Paper Structure

Contents

1 Introduction **2**

2 Methods **2**

2.1 Preprocessing 2

2.1.1 Feature Engineering 2

2.1.2 Manual Feature Selection 5

2.1.3 Encoding and Automatic Feature Selection 5

2.2 Models 6

2.2.1 Classical Models 6

2.2.2 Neural Network 6

2.2.3 Price Distribution 6

3 Results **7**

3.1 Predictive Performance 7

3.2 Explainability / Interpretability 8

3.2.1 Feature Importance 8

3.2.2 Sensitivity of Neural Network Performance on Outliers 8

4 Conclusion **8**

5 Appendix **8**

6 References **8**

1 Introduction

2 Methods

2.1 Preprocessing

2.1.1 Feature Engineering

Images

Strategy

- Use number of available images for each apartment as well as price predictions solely based on the image *contents* as numeric features for the final model

Webscraping

- Dataset contains link to websites of each listing
- Use the `requests` library to get `HTML` Source Code of each website
- Use the `beautifulsoup` library as HTML parser to find and extract embedded weblinks to all images located on this front page
  With this strategy we could extract the first 5 images (if 5 or more were available) that could be directly accessed from the front page source code
- Use `requests` again in combination with the `pillow` library to decode the content of at all image adresses to two dimensional images which serve as input to a Convolutional Neural Network

Preprocessing

- In contrast to classification tasks where *Data Augmentation* is commonly used in order to expand the training set and improve generalization, this approach is not immediately transferable to a regression context since we have to guarantee that the label (i.e. the price) remains unchanged for each image transformation
- Thus, we decided against rotating the images or manipulating the color composition

- We **did** use image *cropping* however, which is in our opinion one of the few reasonable augmentations for regression
- After resizing all images to 256×256 pixels we randomly cropped a square area of 224×224 out of each image in the training set and cropped an equally sized area out of the image **center** in the validation set to avoid any randomness during inference
- At a final step all images were normalized for each color channel separately with the same values that were used during training on `ImageNet` The mean values and standard deviations for each color channel are provided in the `PyTorch` documentation

Model

- Strategy: Fit one small custom CNN as benchmark model and one large pretrained Model with some custom final layers as main model
- Idea: The large model is pretrained on `ImageNet` and thus capable of extracting common features, use these features as input to custom final layer(s) that output a price prediction, i.e. the last layer directly maps to a single node
- Potential Issue: If the pretrained network is very deep, the learned features before the final layer could be very specific to the *Output Classes* of `ImageNet` and not generalize well to our images Some possible options:
  - Out of the collection of available very large pretrained models, choose one that is not extremely deep:
    We chose `ResNet18` with roughly 11 million parameters
  - Do not freeze the weights of the pretrained model completely but fine tune them (i.e. modify the weights by backpropagating through the entire network):
    We did not investigate this option further due to its high computational cost
  - Cut the pretrained model before the last layer (with the hope that at this point very generic and widely applicable features of images are extracted which therefore generalize better) and append the custom output layer

This option did not improve our results significantly

- It turned out that a single custom layer mapping from 512 to a single neuron was not expressive enough
- The performance improved slightly by adding a Fully Connected Network with three layers and $ReLU$ activation functions at the end

Results

- Using only the content of the available images, the pretrained `ResNet18` achieved a Mean Absolute Error of 579 NOK (approx. 58 Euros) on the Validation Set
- The 'Null' Model of always predicting the mean price achieved an MAE of 630 NOK without a log-transformation of the price and a MAE of 569 NOK with a log-transformation, so the predictive power of the images alone was very small
- However, the correlation with the CNN predictions with the true price was 0.41:
  This indicates some limitations of the correlation as useful metric on the one hand but positive tendencies of the CNN predictions on the other hand
- In fact, the network struggled the most with capturing the wide range of prices and almost always predicted values close to the center of the (log) price distribution
- Considering the difficulty of the task it is actually highly doubtful that humans could provide much more accurate predictions
- Show **Figure** of sample images with true and predicted price

Reviews

- Description of Sentiment Analysis, stating procedure and results and including **Figure** with Wordcloud, either only English Words or Side-by-Side Wordclouds of English and Norwegian Words
- In addition: Language Detection to include the *number of different languages* and the *fraction of norwegian languages* and Analyzing the reviews lengths to include the *median review length*

- Since there are multiple reviews per apartment the results for each review were averaged for each apartment separately.

Others

- Optionally mention all other features that we added to the dataset

2.1.2 Manual Feature Selection

- Describe process of combining features from images, reviews and numeric features into one dataframe
- Decision to select features for the final model were based on:
  - Human Background/Context Knowledge (Apartment *Size* is a sensible predictor for the price by intuition)
  - marginal relationships to price detected by visualization (barplot for categorical variable vs. price or scatterplot for numeric variable vs. price)
  - Small Dataset with around 3000 observations: If no reasonable imputation is possible, consider tradeoff between additional predictive value of a variable and number of lost data points due to missing values
  - Final Step: Built Linear Regression Model with (almost) all features and investigate features with high coefficients in absolute value. Since variables are standardized, coefficient magnitudes have meaning.

2.1.3 Encoding and Automatic Feature Selection

- One-Hot Encoding of Categorical Variables, Standardization of Numeric Variables
- Experimenting with different ways of algorithm-based feature selection / dimensionality reduction
- Focus on `PCA` as most theoretically supported procedure and `RFE` as procedure we chose

- PCA has advantage of reducing dimensionality and simultaneously producing *uncorrelated* features, disadvantage of producing linear combinations of original features which are harder to interpret
- `RFE` showed best performance and selects subset of original features, can be immediately interpreted as potentially most important features
- Briefly explain how `RFE` works

2.2 Models

2.2.1 Classical Models

- serve as benchmark models to better evaluate performance of custom neural network
- selected with increasing degrees of complexity and corresponding decreasing degree of interpretability
- Focus on 4 models: `LinearRegression`, `Ridge`, `RandomForest` and `HistGradientBoosting`
- Describe Model Fitting process and hyperparameter tuning with Randomized Search Cross Validation

2.2.2 Neural Network

- Explain Architecture of Neural Network
- **Figure** with diagram of `Linear Block`
- Explain Choice of Model Parameters and Hyperparameters and emphasize which components were most important
- **Figure** with impact of `Dropout`

2.2.3 Price Distribution

- **Figure** of Side-by-Side Histograms of Price and Log-Price Distribution
- Price Distribution right-skewed with some very large outliers

- Explain benefits of normally distributed dependent variable, particular for models with distributional assumptions (e.g. Linear Regression vs. Neural Network)
- State that all classical Machine Learning Models benefitted from log transformation
- Briefly discuss why we did not transform the price variable for the Neural Network

3 Results

3.1 Predictive Performance

- **Figure** of performance comparison between selected classical models and neural network for given feature selector (e.g. RFE) and different number of selected features
- Interpret Differences in Training and Validation Performance between different models
- Interpret Differences in Performance for different number of selected features
- Compare Performance on Validation Set with Performance on Test Set for the best model of each class by means of a table
  $\Rightarrow$ Models whose hyperparameters were tuned on validation set generalize worse to test set, e.g. `HistGradientBoosting`, `RandomForest` and `Ridge`
- Include average predictions of top 2/3/4/5 models, where models are selected based on validation set performance and Test Set predictions are averaged
- Potentially mention which models contributed to predictions on new, unseen dataset from challenge

3.2 Explainability / Interpretability

3.2.1 Feature Importance

- **Figure** of Coefficient Plot for Linear Regression with e.g. 25 selected features
- Interpret Figure

3.2.2 Sensitivity of Neural Network Performance on Outliers

- State shortcomings of Neural Net to predict prices in the tails of the distribution, error metrics thus largely impacted by outliers
- State (maybe with a table) drastic increase in predictive performance when excluding largest quantiles of price distribution from dataset
- Discuss if the task itself is theoretically feasible for any kind of model
- **Figure** of latent space representation
- Discuss that the data is not expressive enough to capture all features that determine the price in reality, particularly for apartments with very high prices, that do not differ from lower priced apartments based on their feature set
- Question underlying assumptions that all apartments are reasonably priced, difficult to detect overpriced listings that bias model predictions

4 Conclusion

5 Appendix

- include link to repository with codebase to reproduce all findings

6 References