Scalable ML - final project

Egill Friðriksson & Gard Aasness

January 7, 2024

1 How to run code

The code consists of three different main pipelines, which are all connected to GitHub actions. The GitHub actions pipeline runs the code automatically every month, as the data gets updated monthly. It triggers all pipelines to end up with the same result as if we were to run it manually. If you want to run the code manually, you have to start by running the *fetch_ data_ monthly.py* file. That will download the data from the website and store it in a CSV file in your file system. After that, you have to run *feature_ engineering_ pipeline.py*, which prepares the data and stores the cleaned CSV file locally, before uploading it to Hopsworks as a feature group. Thereafter, you have to run the *training_ pipeline.py*, which downloads the feature view (the data) from Hopsworks and uses it to train, evaluate, and tune the chosen model before uploading the model back up to Hopsworks. Lastly, you have to run *batch_ inference_ pipeline.py*, which retrieves the model from Hopsworks, and hosts the predictive model as a web user interface using Gradio. The user interface of our stand-alone serverless ML system, which is hosted at Huggingface, can be found **here**.

2 Introduction

This report studies a machine learning initiative aimed at predicting property prices in Iceland through a comprehensive dataset from the Icelandic Housing and Construction Authority. Their dataset comprises every property sold in Iceland since 2006. The dataset contains features like postal codes, area, number of rooms, and year of construction. A monthly database update script ensures real-time adaptability, as the authority updates its database on the 22nd of each month. The report delves into thorough feature engineering, emphasizing variables crucial to Icelandic property valuation. An algorithm was implemented to autonomously select the most suitable machine learning algorithm for model training, i.e. to minimize the RSME. Additionally, a user interface was implemented, using Gradio, which provides an intuitive way for users to input property details and receive instant price predictions, making complex insights accessible.

3 Feature Engineering

Initially the dataset included 24 columns, most of which were either irrelevant or had negligible effect on the price of the property. Of the columns that got dropped: seven were different types of id numbers, two were related to the size of the plot of land, one was the full address, one evaluated the validity of the contract and five were redundant. The following eight columns remained and were translated to English: **postal code**, **date** (of contract), **price**, **year** (of construction), **area**, **rooms**, **type** (apartment, semi-detached or house) and **complete** (whether it's under construction or not).

4 Filtering & Data Cleaning

After extracting the relevant columns, it was time to filter and clean the data. First of all, since the housing prices in Iceland have been continually rising since 2011, it's difficult to pick a dataset that has enough data to make accurate predictions while simultaneously representing the current market. It was decided that data from 01/01/2021 to today would suffice, even though some properties might be up to more than 20% more expensive today compared to then (see Figure 1).

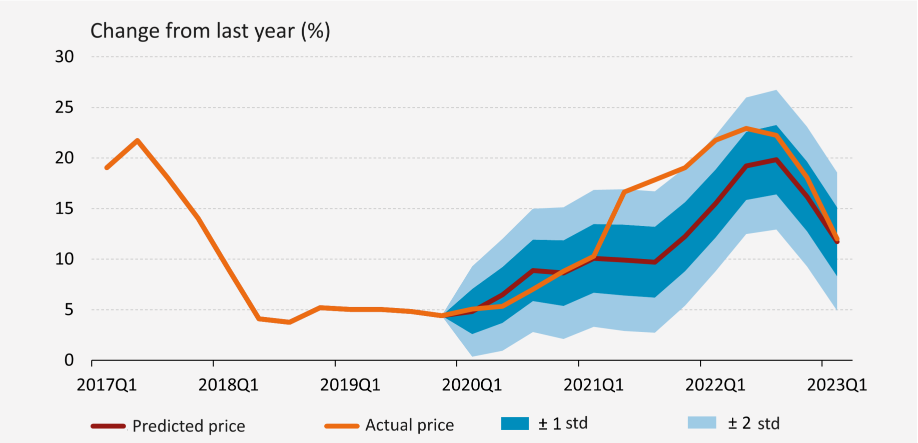
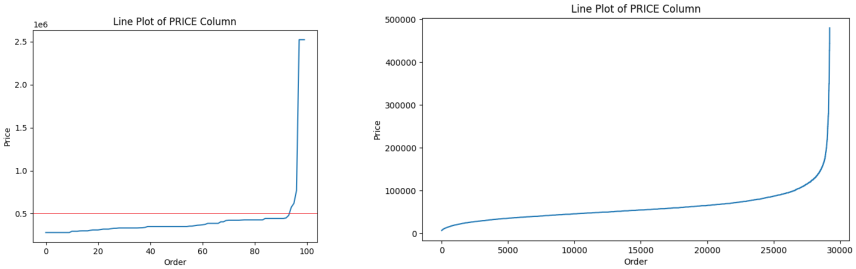


Figure 1: Change in % in price compared to the previous year

Thereafter all rows with missing values were dropped, only residential properties were kept, only finished houses were kept and then outliers were filtered out. Since the database included contractors buying whole apartment buildings, there needed to be a criteria that defined a "normal" transaction. An apartment can't have 0 rooms and the biggest private home in Iceland is 932 m^2 and has 11 rooms, so that was used as the benchmark (with some degrees of freedom). Also the selling price was restricted to being between 7 million and 500 million, as the cheapest current property in Iceland goes for 9.9 million and the largest private sale tallied up to 620 million. It was decided to lower the highest price as the data included a lot of contractor contracts above 500 million that severely skewed the data. Figure 2 shows the distribution of selling prices in the unfiltered and filtered dataset with the red line showing the 500 million mark.



(a) Top 100 before filtering

(b) Distribution of all properties after filtering

Figure 2: Overall Caption

5 Model selection and hypertuning

After performing feature engineering and cleaning the data, we uploaded the data to our feature store, Hopsworks, as a feature group. The data was not ready to be used to train a model.

Next, we implemented the training pipeline. The training pipeline consists of two main tasks: finding the best machine-learning algorithm for our specific problem and then tuning that model to find the optimal hyperparameters. Therefore, we started by testing the following baseline machine learning algorithms using the default hyperparameters: Linear regression, Decision trees, Random forest, SVR, K-nearest neighbors, Elastic Net, Lasso regression, and Ridge regression. We trained all the algorithms using 5-fold cross-validation and evaluated them using the root mean squared error (RMSE) as a metric. The three algorithms with the best scores were Random forest, Decision trees, and K-nearest neighbors. Therefore, we decided to hypertune all three algorithms using grid search, to find the optimal hyperparameters for all algorithms and proceed with the algorithm with the best RMSE score after being hypertuned. The best-performing model was the Random Forest, so we uploaded the model to Hopsworks, along with info about the RMSE score, best hyperparameters, and training data.

6 Hosting with Gradio

Once the model had been chosen and tuned, an intuitive user interface was created for the user to enter values for the different features. The script for the interface fetches the trained model from Hopsworks to predict the price. The interface can be found **here**

7 Results and Discussion

The final RMSE score of our model after tuning it was around 15000000 (15M ISK), and with the current conversion being 1 SEK=13,50 ISK, it results in a bit more than 1M. However, we noticed that the performance of the model depended on some factors, as it in some cases gave a very accurate price estimation and in other cases was a bit further off. Our hypothesis for what caused that is the inflation that has severely affected the housing prices in Iceland. As seen in Figure 1, housing prices have risen by roughly 20-30% since the beginning of 2021. That means that the model is not accurate for properties most affected by the inflation (namely in the capital region). However, since Iceland is a small country, it was not possible to omit the 2021 and 2022 data since the dataset would be too small to train and give accurate results, especially for rural areas. A solution might be to either artificially inflate the prices of properties in the older data by the corresponding housing index, or give more significance to the more recent data. Despite that we are pleased with the results of our final project, as we consider the prediction to be highly accurate considering the comprehensiveness of the source data.