Application of methods based on ensembles and deep neural networks to estimating the cost of commercial real estate

Sergey Tkachenko*, Valery Burmistrov, and Svetlana Shpilevaya

Immanuel Kant Baltic Federal University, Educational and scientific cluster "Institute of High Technologies", 236041 Kaliningrad, Russia

Abstract. The paper considers the possibility of using ensemble machine learning models and artificial neural networks to solve the problem of assessing the value of commercial real estate. There are some models such as the gradient boosting model and the TabNet model have been trained. The main goal of these models is predict the value of commercial real estate without creating dependencies between data by the analyst. The proposed solutions are considered from the point of view of the banking sector. The best predictive model is the gradient boosting model implemented using the LightGBM library. The advantages of this model are associated with its ability to "resist" the presence of outliers in the data and a low propensity for retraining.

1 Introduction

Today, the banking industry is the most important part of the economy, which provides a large number of opportunities and tools for financial management of both large organizations and individuals.

Separately, it is worth noting the sphere of crediting. A bank loan is the issuance by a credit institution of funds to an individual or legal entity, which the borrower undertakes to return within the prescribed period with certain percentages [1]. Lending is especially useful for legal entities because it allows getting funds for operating activities, the purchase of expensive raw materials, the diversification of production or the purchase of equipment.

In such cases, the loan size can reach hundreds of millions of roubles. The bank forms an offer based on the current turnover of the company and account balances, and often the bank loans only secured by commercial real estate. Thanks to the collateral, the bank can provide customers with more favourable credit conditions and, in case of loss of the customer's solvency, can sell the collateral property to repay the debts.

The assessment of commercial real estate is a separate task, for which the bank can use both its own specialists and use the services of appraisal companies with which the bank cooperates. To determine the value of a real estate object, as a rule, the liquidation value is used, i.e. the value that allows the real estate object to be sold in 3-6 months. The valuation

* Corresponding author: tkasergey@yandex.ru

© The Authors, published by EDP Sciences. This is an open access article distributed under the terms of the Creative Commons Attribution License 4.0 (https://creativecommons.org/licenses/by/4.0/).
of the collateral object makes it possible to establish a relationship between the value of the mortgaged property and the size of the loan provided.

The main approach to assessing the value of real estate is a comparative treatment. The main difficulty is the analysis of the real estate market, as well as the availability of information for analysis. There are companies that publish information about the commercial real estate market, but detailed data applies only to large cities, while other towns are absent (for example, Knight Frank [2]). In addition, it is worth noting that appraisers select from three to five analogue objects, which may not always be enough. Thus, this approach requires highly qualified specialists with extensive experience.

The evaluation time of one commercial real estate object can vary from several hours to several days. This time with the help of an automated tool can be significantly less, which in conditions of high competition in the banking sector can be an important advantage. To develop such a tool, it is necessary to process a huge array of data and perform high-quality analytics. Machine learning is perfect for solving such a problem.

2 Data for machine learning

The work uses data from the online hackathon on Data Science, which was conducted by Raiffeisenbank. The dataset is publicly available on the Kaggle platform [3]. This dataset is distributed under the CC BY-NC-SA 4.0 license, which allows you to copy, distribute, modify and create new ones based on data. Data provided for 2020.

The data is provided for the period from January 5 to December 23, 2020 and contains descriptions of commercial real estate objects presented in 47 subjects of the Russian Federation. The description of the property contains its various characteristics, including: region, city, coordinates in WGS84 format, type of real estate, floors on which the building is located, the area of the object, various characteristics describing the location of the object (distance to the nearest metro station, number of buildings in a certain radius, etc.), various characteristics describing the buildings surrounding the object (average number of floors of buildings in a certain radius, average year construction of buildings in a certain radius, etc.). The target feature is "per_square_meter_price" - the cost per square meter expressed in roubles.

In total, the data set contains 279792 real estate objects, which are described by 77 features. Particular attention should be paid to the attribute “price_type” which takes values of 0 and 1. Properties that were manually priced by an appraiser have a “price_type” equal to 1. Prices for properties with a “price_type” equal to zero were obtained from advertisements with offers to purchase commercial real estate. Thus, the dataset contains 4493 objects marked up by experts and 275299 objects with prices from ads. From this, we can conclude that despite the rather large size of the entire dataset, the number of relevant data obtained by experts is less than 5000. The rest of the data, because the cost was indicated not by appraisers, but by the owners of real estate objects interested in selling them, contain biased estimates and much more noise.

Since the cost per square meter largely depends on the type of object (for example, the cost per square meter in a workshop on the outskirts will be less than the cost of a similar area in a boutique in a shopping center), it is worth noting the “realty_type” attribute, which is responsible for the type of commercial property. The dataset contains three different types of real estate: office space, retail space, and free-use premises.

Another feature is that the data was noisy for depersonalization purposes. A little normal noise was added to the total area of the object and its coordinates, and the “realty_type” attribute was encoded.

It required a complex data preprocessing before conducting experiments and creating a machine-learning model.
The target variable is "per_square_meter_price", which takes values in the range from 800 to 1,791,000 rubles (for objects evaluated by an expert). From the distribution of the target value, it can be seen that there are outliers in the data, as well as the fact that most of the data has a target value in the range of 400,000.

Note that there is an explicit dependence of the target variable on the area of the object. The larger the area, the lower the cost per square meter, and vice versa, which is logical from a business point of view because the smaller the object, the greater the demand for it from buyers and vice versa. For example, potential buyers for a room in 10000 m² will be much smaller than an office with an area of 200 m².

It can be assumed that in the solution, one of the important signs will be signs related to the area of the object, signs characterizing the population and signs describing the location of the property.

Missing values of numerical features were replaced by the median value of this feature in the region.

Outliers were removed for q_0.99 and q_0.01 quantiles, since outliers negatively affect the training of machine learning models. Q_0.99 and q_0.01 were chosen due to the small amount of data received from experts, since less than 2% of data will be lost with this approach.

New features were created that are responsible for the number of floors, the presence of a basement, as well as features that describe the range of floors in which the building is located. Signs of the general, rural and urban population of the region in which the object is located were created.

It is known that in different cities there are streets with the same names. To train a machine-learning model, it is necessary that the “street” feature have a different value for the actual location of objects in different cities. Therefore, to avoid a collision, it is necessary to create a new feature that combines a city and a street, but in this case, more than forty thousand unique values will be obtained, which, with a small dataset with prices received from appraisers, will not carry any information for the model, but will only hinder its learning. Therefore, the "street" feature was removed.

The experiments were carried out using data that were manually assessed by experts, since this approach will significantly speed up experiments that can take hours or even days. The data were divided into training, validation and test sets. The validation and test sets each account for 10% of the total data volume.

Since not all models are resistant to different feature scales, for those models that are sensitive to feature scales, the StandardScaler class of the sklearn library was used to normalize numeric features.

3 Gradient boosting model

Boosting is a machine-learning algorithm for sequentially building a composition of predictive models. The first widely known implementation of this approach, which has been successfully used to solve various problems, is AdaBoost (AdaptiveBoosting) [4].

In AdaBoost, a composition is iteratively built from simple predictors, which in the general case can be represented by any machine-learning model. A simple predictor is a model whose performance is better than random guessing. The algorithm also uses weights for each object from the original dataset, which affect how often this object will be used for training. At each iteration, a simple predictor is trained and then feature weights are recalculated for each instance, such that larger weights are assigned to misclassified features. Also, weights are assigned to the predictors themselves, so that predictors with better quality have more weight.
This algorithm was a great success, but still had drawbacks. For example, AdaBoost was sensitive to outliers because outliers were classified incorrectly and received a large weight value. To solve this problem, BrownBoost was later created [5].

Soon a generalization of these algorithms appeared under the name Gradient Boosting Machine (GBM) [6]. The idea of the algorithm is that at each iteration, a simple predictor is trained to predict the errors of the previous approximation, namely the negative gradient. This algorithm is implemented in most modern machine learning frameworks.

To solve the problem of assessing the value of commercial real estate using gradient boosting, the LightGBM framework from Microsoft was used [7]. LightGBM is a gradient boosting framework using decision tree learning algorithms implemented in C++. This framework was chosen due to its high performance, which is achieved through many engineering improvements to the original algorithm.

The main innovations of LightGBM are Gradient-based One-Side Sampling (GOSS) and Exclusive Feature Bundling (EFB). Note that the GBM algorithm does not use weights for objects used in the AdaBoost algorithm. The GOSS technique introduces the concept of the importance of objects based on the magnitude of the gradient for a particular object, and selects for training a predetermined percentage of objects with the highest gradient value (this value is controlled by the “top_rate” hyperparameter).

The Optuna framework [8] was used to select training hyperparameters. Optuna is an automatic hyperparameter-fitting framework specifically designed for machine learning problems.

The MAE and MAPE metrics were used to evaluate the quality of the models. The MAE metric is the average absolute error, which shows how much, on average, the prediction of the model deviates from the true value in absolute value. MAE is less sensitive to outliers than classical MSE, but it is also sensitive to the scale of the error.

The MAPE metric is the average absolute error in percentages, which shows by how many percent on average the prediction of the model deviates from the true value. The main advantage of this metric is interpretability. In addition, the advantages of MAPE include measuring the error not in absolute values, but in relative ones.

After training the model on the data marked by the estimator, the following results were obtained on the validation set:

MAE=14440.0; MAPE=0.223.

For comparison, the base model was trained using a random forest and the following results were obtained on the validation set:

MAE=18738.7; MAPE=0.308.

A random forest is an ensemble of several decision trees. Algorithm for building a tree in a random forest:

- a subsample of the training set is selected; a decision tree is built on this subsample.
- when building a tree, we look at a fixed number of random features for each partition.

With this approach, each individual decision tree learns from its own sample, which is good for the ensembling of such models.

Based on the obtained results, it can be seen that LightGBM is superior to the base model, and, therefore, the gradient boosting model can be successfully applied to solve the problem of estimating the value of commercial real estate.

4 TabNet model

Although tree ensemble based algorithms outperform deep learning models on most tabular data problems, there are a large number of deep learning approaches [9] that perform better on some datasets.
Within the framework of this work, to solve the problem of estimating the value of commercial real estate, the TabNet architecture is considered [10]. TabNet is a neural network with a sequential attention mechanism that has a sequential multi-step architecture where each step contributes to a solution based on selected features. The TabNet architecture is shown in Figure 1.

Fig. 1. TabNet encoder architecture [11].

Since, unlike decision tree-based algorithms, neural networks are sensitive to the scale of features, the first layer of the TabNet architecture is data normalization using Batch Normalization [11].

To use categorical features, TabNet uses learnable embeddings. Embedding is a compact vector representation of an object, in which objects similar in some sense are located side by side in the constructed vector space. Denote by $N_{\text{emb}}$ the dimension of the embedding vector.

Denote by $f \in \mathbb{R}^{B \times D}$ the input tensor after normalization, where $B$ is the dimension of the batch, $D$ is the dimension of the feature space.

TabNet works in such a way that each step uses the results of the previous one for all $N_{\text{steps}}$. The $i$-th step uses the information from the $(i-1)$-th step, and as a result creates a feature representation that will be used in the overall aggregated solution.

The main elements of the TabNet architecture are:

- Attentive Transformer block;
- Feature Transformer block;
- Split block.

Feature Transformer is a block that consists of four layers, where the weights of the two layers are common for all processing steps, and the remaining weights are specific to each individual step. Each of the four layers is a combination of a fully connected layer, a batch normalization layer, and a GLU (Gated Linear Unit) activation function.

The output tensor from the Feature transformer is the input tensor for the Split block, which splits the tensor into two equal parts, one of which will later be used in aggregation to get the final prediction, and the other part will be the input tensor to the Attentive Transformer block.

The role of the Attentive Transformer is to select the most important features for learning at $i$-th step, so that the training capacity of the model is not wasted on irrelevant features, and thus the model becomes more efficient in terms of parameters.

To get the final prediction, the model aggregates all the answers in $N_{\text{steps}}$ and applies a linear transformation to get the answer.
To train the TabNet model to solve the problem of predicting the value of commercial real estate, the PyTorch machine learning framework and the pytorch-tabnet library were used [12].

Since neural networks are sensitive to the scale of features, including the scale of the target variable, it was decided to train the model to predict the logarithm of the target variable and exponentialize the model responses at the prediction stage.

The loss function SmoothL1Loss was chosen to train the model. A feature of this function is that it behaves like an L1 regularization function for large values of the loss function, which has a positive effect on the learning dynamics in the presence of outliers. In the case of small values, it behaves like a standard L2 function.

TabNet hyperparameters:
- $N_a = N_d = 128$;
- $N_{steps} = 2$;
- $N_{emb} = 6$;
- $\lambda_{sparse} = 10^{-4}$;
- $\gamma = 1.264$;
- $lr = 0.086$.

The dependence of the metric on the training time of the model is shown in Figure 2.

![Loss over epochs](image)

**Fig. 2.** The value of the loss function over time.

The Optuna framework [8] was used to select hyperparameters and train the model. After training the model on the data marked by the estimator, the following results were obtained on the validation set:

\[
\text{MAE}=14777.6; \quad \text{MAPE}=0.238
\]

Based on the results obtained, it can be concluded that the TabNet architecture based on neural networks is superior to the basic solution (random forest) and, therefore, can be successfully applied to solve the problem of commercial real estate valuation.

### 5 Conclusions

As part of the work done, an analysis was made of the applicability of the gradient boosting model and the TabNet architecture, and the LightGBM and TabNet models were trained. The final values of the metrics on the test set are shown in Table 1.
Table 1. The results of the models on the validation and test set.

<table>
<thead>
<tr>
<th>Model</th>
<th>MAE, val</th>
<th>MAPE, val</th>
<th>MAE, test</th>
<th>MAPE, test</th>
</tr>
</thead>
<tbody>
<tr>
<td>Random forest</td>
<td>18738.7</td>
<td>0.308</td>
<td>21220.8</td>
<td>0.404</td>
</tr>
<tr>
<td>LightGBM</td>
<td>14440.0</td>
<td>0.223</td>
<td>15173.2</td>
<td>0.333</td>
</tr>
<tr>
<td>TabNet</td>
<td>14777.6</td>
<td>0.238</td>
<td>19418.3</td>
<td>0.348</td>
</tr>
</tbody>
</table>

Based on the results obtained, we can conclude that the gradient boosting and TabNet models are applicable to solving the problem of estimating the value of commercial real estate. However, it is worth noting that with a relatively small loss of the TabNet model in terms of the MAPE metric, in terms of the MAE metric, the gradient boosting model outperforms TabNet by 4245. This fact suggests that the gradient boosting model was able to better generalize the outliers in the data, which due to a small number have little effect on the MAPE metric, but at the same time, due to the large absolute value, they make a significant contribution to the MAE metric, which is not resistant to outliers and is measured in absolute values. Therefore, the most preferred approach for solving the problem under consideration is the use of the gradient boosting model.

References

1. A. Sikochi, Journal of Corporate Finance 64 (2020)