Stockholm School of Economics
Center for Data Analytics

Joel Smedberg
joel.smedberg@hhs.se

Swedish Apartments

A gentle reminder from the philosophers. That beauty is not truth. And truth is not beauty. But location is location is location. -Eric Jarosinski

Prelude

Real estate in many prosperous cities worldwide has been the subject of endless debates, speculation, riches, frustration, and agony for decades, if not longer. The Swedish housing market is no different. On a macro level, whether it's an overpriced religion so crazy not even Tom Cruise would join or something completely different is still an open question. Despite knowing fully that relying too heavily on simple forecasts in a complex system is unwise, we still need to base our decisions on something rather than rolling dies.

The lion part of apartments bought and sold in Sweden are part of housing associations. Despite only sometimes being taken into consideration by home buyers, the financial situation of these associations could have a tremendous impact on the buyer's personal finances. Steep increases in fees, dropping property value, and, in extreme cases, bankruptcy are some of the woes awaiting the unlucky and uninformed buyer.

Your company wishes to use machine learning to predict property sales prices in Sweden. The database includes historical sales dating back almost ten years and some annual reports for the corresponding housing associations. Like most data sources, the data contains flaws, inconsistencies, and other defects due to flawed processes, human input, and poor underlying system design. The company operates in a regulated environment, so a validation committee must review statistical models before they can go into production. It is, therefore, of equal importance to the model performance that it is well documented and tested and that the code can be read and understood by people outside your team.

Stockholm School of Economics
Center for Data Analytics

Joel Smedberg
joel.smedberg@hhs.se

This assignment

The data is authentic but modified to fit the assignment's scope. It should be considered neither exhaustive nor entirely valid outside the course.

Reverse engineering sales prices from online sources is not kosher for obvious reasons. Utilizing other open data sources could potentially add predictive power, but it is not necessarily needed.

The grades will be based on the performance of the predictions made, the quality of the engineering (code and design decisions), and the final report. You may work in teams of two, and it is highly encouraged to do so. Groups are graded as one unit but can, on very rare occasions, be graded individually at the course director's discretion.

To provide a glimpse of forecasting, the final weeks of the dataset do not include sale prices. Part of the submission at the end will be a prediction for the missing sales prices. We are aware that relying solely on this type of data to make longer predictions is not a reliable approach. It should be noted that the precise evaluation of macroeconomic trends and black swan events with fat tails falls beyond the scope of this assignment.

scikit-learn is the most popular framework for dealing with machine learning in Python, and it is highly recommended for use in this course and assignment. What does not come out of the box can be easily extended by installing algorithms from PIP or Conda.

You are free to use any library or framework, but any external dependencies need to open source and installable through standard repositories such as PIP or Conda.

All data used for this assignment can be found on the MySQL database:

Host: mysql-1.cda.hhs.se
Username: 7313
Password: data
Port: 3306
Schema/database: SwedishHousingMarket

Some tips:

- Reading through the entire assignment before starting to code is also highly recommended. This is especially true for documentation requirements.
- Iterate – start with a small and simple model and gradually improve features, data, and tuning.
- If it’s stupid but works, it isn’t stupid.
- Creativity is always encouraged.

Stockholm School of Economics
Center for Data Analytics

Joel Smedberg
joel.smedberg@hhs.se

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 would 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 is 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 soundly evaluate the model's performance and robustness.

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 play around with 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 part 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).

Stockholm School of Economics
Center for Data Analytics
Joel Smedberg
joel.smedberg@hhs.se

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 are already 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 of good performance, no guarantees are made that what we are dealing with won't be better solved by some entirely different method. It would, as such, 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 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 demons will need to have a conversation about which rabbit holes are actually worthwhile diving deep into.

Step 3 – Validation Documentation

As part of the company's governance structure, models must be documented and externally validated. This ensures that no 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 more advanced A.I models are opaque black boxes, and explainable and interpretable AI are big research fields. The FSA [Financial Services Authority] performs regular checks and punishes compliance failures with hefty fines. Fines are handed out to institutions regularly, so the threat is not just 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 things.

Stockholm School of Economics
Center for Data Analytics

Joel Smedberg
joel.smedberg@hhs.se

Step 3B- Checklist of things that should be considered

- Feature engineering:
  - What features are used and not, and why?
  - For the 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 done additional feature engineering that needs to be mentioned?

- Model selection:
  - What machine learning algorithm did you settle for and why?
  - What baseline model did you use for comparison and why?
- What Champion challenger did you settle for?
- Model evaluation and testing
  - What metric(s) to evaluate did you decide on and why?
  - How well is the model performing according to your estimates?
  - How do you avoid over 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 new ones?
  - Have you made any other assumptions that need to be brought up and discussed?
- Tuning
  - Which hyperparameters did you tune? Why did you include /exclude specific parameters?
  - Can you be sure you have not introduced additional overfitting while tuning?
  - How much performance did you gain by tuning the model?
- Robustness
  - Does the model freak out by 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 gets lost? For example, some column(s) start containing only 0/null values instead of original values.
- Evaluation
  - How well are your models performing?
  - Is using a complex model justified, compared to using a simpler model?
  - With how much predictive power did each feature contribute?
  - Did some features turn out to be better or worse than one would expect?
  - 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 miserably in some area, price category, or similar that the company should worry about?
  - Are there any other concerns with the data or model that you believe should be flagged and brought to the attention of validation or senior management?

Stockholm School of Economics
Center for Data Analytics

Joel Smedberg
joel.smedberg@hhs.se

Step 4 – Submission

Code for generating plots and experiments outside the final model does not need to be submitted.

I. A JSON file containing the predicted sale prices (See example output below)
II. The Python code for training the primary model and generating the JSON results
III. The model validation documentation in .pdf

Name code and data files

1) assignment2_123456.json
2) assignment2_123456.py (or .ipynb also ok)

(replace 123456 with your enrollment number)

Example output for the prediction file (a list of dictionaries in JSON format)

```
[
    {
        "id": "00015e9ff97f3e12967d8551251c474a",
        "predicted_sell_price": 3600000
    },
    {
        "id": "0004952a432928d864eac5ff882d6124",
        "predicted_sell_price": 4710000
    }
   …More data goes here
]
```