Gentrification Exploration in Zurich

Apolinarska Aleksandra Yanick Bachmann Xiyi Chen Florian Schweingruber

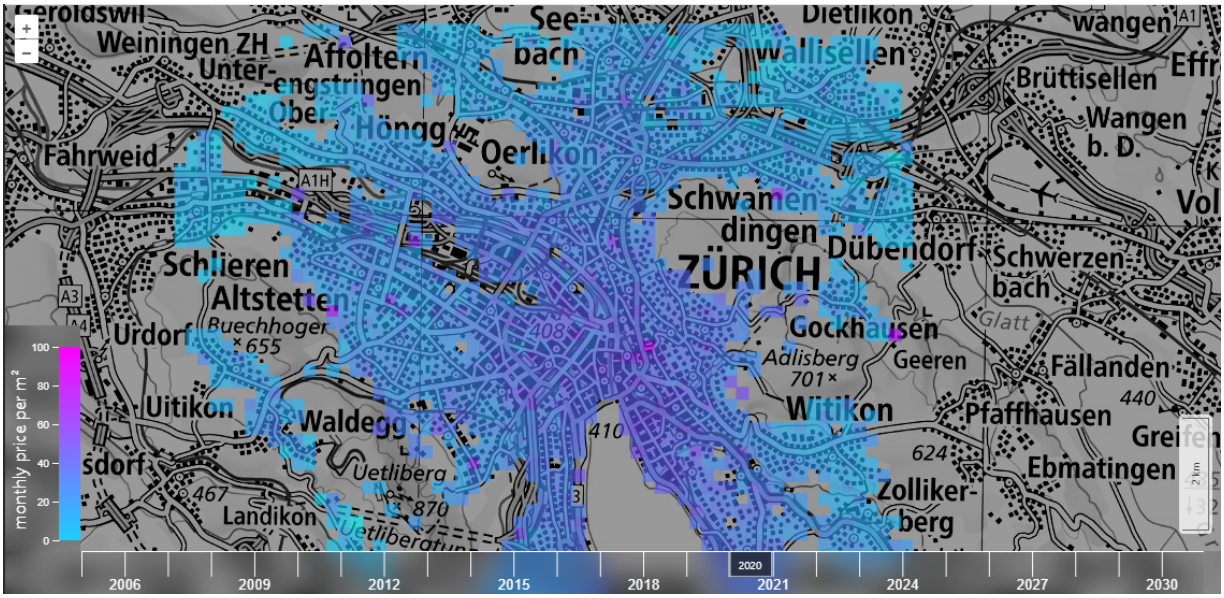

Figure 1: Animated and interactive color Map showing predicted house rent prices in Zurich over time.

Abstract

Our project seeks to uncover and visualize pricing dynamics and gentrification developments in Zurich's housing rental market, along with predicting future trends, emphasizing uncertainty. We developed a machine learning model that generalizes past price values to forecast future trends. Users can interact with the tool to explore rental price developments and potential gentrification through an animated color map overlay on Zurich's map on a single page web interface. Precomputed data patches and the use of confidence intervals enhance both performance and transparency. Our approach to explainability includes overlaying predictions with original data and providing visual explanations. This project offers a comprehensive resource aimed at lay users. It strives to make complex data accessible and actionable, highlighting the impacts of urban development. This project was conducted as a semester project for the course "Explainable and Interactive Artificial Intelligence" at ETH Zurich.

Index Terms: Gentrification, Rental Price Prediction, Model used

1 Introduction

Our approach is rooted in the application of the Variational Nearest Neighbour Gaussian Process (VNNGP) model, which allows us to capture spatiotemporal relationships and provide reliable predictions along with confidence intervals. This model is particularly suited for our raw dataset, which comprises approximately 1.2 million property records collected from various online advertisement websites between 2005 and 2024. To facilitate user interaction and exploration, we developed a web application using Svelte for the front-end, OpenLayers for the interactive map, and D3.js for data visualizations. The map utilizes resources from OpenStreetMap, providing a familiar and detailed geographical base. A color map overlay and timeline for a chosen point enable an inuitive entry point for lay users to a large and high dimensional data set. Further, several explainability features were implemented to enhance transparency and interpretability of our model.

1.1 Gentrification

Our initial inspiration was the phenomenon of gentrification and the idea to detect which neighbourhoods in Zurich are affected by it. Gentrification broadly refers to a phenomenton in urban development where poor neighbourhoods are gradually transformed under influence of wealthier people moving in, attracting new businesses and changing the character of the neighbourhood. One of the most visible effects of gentrification is the rise in housing prices, which may push previous residents out of the neighbourhood. There is no consensus on the exact definition of gentrification, and the process is often difficult to spot. Nevertheless, following indicators are often used to detect an ongoing gentrification process: sharp increase of rental prices, sharp increase of incomes in low-price neighbourhoods, densification including demolishing old housing and replacing it by new, exchange of population, i.e. young replacing old residents.

In our project, due to limited time and data, we focus primarily on development of rental prices as a proxy for gentrification analysis and visualize them as an overlay over the map of Zurich.

1.2 Related Work

We found two notable examples of similar work as commercial services: Novelletta Maps [9] and Urban Data Labs [12], which target property owners or investors as main users/customers and provide interactive maps for filtering, prediction and analysis of rental prices in Zurich and/or Switzerland, which however do not disclose the underlying methods in much detail. By contrast, [7] provides a

comparison of different methods for price modelling (i.a. Random Forests, Bayesian Regression, Linear Regression) exemplified for the Swiss property market. Further, [4] demonstrated applicability of the Stochastic Variational Gaussian Process for spatiotemporal predictions of rental prices in England and Wales. Overall, the history of statistical methods for modelling of rent prices goes back to 1970s, and includes methods based on hedonic regression and kriging (Gaussian Process Regression).

1.3 Project goals

1.3.1 Users

The website is intended primarily for lay users from the general public, such as residents of the city of Zurich. We assume the users are generally familiar with geography of the city and roughly know the city districts. They also have some idea about the housing situation and rent prices, but are interested in understanding the development of the housing market over the past two decades, compare different districts and are curious about future trends.

1.3.2 Purpose

Our tool allows users to explore rental price development and uncover gentrification processes in the city of Zurich through an interactive map showing an animated visualization of rental prices as an overlayed color map, allowing users to query the locations of interest on the map and to observe the price changes across multiple years, including predictions into the future.

1.3.3 Model

Our machine learning backend captures spatiotemporal relationships and is able to predict rental prices at different locations and time points, including in the future.

1.3.4 Explainability

We want to expose to the user the how our model makes the predictions, by modelling uncertainty and overlaying predictions with the distribution of original data.

2 Data

2.1 Raw data

The raw dataset consists of ca. 1'200'000 records of property ads crawled from the internet, from different online ad websites, between 2005 and 2024, and has been kindly shared with us by Vahid Moosavi from the Noveletta DataHub [8], a Zurich-based startup. The features captured in the dataset include: ad title, date and description, geo-coordinates of the property (latitude and longitude), zip code, postal address, price and price type (for sale or for rent, price per day, month or year), property type (various types of houses and appartments, but also workshop studios, holiday homes, car parking spots, single rooms etc.), floor area, number of rooms, floor number, year of construction and some others.

2.2 Preprocessing

For the project purposes, we have only kept records of houses and apartments for long-term rent. We also de-duplicated the data to remove identical ads coming from different ad portals. Further, we removed invalid records, i.e. those where price, floor area or geo-coordinates were missing or invalid, and also removed outliers and/or restricted the value ranges to avoid long tails in the distributions (e.g.: price/month/m^2 $\in [5, 100]$ CHF). For the project's purposes, we recalculated all prices as price/month and price/month/m^2.

2.3 Final dataset

After these measures, our dataset entailed ca. 325'000 records. Around half of the records had a missing value for the year of construction, and most of the records were apartments (as opposed to houses). Hence, in the end we only used the features: location (continuous), date, price/month/m^2 (continuous), number of rooms (discrete), year of construction (categorical) and floor area (continuous).

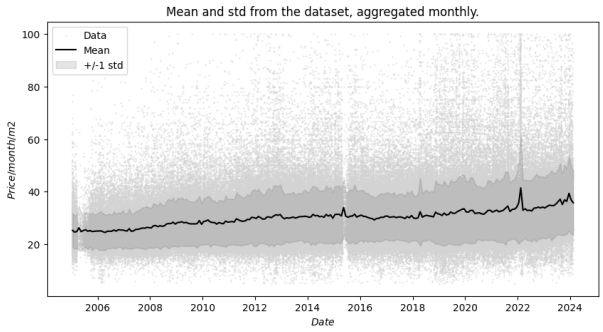

Figure 2: All data points used in the project: rent prices per month per m^2 according to the ad date, and their mean and standard deviation (aggregated monthly).

3 ML Model

The goal of the underlyting machine learning model in our project is to predict rental prices (price/month/m^2) over time for given locations and property features. This problem setting implies capturing spatio-temporal relationships; additionally, we wanted to visualize model's uncertainty and use it for what-if scenarios. We have looked in different neural network architectures (mainly RNNs and Gaussian Process methods) that could be applied to our problem.

3.1 LSTM

With the aim predict future rental prices, we considered autoregressive models like LSTM [5]. However, they require consistent data across the time axis. By nature, our dataset does not have such consistency, neither in spatial (locations) nor temporal dimension, i.e. we don't have a fix number of records for a given location for every timestep. As a workaround, we have tried aggregating the data per area (using webmercator zoom level tiles [10])) and timesteps (e.g. yearly, quarterly or monthly, including sliding-window aggregation), at the cost of losing granularity due to averaging of the features. Also, we could only aggregate at low resolutions (e.g. zoom level 16 and a montly sliding window of 6 months) to obtain tiles with values for every timestep. With this, we could train an LSTM model mapping from tile and timestep indices and property features to a mean and standard deviation of rental prices.

3.2 VNNGP

Our other approach focussed more on modelling uncertainty. Due to the size of our dataset, we could not use a standard Gaussian Process Regressor model which requires computation of a covariance matrix of all datapoints. Variational Nearest Neighbour Gausshian Process (VNNGP) [14] can alleviate this problem by using only a small number of nearest observations to calculate the conditionals (at the cost of finding nearest neighbours for all data points). This makes it also naturally suitable for modelling spatiotemporal relationships. The VNNGP model was able to make reasonable predictions into the future.

3.3 Implementation

LSTM showed slightly worse predictions ($R^2 = 0.53$) than VNNGP ($R^2 = 0.62$), and it required to aggregate (average) data to construct the time series, which would have obfuscated the data for explainability. Hence we opted for VNNGP, which also has built in uncertainty (confidence intervals) on top.

The VNNGP model is trained to predict price/month/m^2 from latitude, longitude, date, floor area and number of rooms features. To train it, we used GPyTorch library [1] and adapted the code snippet provided in the library [3]. The trained model is used then to both create the colored map and the line plot (see Section 4.1).

To create the colored map, we only want to display areas that contained data. We pick 20k random records from the dataset to get representative locations, and ask the model to predict prices for these locations at different timesteps. For the inference, however, we need to provide also the remaining features (rooms and areas) as input. Initially, we wanted to predict these using an additional, very simple model. However, as we did not recognize any temporal trends in these features, we decided to simply use the average of the respective tile over all years. The procedure to create the line plot is similar but the location is given by the point on the map.

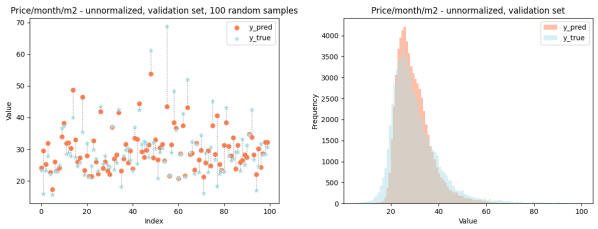

Figure 3: Predictions of the VNNGP model vs. ground truth (validation set). Left: 100 random samples from the dataset. Right: distribution of predicted and true values.

4 User Interface

4.1 Tasks, Navigation, Interaction

Our interface is a single interactive webpage. The initial state, when the user first arrives on the site, shows a map of Zurich and an overlayed color map of rental prices today (i.e. year = 2024). We follow the principle of "overview first, details on demand". The user can zoom and pan the map, and select a location through a mouse-click on the map. When hovering over a point in the highlighted tile, a tooltip shows the respective price, room number and area. When a point in the highlighted tile is selected, a timeline plot is revealed at the bottom edge of the window showing detailed price evolution for that location. The graph shows a line plot of the mean price and confidence interval predicted by the model, as well as scatter plot of the original data points corresponding to the selected area. By this we show multidimensional data in different "views". The user can also select a point on the time axis to set the overall color map to a chosen timestep.

4.2 Visual concerns

Color is our main visual channel, which we use to encode the rental price values in the line plot and the color map. We decided to use a sequential color map ("cool" from MatPlotLib colormaps collection [6]). The underlying map of Zurich is toned down to dark grey scale to improve legibility.

We use the same color scheme for the color map overlay, for the line plot and for the scatter plot of the data points, to visually connect the price values across views and marker types. To make panels stand out from the map in the background while retaining a modern and appealing design, it was decided to place them directly on the map with a Gaussian blur to offset them.

To effectively visualize the data, the following marks and channels are used:

- **Marks:**
  - Data points on Map: Individual Property records are represented as points
  - Data Points in Line Graph: Rental Prices over Time are displayed as points
  - Tiles: Aggregated data for specific geographic areas are encoded as tiles on the map, representing a trend in that area
- **Channels:**
  - Data Points: The price attribute is encoded via the color grading of the point, on the map and also on the line graph
  - Tiles: Color Grading is used to encode the scale of rental prices within each tile, providing a visual representation of price variations

4.3 Explainability

Explainability is a crucial part of this project, aimed at ensuring that users put trust and understand the predictions made by our model. We implemented the following measures to establish transparency and interpretability of our model:

- **Uncertainty Modelling:** Our primary approach to explainability involves modeling the uncertainty of predictions. By using the Variational Nearest Neighbour Gaussian Process (VNNGP), we provide not only point estimates of rental prices but also confidence intervals. This allows users to see the range of possible outcomes and understand the reliability of the predictions. The confidence intervals are visualized on the timeline plot as shaded regions around the predicted mean values, giving users a clear indication of the prediction uncertainty at different time points.
- **Overlaying Predictions with Original Data:** To help users compare the models with actual data, the users can see data points from the original data overlaid on the predicted tiles upon clicking the map. Additionally, the scatter plot of original data points alongside the timeline plot for a chosen point provides a direct comparison.
- **Visual Explanations:** The color-coded map to visualize rental price distribution across Zurich makes it intuitive for users to identify trends and anomalies. Further, the use of interactive features such as tooltops and clickable map points provide additional context, including the underlying features used by the model, such as room numbers and area.
- **Feature Importance:** Although our final model focuses on a limited set of features (location, date, price/month/m², number of rooms, and floor area), we ensure that users understand the influence of these features on the predictions. Future enhancements could be made by not only highlighting the features but also providing an importance ranking to explain in more detail what drove the prediction.

5 Implementation

5.1 Frontend

The front-end was developed using Svelte, chosen for its simplicity and performance. For implementing the interactive map and color map, we used the OpenLayers library due to its versatility and optimization for handling large amounts of data. Swiss Geoportal [11] was used for map resources, providing a reliable and detailed geographical base. For graph visualizations, we employed D3.js, a library for creating dynamic and interactive data visualizations. Our map uses the conventional web mercator projection [13]; we use `mercantile` [2] library to convert between latitude-longitude and the 2D point on the screen, and the x, y, z coordinates for the map graphics (where x, y are tile indices and z is the zoom level.

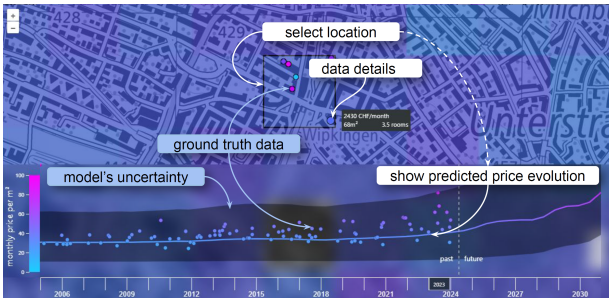

Figure 4: Details on demand: 1. user selects a point/area on the map, 2. price evolution plot appears at the bottom, 3. data samples can be queried for details

5.2 Backend

Our backend is designed with three key endpoints that facilitate querying the database and returning data in structured formats to enhance our web application's functionality.

5.2.1 avgm2price_grid

This endpoint accepts a single parameter, year, and queries a pre-computed dataframe generated using the VNnGP model for the specified year. Here's an overview of the process:

• **Pre-computation:** We pre-compute predictions for the years 2005 to 2030. For each year, we take a balanced sample from our database, ensuring each tile at zoom level 17 is covered. This results in approximately 20,000 data points per year, with timestamps randomized within the year.

• **Model Execution:** The VNNGP model is run to generate predicted prices. These predictions, along with the features, are stored as dataframes.

• **Data Storage:** The data is precomputed once and stored as JSON files on the server.

• **Recursive Helper Function:** The endpoint utilizes a recursive helper function to combine tiles from finer granularity. The smallest tile is at zoom level 17[1], and larger tiles are recursively combined from this base.

• **Response Format:** The backend returns prices in a grid format (an $n \times n$ list), which the frontend converts into a color-coded map, always maintaining a 64×64 grid at the lowest zoom level.

5.2.2 get_sample

This endpoint is designed for obtaining sample data for visualization when a user clicks on the map. It accepts the following parameters: start_date, end_date, n_samples, x, y, and z, where the z coordinate represents the zoom level. The endpoint returns a sample of data points within the specified date range and geographic area, facilitating detailed visual analysis on the frontend.

5.2.3 timeseries_at_point

This endpoint takes arguments lon, lat, start_date, end_date and n_samples. These are used to calculate the hidden features (nrooms, area, age category) based on existing datapoints in the tile (average for nrooms and round up to next half, average for area, and mode for the age category). It then queries the model using these hidden features and the interpolated n timestamps, to get mean as well as upper and lower confidence values from the VNNGP. Using that, the frontend builds the timeseries plot in the bottom for the selected tile.

[1] At zoom level 17, a tile corresponds to an area of ca. 227×227 m.

These endpoints collectively enhance the functionality of our web application, enabling efficient data retrieval and visualization for end-users. The pre-computed and recursive nature of the data processing ensures fast and accurate responses, significantly improving user experience.

5.3 Optimization for Low Latency

To ensure a smooth user experience with minimal latency, we optimized the front-end and back-end interaction by precomputing and caching data. Here are the specifics:

• **Pre-computation and Storage:**

– **Zoom Level 17 tiles:** We precompute data tiles at zoom level 17 in the backend upon initialisation and store them there. These tiles were aggregated based on predictions from the model for each square area.

– **Data Storage:** The precomputed data is stored as JSON files on the server, ensuring that the client can quickly access necessary data without requiring on-the-fly computations.

• **Client-side Operations:**

– **Rendering:** The client-side is responsible for rendering the interactive map, color map, and graphs. This includes displaying precomputed tiles and ensuring smooth panning and zooming functionalities.

– **Dynamic Interactions:** When users interact with the map (e.g., selecting a location or time period), the client-side dynamically fetches and displays the relevant precomputed data.

– **Visualization:** D3.js is used on the client side to generate and update visualizations based on user interactions.

• **Server-side Operations:**

– **Data Processing:** The server handles intensive data processing tasks, such as model predictions and aggregations. This processing is done periodically to update the precomputed tiles.

– **API Endpoints:** The server provides API endpoints that the client can query to fetch the necessary data for rendering the map and visualizations. These endpoints include querying for specific tiles, time series data, and sample data points.

– **Random Selection of Points:** For areas shown on the map, the server performs a random selection of data points to ensure a representative sample is provided for visualizations.

By dividing the workload efficiently between the client and server, we ensure that the user experience is responsive and smooth, even when handling large datasets and complex visualizations. This architecture also allows for scalability, as additional data can be precomputed and stored on the server without affecting the performance of the client-side application.

6 Conclusion

In conclusion, our project successfully developed a comprehensive tool to analyze and predict rental price trends and gentrification developments in Zurich. By leveraging a robust machine learning model, we provided insights into the city's housing market, addressing the complexities of spatiotemporal relationships and uncertainties in predictions. Our interactive web application allows lay users to dynamically explore these trends, offering a valuable resource for anyone interested in the impacts of urban development and gentrification. The integration of advanced modeling techniques with user-friendly visualization marks a significant step in making complex data accessible and actionable for a broader audience.

References

[1] J. R. Gardner, G. Pleiss, D. Bindel, K. Q. Weinberger, and A. G. Wilson. GPyTorch: Blackbox Matrix-Matrix Gaussian Process Inference with GPU Acceleration, 2021. 3

[2] S. Gillies. Mercantile. Web mercator XYZ tile utilities. https://mercantile.readthedocs.io/en/stable/api/mercantile.html, 2024. 3

[3] GPyTorch. VNNGP: Variational Nearest Neighbor Gaussian Procceses. https://docs.gpytorch.ai/en/latest/examples/04_Variational_and_Approximate_GPs/VNNGP.html. 3

[4] J. Hensman, N. Fusi, and N. D. Lawrence. Gaussian processes for big data, 2013. 2

[5] S. Hochreiter and J. Schmidhuber. Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780, 11 1997. doi: 10.1162/neco.1997.9.8.1735 2

[6] Matplotlib. Choosing Colormaps in Matplotlib: Sequential2. https://matplotlib.org/stable/users/explain/colors/colormaps.html#sequential2, 2024. 3

[7] V. Moosavi. Urban data streams and machine learning: A case of swiss real estate market, 2017. 1

[8] V. Moosavi. Noveletta DataHub. https://www.linkedin.com/company/noveletta/posts/?feedView=all, 2024. 2

[9] Noveletta Data Hub. Multi-criteria search engine for real estate developers. https://www.maps.noveletta.com/, 2024. 1

[10] OpenStreetMap developers. OpenStreetMap: Zoom levels. https://wiki.openstreetmap.org/wiki/Zoom_levels, 2024. 2

[11] Swiss Geoportal. Maps of Switzerland. https://map.geo.admin.ch/, 2024. 3

[12] UrbanDataLab AG. Rent price predictions in Switzerland. https://www.linkedin.com/pulse/udl-rent-price-pred-urbandatalab/, 2022. 1

[13] Wikipedia. Web Mercator projection. https://en.wikipedia.org/wiki/Web_Mercator_projection, 2024. 3

[14] L. Wu, G. Pleiss, and J. Cunningham. Variational nearest neighbor gaussian process, 2022. doi: 10.48550/arXiv.2202.01694 2