2015 Nepal Earthquake Reconstruction Need Prediction

Guenter W.

Feb 15, 2019

1 Executive summary / Overview

The goal of this report is to present an analysis of data regarding need for reconstruction of buildings collected by the Central Bureau of Statistics after the 2015 Gorkha earthquake in Nepal. The data set consists mainly of aspects of building location and construction.

The initial descriptive statistics of the data is followed by a presentation of potential relationships between building features and the need for reconstruction afer the earthquake. Based on the findings from this analysis and feature importance indicators, various categorical prediction models are created and compared in regards to their accuracy. Finally, an ensemble model is created.

With our analysis, we show that it is possible to create a classification model that allows to reasonably predict the need for reconstruction of buildings caused by the 2015 Nepal earthquake based on its characteristics.

2 Introduction

In April 2015, an earthquake in Nepal killed about 9,000 people and caused severe damage to existing buildings. More information about this disaster can be found here: https://en.wikipedia.org/wiki/April_2015_Nepal_earthquake

2.1 Dataset source

The website https://opendata.klldev.org hosts datasets regaring the 2015 Nepal Earthquake including data about affected individuals, buildings and households. For this analysis, we focus on the Structural Data dataset from the Buildings section. The data can be obtained by manual download from the website https://opendata.klldev.org/#/download or via direct download link: https://opendata.klldev.org/statics/building_structure.zip

2.2 Dataset description

The data set consists of 31 variables and 1052948 rows.

```
dim(buildings)
```

```
## [1] 1052948      31
```

The dataset consists of information on earthquake impacts, household conditions, and socio-economic-demographic statistics and has the following columns:

- building_id: A unique ID that identifies a unique building from the survey(Datatype: Text)
- district_id: District where the building is located(Datatype: Text)
- vdcmun_id: Municipality where the building is located(Datatype: Text)
- ward_id: Ward Number in which the building is located(Datatype: Text)
- count_floors_pre_eq: Number of floors that the building had before the earthquake(Datatype: Number)
- count_floors_post_eq: Number of floors that the building had after the earthquake(Datatype: Number)
- age_building: Age of the building (in years)(Datatype: Number)

- plinth_area_sq_ft: Plinth area of the building (in square feet)(Datatype: Number)
- height_ft_pre_eq: Height of the building before the earthquake (in feet)(Datatype: Number)
- height_ft_post_eq: Height of the building after the earthquake (in feet)(Datatype: Number)
- land_surface_condition: Surface condition of the land in which the building is built(Datatype: Categorical)
- foundation_type: Type of foundation used in the building(Datatype: Categorical)
- roof_type: Type of roof used in the building(Datatype: Categorical)
- ground_floor_type: Ground floor type (Datatype: Categorical)
- other_floor_type: Type of construction used in other floors (except ground floor and roof)(Datatype: Categorical)
- position: Position of the building(Datatype: Categorical)
- plan_configuration: Building plan configuration(Datatype: Categorical)
- has_superstructure_adobe_mud: Flag variable that indicates if the superstructure of the building is made of Adobe/Mud(Datatype: Boolean)
- has_superstructure_mud_mortar_stone: Flag variable that indicates if the superstructure of the building is made of Mud Mortar - Stone(Datatype: Boolean)
- has_superstructure_stone_flag: Flag variable that indicates if the superstructure of the building is made of Stone(Datatype: Boolean)
- has_superstructure_cement_mortar_stone: Flag variable that indicates if the superstructure of the building is made of Stone(Datatype: Boolean)
- has_superstructure_mud_mortar_brick: Flag variable that indicates if the superstructure of the building is made of Cement Mortar - Stone(Datatype: Boolean)
- has_superstructure_cement_mortar_brick: Flag variable that indicates if the superstructure of the building is made of Mud Mortar - Brick(Datatype: Boolean)
- has_superstructure_timber: Flag variable that indicates if the superstructure of the building is made of Timber(Datatype: Boolean)
- has_superstructure_bamboo: Flag variable that indicates if the superstructure of the building is made of Bamboo(Datatype: Boolean)
- has_superstructure_rc_non_engineered: Flag variable that indicates if the superstructure of the building is made of RC (reinforced concrete) (Non Engineered)(Datatype: Boolean)
- has_superstructure_rc_engineered: Flag variable that indicates if the superstructure of the building is made of RC (reinforced concrete) (Engineered)(Datatype: Boolean)
- has_superstructure_other: Flag variable that indicates if the superstructure of the building is made of any other material(Datatype: Boolean)
- condition_post_eq: Actual contition of the building after the earthquake(Datatype: Categorical)
- damage_grade: Damage grade assigned to the building by the surveyor after assessment(Datatype: Categorical)
- technical_solution_proposed: Technical solution proposed by the surveyor after assessment(Datatype: Categorical)
- id: A unique ID that identifies a unique information from all table(Datatype: Number)

2.3 Goal of Analysis

The goal of this analysis is to develop a suitable supervised prediction model to predict if a building needs reconstruction or not.

3 Methods

In order to tackle the defined analysis goal, we lay out the following analysis process:

- Data preparation / Cleansing: In the data preparation step, we remove missing rows and all features that were unknown at the time of the earthquake. In addition, we create the label of interest

reconstruction_needed and normalize continuous features.

- Explorative data analysis (EDA) including feature statistics and visualization: We use visualization and statistical indicators to get and overview of the data set. In particular, we examine the relationship of the data set features with the need for reconstruction which represents the label of interest.
- Feature selection based on findings from EDA and importance settings of models: In order to create an efficient and simple prediction model for the prediction of the need for reconstruction, we use our findings from the EDA and the importance features from various models. We limit the features in the data set to those that seem to have an influence on *reconstruction_needed*. Therefore, we take a two-steps approach. First, we select features based on our findings from the EDA. Second, we employ a handful of models from the Caret package that include feature importance
- Prediction model creation using the Caret library in R: We create a variety of models as well as an ensembled model and compare the accuracy of these models in order to find the best performing model.

4 Data preparation / Cleansing

The data set doesn't contain missing values.

```
sum(is.na(buildings))
```

```
## [1] 0
```

It includes the following variables.

```
str(buildings)
```

```
## 'data.frame':    1052948 obs. of  31 variables:
##  $ building_id                           : num  7.01e+10 7.01e+10 7.01e+10 7.01e+10 7.01e+10 ...
##  $ district_id                           : int  7 7 7 7 7 7 7 7 7 7 ...
##  $ vdcmun_id                             : int  701 701 701 701 701 701 701 701 701 701 ...
##  $ ward_id                               : int  70102 70102 70102 70103 70103 70103 70103 70105 70105
##  $ count_floors_pre_eq                   : int  1 1 2 2 1 2 1 1 2 2 ...
##  $ count_floors_post_eq                  : int  1 1 2 2 0 0 1 1 0 0 ...
##  $ age_building                          : int  28 32 34 20 25 35 44 25 24 27 ...
##  $ plinth_area_sq_ft                     : int  454 324 456 452 542 589 546 324 548 574 ...
##  $ height_ft_pre_eq                      : int  9 9 18 18 9 18 9 9 18 18 ...
##  $ height_ft_post_eq                     : int  9 9 18 18 0 0 9 9 0 0 ...
##  $ land_surface_condition                : Factor w/ 3 levels "Flat","Moderate slope",..: 2 2 2 2 2 1
##  $ foundation_type                       : Factor w/ 5 levels "Bamboo/Timber",..: 3 3 3 3 3 3 3 3 3 3
##  $ roof_type                             : Factor w/ 3 levels "Bamboo/Timber-Heavy roof",..: 2 2 2 2
##  $ ground_floor_type                     : Factor w/ 5 levels "Brick/Stone",..: 2 2 2 2 2 2 2 2 2 2
##  $ other_floor_type                      : Factor w/ 4 levels "Not applicable",..: 1 1 4 4 1 4 1 1 4
##  $ position                              : Factor w/ 5 levels "","Attached-1 side",..: 2 2 2 2 2 2 2
##  $ plan_configuration                    : Factor w/ 11 levels "","Building with Central Courtyard",
##  $ has_superstructure_adobe_mud          : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ has_superstructure_mud_mortar_stone   : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ has_superstructure_stone_flag         : int  0 0 0 0 0 0 0 0 1 1 ...
##  $ has_superstructure_cement_mortar_stone: int  0 0 0 0 0 0 0 0 0 0 ...
##  $ has_superstructure_mud_mortar_brick   : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ has_superstructure_cement_mortar_brick: int  0 0 0 0 0 0 0 0 0 0 ...
##  $ has_superstructure_timber             : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ has_superstructure_bamboo             : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ has_superstructure_rc_non_engineered  : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ has_superstructure_rc_engineered      : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ has_superstructure_other              : int  1 1 1 1 1 0 0 1 0 1 ...
```

```
##  $ condition_post_eq                    : Factor w/ 8 levels "Covered by landslide",..: 3 3 3 3 6 5
##  $ damage_grade                         : Factor w/ 6 levels "","Grade 1","Grade 2",..: 3 3 4 4 6 6
##  $ technical_solution_proposed          : Factor w/ 5 levels "","Major repair",..: 3 3 2 2 5 5 5 2 5
```

Since the variables *damage_grade*, *count_floors_post_eq*, *height_ft_post_eq* and *condition_post_eq* contain data that was unknown before the disaster, we exclude those variables as well as the building ID from the data set.

```
buildings <- buildings[ , -c(1, 3, 4, 6, 10, 29, 30)]
```

As a next step, we create the label variable based on the column technical_solution_proposed. First, we filter out all rows with an unknown *technical_solution_proposed*. Then, we rename the column *technical_solution_proposed* to *reconstruction_needed* and create two new levels that indicate that the building needs reconstruction or not.

```
# we filter out all rows with an unknown technical_solution_proposed
buildings <- buildings[  buildings$technical_solution_proposed != '', ]
# rename the column
colnames(buildings)[24] <- 'reconstruction_needed'
# re re-encode the values and combine values to have a binary classification
levels(buildings$reconstruction_needed)[5] <- 'Yes'
levels(buildings$reconstruction_needed)[1:4] <- 'No'
buildings$reconstruction_needed <- factor(buildings$reconstruction_needed)
```

Finally, we make sure that all factor variables are encoded as such.

```
buildings$has_superstructure_adobe_mud <-
  as.factor(buildings$has_superstructure_adobe_mud)
buildings$has_superstructure_mud_mortar_stone <-
  as.factor(buildings$has_superstructure_mud_mortar_stone)
buildings$has_superstructure_stone_flag <-
  as.factor(buildings$has_superstructure_stone_flag)
buildings$has_superstructure_cement_mortar_stone <-
  as.factor(buildings$has_superstructure_cement_mortar_stone)
buildings$has_superstructure_mud_mortar_brick <-
  as.factor(buildings$has_superstructure_mud_mortar_brick)
buildings$has_superstructure_cement_mortar_brick <-
  as.factor(buildings$has_superstructure_cement_mortar_brick)
buildings$has_superstructure_timber <-
  as.factor(buildings$has_superstructure_timber)
buildings$has_superstructure_bamboo <-
  as.factor(buildings$has_superstructure_bamboo)
buildings$has_superstructure_rc_non_engineered <-
  as.factor(buildings$has_superstructure_rc_non_engineered)
buildings$has_superstructure_rc_engineered <-
  as.factor(buildings$has_superstructure_rc_engineered)
buildings$has_superstructure_other <-
  as.factor(buildings$has_superstructure_other)

buildings$district_id <- as.factor(buildings$district_id)
```

5 Results

In this section, we present the results from our explorative data analysis, feature selection process and model creation.

Figure 1: Distribution Reconstruction needed

5.1 Explorative data analysis

5.1.1 Label reconstruction_needed

As we can see in Figure 1 below, the need for reconstruction is approximately equally distributed with 58% of buildings that need reconstruction after the earthquake and 42% of buildings that do not need any reconstruction.

```
tbl <- table(buildings$reconstruction_needed)
prop.table(tbl)
```

```
##
##        No       Yes
## 0.4200198 0.5799802
```

```
## Warning: Ignoring unknown parameters: binwidth, bins, pad
```

5.1.2 Features description

The data set contains four numerical features: *count_floors_pre_eq*, _height_ft_pre_eq, *age_building* and *plinth_area_sq_ft*. The table below provides an overview of the most important statistical indicators.

```
summary(buildings[, c('count_floors_pre_eq', 'height_ft_pre_eq',
                      'age_building', 'plinth_area_sq_ft')])
```

```
##  count_floors_pre_eq height_ft_pre_eq  age_building    plinth_area_sq_ft
##  Min.   :1.000       Min.   :  6.00   Min.   :  0.00   Min.   :  70.0
##  1st Qu.:2.000       1st Qu.: 13.00   1st Qu.: 10.00   1st Qu.: 284.0
##  Median :2.000       Median : 16.00   Median : 18.00   Median : 364.0
##  Mean   :2.131       Mean   : 16.31   Mean   : 27.75   Mean   : 417.9
##  3rd Qu.:2.000       3rd Qu.: 19.00   3rd Qu.: 30.00   3rd Qu.: 493.0
##  Max.   :9.000       Max.   :305.00   Max.   :999.00   Max.   :5220.0
```

5.1.2.1 count_floors_pre_eq

The median number of floors of buildings in our data set is 2. The building with the most number of floors has 9 floors. Figure 2 displays the difference of number of floors by *reconstruction_needed* of buildings with less than 5 floors as boxplots. We can identify a slight tendency that buildings with a higher number of floors may be more likely to need recontruction after the earthquake.

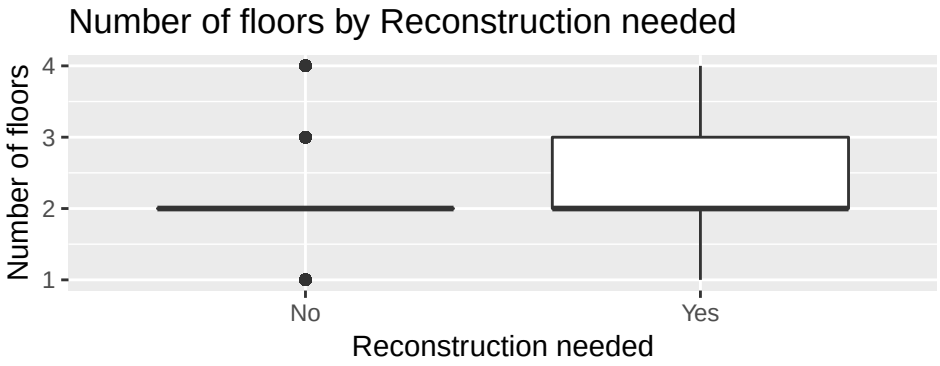

Figure 2: Number of floors by Reconstruction needed

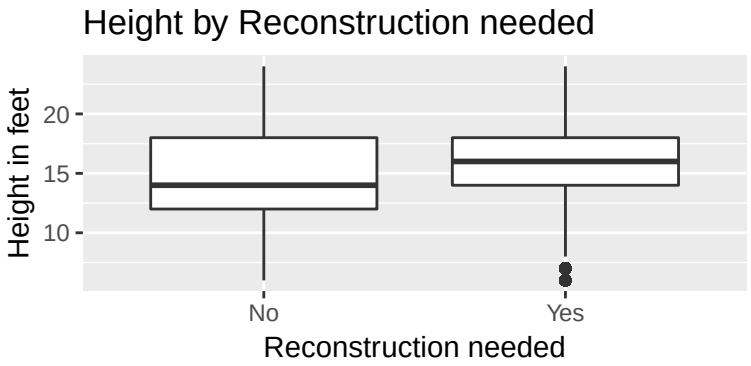

Figure 3: Height by Reconstruction needed

5.1.2.2 height_ft_pre_eq

While the average building is 16.31 feet tall, the highest building is 305 feet tall. In Figure 3, we can see the height of buildings in feet grouped by the need for reconstruction. In order to allow a visual comparison, we restricted the height to all buildings below 25 feet. It seems that the majority of buildings that need reconstruction are tending to be higher. This correspondes to our findings regarding the number of floors.

5.1.2.3 age_building

Another numerical feature of our data set is the age of buildings. The average age of a building is 27.75 years. However, the median age is only 18 years. From this difference, we can see that there are probably some outliers which is also confirmed by Figure 4 which shows the distribution of age.

As a consequence, we limit our graphical comparison of the age of buildings by *reconstruction_needed* to buildings with an age below 50 years. In Figure 5, we can see a slight tendency of older buildings to need reconstruction after the earthquake.

Besides these numerical features, the data set also contains a rich categorical variable set that mainly refers to construction features of the buildings. In these section, we only introduce the ones that seem to have an impact on the need for reconstruction in order to focus on the important information and to keep the section short. All other variables can be found in Appendix A.

5.1.2.4 district_id

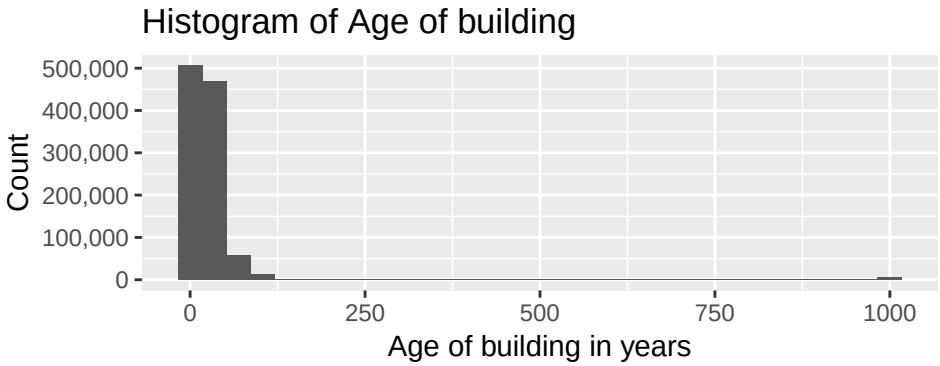

Figure 4: Distribution of age

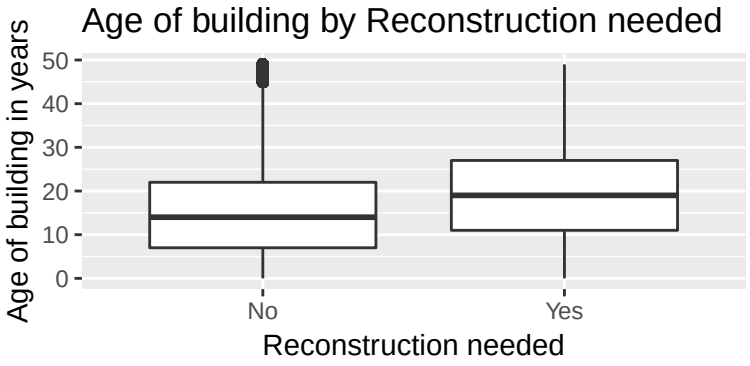

Figure 5: Age of building by Reconstruction needed

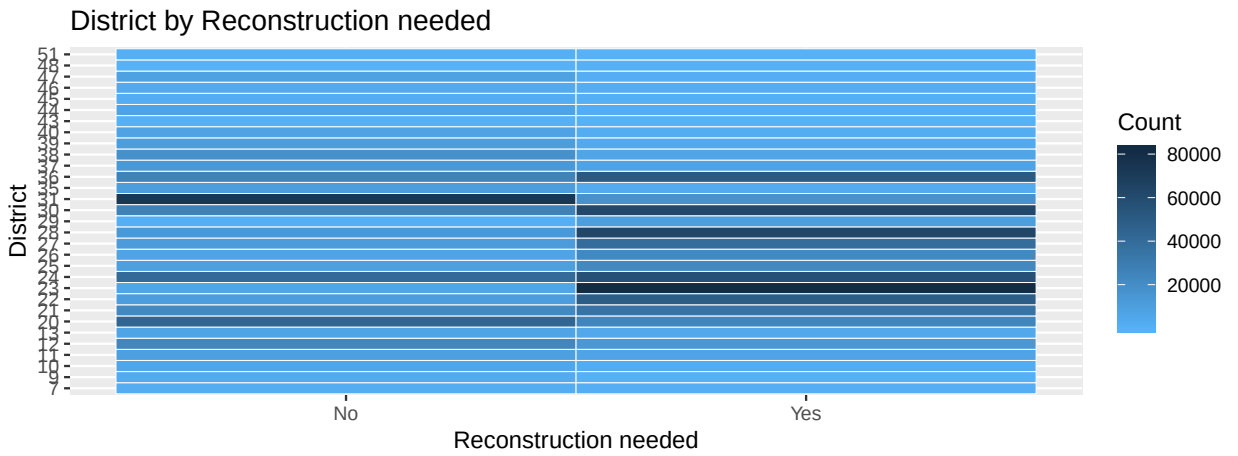

Figure 6: District by Reconstruction needed

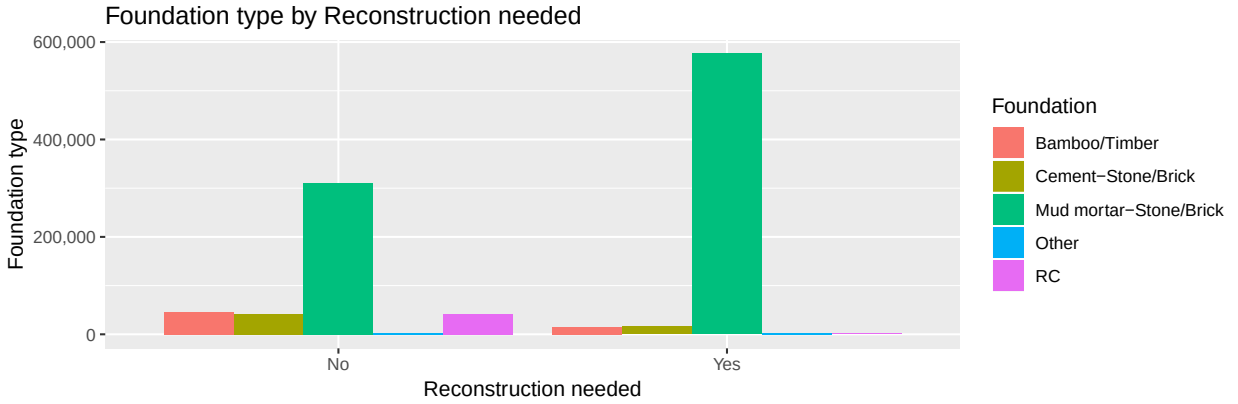

Figure 7: District by Reconstruction needed

district_id is a categorical variable with an identifier of the district in which the building is located.Figure 6 shows the number of buildings in each district grouped by *reconstruction_needed*. As we can see, there seems to be an impact of the district location on the need for reconstruction.

5.1.2.5 foundation_type

The type of foundation used in the building seems also to have an impact on the need for reconstruction as shown in Figure 7. Specifically, the RC (reinforced concrete) foundation type shows a different pattern if reconstruction is need, i.e. a lower number of buildings with an RC foundation type need reconstruction.

5.1.2.6 roof_type

In Figure 8, we cannot see a difference in the pattern for heavy or ligh bamboo or timer roofs depending on the need for reconstruction. However, the RCC/RB/RBC shows a divergent pattern. This may indicate that buildings with this type of roof are more likely to not need reconstruction.

5.1.2.7 ground_floor_type

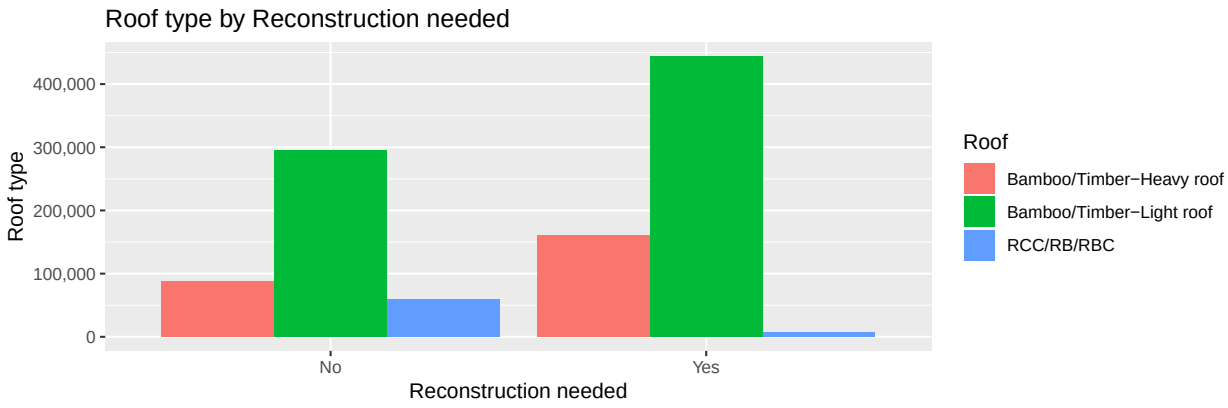

Figure 8: Roof type by Reconstruction needed

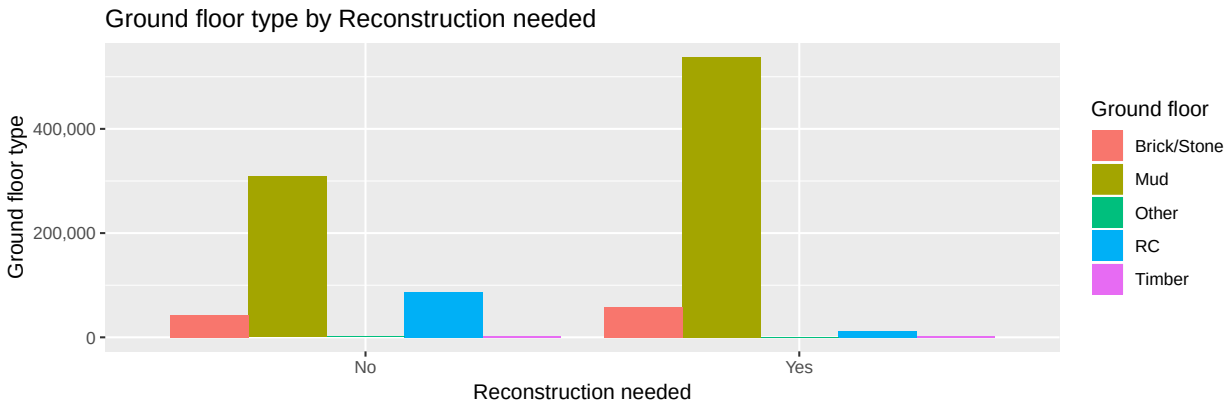

Figure 9: Ground floor type by Reconstruction needed

As shown in Figure 9, the ground floor type shows a different pattern depending if the buildings needs reconstruction or not. Specifically, buildings built on mud seem to be more likely to have a need for reconstruction. On the contrary, buildings built on a RC (reinforced concrete) ground floor type do not tend to need reconstruction.

5.1.2.8 other_floor_type

The *other_floor_type* variable represents the type of construction used in other floors except ground floor and roof. In Figure 10, we see a slightly different pattern between other_floor_type grouped by *reconstruction_needed*. It seems buildings that have RCC/RB/RBC floor types tend to need less reconstruction.

5.1.2.9 has_superstructure_mud_mortar_stone

Figure 11 depicts the relationship of a superstructure of mud, mortar and stone with the need for reconstruction. As we can see, there is a difference in the groups that indicate that buildings with a superstructure of mud, mortar and stone might need reconstruction more likely.

5.1.2.10 has_superstructure_cement_mortar_brick

In Figure 12, we can see a divergent pattern between buildings that need reconstruction and the ones that do not need based on the use of cement, mortar and bricks in the superstructure. Buildings that are based on

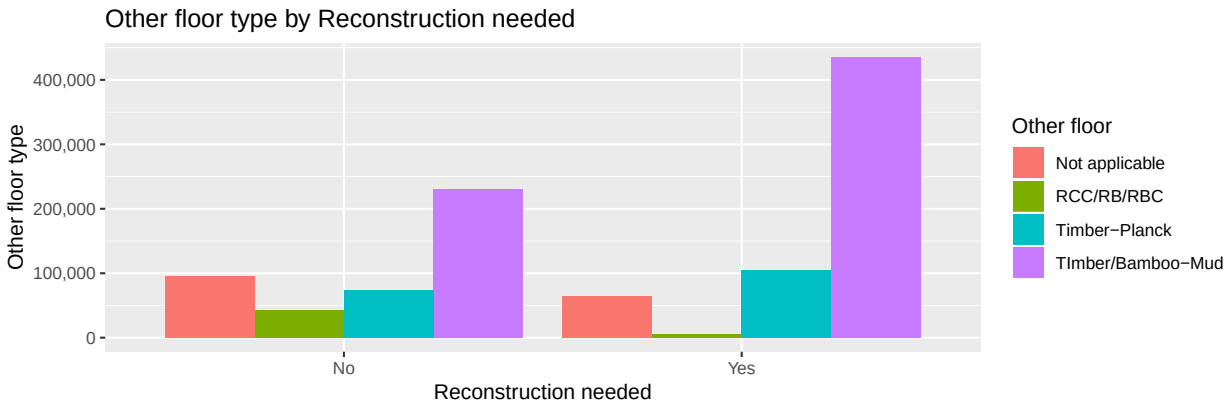

Figure 10: Other floor type by Reconstruction needed

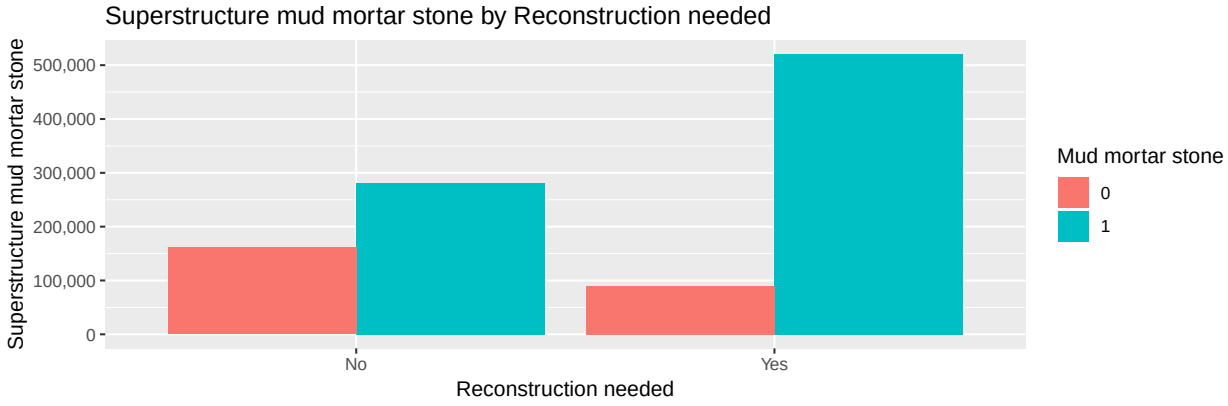

Figure 11: Superstructure mud mortar stone by Reconstruction needed

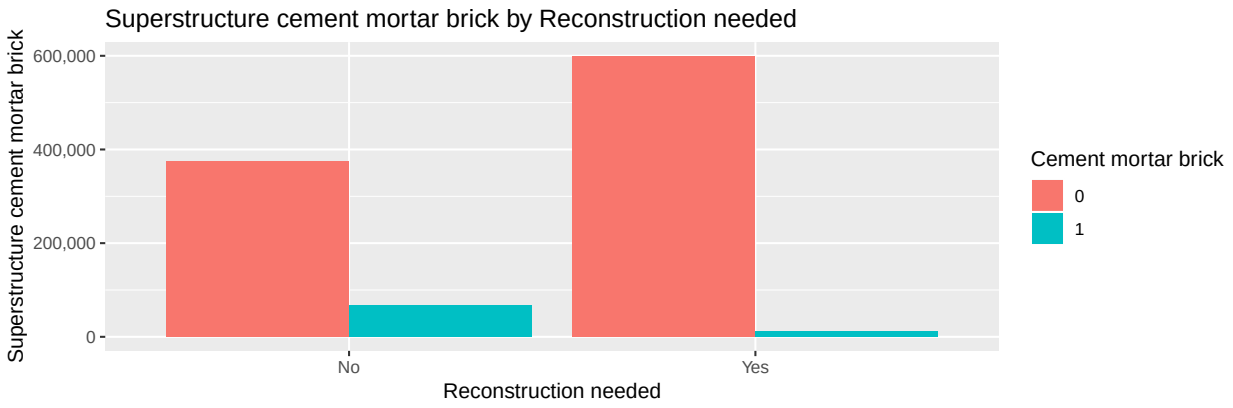

Figure 12: Superstructure cement mortar brick by Reconstruction needed

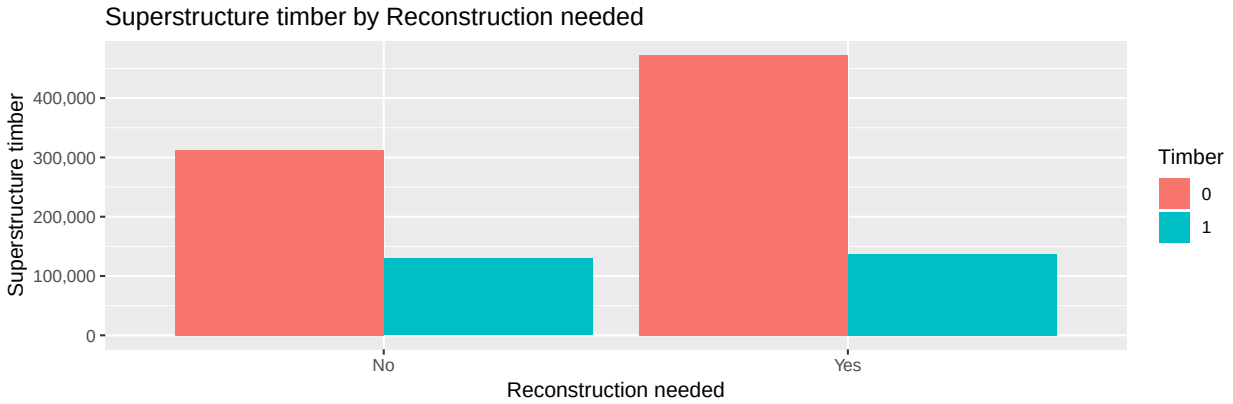

Figure 13: Superstructure timber by Reconstruction needed

cement, mortar and bricks seem to be less likely to need reconstruction.

5.1.2.11 has_superstructure_timber

Figure 13 shows the relationship between a superstructure of timber and the need for reconstruction. As we can see, buildings that don't have a superstructure of timber show a slight tendency to have a need for reconstruction.

5.1.2.12 has_superstructure_timber

Similar to superstructure of timber, buildings with superstructure of bamboo seem to be slightly less likely to have a need for reconstruction as shown in Figure 14.

5.1.2.13 has_superstructure_rc_non_engineered and has_superstructure_rc_engineered

For buildings with a superstructure of RC (reinforced concrete), there seems to be a trend that those buildings are less likely to have a need for reconstruction (independently whether they are engineered or not) as shown in Figure 15 and Figure 16.

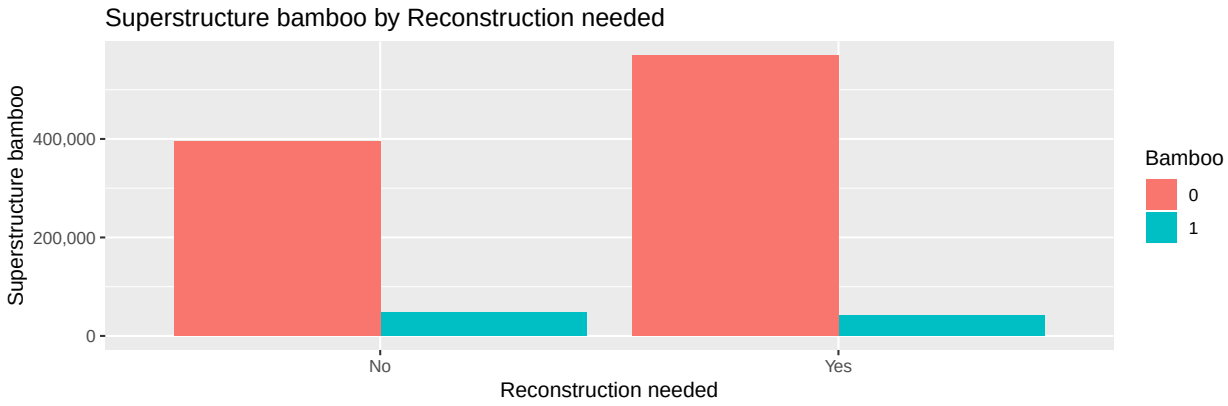

Figure 14: Superstructure bamboo by Reconstruction needed

Figure 15: Superstructure RC non engineered by Reconstruction needed

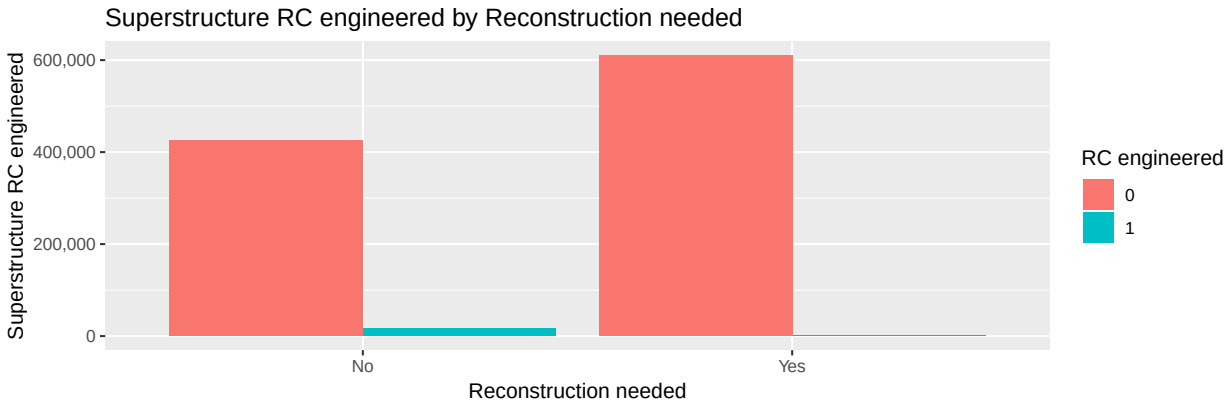

Figure 16: Superstructure RC engineered by Reconstruction needed

5.2 Feature selection process

As already mentioned in Section 3, we limit the features in the data set to those that seem to have an influence on *reconstruction_needed*. Therefore, we take a two-steps approach.

- First, we select features based on our findings from the EDA in the section before
- Second, we employ a handful of models from the Caret package that include feature importance

From our results in the EDA, we identify the following variables that might be useful for a prediction model.

- count_floors_pre_eq
- height_ft_pre_eq
- age_building
- plinth_area_sq_ft
- district_id
- foundation_type
- roof_type
- ground_floor_type
- other_floor_type
- has_superstructure_mud_mortar_stone
- has_superstructure_cement_mortar_brick
- has_superstructure_timber
- has_superstructure_bamboo
- has_superstructure_rc_non_engineered
- has_superstructure_rc_engineered

As a next step, we use the following models from the Caret package (see https://rdrr.io/cran/caret/man/models.html) to determine feature importance. [1]

- xgbTree
- rf
- kknn
- avNNet
- wsrf

First, we select a subset of the data set and train the selected models. Then, we obtain the feature importance results.

The results of the most important features are shown below and basically confirm our feature selection results from the EDA. The table below shows the 15 most important features for each of the selected models.

```
##                                    xgbTree
## 1                       ground_floor_type
## 2                             district_id
## 3                         foundation_type
## 4                            age_building
## 5                       plinth_area_sq_ft
## 6  has_superstructure_cement_mortar_brick
## 7     has_superstructure_mud_mortar_stone
## 8                      count_floors_pre_eq
## 9                         other_floor_type
## 10                        height_ft_pre_eq
## 11                               roof_type
## 12  has_superstructure_rc_non_engineered
## 13              has_superstructure_bamboo
## 14       has_superstructure_rc_engineered
```

[1] For detailed code please see the accompanying R script if you are interested

```
## 15    has_superstructure_mud_mortar_brick
##                                         RF
## 1                             district_id
## 2     has_superstructure_mud_mortar_stone
## 3                         ground_floor_type
## 4                       count_floors_pre_eq
## 5                           foundation_type
## 6                              age_building
## 7                          height_ft_pre_eq
## 8  has_superstructure_cement_mortar_brick
## 9                         plinth_area_sq_ft
## 10                         other_floor_type
## 11                               roof_type
## 12           has_superstructure_adobe_mud
## 13             has_superstructure_bamboo1
## 14    has_superstructure_mud_mortar_brick
## 15 has_superstructure_cement_mortar_stone
##                                       KKNN
## 1                          other_floor_type
## 2                              age_building
## 3     has_superstructure_mud_mortar_stone
## 4                       count_floors_pre_eq
## 5                         ground_floor_type
## 6                          height_ft_pre_eq
## 7                         plinth_area_sq_ft
## 8                                roof_type
## 9  has_superstructure_cement_mortar_brick
## 10                            district_id
## 11   has_superstructure_rc_non_engineered
## 12              has_superstructure_timber
## 13              has_superstructure_bamboo
## 14       has_superstructure_rc_engineered
## 15                                position
##                                     avNNet
## 1                          other_floor_type
## 2                              age_building
## 3     has_superstructure_mud_mortar_stone
## 4                       count_floors_pre_eq
## 5                         ground_floor_type
## 6                          height_ft_pre_eq
## 7                         plinth_area_sq_ft
## 8                                roof_type
## 9  has_superstructure_cement_mortar_brick
## 10                            district_id
## 11   has_superstructure_rc_non_engineered
## 12              has_superstructure_timber
## 13              has_superstructure_bamboo
## 14       has_superstructure_rc_engineered
## 15              has_superstructure_timber
##                                       WSRF
## 1                          other_floor_type
## 2                              age_building
## 3     has_superstructure_mud_mortar_stone
## 4                       count_floors_pre_eq
```

```
## 5                        ground_floor_type
## 6                         height_ft_pre_eq
## 7                        plinth_area_sq_ft
## 8                                roof_type
## 9  has_superstructure_cement_mortar_brick
## 10                             district_id
## 11   has_superstructure_rc_non_engineered
## 12              has_superstructure_timber
## 13              has_superstructure_bamboo
## 14       has_superstructure_rc_engineered
## 15                                position
```

5.3 Model creation

5.3.1 Data preparation

Based on our findings from the feature selection procedure, we limit the features in the data set to the relevant ones.

```
limited_buildings <- buildings[ , c('reconstruction_needed',
                                    'count_floors_pre_eq',
                                    'height_ft_pre_eq',
                                    'age_building',
                                    'plinth_area_sq_ft',
                                    'district_id',
                                    'foundation_type',
                                    'roof_type',
                                    'ground_floor_type',
                                    'other_floor_type',
                                    'has_superstructure_mud_mortar_stone',
                                    'has_superstructure_cement_mortar_brick',
                                    'has_superstructure_timber',
                                    'has_superstructure_bamboo',
                                    'has_superstructure_rc_non_engineered',
                                    'has_superstructure_rc_engineered')]
```

Then, we Z-normalize the continuous variables to prevent over-weighting the importance of variables.

```
znorm <- function(d){
  d.mean <- mean(d)
  d.dev <- sd(d)
  (d - d.mean)/d.dev
}

buildings$count_floors_pre_eq <- znorm(buildings$count_floors_pre_eq)
buildings$age_building <- znorm(buildings$age_building)
buildings$plinth_area_sq_ft <- znorm(buildings$plinth_area_sq_ft)
buildings$height_ft_pre_eq <- znorm(buildings$height_ft_pre_eq)
```

5.3.2 Model comparison and selection

For the model comparison and selection, we take again only a small proportion (1%) of the data set to speed up execution time.

```
model_selection_index <- createDataPartition(limited_buildings$reconstruction_needed,
                                             times = 1, p = ratio, list = FALSE)
ms_buildings <- limited_buildings[model_selection_index, ]
```

Then, we create the needed train and test sets.

```
test_index <- createDataPartition(ms_buildings$reconstruction_needed,
                                  times = 1, p = 0.5, list = FALSE)
test_set <- ms_buildings[test_index, ]
train_set <- ms_buildings[-test_index, ]
```

As a next step, we use a selection of models that we train on the train set. The models were selected with the intention to represent different model classes like decision trees and forests, boosting models, SVM, neural nets or weighted k-Nearest Neighbor.[2]

```
ms.gbm <- train(reconstruction_needed ~ ., method = "gbm", data = train_set)
ms.ranger <- train(reconstruction_needed ~ ., method = "ranger", data = train_set)
ms.xgbTree <- train(reconstruction_needed ~ ., method = "xgbTree", data = train_set)
ms.rf <- train(reconstruction_needed ~ ., method = "rf", data = train_set)
ms.svmLinear <- train(reconstruction_needed ~ ., method = "svmLinear", data = train_set)
ms.kknn <- train(reconstruction_needed ~ ., method = "kknn", data = train_set)
ms.avNNet <- train(reconstruction_needed ~ ., method = "avNNet", data = train_set)
ms.svmRadialCost <- train(reconstruction_needed ~ ., method = "svmRadialCost", data = train_set)
ms.naive_bayes <- train(reconstruction_needed ~ ., method = "naive_bayes", data = train_set)
ms.wsrf <- train(reconstruction_needed ~ ., method = "wsrf", data = train_set)
```

Then, we create predictions for all trained models.[3]

```
ms.gbm.pred <- predict(ms.gbm, newdata = test_set)
ms.ranger.pred <- predict(ms.ranger, newdata = test_set)
ms.xgbTree.pred <- predict(ms.xgbTree, newdata = test_set)
ms.rf.pred <- predict(ms.rf, newdata = test_set)
ms.svmLinear.pred <- predict(ms.svmLinear, newdata = test_set)
ms.kknn.pred <- predict(ms.kknn, newdata = test_set)
ms.avNNet.pred <- predict(ms.avNNet, newdata = test_set)
ms.svmRadialCost.pred <- predict(ms.svmRadialCost, newdata = test_set)
ms.naive_bayes.pred <- predict(ms.naive_bayes, newdata = test_set)
ms.wsrf.pred <- predict(ms.wsrf, newdata = test_set)
```

As a next step, we determine the confusion matrices for all models.

```
cm <- c(
  gbm = confusionMatrix(ms.gbm.pred,
                        test_set$reconstruction_needed)$overall["Accuracy"],
  ranger = confusionMatrix(ms.ranger.pred,
                           test_set$reconstruction_needed)$overall["Accuracy"],
  xgbTree = confusionMatrix(ms.xgbTree.pred,
                            test_set$reconstruction_needed)$overall["Accuracy"],
  rf = confusionMatrix(ms.rf.pred,
                       test_set$reconstruction_needed)$overall["Accuracy"],
  svmLinear = confusionMatrix(ms.svmLinear.pred,
                              test_set$reconstruction_needed)$overall["Accuracy"],
  kknn = confusionMatrix(ms.kknn.pred,
                         test_set$reconstruction_needed)$overall["Accuracy"],
```

[2]NOTE: Due to performance reasons, we do not employ an *apply* statement.
[3]AGAIN: Due to performance reasons, we do not employ an *apply* statement.

```
  avNNet = confusionMatrix(ms.avNNet.pred,
                           test_set$reconstruction_needed)$overall["Accuracy"],
  svmRadialCost = confusionMatrix(ms.svmRadialCost.pred,
                                  test_set$reconstruction_needed)$overall["Accuracy"],
  naive_bayes = confusionMatrix(ms.naive_bayes.pred,
                                test_set$reconstruction_needed)$overall["Accuracy"],
  wsrf = confusionMatrix(ms.wsrf.pred,
                         test_set$reconstruction_needed)$overall["Accuracy"]
)
```

Subsequently, we create an ensemble model to combine all models. We start by creating a dataframe with all predictions from the different models.

```
ensemble <- data_frame(
  ms.gbm.pred = ms.gbm.pred,
  ms.ranger.pred = ms.ranger.pred,
  ms.xgbTree.pred = ms.xgbTree.pred,
  ms.rf.pred = ms.rf.pred,
  ms.svmLinear.pred = ms.svmLinear.pred,
  ms.kknn.pred = ms.kknn.pred,
  ms.avNNet.pred = ms.avNNet.pred,
  ms.svmRadialCost.pred = ms.svmRadialCost.pred,
  ms.naive_bayes.pred = ms.naive_bayes.pred,
  ms.wsrf.pred = ms.wsrf.pred
)
```

We define an ensemble function. For every model, we take the mean value and then set to 'Yes' or 'No' based on the majority of values.

```
calc_ensemble_val <- function(r) {
  ifelse(mean(r == 'Yes') > 0.5, 'Yes', 'No')
}
```

We determine the predictions and calculate the confusion matrices of the ensemble model.

```
ensemble.pred <- apply(ensemble, 1, calc_ensemble_val)
ensemble.pred <- as.factor(ensemble.pred)
ensemble.cm <- confusionMatrix(ensemble.pred, test_set$reconstruction_needed)
cm <- c(cm, ensemble = ensemble.cm$overall["Accuracy"])
```

Finally, we determine the model with the best accuracy.

```
results <- sort(cm, decreasing = TRUE)
results
```

```
##      ensemble.Accuracy        avNNet.Accuracy svmRadialCost.Accuracy
##              0.7734523              0.7687049              0.7687049
##           gbm.Accuracy       xgbTree.Accuracy          wsrf.Accuracy
##              0.7673756              0.7643373              0.7586403
##     svmLinear.Accuracy        ranger.Accuracy            rf.Accuracy
##              0.7556020              0.7540828              0.7514242
##          kknn.Accuracy   naive_bayes.Accuracy
##              0.7345234              0.7314850
```

All models are characterized by an accuracy between 0.77 and 0.73 and seem to be sufficient for prediction of the need for reconstruction. The results show that the ensemble model with an accuracy of about 0.77 is slightly better than the other models. Below wepresent a summary of the confucion matrix and various model indicators.

```
ensemble.cm
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   No  Yes
##        No  1528  509
##        Yes  684 2545
##
##                Accuracy : 0.7735
##                  95% CI : (0.7619, 0.7847)
##     No Information Rate : 0.5799
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.5299
##  Mcnemar's Test P-Value : 4.713e-07
##
##             Sensitivity : 0.6908
##             Specificity : 0.8333
##          Pos Pred Value : 0.7501
##          Neg Pred Value : 0.7882
##              Prevalence : 0.4201
##          Detection Rate : 0.2902
##    Detection Prevalence : 0.3868
##       Balanced Accuracy : 0.7621
##
##        'Positive' Class : No
##
```

6 Conclusion

This analysis has shown that the need for reconstruction of buildings caused by the 2015 Nepal earthquake can be confidently predicted from its features. In particular, the height and the count of floor before the earthquake, the age of the building, the plinth area, the district ID, the foundation, roof type and ground floor types, as well as categorical variables regarding the superstructure have a significant effect on the need for reconstruction. The findings from this analysis could be used for better future construction of earthquake-proof buildings in Nepal.

Appendix A: Additional feature description

The following variables of the data set seem to have no relationship with the label reconstruction_needed, i.e. they have no impact if the buildings need reconstruction after the earthquake or not. Visualizations of the relationship are included in this Appendix for reasons of completeness.

- land_surface_condition (see Figure 17)
- position (see Figure 18)
- plan_configuration (see Figure 19)
- has_superstructure_adobe_mud (see Figure 20)
- has_superstructure_stone_flag (see Figure 21)
- has_superstructure_cement_mortar_stone (see Figure 22)
- has_superstructure_mud_mortar_brick (see Figure 23)
- has_superstructure_other (see Figure 24)

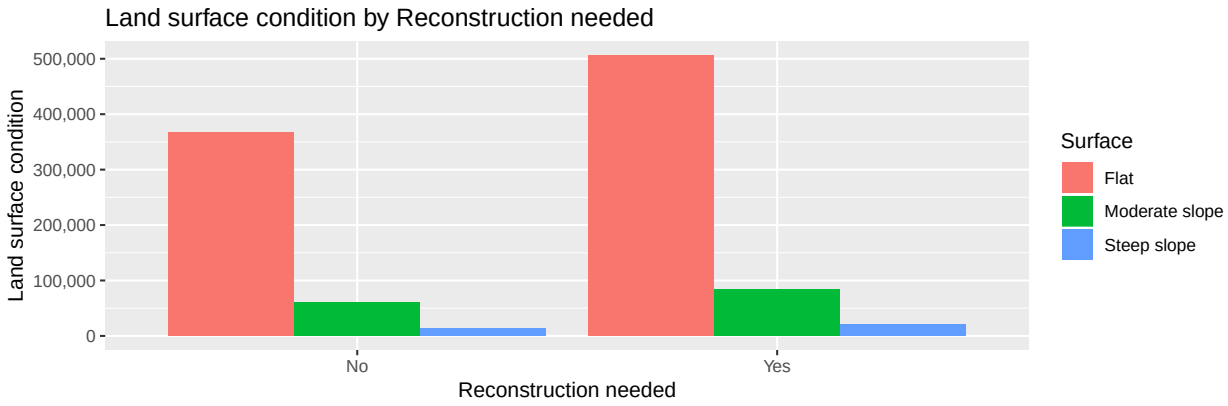

Figure 17: Land surface condition by Reconstruction needed

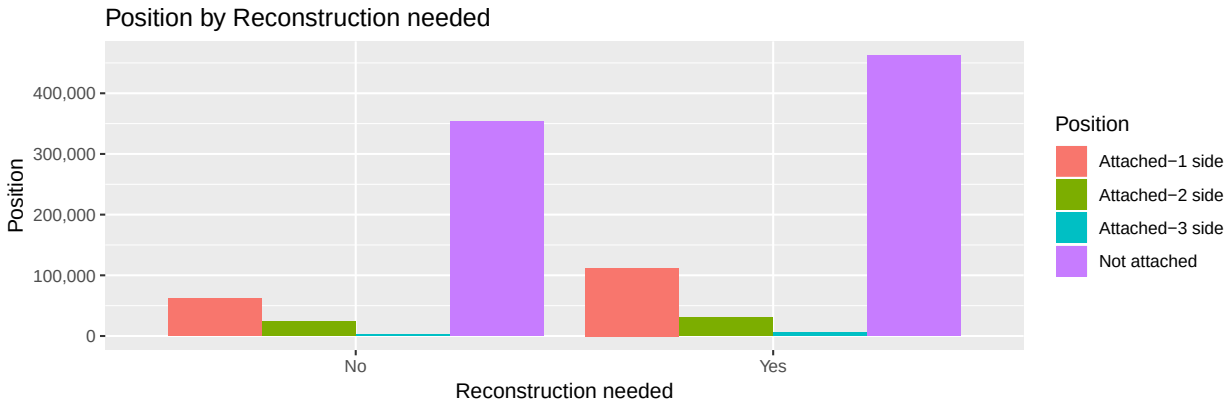

Figure 18: Position by Reconstruction needed

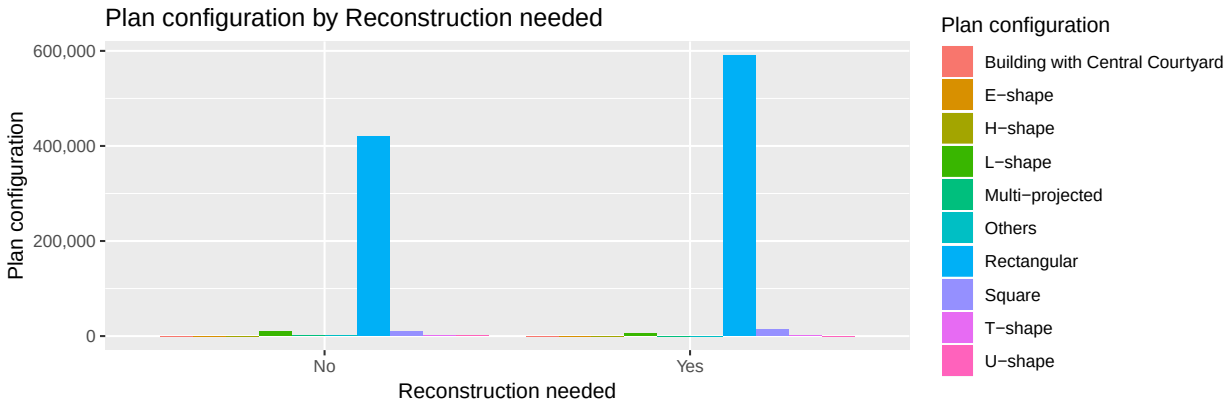

Figure 19: Plan configuration by Reconstruction needed

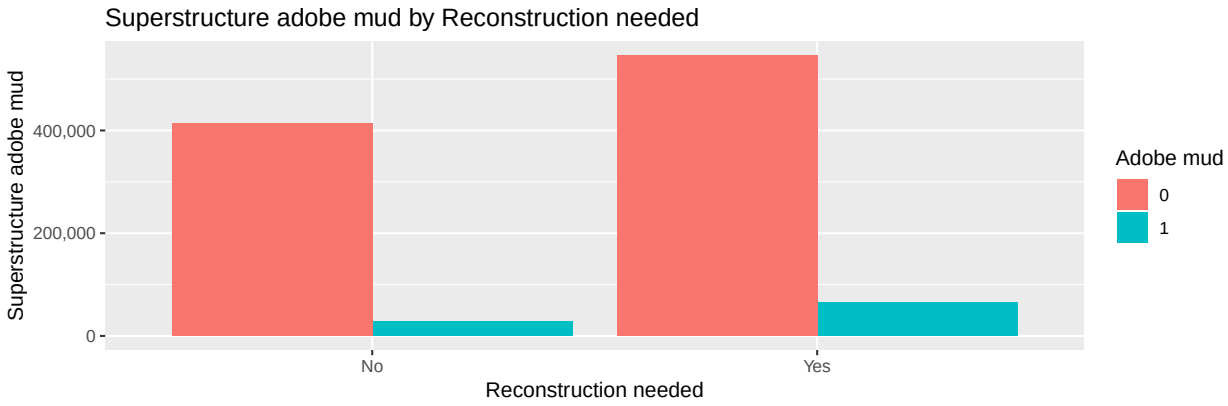

Figure 20: Superstructure adobe mud by Reconstruction needed

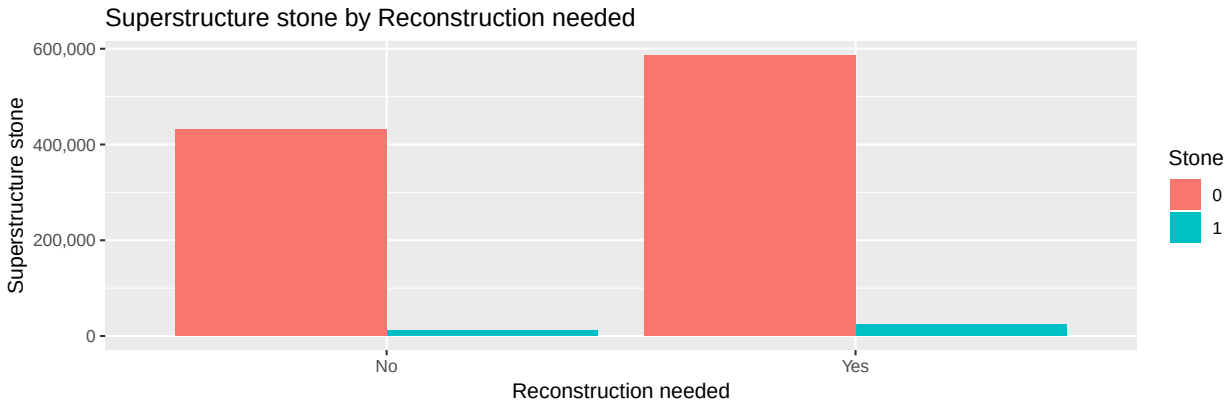

Figure 21: Superstructure stone by Reconstruction needed

Figure 22: Superstructure cement mortar stone by Reconstruction needed

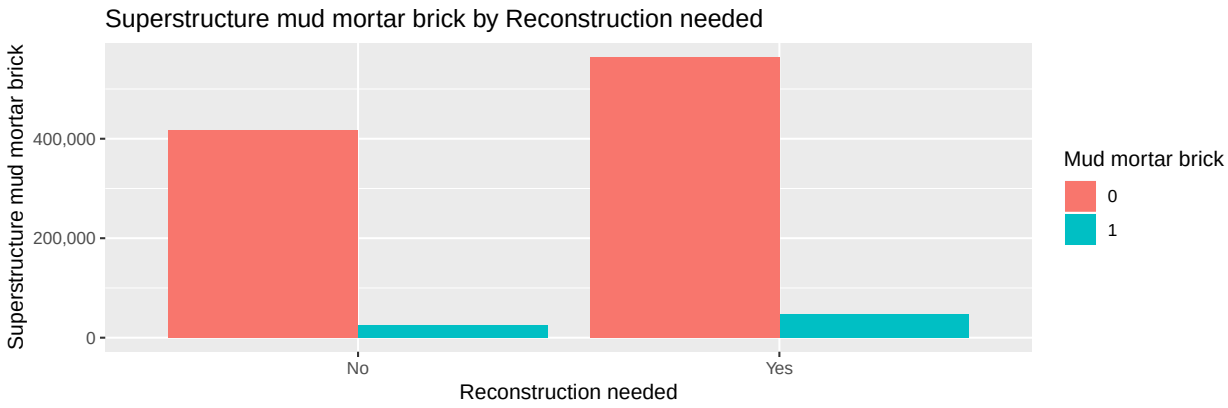

Figure 23: Superstructure mud mortar brick by Reconstruction needed

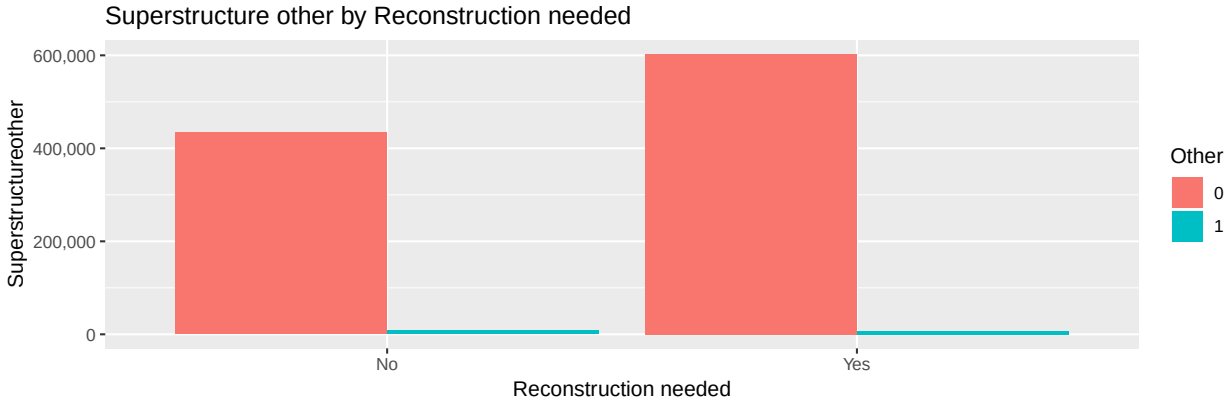

Figure 24: Superstructure other by Reconstruction needed