Swedish Housing Prediction - 2024

Step 1 – Data Understanding & Feature Engineering

Step one of any machine learning task is understanding what we wish to predict and what data we have at hand or may need. In this case, we have a straightforward regression problem, which involves predicting the sales price using some historical data in labeled form.

So, what kind of data do we have exactly? The schema includes three tables:

- **Apartment**
- **HousingAssociation**
- **AnnualReport**

Each apartment may belong to one housing association, and each housing association can have zero or more annual reports. The sell_price column in the apartment table is our target – what we wish to predict. This is not to be confused with asking_price, which represents how much the seller has listed the house for.

What metric should be used to evaluate the results in the end? Would it be beneficial to examine more than one metric? It is up to you to evaluate the model's performance and robustness soundly.

Most columns in the dataset should be self-explanatory. Some columns have high predictive power, and others do not.

Some categorical variables are relatively easy to handle, but categorical data with high cardinality (many different values) may require another approach.

Some categories have missing data in a few cases, and some columns are sparsely populated. New features could be built by combining or modifying existing columns intelligently.

The exact difference between locality, brokers_description, and legal_district, which all seem to refer to areas within the municipality, remains a mystery even to the creators of this assignment.

It is recommended to explore the dataset in MySQL to get a good feel for what you are dealing with. Plotting how columns relate to the target is an excellent way to understand each feature's predictive aspects.

Getting the data into a format that can be plugged into a machine learning algorithm is usually the lion's share of the work. Some columns can be used right out of the box, while others may require a certain amount of scrubbing.

As the dataset is mid-sized, you could benefit from downloading it once and storing it locally as a pickle or parquet file (similar to CSV/JSON).

Step 2 – Model Design and Tuning

You are unlikely to wish to implement a machine-learning algorithm from scratch. Instead, stand on the shoulders of giants and pull an already battle-tested library from the internet.

Heaps of different algorithms have already been invented, but knowing which one to choose can be challenging. This is especially true since every dataset is unique. Although some fancier new algorithms generally boast good performance, no guarantees are made that what we are dealing with won't be better solved by some entirely different method. As such, it would be wise to try a handful of different algorithms to see how they compare.

Establishing a baseline is generally a good idea. This is usually a simple or naïve model to which future optimization may be compared. It doesn't have to be a model if you can establish a sound yardstick for comparing performance, but a simple model is usually an easy way to approach this. Depending on the problem, KNN, Linear Regression, Logistic Regression, or similar algorithms could be good candidates for such a role.

Except for the baseline, compare your primary model with at least one other "champion challenger" model, which could have been your second choice.

There are many strategies to choose from, models to try, and hyperparameters to test. Not all sections of the data may be relevant for all models, and sometimes, multiple smaller models may be superior to one model that tries to rule them all. But you have limited amounts of time, so you and your team will need to prioritize which rabbit holes are worthwhile to dive into.

Step 3 – Validation Documentation

As part of the company's governance structure, models must be documented and externally validated. This ensures that the risks brought on by statistical modeling are fully understood and within the risk appetite set out by the company's strategy.

This may be challenging as many advanced AI models are opaque black boxes, and explainable/interpretable AI is a significant research field. The FSA (Financial Services Authority) performs regular checks and punishes compliance failures with hefty fines. These fines are handed out to institutions regularly, so the threat is not merely theoretical.

There is no need to include descriptions of concepts readily available on Wikipedia, but model documentation typically requires additional experiments to back up claims with numbers, plots, or figures. You need to prove that the engineering is sound, that the company doesn't take on hidden risks, and that performance is well understood. Below is a checklist of considerations:

Step 3B – Checklist of Considerations

- **Feature Engineering:**
  - What features are used and not, and why?
  - For features that aren't straightforward numbers, how did they fit into the machine learning model?
  - How do you handle missing values?
  - How do you handle sparse data?
  - How did you handle annual reports, which map one to many?
  - Have you performed additional feature engineering that needs to be mentioned?
- **Model Selection:**
  - What machine learning algorithm did you settle on and why?
  - What baseline model did you use for comparison and why?
  - What champion challenger did you select?
- **Model Evaluation and Testing:**
  - What metric(s) did you decide to use for evaluation and why?
  - How well is the model performing according to your estimates?
  - How do you avoid overfitting or underfitting?
  - What data are you using to test the model?
  - Did you use all historical data to train? Are older data points as valid as newer ones?
  - Have you made any other assumptions that need to be discussed?
- **Tuning:**

  - Which hyperparameters did you tune? Why did you include/exclude specific parameters?
  - Can you ensure you have not introduced additional overfitting while tuning?
  - How much performance improvement did you achieve by tuning the model?
- **Robustness:**
  - Does the model fail or behave unexpectedly with extreme values caused by errors or statistical artifacts?
  - If the database is populated by an upstream service, how would the model's predictive ability be affected if data in some columns is lost? For example, what if some column(s) only contain 0/null values instead of the original values?
- **Evaluation:**
  - How well are your models performing?
  - Is using a complex model justified compared to using a simpler model?
  - What predictive power did each feature contribute?
  - Did some features perform better or worse than expected?
  - Is the model likely to drift (degrade) over time? If so, what can be done about it?
  - Does the model perform equally well across all real estate, or does it fail in specific areas, price categories, or other aspects the company should worry about?
  - Are there any other concerns with the data or model that should be flagged and brought to the attention of validation or senior management?