Richter's Predictor: Modeling Earthquake Damage

Makhmutshina Kamila, Chernysheva Mariya, Severinov Yuriy

Case overview

Situation

- The April 2015 Gorkha earthquake killed ~ 9,000 people and injured ~ 22,000, millions of people were instantly made homeless, and $10 billion in damages – about half of Nepal’s nominal GDP
- After that disaster has been generated one of the largest post-disaster datasets ever collected, containing valuable information on earthquake impacts, household conditions, and socio-economic-demographic statistics

Purpose of the research

- Our work on modeling the prediction algorithm may help in the future, to customize insurance, to choose construction type of buildings in seismic dangerous districts which can help to save people’s lives, money and make the situation with earthquakes less disastrous

Task

We need to predict the ordinal variable `damage_grade`, which represents a level of damage to the building that was hit by the earthquake.

There are 3 grades of the damage:

- 1 - represents low damage
- 2 - represents a medium amount of damage
- 3 - represents almost complete destruction

EDA

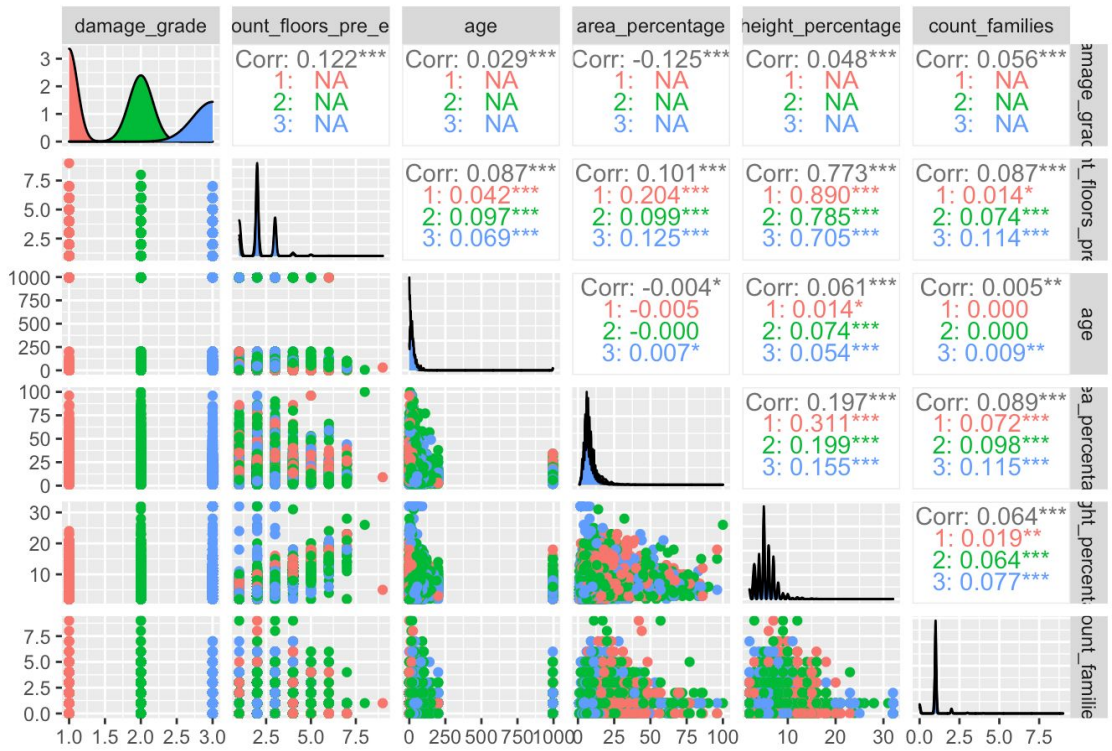
Dataset structure

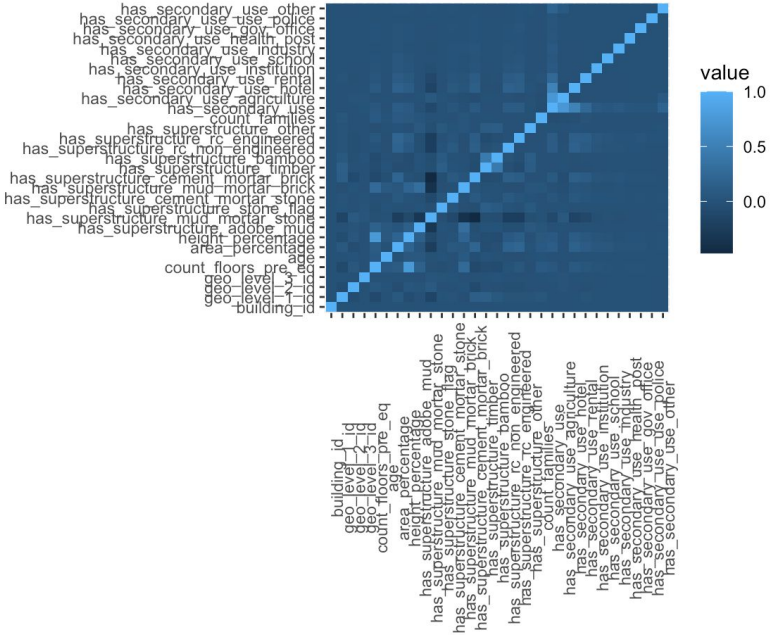
The dataset mainly consists of information on the buildings’ structure and their legal ownership

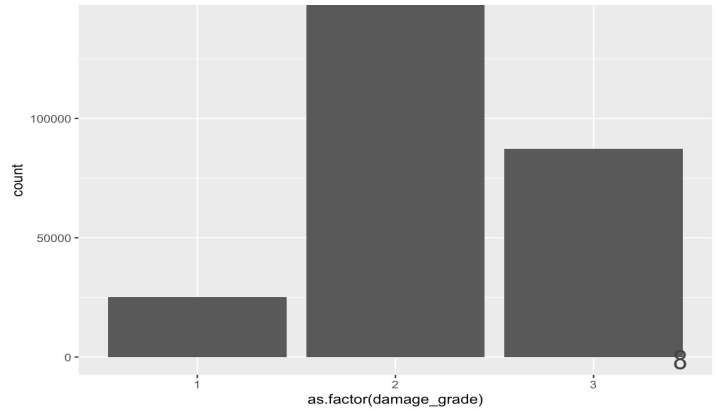
There are 39 columns in this dataset in four groups:

- basic information about building
- its location
- information about building construction
- information about ownership

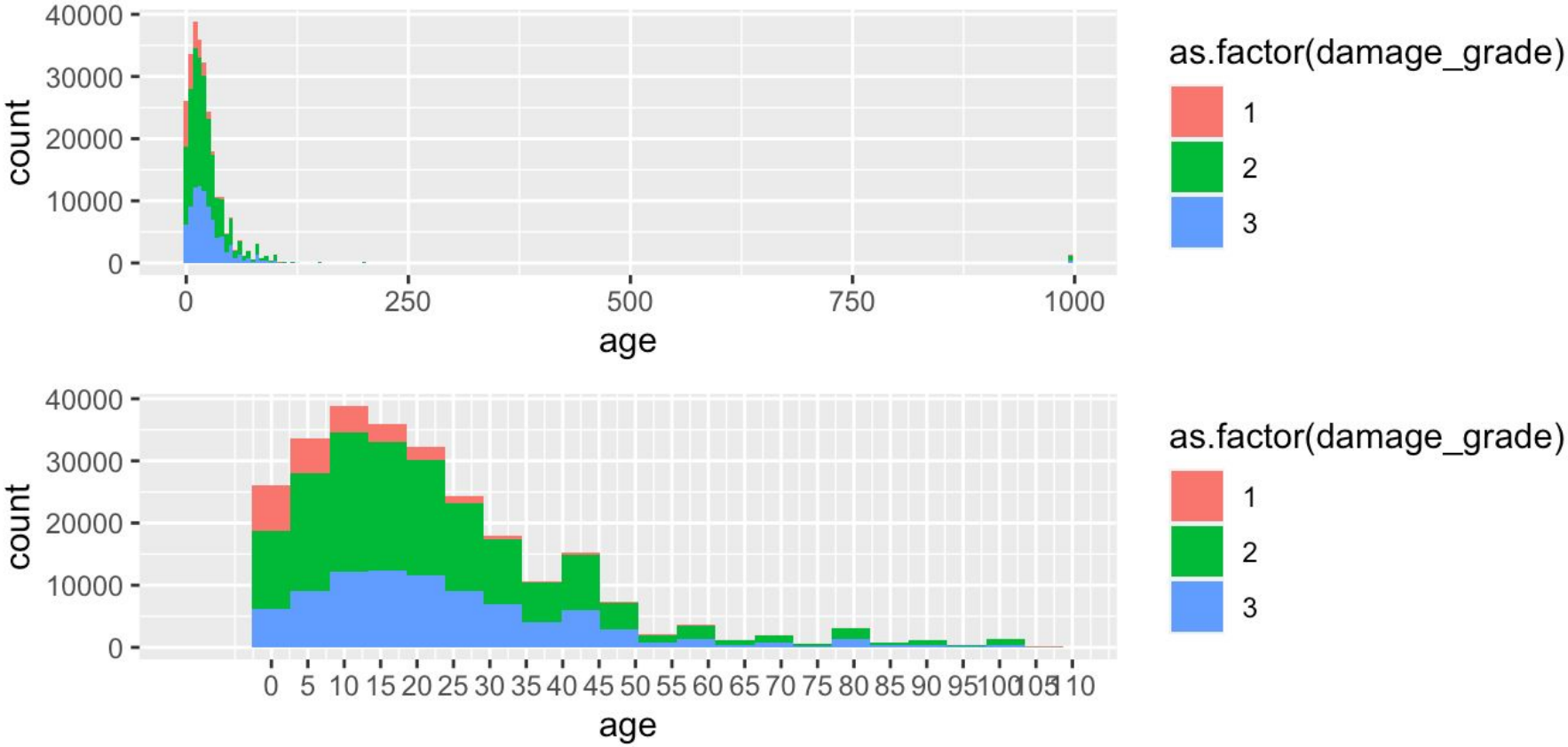
Quick look at the data



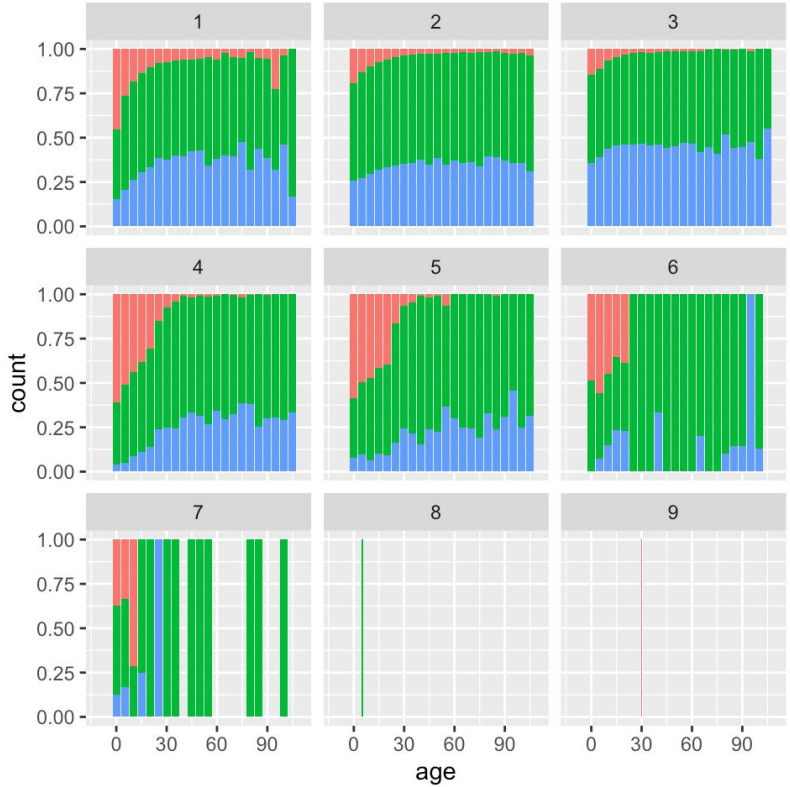


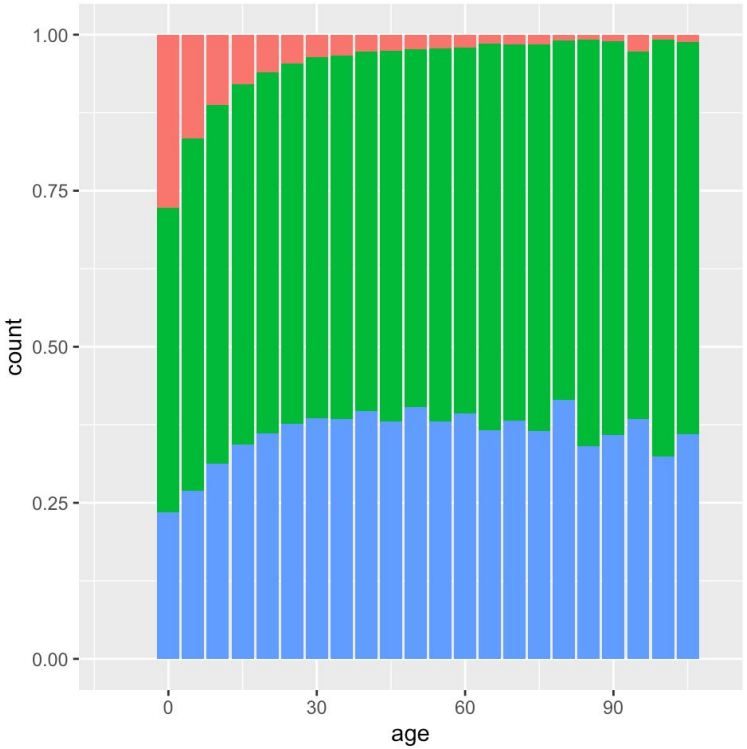


Distribution of damage across age

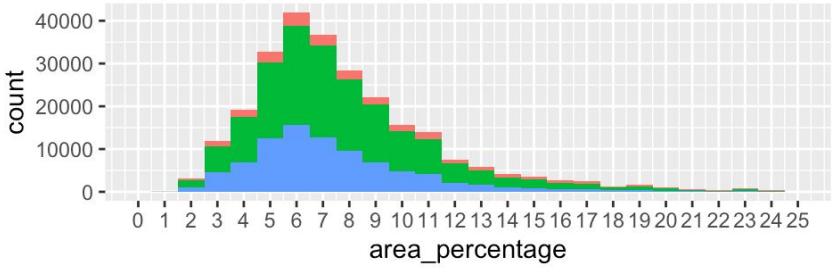


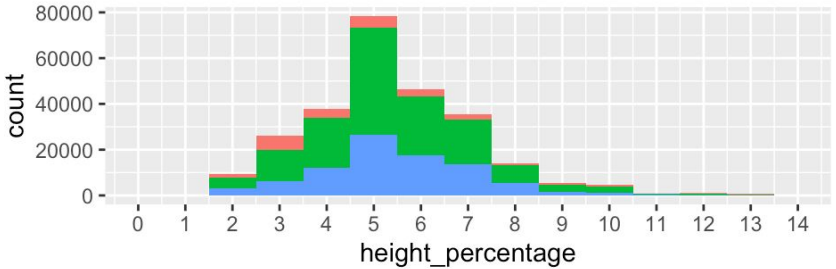
Distribution of damage across age

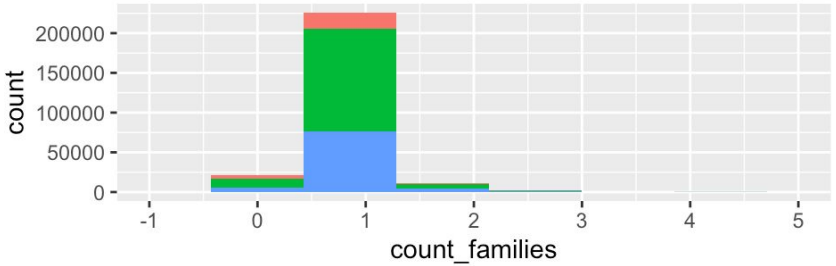


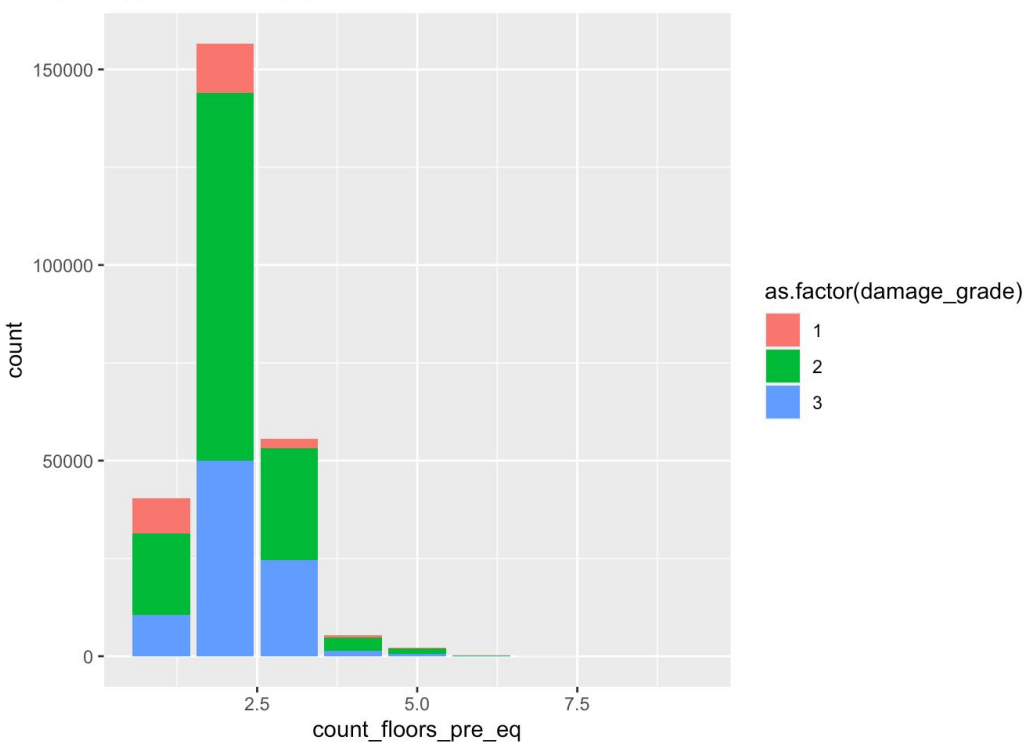


Distribution of damage across other variables

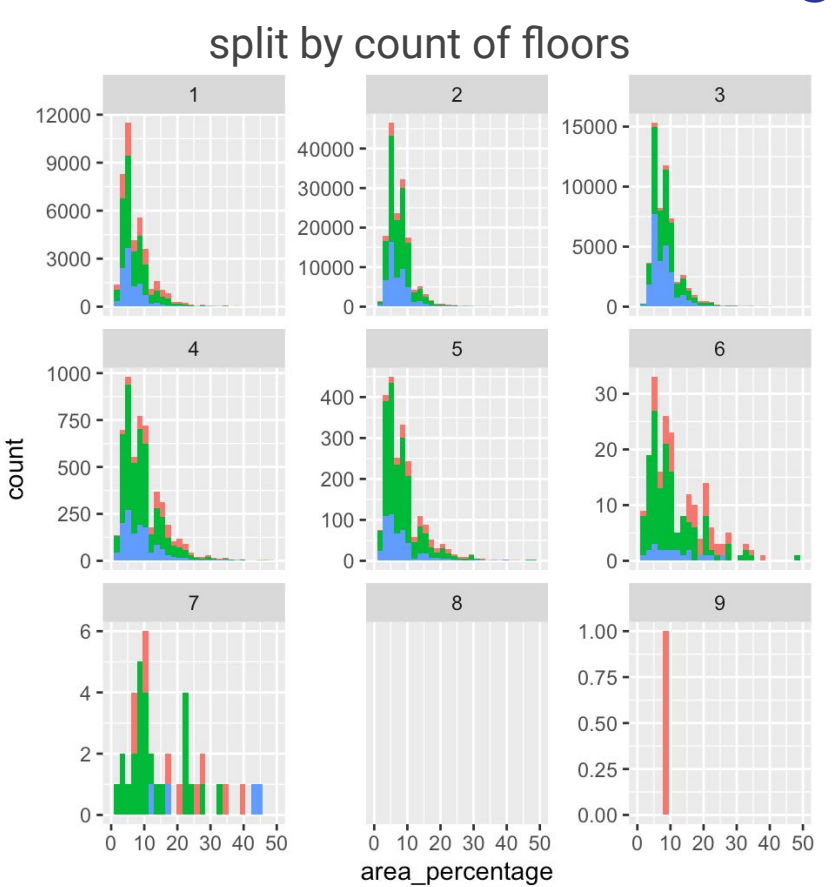


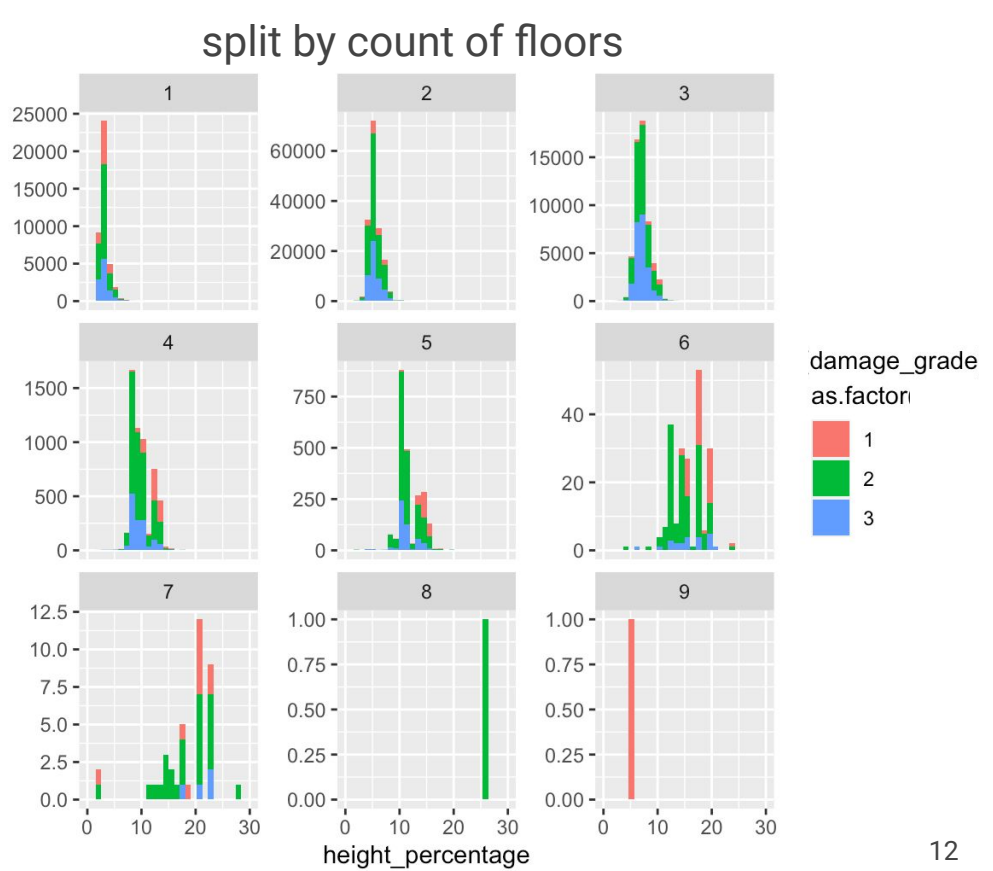




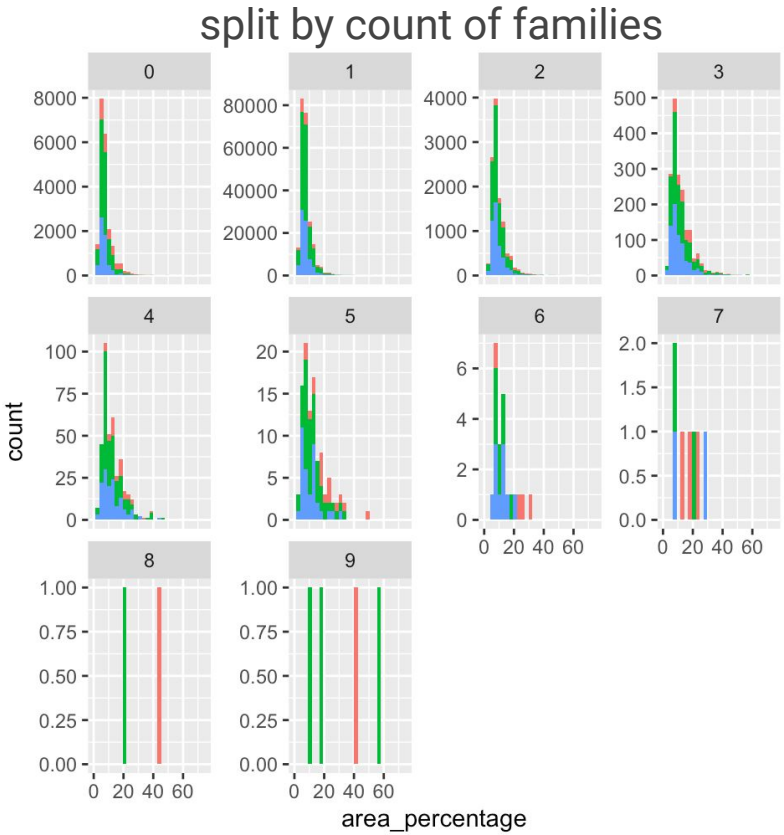


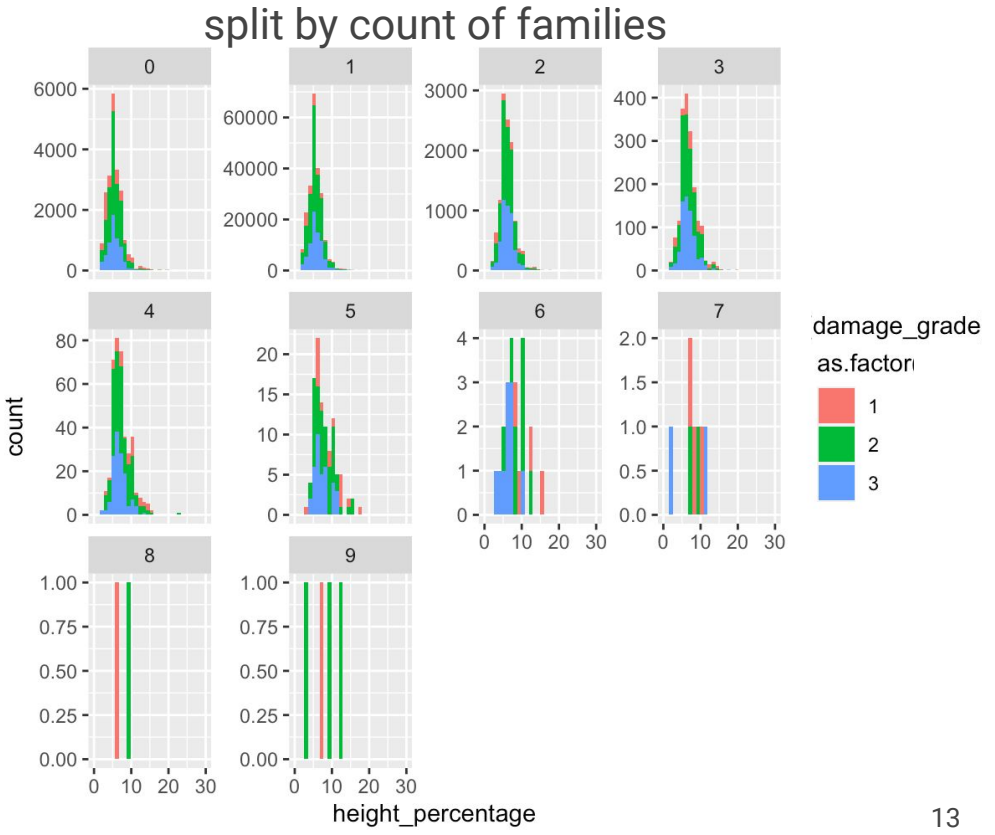
Distribution of damage across other variables





Distribution of damage across other variables





What we concluded after EDA

- `area_percentage` and `height_percentage` are correlating
- `has_secondary_use` is correlating with its subtypes
- `height_percentage` is highly correlating with `count_floors_pre_eq`
- `area_percentage` and `height_percentage` are correlating with `has_super_structure` features and secondary use of buildings
- the older buildings had more damage, than the newer ones, but more newer buildings were damaged
- percent of seriously damaged higher buildings is much more than of the lower ones

How we prepared the data

We made the following transformations with the data to make them eligible for the applied models in order to ultimately optimize the final result:

- converted categorical variables into factors
- converted categorical binary variables into dummies
- relevelled dataset on the basic damage value of 2

How we prepared the data

We tried to implement the following extensions, but with them dataset expanded up to 79 variables and we were unable to calculate any model on the subset larger than 1% of all data:

- count_of_floors_per_age <- count_floors_pre_eq/(age+1)
- count_of_floors_per_area <- count_floors_pre_eq/area_percentage
- count_of_floors_per_height <- count_floors_pre_eq/height_percentage
- area_per_age <- area_percentage/(age+1)
- height_per_age <- height_percentage/(age+1)
- families_on_floor <- count_families/count_floors_pre_eq
- families_on_area <- count_families/area_percentage
- families_on_height <- count_families/height_percentage

With that implementations final result might have been much higher

Making models

What models we’ve chosen

To automatize the modeling process, we decided to use mlr package with next set of models:

- randomForest
- xgboost
- multinom
- rpart
- ctree
- C50

What is the best

We have made two benchmark models – full stack with six models on 10% of data and stack of following models made on 50% of data:

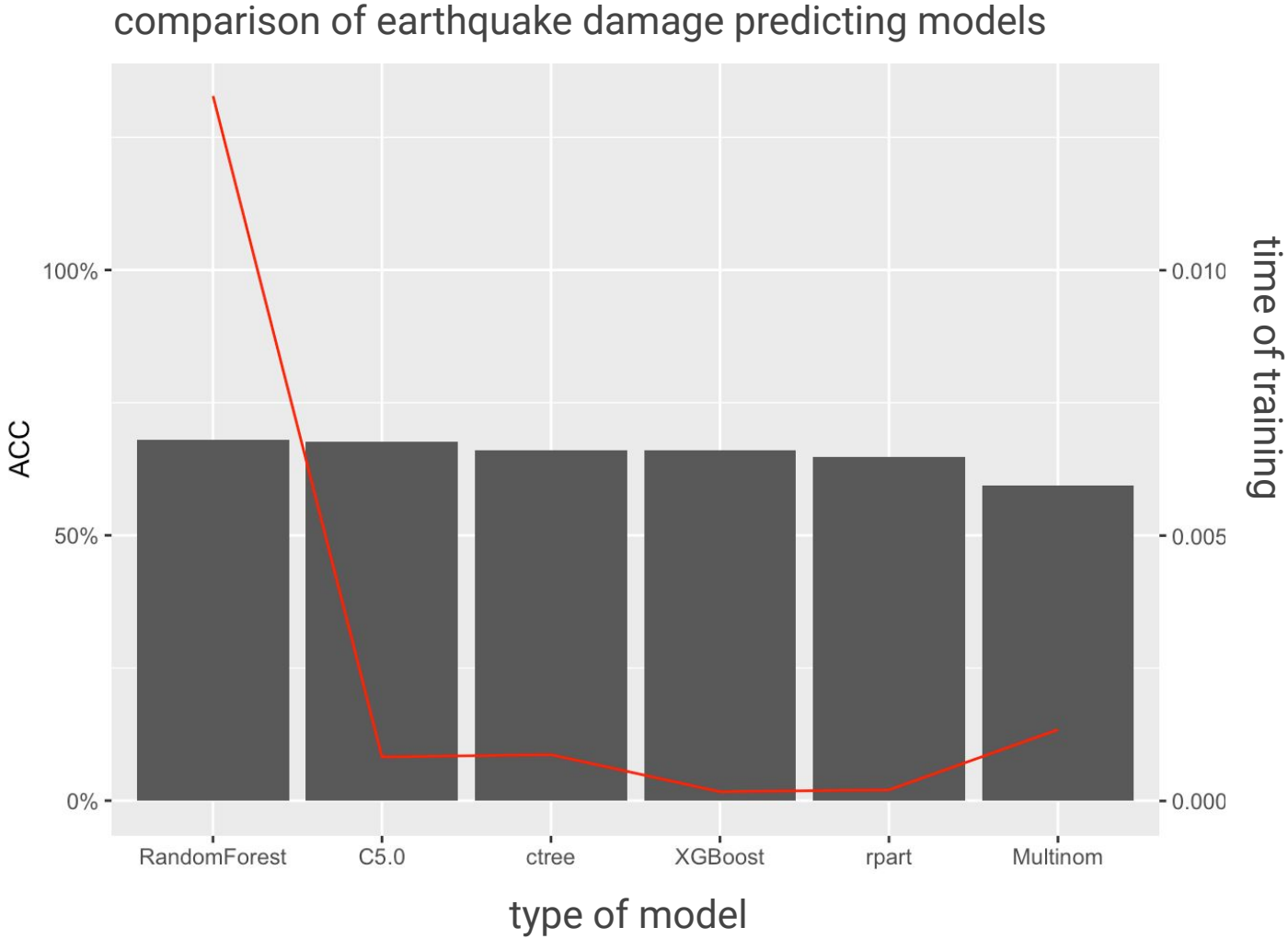
- randomForest
- xgboost
- rpart
- C50

What is the best

After the first benchmark, we found that some of the models are better for that project

	learner.id	timetrain.test.mean	acc.test.mean	kappa.test.mean	mmce.test.mean
1	RandomForest	132.789	0.680	0.366	0.320
2	C5.0	8.239	0.676	0.381	0.324
3	ctree	8.660	0.661	0.333	0.339
4	XGBoost	1.680	0.660	0.304	0.340
5	rpart	2.025	0.647	0.260	0.353
6	Multinom	13.386	0.594	0.160	0.406

So we decided to train them on half of the dataset


comparison of earthquake damage predicting models
ACC
100%
50%
0%
0.010
0.005
0.000
time of training
RandomForest
C5.0
ctree
XGBoost
rpart
Multinom
type of model

What is the best

After the second benchmark, C5.0 and Random Forest are the best

```
##      learner.id timetrain.test.mean acc.test.mean kappa.test.mean mmce.test.mean
## 1          C5.0                2.528         0.667            0.351          0.333
## 2 RandomForest               59.403         0.662            0.315          0.338
## 3       XGBoost                0.372         0.651            0.275          0.349
## 4         rpart                0.492         0.643            0.224          0.357
```

But C5.0 is much higher, so for the next analysis we use it

So we have trained it on the full dataset

Predicting

Using model on the new data

After predicting values on the new data from the competition we got the following result

Submissions

BEST	CURRENT RANK	# COMPETITORS	SUBS. MADE
0.7177	676	4322	0 of 3

We think that the result is pretty good for our model, made in learning purposes, trained on bad performing machines. After tuning the model and data transformations, result might be much higher.

Limitations and following usage

With proper modifications and adjustments our model can be used for different purposes in different countries to prevent economical, social, demographic consequences after different earthquake and may be other disastrous.

Limitations of implementing that model in future projects are the same dataset structure, similar conditions and characteristics of the location.

Questions?