Assignment 2 By Team 4

Salma Bouzid, Shun-Lung Chang, Savitha Singh

Import data

We first downloaded the data sets from **here**, and then we imported and stored them as R dataframes (**properties** and **transaction** for "properties_2016.csv" and "train_2016.csv" respectively). In addition, the two dataframes were joined as a new dataframe, **joined_df**.

```
properties <- fread("./data/properties_2016.csv")
transaction <- fread("./data/train_2016.csv")
joined_df <- merge(transaction, properties, all.x = TRUE)
```

1. Explain why it is or why it is not a big data problem.

We can safely conclude that analyzing this data set is not a big data problem, since it fails to satisfy the volume, variety, velocity, veracity argument (Shafer, 2017).

A closer look at the 4Vs of big data will enable us to better understand this problem:

1. The dataset is not voluminous
   After using `pryr::object_size(properties)` in R, we know that properties's memory is merely 0.908 GBs. According to the table below[1], we know that handling this dataset can be done on a consumer PC and does not require extra cores or machines.

Big	Can't fit in memory on one computer: **>1 TB**
Medium	Fits in memory on a server: **10 GB-1 TB**
Small	Fits in memory on a laptop: **<10 GB**

Big	Many machines, many cores
Medium	Many cores
Small	One core

2. The dataset is structured
   The dataset is well-defined in labeled rows and columns. In fact, the dataset comes with a dictionary that clearly explains the 58 attributes in the properties data.

3. The dataset is static
   Most of the datasets' attributes come from government agencies that publish yearly or bi-monthly statistics. Moreover, this dataset does not include the newly user generated input, although users can update their housing information on Zillow's portal anytime to replects changes of their households (Zillow, 2017).

4. We can trust the data
   The dataset's observations come from public records regarding location and property characteristics. Although it can be missing or outdated (Zillow, 2017), we can safely assume that this dataset has

not been manipulated to reflect bias that favors one housing area or any such fraudulent behaviors. Therefore, the uncertainty of this problem does not achive the level of big data problems.

2. Why is it an analytics problem?

Analytics aims to derive actionable insights from data. One starts by defining the problem at hand. Second, statistical models and computing algorithms are used to solve the issue. (Cooper, 2012). We will rely on this definition to answer this question.

1. Problem defining
   Buyers and sellers are not equally informed about the value of houses. In fact, some players, such as real estate agents, have information advantage. They are more informed about future gentrification and demographic patterns that impact future house prices. (Kurlat and Stroebel, 2014). In order to resolve the information asymmetry in the market, Zillow strive to help house buyers and provide them with precise information of the housing market. (Zillow, 2017)

2. How does the Zillow fight information asymmetry through data analytics?
   By analyzing user input data and public records, Zillow predicts the house values, which is called Zestimate. And Zillow evaluates the log difference between Zestimate and actual prices to offer house buyers a clearer picture of future house market. Moreover, during the analytics process, Zillow relied on statistical models for precise prediction, and launched this challenge to improve its housing valuation algorithm by learning from the most performing models submitted by Kaggle users

3. How many data attributes are there?

As can be seen from the result of `colnames(joined_df)`, the whole data set contains 60 attributes. Also, properties contains 58 attributes and transaction contains 3 attributes.

```r
colnames(joined_df)
```

```
 [1] "parcelid"                     "logerror"
 [3] "transactiondate"              "airconditioningtypeid"
 [5] "architecturalstyletypeid"     "basementsqft"
 [7] "bathroomcnt"                  "bedroomcnt"
 [9] "buildingclasstypeid"          "buildingqualitytypeid"
[11] "calculatedbathnbr"            "decktypeid"
[13] "finishedfloor1squarefeet"     "calculatedfinishedsquarefeet"
[15] "finishedsquarefeet12"         "finishedsquarefeet13"
[17] "finishedsquarefeet15"         "finishedsquarefeet50"
[19] "finishedsquarefeet6"          "fips"
[21] "fireplacecnt"                 "fullbathcnt"
[23] "garagecarcnt"                 "garagetotalsqft"
[25] "hashottuborspa"               "heatingorsystemtypeid"
[27] "latitude"                     "longitude"
[29] "lotsizesquarefeet"            "poolcnt"
[31] "poolsizesum"                  "pooltypeid10"
[33] "pooltypeid2"                  "pooltypeid7"
[35] "propertycountylandusecode"    "propertylandusetypeid"
[37] "propertyzoningdesc"           "rawcensustractandblock"
[39] "regionidcity"                 "regionidcounty"
[41] "regionidneighborhood"         "regionidzip"
[43] "roomcnt"                      "storytypeid"
[45] "threequarterbathnbr"          "typeconstructiontypeid"
```

```
[47] "unitcnt"                     "yardbuildingsqft17"
[49] "yardbuildingsqft26"          "yearbuilt"
[51] "numberofstories"             "fireplaceflag"
[53] "structuretaxvaluedollarcnt"  "taxvaluedollarcnt"
[55] "assessmentyear"              "landtaxvaluedollarcnt"
[57] "taxamount"                   "taxdelinquencyflag"
[59] "taxdelinquencyyear"          "censustractandblock"
```

4. Identify the type of the 15 attributes you find most relevant in this context

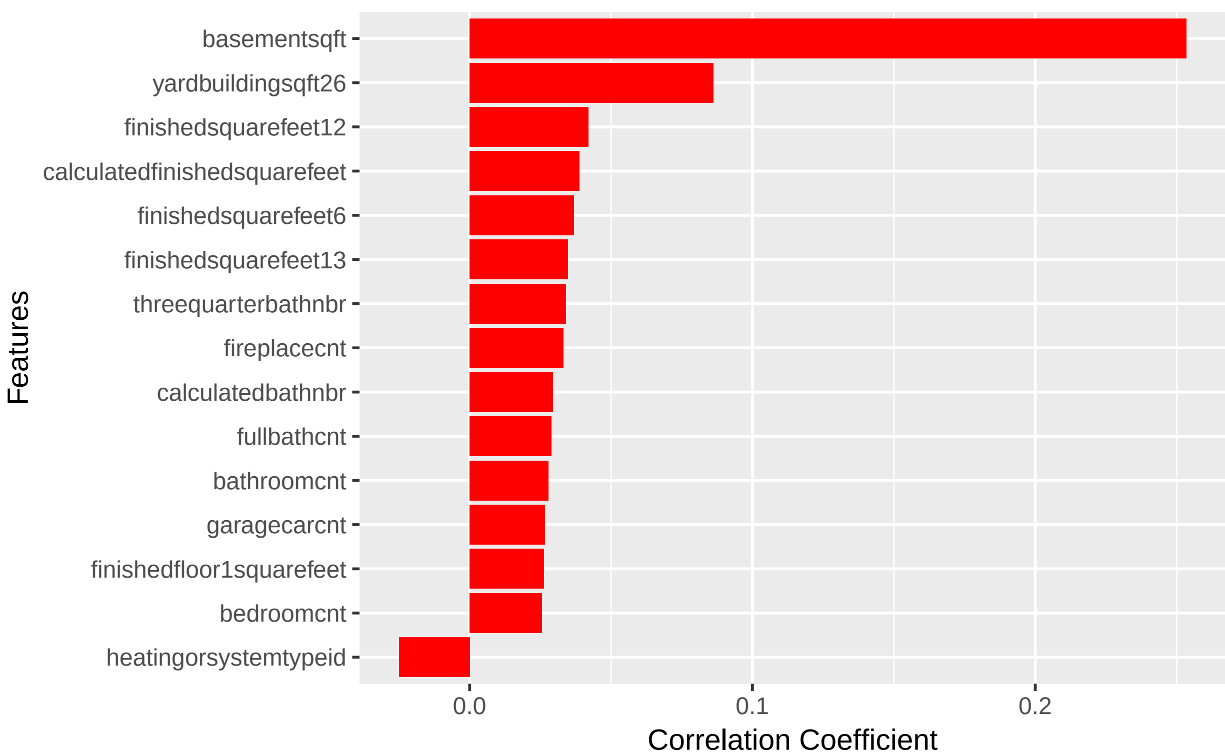
Since the goal of this problem is to find minimal *log(error)*, and we assume that "relevant" means "linearly correlated". We show the first 15 attributes that has highest absolute correlation coefficients between logerror below.

Firstly, we pickcd those features that are numeric, and then used `Hmisc::rcorr()` to get the correlation coefficient matrix. At last we sorted the correlation coefficients between logerror decreasingly, and chose the first 15 items. The result is shown as the barplot.

```r
numeric_df <- joined_df[, sapply(joined_df, is.numeric), with = FALSE]

corr_mat <- Hmisc::rcorr(as.matrix(numeric_df))

top15 <- abs(corr_mat$r[2, ]) %>%
    sort(decreasing = TRUE) %>%
    .[2:16] %>%
    names()
```



5. Determine whether the task refers to a supervised or unsupervised learning problem

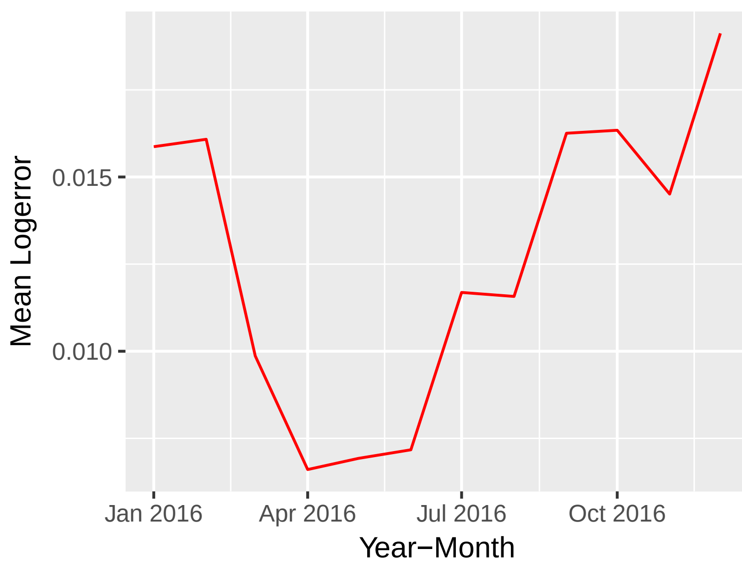
This should be a supervised learning task. As noted by Mehryar Mohri et al, the goal of supervised learning proplem is to use labeled data for prediction, namely, there is a dependant variable, which is usually denoted as y, and one uses other independant variables, which is also known as feature, to predict this dependant variable. Given that we have to predict *log(error)* variable using features in properties, so *log(error)* is the labeled variable we need to prediction in a supervised problem.

6. Find out what the standard analysis algorithms are for this analytics problem

In order to find the optimal logerror, a standard analysis algorithm that can be used is the regression analysis. Regression analysis is for estimation the relationship among variables. for this task, one can build regression models to find the relationship between logerror and other variables.

7. Download the data and provide descriptive summaries of the sample data

First, we summarised transaction by ploting its trend. One can see an uprising trend of the average logerror of each month.



Also, we used `summary()` in R to summarise properties. One can see a noticeable number of missing values (NAs) in many of the features, and we would carry out a deeper analysis of these missing values in the next question.

```r
summary(properties)
```

```
    parcelid         airconditioningtypeid architecturalstyletypeid
 Min.   : 10711725   Min.   : 1.0          Min.   : 2.0
 1st Qu.: 11643707   1st Qu.: 1.0          1st Qu.: 7.0
 Median : 12545094   Median : 1.0          Median : 7.0
```

```
Mean   : 13325858   Mean   : 1.9         Mean   : 7.2
3rd Qu.: 14097122   3rd Qu.: 1.0         3rd Qu.: 7.0
Max.   :169601949   Max.   :13.0         Max.   :27.0
                    NA's   :2173698      NA's   :2979156
 basementsqft      bathroomcnt       bedroomcnt     buildingclasstypeid
Min.   :  20.0    Min.   : 0.000   Min.   : 0.000   Min.   :1.0
1st Qu.: 272.0    1st Qu.: 2.000   1st Qu.: 2.000   1st Qu.:3.0
Median : 534.0    Median : 2.000   Median : 3.000   Median :4.0
Mean   : 646.9    Mean   : 2.209   Mean   : 3.089   Mean   :3.7
3rd Qu.: 847.2    3rd Qu.: 3.000   3rd Qu.: 4.000   3rd Qu.:4.0
Max.   :8516.0    Max.   :20.000   Max.   :20.000   Max.   :5.0
NA's   :2983589   NA's   :11462    NA's   :11450    NA's   :2972588
buildingqualitytypeid calculatedbathnbr   decktypeid
Min.   : 1.0          Min.   : 1.0      Min.   :66
1st Qu.: 4.0          1st Qu.: 2.0      1st Qu.:66
Median : 7.0          Median : 2.0      Median :66
Mean   : 5.8          Mean   : 2.3      Mean   :66
3rd Qu.: 7.0          3rd Qu.: 3.0      3rd Qu.:66
Max.   :12.0          Max.   :20.0      Max.   :66
NA's   :1046729       NA's   :128912    NA's   :2968121
finishedfloor1squarefeet calculatedfinishedsquarefeet
Min.   :    3            Min.   :     1
1st Qu.: 1012            1st Qu.:  1213
Median : 1283            Median :  1572
Mean   : 1381            Mean   :  1827
3rd Qu.: 1615            3rd Qu.:  2136
Max.   :31303            Max.   :952576
NA's   :2782500          NA's   :55565
finishedsquarefeet12 finishedsquarefeet13 finishedsquarefeet15
Min.   :     1       Min.   : 120        Min.   :   112
1st Qu.:  1196       1st Qu.: 960        1st Qu.:  1694
Median :  1539       Median :1296        Median :  2172
Mean   :  1760       Mean   :1179        Mean   :  2739
3rd Qu.:  2070       3rd Qu.:1440        3rd Qu.:  2976
Max.   :290345       Max.   :2688        Max.   :820242
NA's   :276033       NA's   :2977545     NA's   :2794419
finishedsquarefeet50 finishedsquarefeet6      fips
Min.   :    3        Min.   :   117      Min.   :6037
1st Qu.: 1013        1st Qu.:  1079      1st Qu.:6037
Median : 1284        Median :  1992      Median :6037
Mean   : 1389        Mean   :  2414      Mean   :6048
3rd Qu.: 1618        3rd Qu.:  3366      3rd Qu.:6059
Max.   :31303        Max.   :952576      Max.   :6111
NA's   :2782500      NA's   :2963216     NA's   :11437
 fireplacecnt      fullbathcnt      garagecarcnt     garagetotalsqft
Min.   :1.0       Min.   : 1.00    Min.   : 0.0      Min.   :   0.0
1st Qu.:1.0       1st Qu.: 2.00    1st Qu.: 2.0      1st Qu.: 324.0
Median :1.0       Median : 2.00    Median : 2.0      Median : 441.0
Mean   :1.2       Mean   : 2.24    Mean   : 1.8      Mean   : 383.8
3rd Qu.:1.0       3rd Qu.: 3.00    3rd Qu.: 2.0      3rd Qu.: 494.0
Max.   :9.0       Max.   :20.00    Max.   :25.0      Max.   :7749.0
NA's   :2672580   NA's   :128912   NA's   :2101950   NA's   :2101950
hashottuborspa     heatingorsystemtypeid    latitude
Length:2985217     Min.   : 1            Min.   :33324388
```

```
Class :character   1st Qu.: 2           1st Qu.:33827685
Mode  :character   Median : 2           Median :34008249
                   Mean   : 4           Mean   :34001469
                   3rd Qu.: 7           3rd Qu.:34161860
                   Max.   :24           Max.   :34819650
                   NA's   :1178816      NA's   :11437
  longitude         lotsizesquarefeet      poolcnt
Min.   :-119475780   Min.   :      100   Min.   :1
1st Qu.:-118392983   1st Qu.:     5688   1st Qu.:1
Median :-118172540   Median :     7000   Median :1
Mean   :-118201934   Mean   :    22823   Mean   :1
3rd Qu.:-117949468   3rd Qu.:     9898   3rd Qu.:1
Max.   :-117554316   Max.   :328263808   Max.   :1
NA's   :11437        NA's   :276099      NA's   :2467683
 poolsizesum       pooltypeid10      pooltypeid2      pooltypeid7
Min.   :   19.0   Min.   :1         Min.   :1         Min.   :1
1st Qu.:  430.0   1st Qu.:1         1st Qu.:1         1st Qu.:1
Median :  495.0   Median :1         Median :1         Median :1
Mean   :  519.7   Mean   :1         Mean   :1         Mean   :1
3rd Qu.:  594.0   3rd Qu.:1         3rd Qu.:1         3rd Qu.:1
Max.   :17410.0   Max.   :1         Max.   :1         Max.   :1
NA's   :2957257   NA's   :2948278   NA's   :2953142   NA's   :2499758
propertycountylandusecode propertylandusetypeid propertyzoningdesc
Length:2985217            Min.   : 31           Length:2985217
Class :character          1st Qu.:261           Class :character
Mode  :character          Median :261           Mode  :character
                          Mean   :260
                          3rd Qu.:261
                          Max.   :275
                          NA's   :11437
rawcensustractandblock  regionidcity    regionidcounty
Min.   :60371011       Min.   :  3491   Min.   :1286
1st Qu.:60373203       1st Qu.: 12447   1st Qu.:2061
Median :60375712       Median : 25218   Median :3101
Mean   :60483450       Mean   : 34993   Mean   :2570
3rd Qu.:60590423       3rd Qu.: 45457   3rd Qu.:3101
Max.   :61110091       Max.   :396556   Max.   :3101
NA's   :11437          NA's   :62845    NA's   :11437
regionidneighborhood  regionidzip        roomcnt        storytypeid
Min.   :  6952       Min.   : 95982   Min.   : 0.000   Min.   :7
1st Qu.: 46736       1st Qu.: 96180   1st Qu.: 0.000   1st Qu.:7
Median :118920       Median : 96377   Median : 0.000   Median :7
Mean   :193476       Mean   : 96553   Mean   : 1.475   Mean   :7
3rd Qu.:274800       3rd Qu.: 96974   3rd Qu.: 0.000   3rd Qu.:7
Max.   :764167       Max.   :399675   Max.   :96.000   Max.   :7
NA's   :1828815      NA's   :13980    NA's   :11475    NA's   :2983593
threequarterbathnbr typeconstructiontypeid    unitcnt
Min.   :1           Min.   : 4            Min.   :  1.0
1st Qu.:1           1st Qu.: 6            1st Qu.:  1.0
Median :1           Median : 6            Median :  1.0
Mean   :1           Mean   : 6            Mean   :  1.2
3rd Qu.:1           3rd Qu.: 6            3rd Qu.:  1.0
Max.   :7           Max.   :13            Max.   :997.0
NA's   :2673586     NA's   :2978470       NA's   :1007727
```

```
 yardbuildingsqft17 yardbuildingsqft26   yearbuilt     numberofstories
 Min.   :  10.0     Min.   :  10.0     Min.   :1801    Min.   : 1.0
 1st Qu.: 190.0     1st Qu.:  96.0     1st Qu.:1950    1st Qu.: 1.0
 Median : 270.0     Median : 168.0     Median :1963    Median : 1.0
 Mean   : 319.8     Mean   : 278.3     Mean   :1964    Mean   : 1.4
 3rd Qu.: 390.0     3rd Qu.: 320.0     3rd Qu.:1981    3rd Qu.: 2.0
 Max.   :7983.0     Max.   :6141.0     Max.   :2015    Max.   :41.0
 NA's   :2904862    NA's   :2982570    NA's   :59928   NA's   :2303148
 fireplaceflag      structuretaxvaluedollarcnt taxvaluedollarcnt
 Length:2985217     Min.   :        1          Min.   :        1
 Class :character   1st Qu.:    74800          1st Qu.:   179675
 Mode  :character   Median :   122590          Median :   306086
                    Mean   :   170884          Mean   :   420479
                    3rd Qu.:   196889          3rd Qu.:   488000
                    Max.   :251486000          Max.   :282786000
                    NA's   :54982              NA's   :42550
 assessmentyear  landtaxvaluedollarcnt   taxamount
 Min.   :2000    Min.   :        1      Min.   :      1
 1st Qu.:2015    1st Qu.:    74836      1st Qu.:   2461
 Median :2015    Median :   167042      Median :   3992
 Mean   :2015    Mean   :   252478      Mean   :   5378
 3rd Qu.:2015    3rd Qu.:   306918      3rd Qu.:   6201
 Max.   :2016    Max.   :90246219       Max.   :3458861
 NA's   :11439   NA's   :67733          NA's   :31250
 taxdelinquencyflag taxdelinquencyyear censustractandblock
 Length:2985217     Min.   : 0.0       Min.   :0
 Class :character   1st Qu.:14.0       1st Qu.:0
 Mode  :character   Median :14.0       Median :0
                    Mean   :13.9       Mean   :0
                    3rd Qu.:15.0       3rd Qu.:0
                    Max.   :99.0       Max.   :0
                    NA's   :2928753    NA's   :3
```

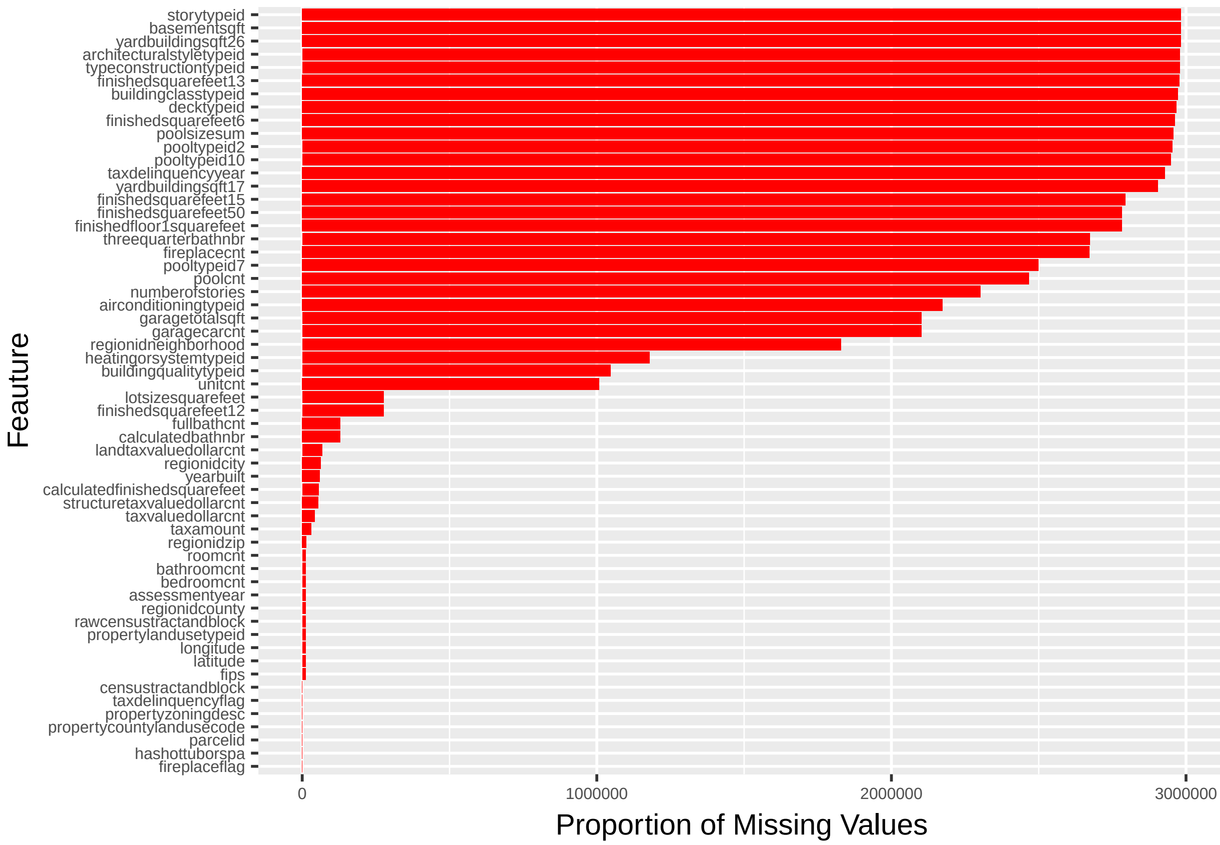
8. Check for completeness of the data! Are there any missings? How are the missings distributed?

For transaction, we first use `complete.cases()` to check whether each observation is complete or not. The result is shown below, and there is no incomplete row in transaction.

```r
sum(!complete.cases(transaction))
```

```
[1] 0
```

For properties, as can be seen from question 7, there exist a great amount of missing values in many features of The plot below shows the counts of missing values of each feature and how they distribute. One can see that 18 features lack 95% of values. For variable 'basementsqft', 'buildingclasstypeid', 'finishedsquarefeet13', 'storytypeid', they miss more that 99% of observations. Thirteen features, however, have zero missing values, such as geographical and house room attributes.



References

[1] Adalbert F.X. Wilhelm (2017), The Big Data Challenge: Topics, Applications, Perspectives [Powerpoint slides]