Predicting Estonian apartment rent prices based on KV.ee listings

Sebastian Mais, Caroline Markov, Rainer Talvik

Project repository: https://github.com/S33bu/DS_RENT

Business understanding

Business goals

The real estate market in Estonia, particularly in cities like Tallinn and Tartu, has seen significant growth and fluctuation in recent years. With the increasing demand for rental properties, both landlords and tenants face challenges in determining fair rental prices. Accurate rental price predictions can help landlords set competitive prices and assist tenants in finding affordable housing options.

The project aims to analyse long-term apartment rental listings from the Estonian real estate sale and rental site KV.ee [1]. We will gather information on various features of rental properties to develop a model that can accurately predict rental prices based on these features, providing valuable insight to interested parties in the real estate market.

Our goal is to create a model which could achieve an RMSE (Root Mean Squared Error) of 50, indicating that the square root of the average squared differences between our model's predictions and the actual rental prices is 50€.

Situation assessment

Our primary data source is the Estonian real estate site KV.ee, from where we will scrape the data needed.

We will use the Web Scraper [2] browser extension to gather data from KV.ee, Python for data analysis and different libraries such as NumPy, Pandas and Scikit-learn for data processing and modelling.

We will use large language models (LLMs) to analyze unstructured descriptions and retrieve information about factors such as available amenities, apartment condition and proximity to essential services.

Requirements, assumptions, and constraints:

- Accurate and comprehensive data from KV.ee.
- The data from KV.ee is representative of the rental market in Tallinn and Tartu, and the data quality is sufficient for analysis.
- Limited time for project completion, data collection and processing may need manual reviewing (e.g some info is described only in the listing description).

Terminology:

- Web Scraper – A browser extension used to scrape data from websites.
- KV.ee – An Estonian real estate sale and rental site.
- NumPy – A Python library used for numerical computations.
- Pandas – A Python library used for data manipulation and analysis.
- Scikit-learn – A Python library used for machine learning.
- RMSE – Root Mean Squared Error, a metric used to measure the accuracy of a predictive model. It represents the square root of the average squared differences between the predicted and actual values.
- Data cleaning – The process of detecting and correcting (or removing) corrupt or inaccurate records from a dataset.
- Outliers – Data points that differ significantly from other observations.

Costs and benefits:

- Time and effort required for data collection, analysis and modeling.
- Accurate rental price predictions, valuable insights.

Data-mining goals

Our data-mining goals are to develop a predictive regression model to estimate rental prices based on various features such as location, area, number of rooms, rent, etc. These features will be compiled into a separate dataset for analysis. Additionally, we aim to identify significant factors that influence rental prices in Tallinn and Tartu.

Data-mining success criteria includes achieving a RMSE of 50€, ensuring that a majority of price predictions are within ±15% of the actual rental listing prices.

Data understanding

Gathering data

All data will be gathered from the Estonian real estate sale and rental site KV.ee. Our model focuses on long-term apartment rent listings, which are created by users of the site.

While there is no public-facing database for apartment rent listings in Estonia, the KV.ee website follows a rather rigid structure, which makes it viable for scraping. We will use the Web Scraper browser extension because it is easy to use, it has ample documentation and it emulates an user browsing the website in a regular browser window.

Because the listings are created by users and not all fields are required, the consistency of information varies greatly from one post to another. However, the following fields are almost always populated:

- address (text);
- rental price (float, €);

- area (float, m^2);
- price per unit of area (float, €/m^2);
- owner or broker (text);
- number of rooms (integer);
- floor number (integer);
- total number of floors in the building (integer);
- text summary, which typically includes information about the kitchen, sanitary arrangements, heating, ventilation and security (text);
- longer, non-structured description (text).

Almost all listings have images attached, which creates an opportunity for additional features via image processing methods.

We will set Web Scraper to fetch information about as many fields from the website as possible (incl. table on the right side of a listing: condition, utilities, energy mark etc.), which will likely lead to many missing values.

Describing data

The final scraped data can be described with the following table (names adjusted and a few columns, such as listing and image links, omitted):

Feature name	**Description**	**Example data format**
Address	Full apartment address	COU, MUNI?[(1)], CITY, DIST.?, ST
Price	Apartment rental price and price per m^2	X € Y €/m^2 (floats)
Owner_or_broker	Listing owner, either contains "Owner" or the broker firm	"Owner" OR (Broker firm)
Rooms	Number of rooms in the apartment	X (integer)
Bedrooms	Number of bedrooms in the apartment	X (integer)
Area	Total rentable area	X m^2 (float)
Floor_out_of_floors	Apartment floor & total number of floors	X / Y (integers)
Build_year	Year when apartment was built	X (integer)
Condition	General condition	(Ordinal text)
Energy_mark	An energy rating for the apartment	(Ordinal text)
Utility_costs	Utility costs, often divided into summer/winter	X € / Y € (floats)
Images_link_text	Text from the "All images" link on the listing, contains the number of images attached	"All images (X)" (integer)

Prepayment	Prepayment for apartment rental	X € (float)
Summary	A compact overview, includes information about the kitchen, sanitary utilities, heating etc.	(Loosely formatted text)
Description	Long description of the rentable apartment	(Text)
Description_header	First part of the description, often a sales pitch	(Text)
Description_footer	Last part of the description, includes whether brokers are allowed to contact	(Strictly formatted text)

(1) A question mark (?) indicates that value can exist or not

Exploring data

The site was scraped on November 16th 2024.

The dataset contains 2676 listings, with approximately 1470 from Tallinn and 500 from Tartu, making it well-suited for modelling rental prices in these cities. Smaller cities, with fewer listings, might need to be excluded or treated separately, as they could impact the accuracy of predictions for Tallinn and Tartu, where prices are higher and more varied.

General trends are as expected – newer apartments closer to city centers, with more rooms and higher energy ratings, have higher rental prices. Address information needs to be cleaned and segmented. Price data, which initially includes both total rent and price per square meter, must be split into separate columns. Features like floor_out_of_floors require large corrections as scraped data was misinterpreted as dates. The number of images in each listing must be extracted to retrieve a feature that might correlate with the listing's price.

Variables such as condition and energy_mark need transformation into numeric or ordinal formats to reveal correlations with price. Data quality issues, such as missing or unrealistic values in critical fields like price, area and build_year, need resolution to ensure data reliability.

Meaningful insights can be extracted from text fields by focusing on key attributes. For example, if the description mentions that pets are not allowed, this can be converted into a Boolean value (pets_allowed = False). Similarly, information about amenities like access to a gym or a shared washing machine, as well as whether the apartment comes furnished, can be extracted. Collecting such details in a structured manner creates more features for analysis. Alternatively, we could use LLMs to analyze the description and assign a “comfort score” based on key factors such as available amenities, apartment condition, and proximity to essential services.

Verifying data quality

The data seems good enough. While it requires a lot of processing, most fields follow a strict structure and can therefore be parsed with relative ease. Some fields, especially text fields, require the use of advanced tools, such as LLMs, to be viable to process at scale.

Planning your project

Task 1: Data collection (1 hour)

- Description: Use the Web Scraper extension to collect rental data from KV.ee. Ensure data includes key fields like address, price, area, number of rooms, and descriptions.
- Team member contribution:
    - Rainer Talvik 1h (set up and execute scraping)
- Tools and methods: Web Scraper extension, manual review for data consistency.

Task 2: Data cleaning and preprocessing (45h)

- Description:
    - Transform numerical columns (e.g. price, area) into proper formats and remove outliers;
    - Correct floor_out_of_floors data that has misinterpreted as dates;
    - Extract key attributes form text columns (e.g pets, allowed, amenities) into Boolean or categorical columns;
    - Use LLMs to analyse descriptions;
    - Handle missing values in critical fields like rooms, price, area.
- Team member contribution:
    - Everyone ~15h
- Tools and methods:
    - Pandas for data manipulation, NumPy for handling missing values and outliers, Python and LLMs for text processing.

Task 3: Exploratory Data Analysis (21h)

- Description:
    - Look at the data to see how key features like price, area, rooms are spread out;
    - Create graphs to spot patterns and trends in data;
    - Check if hypotheses like “better energy ratings lead to higher price” etc. are true.
- Team member contribution:
    - Everyone 7h
- Tools and methods:
    - Matplotlib and Seaborn to make graphs, Pandas to summarize the data in tables.

Task 4: Model Development (21h)

- Description:
  - Build regression models to predict rental prices based on selected features;
  - Experiment with different algorithms (e.g. linear regression, decision trees);
  - Optimize hyperparameters;
  - Check how good the models are by:
    - Calculating how far predictions are form the actual prices on average (RMSE);
    - Making sure that predictions are within ±15% of the actual prices.
- Team member contribution:
  - Everyone 7h
- Tools and methods:
  - Mainly Scikit-learn.

Task 5: Results Interpretation and reporting (15h)

- Description:
  - Summarize findings, including significant factors influencing rental prices;
  - Prepare a comprehensive report, including visualizations;
  - Highlight any limitations or areas for future work.
- Team member contribution:
  - Everyone 5h

Resources

[1] KV.ee (2024). https://www.kv.ee/

[2] Web Scraper (2024). https://webscraper.io/