Predicting the real estate prices in Estonia

Jüri Jõul, Sten Salumets, Tarvi Tepandi

Institute of Computer Science, University of Tartu

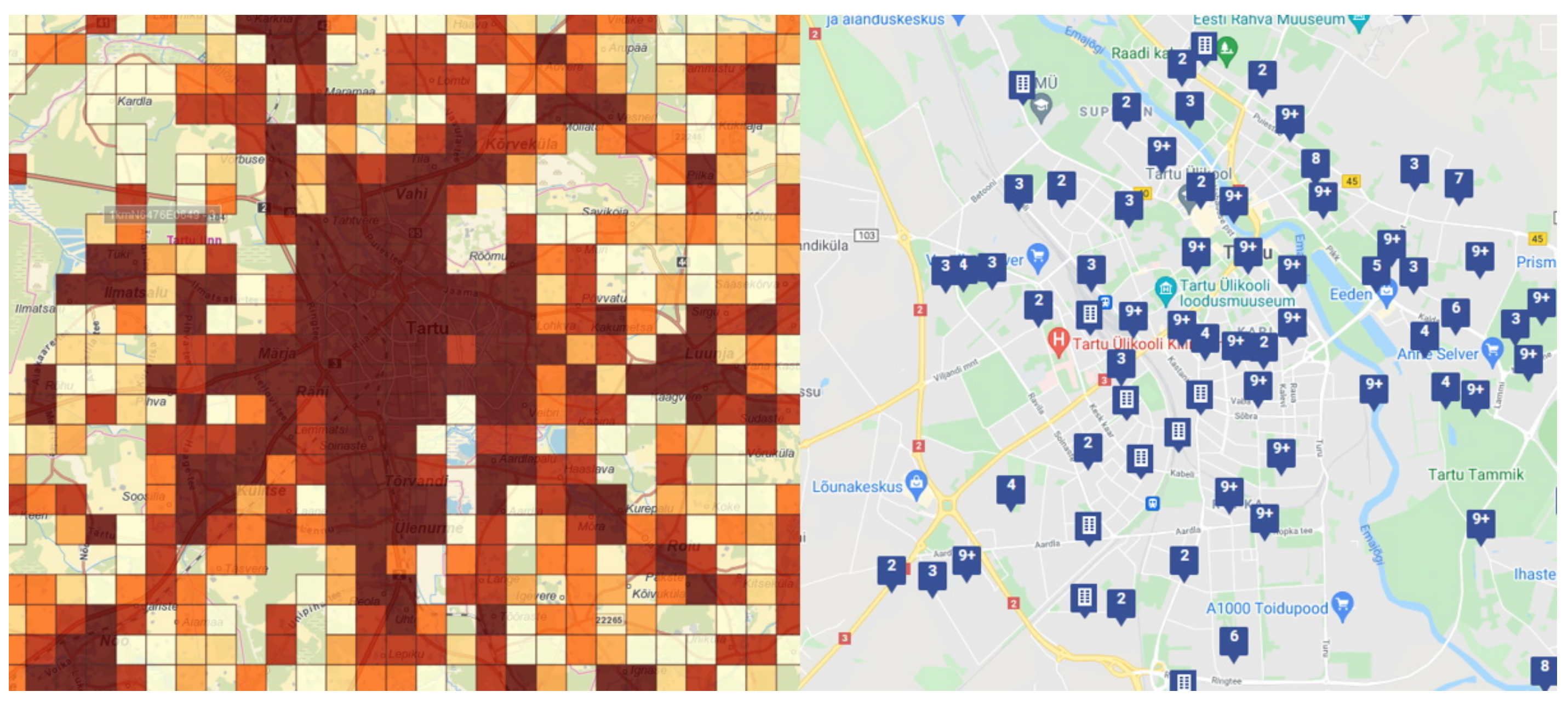
Introduction

The real estate market and its fluctuations often cause real estate appraisals to be difficult. Most commercial real estate sites, such as `www.kv.ee`, allow the sellers to list many distinct features in an advertisement which could all affect the final price [4]. The price could also be affected by external factors such as population density of an area. A machine learning model could be created to give an automated appraisal to any real estate, in order to aid sellers set optimal prices to their advertisements and to help buyers avoid buying overpriced real estate. We have done just that!

Data scraping

The most popular real estate site in Estonia, `www.kv.ee`, has over 7000 advertisements listed at the time of writing [4]. Every listing has distinct features in specific fields. Since there are no public Estonian real estate datasets, all the data needed to be scraped from the internet. This was done using a Python web scraping library BeautifulSoup [1]. As a result, all available advertisements with all their text-based features were included into our dataset.

In addition, based on the location in the listing, daily population metric per average in the 1 km^2 area was added as a feature. This was scraped from the latest (2016) available dataset provided by the Estonian statistical office [2].



Data preprocessing

A number of features were a mix of information about the real estate, separated with commas. One example of this is the 'Additional information' feature. It included information about the size of the balcony, the situation with parking, the existence of a garage, etc. All of these had to be separated.

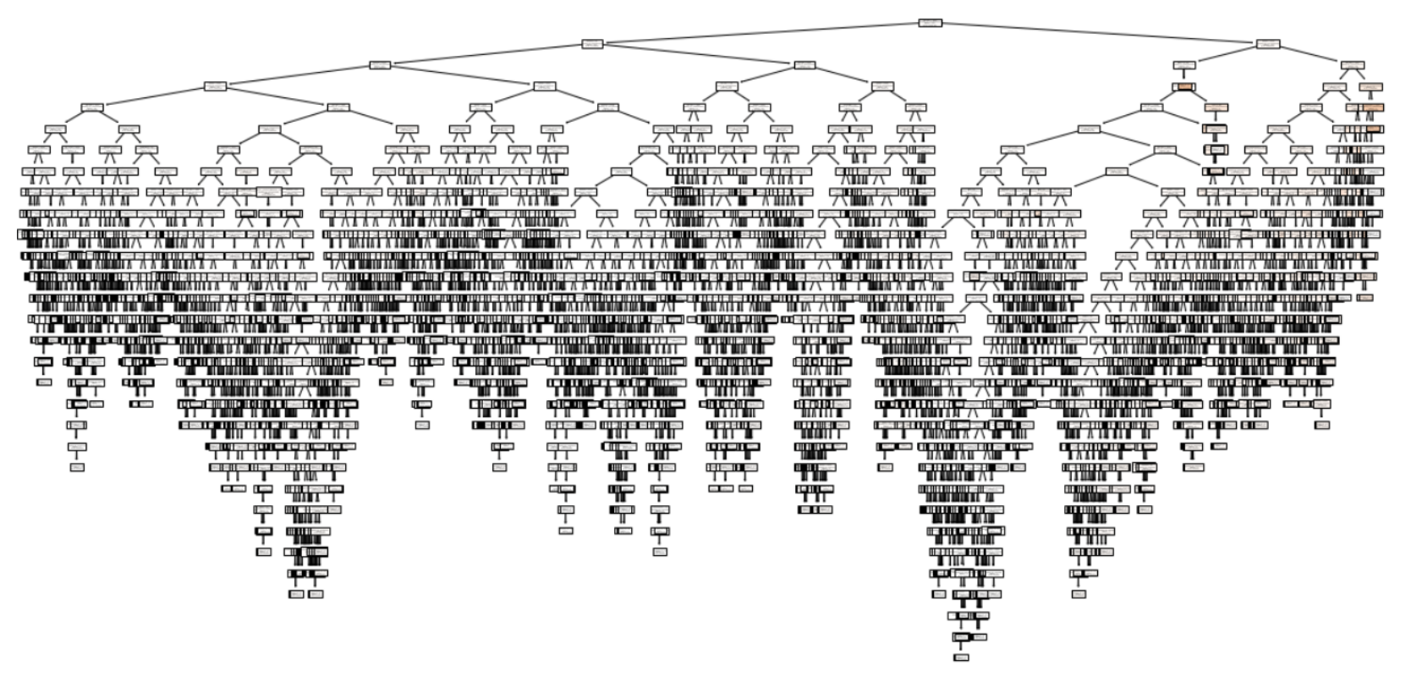
This was followed by simple data type conversions, including removing units from features like 'Total area'. Also, we converted categorical values like 'Energy mark' and 'Heating type' into numerical values or dummy/indicator variables.

Finally, we were left with dealing with the numerous NaN values in the dataset. Since our goal was to predict price, the entries that were missing the real estate's price were removed entirely. Completely removed were the features that had more than 50 percent of their values as NaN. The remaining few NaN were filled with either the features' mode or mean values.

Algorithms

Initially, the Random Forest Regression algorithm was selected because of its suitability for use with tabular data, such as ours [5]. It is also a learning ensemble which means that any errors in a single base model will most likely be mitigated. In addition, the model is relatively fast to train (although still slower than linear models) and robust [3].

The Linear Regression and ElasticNet models were tried next [6]. An important aspect of both of those models is that they assume the relationships in the data to be linear. We, however, realized that this may not always be the case with real estate features. Still, the models were used as a benchmark and also because they are faster compared to the aforementioned random forest regression model. To give an idea of the random forest regressor's complexity, one of its component trees can be seen below.



Evaluation

The random forest regression model returned an R^2 score of **0.72**, which is our best result. The linear regression and ElasticNet were R^2 evaluated at **0.36** and **0.29**, respectively.

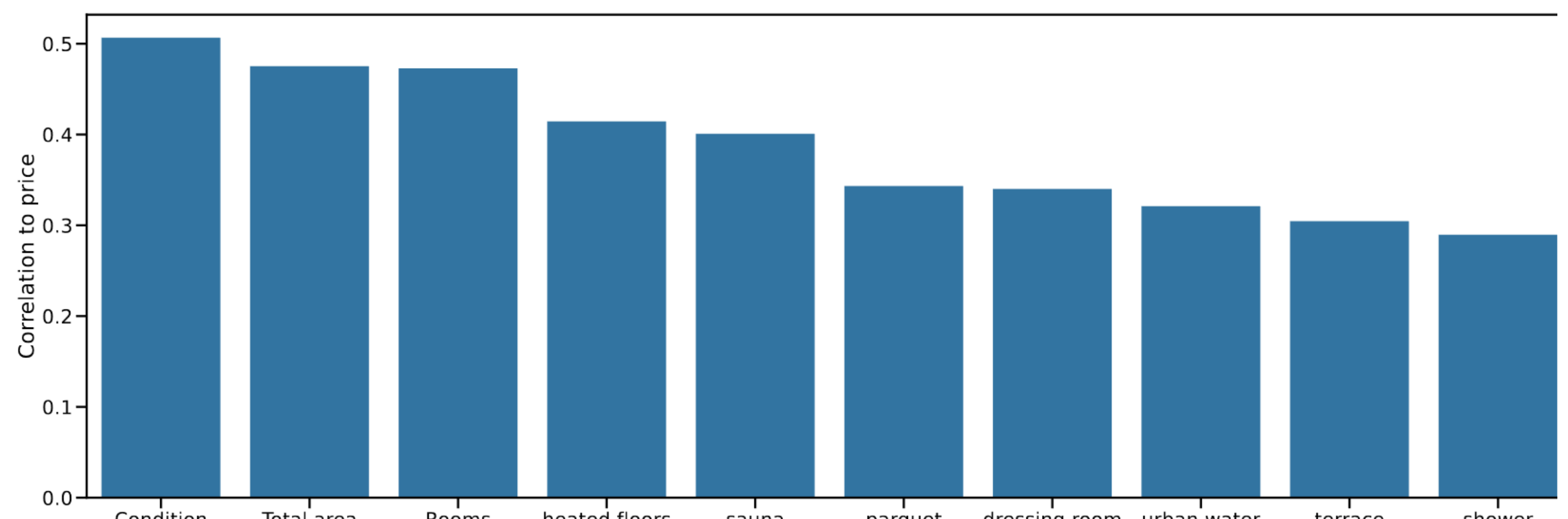
Even our best model was vastly limited by the amount of poorly entered real estate advertisements in `www.kv.ee` [4]. For example, there are entries where the ground area (size of the land) had been entered as the total area (indoors area). This caused the model to think that there were estates with an indoors area of thousands of square meters. Similarly, some of the listings had their condition listed as "brand-new", while in reality they were built many decades ago.

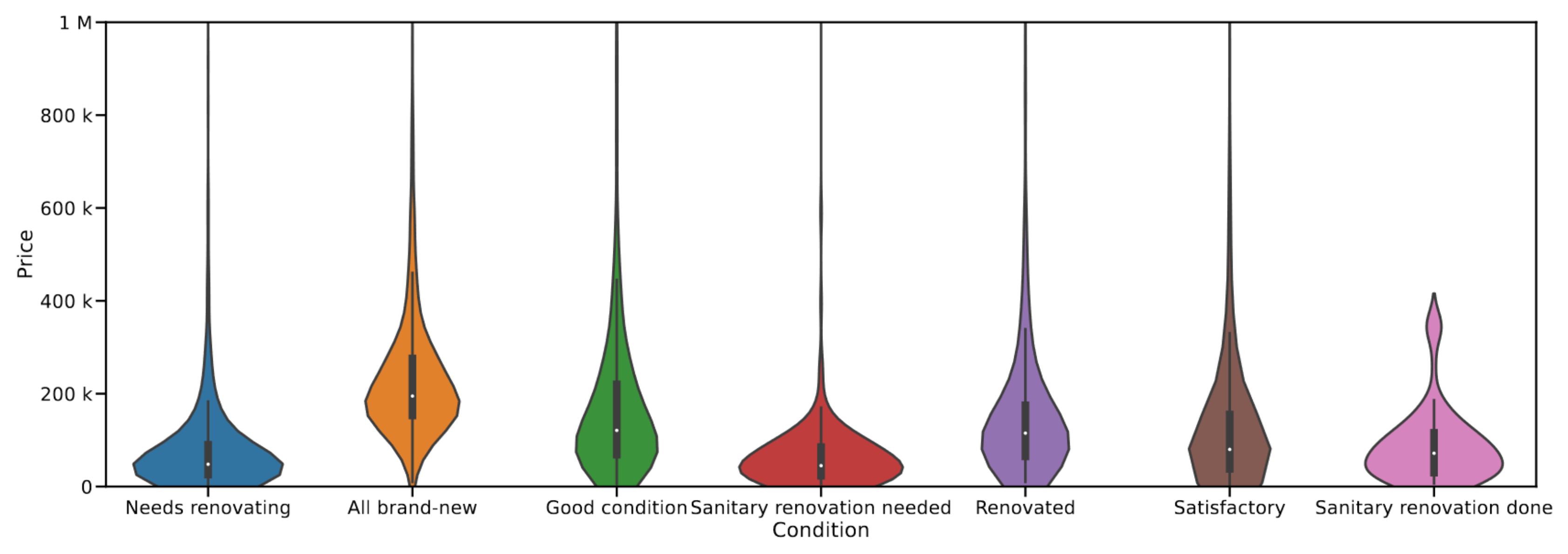
Model	R^2 Score
ElasticNet	0.29
Linear Regression	0.36
Random Forest Regressor	**0.72**
Human average	<0.4

For showing the relevance of our work, we developed a software that lets humans try to guess the price of real estate by the same features that were given to our model. The code takes random entries from the dataset, shows their features' values, and prompts the user to enter a prediction of the price. After entry, the correct answer and the price that our model predicted is presented. In addition, the user can find out their own R^2 score.

In testing it with our team, we achieved human R^2 results in the range of **-0.26** to **0.58**, with the averages being under **0.4**. Everyone is welcome to try it out themselves at the poster session!

Since one of our goals was determining the most relevant features for predicting the price, we have visualized the top 10 features with the highest correlations to the price. Also, the distribution of the price for the values of the most relevant feature is shown.





References

[1] *Beautiful Soup Documentation.* 2015. URL: `https://beautiful-soup-4.readthedocs.io/en/latest/`.

[2] *Estonian Statistical Office's statistical map.* 2021. URL: `https://estat.stat.ee/StatistikaKaart/VKR`.

[3] Vladimir Lyashenko. *How to use random forest for regression.* 2021. URL: `https://cnvrg.io/random-forest-regression/`.

[4] *Real estate KV.EE.* 2021. URL: `www.kv.ee`.

[5] *sklearn.ensemble.RandomForestRegressor in scikit-learn documentation.* 2021. URL: `https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html`.

[6] *sklearn.linear_model in scikit-learn documentation.* 2021. URL: `https://scikit-learn.org/stable/modules/classes.html#module-sklearn.linear_model`.