ML Project Report

# Problem & Dataset

# Preparing the data

Clean data: real dataset with many redundant features, incorrectly typed in features, null values…

Strategy: inspect each feature one by one, determine if worth keeping, impute null values if any, correct typos if possible, some feature engineering if applicable.

Examples:

* Extract information from *date\_recorded* to create new features
* Drop features with too many different values or null values
* Impute null values in some numerical features by taking the mean of the values of its geographical neighbors (that is points in the same subvillage, ward, lga as them)
* Drop features that represent geographical locations as they are redundant with gps coordinates
* Group some feature values together to reduce cardinality and correct possible mistakes
* Correct impossible features (wrong gps coordinates, construction year…)

# Exploring the data

# Comparing baseline models

Compared several classic baseline models for classification: decision tree, random forest, support vector machine, stochastic gradient descent classifier, k-NN classifier, Gaussian naïve Bayes, MLP classifier…

SVM: 1.0/0.535 (please try with another parameter but it takes so much time for this model to train)

kNN: 0.633/0.496, limit: we have high number of input variables here

Naïve Bayes: 0.629 / 0.627, limit: we probably don’t have independence of features

# Tuning our best models

Out of the baseline models, random forest has the best performance which we will try to improve.

It makes sense for this model to be well-suited to our problem, since unimportant features don’t matter in their decision (so no need for feature selection), collinearity is ok, and it is fairly robust to data with many errors or missing values.

We use Grid Search to find the best hyper-parameters for Random Forest. Notice that the training accuracy is always higher than the validation accuracy, so we should use a regularization method to add bias and reduce overfitting, or tune hyperparameters to increase bias (more trees, less features per tree, smaller depth, .

# Analysis of results

# Conclusion