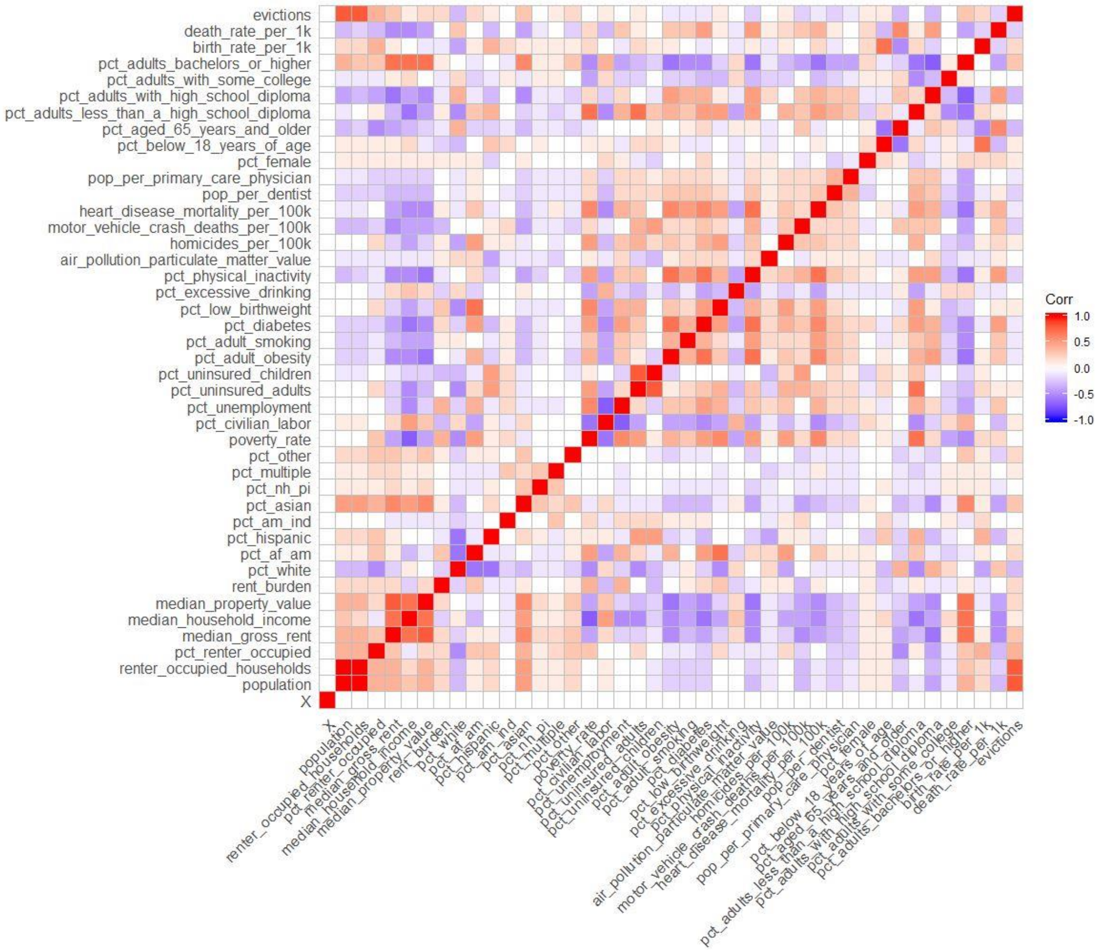

Figure 4: heatmap

According to figure 4, population and renter-occupied households have a high correlation to evictions. On the other hand, most of the columns do not have significantly relative to evictions.

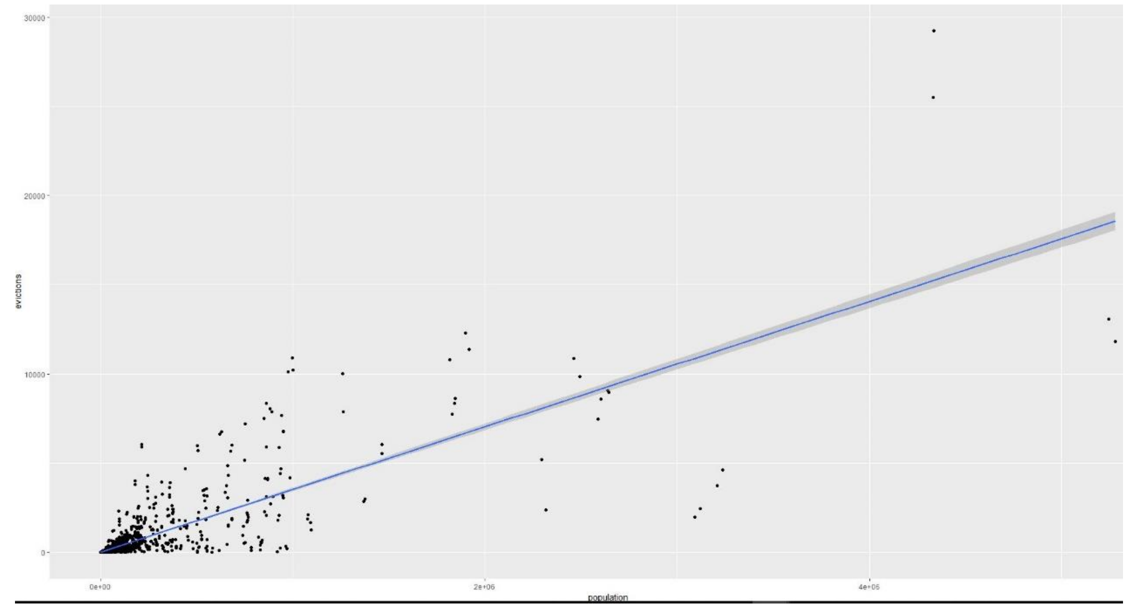

figure 5-1: population - evictions relationship

From figure 5-1, it can find population and evictions have a correlation.

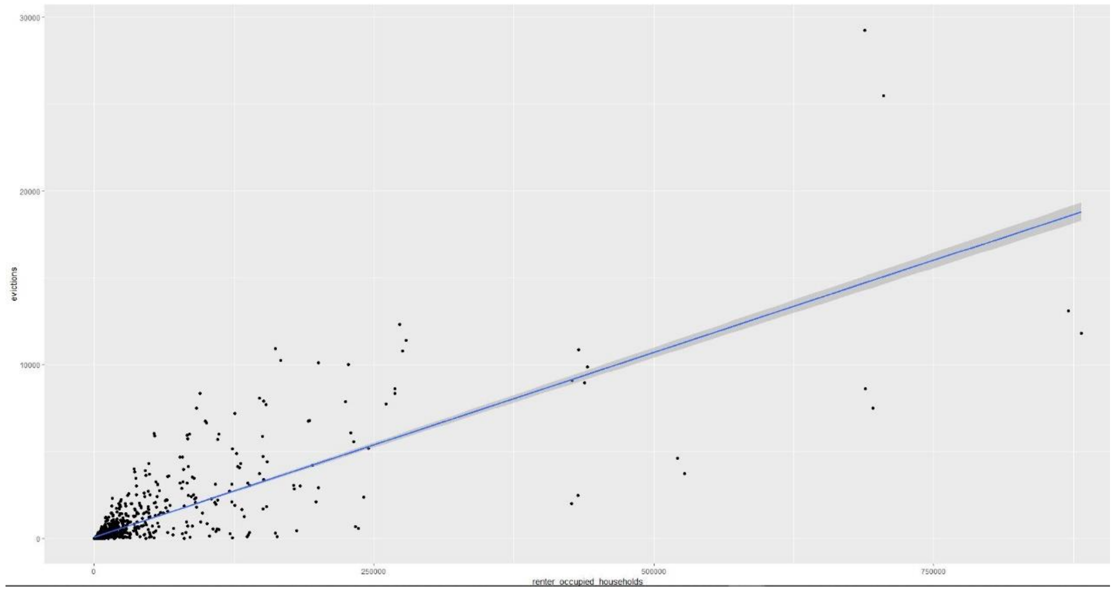

figure 5-2: renter-occupied households – evictions relationship

From figure 5-2, it can find renter-occupied households and evictions have a correlation.

4. The reason for dropping columns

   Because of the ID and State are identical, they don't have significant effect to target column, so remove it out. Including the figure 6, it shows that year a and year b do not have significant different, so drop it. On the other hand, the categorical variable usually to do one hot encoding but it is hard to deal with too many levels, so drop all the other categorical variable.

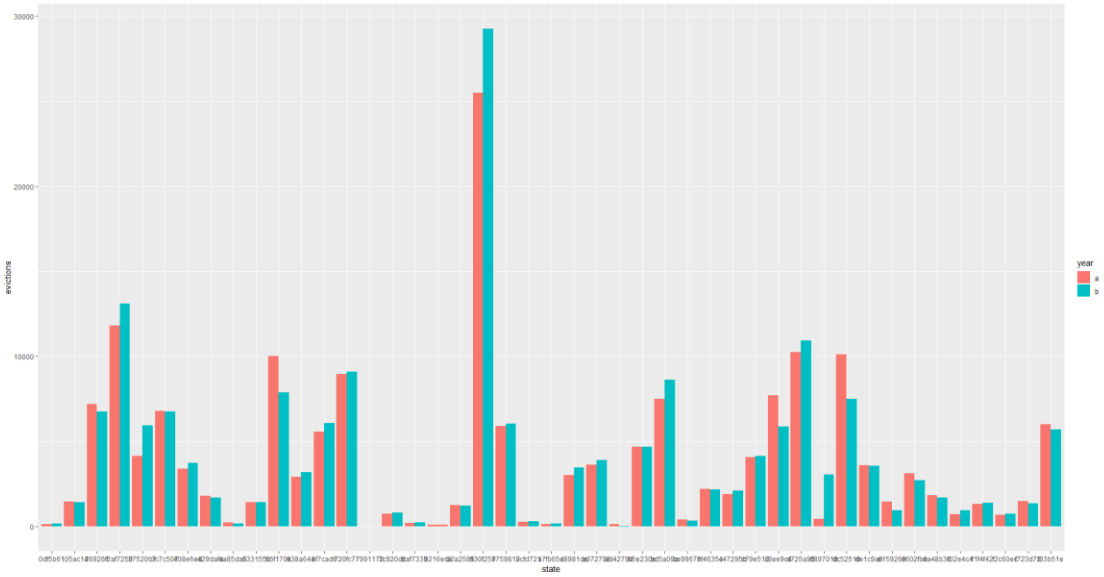

figure 6 Year a and year b compare graphic

From figure 6, after group by state, they are slightly different between year a and year b.

Feature Selection

Feature selection algorithms can be divided into three categories by selecting different evaluation indicators: wrappers, filters and embedded methods.

1. Wrapper

   Wrapper method uses a predictive model to score feature subsets. Each new subset is used to train a model and then tested with a validation data set. The feature subset is scored by calculating the number of errors on the validation dataset. Since the wrapper method trains a new model for each feature subset, the amount of computation is large. However, such methods often find the best performing feature set for a particular type of model.

2. Filter

   Filter methods use a proxy measure instead of the error rate to score a feature subset. Common measures include the mutual information, the pointwise mutual information, Pearson product-moment correlation coefficient, Relief-based algorithms, and inter/intra class distance or the scores of significance tests for each class/feature combinations. Due to the lack of tuning, the feature set selected by the filters method is more general than the feature set selected by the wrapper class, which tends to result in lower prediction performance than the wrapper. However, since the feature set does not contain assumptions about the prediction model, it is more conducive to exposing the relationship between the features.

3. Embedded method

   Embedded method includes all the feature selection techniques used in building the model. An example of such a method is the LASSO method of building a linear model. This method adds an L1 penalty to the regression coefficients, causing many of these parameters to go to zero.

Model Building and Testing

1. Split data

The training data is split to 80% training and 20% validation in order to adjust model parameters in order to get the best model.

2. Performance

   In this case, we are prediction numerical values, which is regression problem. R-squared, also called coefficient of determination is commonly used to measure regression.

$$\mathrm{R}^2 = 1 - \frac{\sum(y-\hat{y})^2}{\sum(y-\bar{y})^2}$$

   The quantity of R-squared is between $-\infty \sim 1$, the higher is better. A value of 1 means that the perdition is complete match the test value.

3. Random Forest

Random forests or random decision forests are an ensemble learning method for classification, regression and other tasks that operates by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees. Random decision forests correct for decision trees' habit of overfitting to their training set.

   Submission result: R-squared = 0.7722

4. XGboost

XGBoost (eXtreme Gradient Boosting) is an open-source software library which provides a gradient boosting. From the project description, it aims to provide a "Scalable, Portable and Distributed Gradient Boosting (GBM, GBRT, GBDT) Library". It has gained much popularity and attention recently as it was the algorithm of choice for many winning teams of a number of machine learning competitions.

Xgboost is a gradient boosting decision tree that can be used for classification or regression problems. Gradient boosting strives to correct the residuals of all the weak learners by adding a new weak learner. Finally, multiple learners are added together for a final prediction, and the accuracy is higher than the single one. It is called Gradient because it uses a gradient descent algorithm to minimize the loss when adding a new model.

Submission result: R-squared = 0.7855

5. Adjust parameters

By using python GridSearchCV package can help us to adjustment our model. Through testing parameters, it is an efficiency to find the best one to boost the model.

Submission result (use RF , after parameters setting): R-squared = 0.8053
Submission result (use XGBoost, after parameters setting): R-squared = 0.8386

After adjusting parameters, R-squared improvement significantly in those two models. This step can increase predictive accuracy and get the best model.

Conclusion

In conclusion, this analysis has shown the eviction of a country can be predicted from its feature. According to this analysis, population and renter-occupied households has a significant influence on the rate of eviction. In addition, after cleaning up other noises of data, the model accuracy can be improved. Moreover, feature engineer can be further improved the model. In the end, XGBoost has the best performance among all models. This algorithm can be used to wide tasks and get excellent accuracy