Shahid Beheshti University
Data Science (2023-2024)

House Price Estimation From Visual and Textual Features

Mehrab Kalantary, Razieh Moslemy

Department of Computer Science

Shahid Beheshti University

Mehrab.nen@gmail.com, r.m.m.i.a.t@gmail.com

Abstract

Most existing automatic house price estimation systems rely only on some textual data like its neighborhood area and the number of rooms. The final price is estimated by a human agent who visits the house and assesses it visually. In this project, we work on extracting visual features from house photographs and combining them with the house's textual information. The combined features are fed to machine learning and then deep learning algorithms to estimates the house price as its single output. To train and evaluate our models, we use this collected dataset that combines both images and textual attributes. The dataset is composed of 535 sample houses from the state of California, USA.

1 Introduction

Housing market plays a significant role in shaping the economy. Housing renovation and construction boost the economy by increasing the house sales rate, employment and expenditures. It also affects the demand for other relevant industries such as the construction supplies and the household durables. The traditional tedious price prediction process is based on the sales price comparison and the cost which is unreliable and lacks an accepted standard and a certification process. Therefore, a precise automatic prediction for the houses' prices is needed to help policy makers to better design policies and control inflation and also help individuals for wise investment plans. So, we need to combine both visual and textual attributes to be used in the price estimation process.

The contribution of this project:

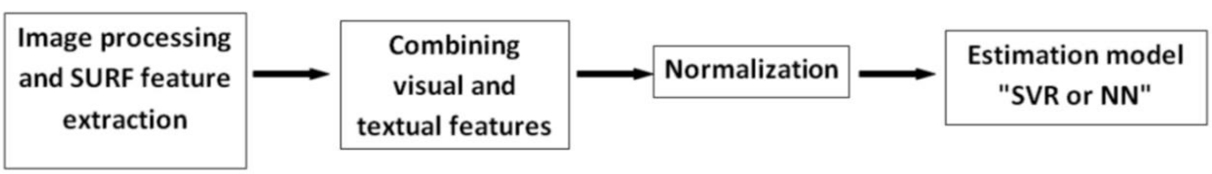
- We use the provided dataset by Eman Ahmed that combines both visual and textual attributes to be used for price estimation.
- We propose both machine learning and deep learning approaches to estimate prices. Then report the results of these proposed models

The remaining of this project is organized as follows:

We start by reviewing related works, followed by a understanding of the dataset provided by Eman Ahmed in detail. We then propose our methods and baselines to solve this problem followed by the results of each baseline. Finally, we close with some concluding remarks.

2 Related Works

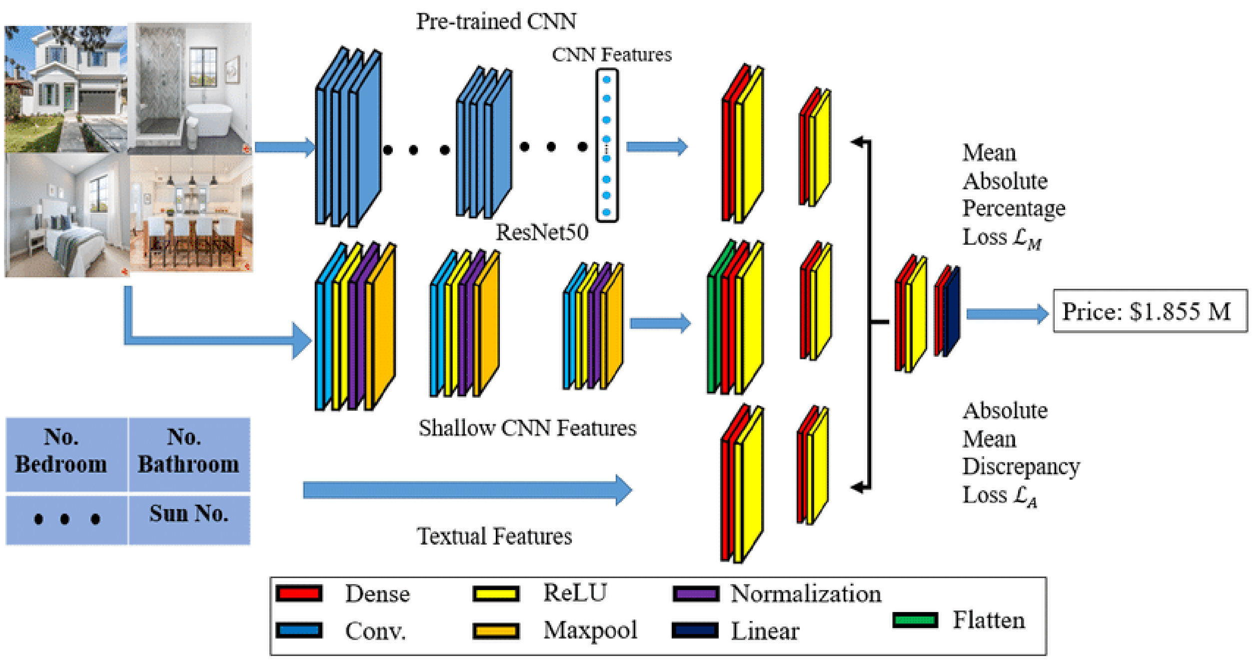
For the first time, estimating houses' prices from visual and textual data was proposed by Eman Ahmed and Mohamed Moustafa in house price estimation from visual and textual features paper in 2016 which focused on feature engineering and preprocessing rather than training complex models. The following pipeline is proposed:



SURF method was used to extract features from each image. This is an example for extracted features:

Also Min-Max normalization was used to normalize combined data. Finally, multilayer perceptron and support vector regressor was used to estimate price.

Later in 2020 in a new paper published by Youshan Zhang, a new deep learning architecture was proposed to estimate houses' prices using a three-channel convolutional neural network.

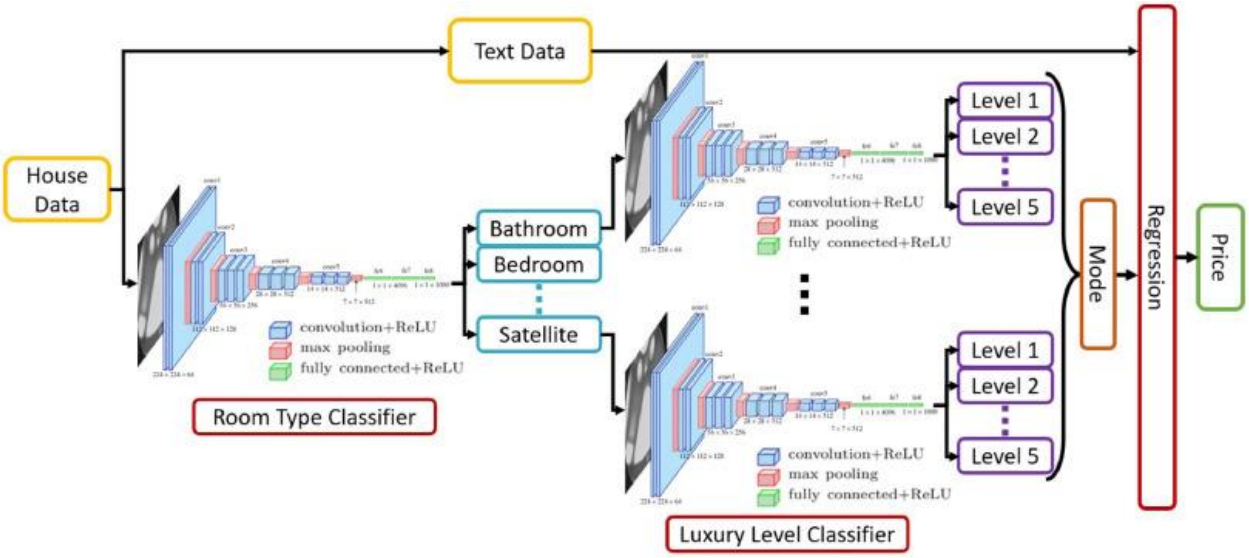


There are three modules in the model:

Deep feature module, shallow CNN feature module and textual feature module. In deep feature module, the feature is obtained from the pre-trained ResNet50 neural network. The shallow CNN feature module directly gets features from raw images via three repeated blocks. The textual feature module majorly contains two layers.

The model consists of two different loss functions, including mean absolute percentage error loss and absolute mean discrepancy loss. The mean absolute percentage error loss measures mean percentage difference between predicted price and actual house price, and absolute mean discrepancy loss ensures that the mean predicted price approximates the mean house price.

Ultimately, in 2022 Vision-based housing price estimation using interior, exterior & satellite images was published to solve a more advanced price estimation problem which can be modified to solve our problem. A new deep learning architecture was proposed to address this issue:



The model includes 6 stages:

1. Classify images to room categories (bathroom, bedroom, kitchen, living room, dining room, front, satellite images)
2. Classify each category of images to luxury levels
3. Take “Mode” of luxury levels of all room types as level feature
4. Combine with text data
5. Input to regression model
6. Compare the performance to with the literature methods.

3 Dataset Description

3.1 Understanding Dataset

The collected dataset is composed of 535 sample houses from California State in the United States. Each house is represented by both visual and textual data. The visual data is a set of 4 images for the frontal image of the house, the bedroom, the kitchen, and the bathroom as shown in the figure below. The textual data represent the physical attributes of the house such as the number of bedrooms, the number of bathrooms, the area of the house, and the zip code for the place where the house is located. This dataset was collected and annotated manually from publicly available information on websites that sell houses. There are no repeated data nor missing ones.

The information about the dataset and its features can be seen in the following tables. There are 5 numerical features and 4 images for each house.

Bedrooms	Bathrooms	Area	Zipcode	Price
4	4.0	4053	85255	869500
4	3.0	3343	36372	865200
3	4.0	3923	85266	889000
5	5.0	4022	85262	910000
3	4.0	4116	85266	971226

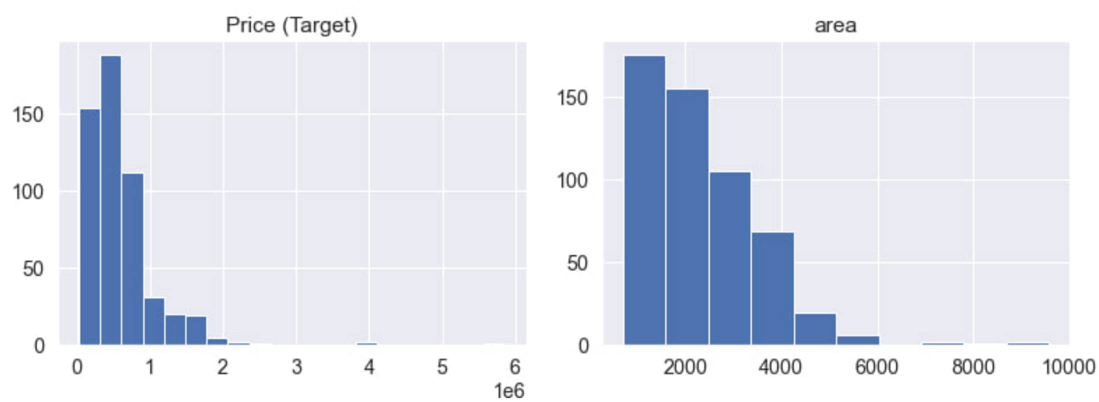
Feature	Dtype	Unique Numbers	Description
bedrooms	int64	9	Number of bedrooms
bathrooms	float64	14	Number of bathrooms
area	int64	435	Area of the house
zipcode	int64	49	Zipcode
price	int64	369	House price

3.2 Statistical Details

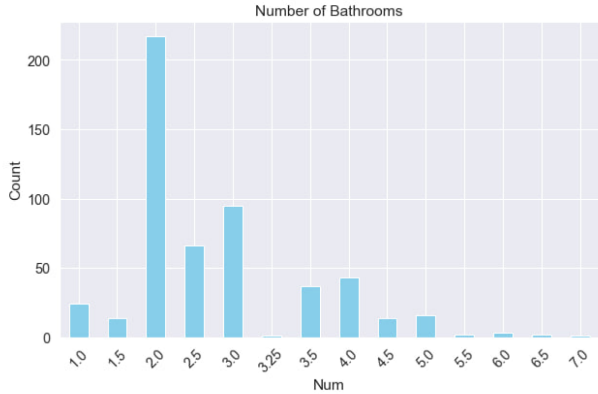
The chart below, provides some statistics related to the dataset. The house price in the dataset ranges from $22,000 to $5,858,000. Also the minimum image resolution is 250 * 187 which is important to us and will be discussed later. It is also notable that number of bathrooms and bedrooms range from 1 to 7 and 10.

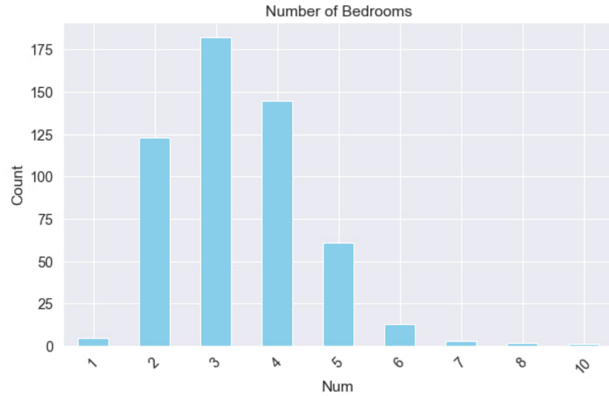
Detail	Average	Minimum	Maximum
House price (USD)	589,360	22,000	5,858,000
House area (sq. ft.)	2364.9	701	9583
Number of bedrooms	3.38	1	10
Number of bathrooms	2.67	1	7
Images resolution	801x560	250x187	1484x1484

The provided histograms also illustrate price and area features distributions those are normal distributions with skewness. The range and average for each feature can be compared with the table. Also there are some huge numbers that may relate to each other.



The bar charts shown also provide information about the distribution of number of bedrooms and bathrooms. It is evident that most houses have 2 bathrooms and 3 bedrooms. Also just one house has 10 bedrooms and 7 bathrooms.



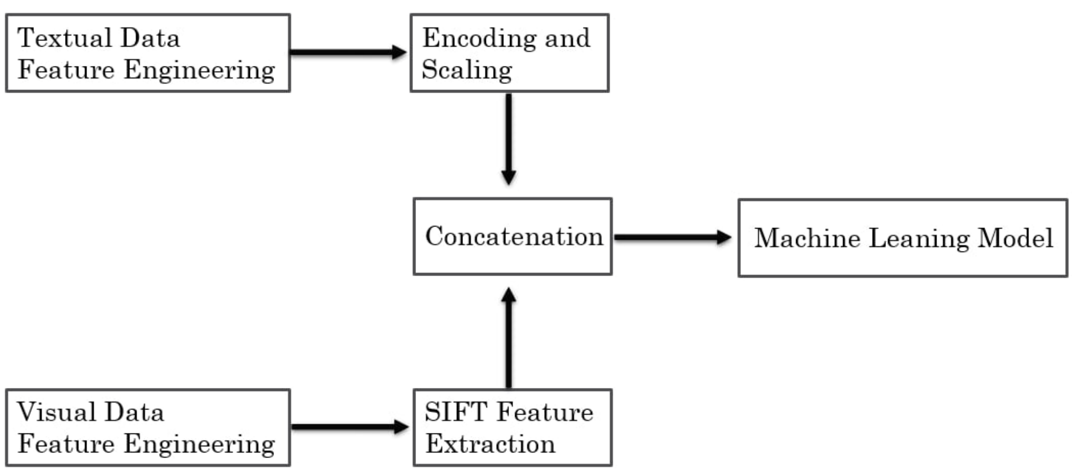


4 Proposed Methods

In this section we talk about methods we used to estimate houses' prices. In general, we can divide our methods into machine learning based approaches and deep learning based approaches.

4.1 Machine Learning Approach

The overall pipeline used in this section can be seen in the following flowchart.



We have used two feature engineering phases for both textual and visual features then applied a feature extraction algorithm for visual data and a preprocessor module for textual data. Finally, the results were concatenated and fed into machine learning algorithms.

Textual data feature engineering includes 3 stages:

Stage 1: Creating and extracting new features from the dataset

- Total rooms: Sum of number of bedrooms and bathrooms plus 1 which is the kitchen
- Average room area: The ratio of area to total rooms
- Area size: The category of area which can be very small, small, medium, large and very large

Stage 2: Ordinal encoding for categorical columns which is only area size. The main reason behind this choice is the characteristics of this feature which is comparable.

Stage 3: Log transformation for skewed features such as area, average room area and price. This choice was made after some tests and it had the best result compared to normalization and standardization.

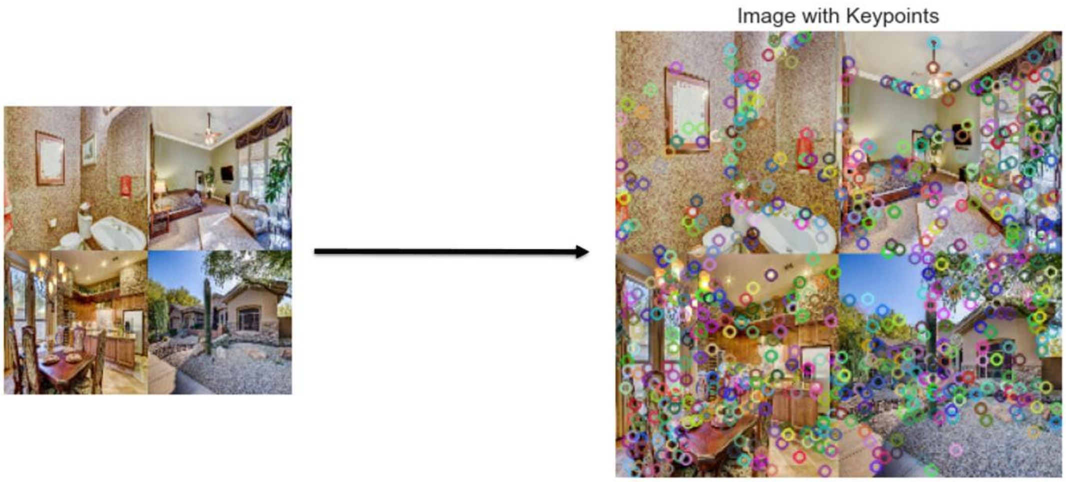
Visual data feature engineering includes 2 stages:

Stage 1: Representation of the images
In this part we decided to create a matrix which represents all 4 images for a house so we can refer to each house by one n * n matrix. The choice of the size was not easy, we had to do some tests and see which one gives the best results. Also we knew that the minimum image resolution is 250 * 187 and we did not want to surpass that number. At first we decided to create a 256 * 256 matrix and put 4 images in it, making each image to be 128 * 128 which is appropriate. Furthermore, we decreased this number and did some tests. Finally, the best results were achieved by the first choice.

After this stage we have 535 matrices with the size of 256 * 256 * 3 each of which contains 4 images with RGB channels. Now it is time for the second stage.

Stage 2: In this stage we apply SIFT method to extract features from each group of images. This part is absolutely essential considering the fact that we need to reduce dimension so we can use machine learning models and achieve a higher score. The main reason for this choice was mostly because SURF were used in previous works and we wanted to try new methods which works better than existing algorithms. By the end of this section we have 535 images with the dimension of 128 each of which represents extracted features from a house.

An example of this method is as follows:



4.1.1 Classic Machine Learning Models

In this section we talk about some classic machine learning models and the way we trained them. By now we have both datasets preprocessed and combined which are ready to be used.

1. **Linear Regression**
   Here only a simple built-in linear regression is trained on the dataset and poor score is achieved.

2. **Polynomial Regression**
   In this part we used a polynomial regression algorithm with the degree of 2 and 3 and the results were absolutely terrible and the worst among all methods.

3. **Ridge Regression**
   This algorithm with a regularization parameter of 1.0 (alpha) reached a better score than the last two methods but not the best among classic methods.

4. **Decision Tree Regressor**
   The best results among classic methods were achieved by decision tree model with a max depth of 3 which was far more better than the other models.

4.1.2 Advanced Machine Learning Models

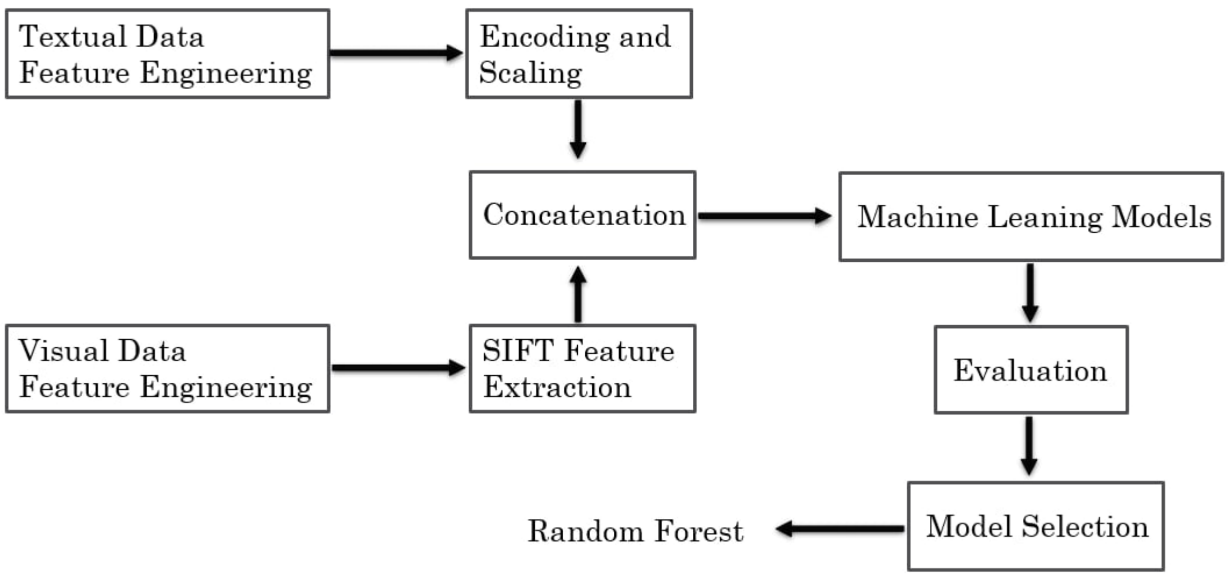
This section focuses on more advanced machine learning models, the ones use bagging and boosting methods. It is notable that we tried plenty of tests and found the best hyper-parameters for these methods and the best result were saved.

1. **Random Forest Regressor**
   This method reached the best result among all methods we used for this task with only 50 estimators.

2. **Support Vector Regressor**
   SVR achieved a relatively high score compared to classic methods but not a good one compared to advanced methods.

3. **CatBoost Regressor**
4. **XGBoost Regressor**
   After lots of tests and evaluations to fine-tune these algorithms, the best score for these two boosting methods were almost equal and high but not better than random forest.

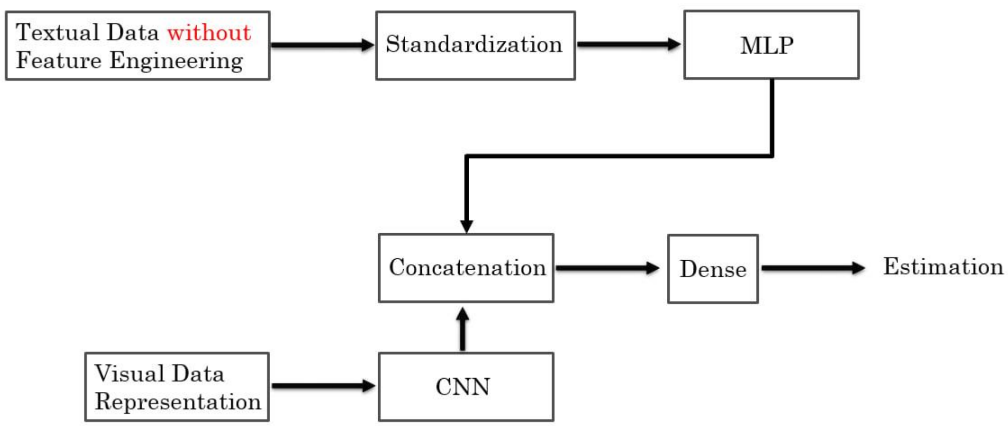
So we can complete our diagram now.



4.2 Deep Learning Approach

In this part we use neural networks and different architectures to estimate houses' prices. Two types of neural networks are used in this section, multilayer perceptron and CNNs. The main reason behind choosing MLP and CNNs is that we need two different networks, one for textual data and the other for visual data and this choice helps us to train a multi-channel neural network.

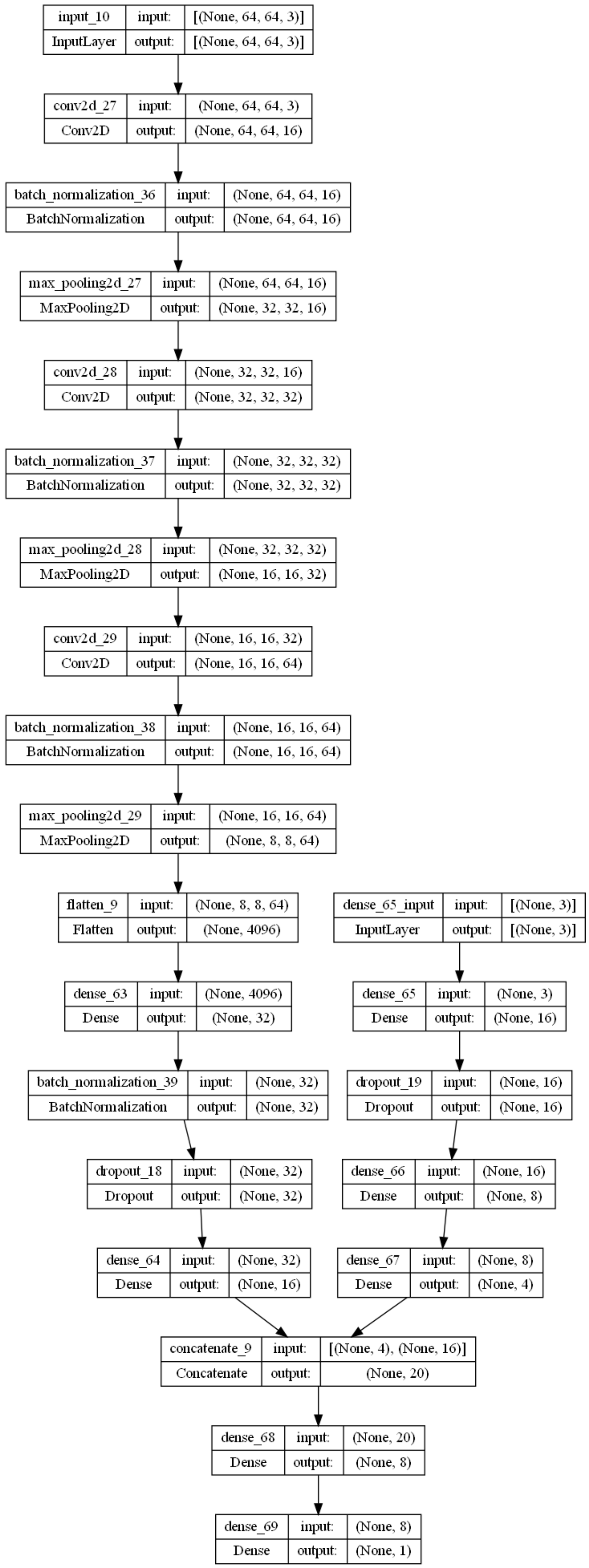
For this approach we do not use the same preprocessing and feature engineering. Instead, we decided to leave it to the network so it extracts features and tries to solve the problem with raw data without any help. In general, we use the following diagram for this part:



All we need is to apply standard scaler on our textual data and find the best way to represent visual data. The answer to the question "Why standardization?" is that we tried many preprocessing methods including the one in machine learning approach, but standardization had the best score for our network among all. Another issue that we faced here was the choice of representation of the visual data. We decided to use matrix representation from the previous section but we needed to find the best size. After some tests, the best score was achieved when we set the size of matrix 64 * 64, meaning that each image would be 32 * 32.

After these steps, textual data is fed through MLP channel and visual data is fed through CNN channel, finally they are concatenated and pass through a hidden dense layer and the output layer estimates the price.

Another main issue we encountered in this section was the choice of architecture. We did not want to create a deep and complex network due to lack of data so we decided to create a shallow network with only 3 convolution layers. Network's architecture is as follows:


input_10 | input: | [(None, 64, 64, 3)]
InputLayer | output: | [(None, 64, 64, 3)]
conv2d_27 | input: | (None, 64, 64, 3)
Conv2D | output: | (None, 64, 64, 16)
batch_normalization_36 | input: | (None, 64, 64, 16)
BatchNormalization | output: | (None, 64, 64, 16)
max_pooling2d_27 | input: | (None, 64, 64, 16)
MaxPooling2D | output: | (None, 32, 32, 16)
conv2d_28 | input: | (None, 32, 32, 16)
Conv2D | output: | (None, 32, 32, 32)
batch_normalization_37 | input: | (None, 32, 32, 32)
BatchNormalization | output: | (None, 32, 32, 32)
max_pooling2d_28 | input: | (None, 32, 32, 32)
MaxPooling2D | output: | (None, 16, 16, 32)
conv2d_29 | input: | (None, 16, 16, 32)
Conv2D | output: | (None, 16, 16, 64)
batch_normalization_38 | input: | (None, 16, 16, 64)
BatchNormalization | output: | (None, 16, 16, 64)
max_pooling2d_29 | input: | (None, 16, 16, 64)
MaxPooling2D | output: | (None, 8, 8, 64)
flatten_9 | input: | (None, 8, 8, 64)
Flatten | output: | (None, 4096)
dense_65_input | input: | [(None, 3)]
InputLayer | output: | [(None, 3)]
dense_63 | input: | (None, 4096)
Dense | output: | (None, 32)
dense_65 | input: | (None, 3)
Dense | output: | (None, 16)
batch_normalization_39 | input: | (None, 32)
BatchNormalization | output: | (None, 32)
dropout_19 | input: | (None, 16)
Dropout | output: | (None, 16)
dropout_18 | input: | (None, 32)
Dropout | output: | (None, 32)
dense_66 | input: | (None, 16)
Dense | output: | (None, 8)
dense_64 | input: | (None, 32)
Dense | output: | (None, 16)
dense_67 | input: | (None, 8)
Dense | output: | (None, 4)
concatenate_9 | input: | [(None, 4), (None, 16)]
Concatenate | output: | (None, 20)
dense_68 | input: | (None, 20)
Dense | output: | (None, 8)
dense_69 | input: | (None, 8)
Dense | output: | (None, 1)

4.3 Performance Evaluation

4.3.1 Mean Absolute Error

MAE measures as the average absolute difference between the predicted values and the actual values and is used to assess the effectiveness of a regression model.

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n} |y_i - \hat{y}_i|$$

4.3.2 R2 Score

The coefficient of determination is a measure of the closeness of the predicted model relative to the actual model. It is calculated a set of various errors:

SSE is the Sum of Squares of Error and SST is the Sum of Squares Total.

$$\text{SSE} = \sum_{i=1}^{n} (\hat{y}_i - y_i)^2$$

$$\text{SST} = \sum_{i=1}^{n} (\bar{y} - y_i)^2$$

The value of R2 ranges between 0 and 1, the higher the value, the more accurate the estimation model. The R-squared value is calculated by:

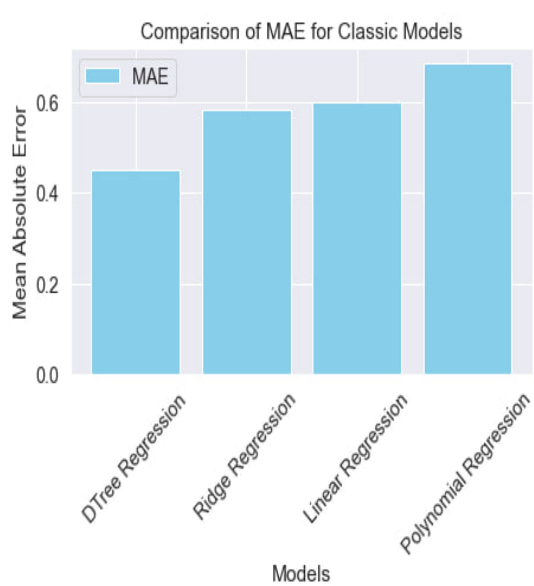
$$R^2 = 1 - \frac{SSE}{SST}$$

5 Results

In this section the results achieved by methods we used is shown and discussed.

5.1 Classic Methods Results

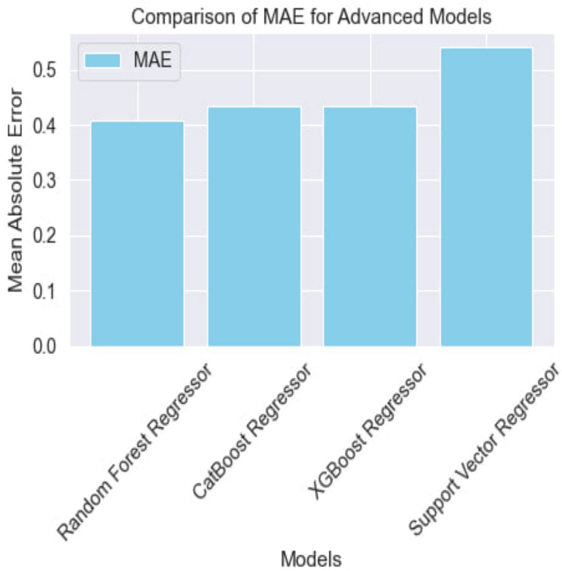
The results for classic machine learning algorithms are as follows:

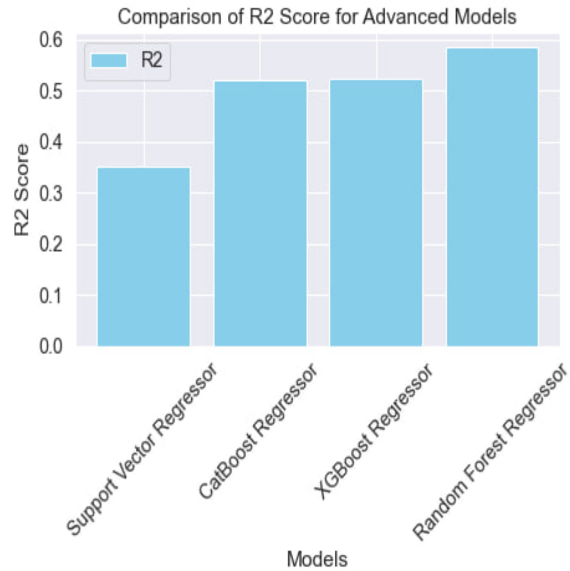


It is evident that the best score was achieved by decision tree and the worst of all was polynomial regression.

5.2 Advanced Methods Results

The results for advanced machine learning algorithms are as follows:

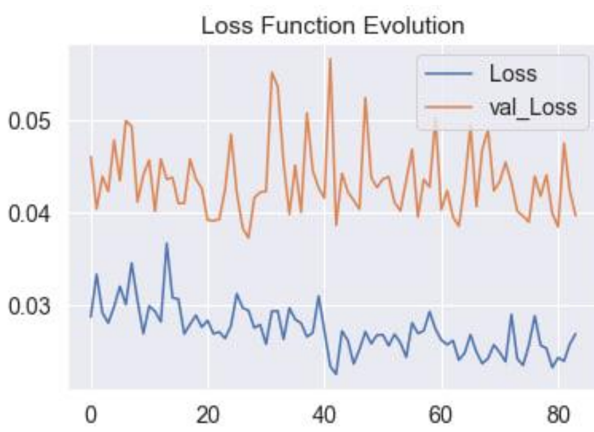




We can see that the best performance of all was achieved by random forest. Also the score of catboost and xgboost methods are nearly equal and better than decision tree. On the other hand, SVR performed weaker than decision tree.

5.3 Multi-Channel Neural Network

For our neural network the best score achieved after 100 epochs was 40% which is not as good as advanced models. This is mostly because of lack of data and simple architecture we used and also it is undeniable that the raw data was fed into the neural network but not machine learning models. The following chart shows train loss and test loss for our network:

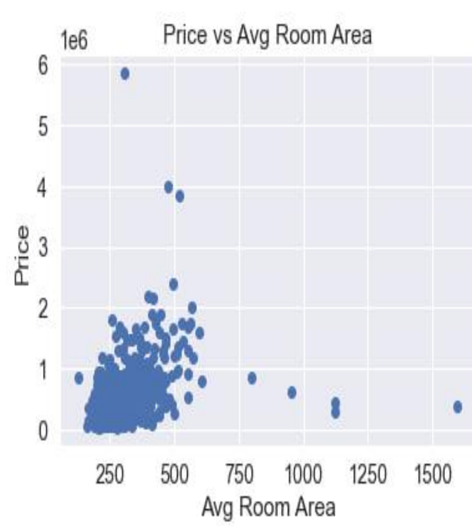


6 Further Analysis

6.1 Feature Importance

In this section we want to talk about feature importance and the impact of each feature on the target. The following charts illustrate the relationship between the target and number of rooms which is bedrooms, bathrooms and total rooms.

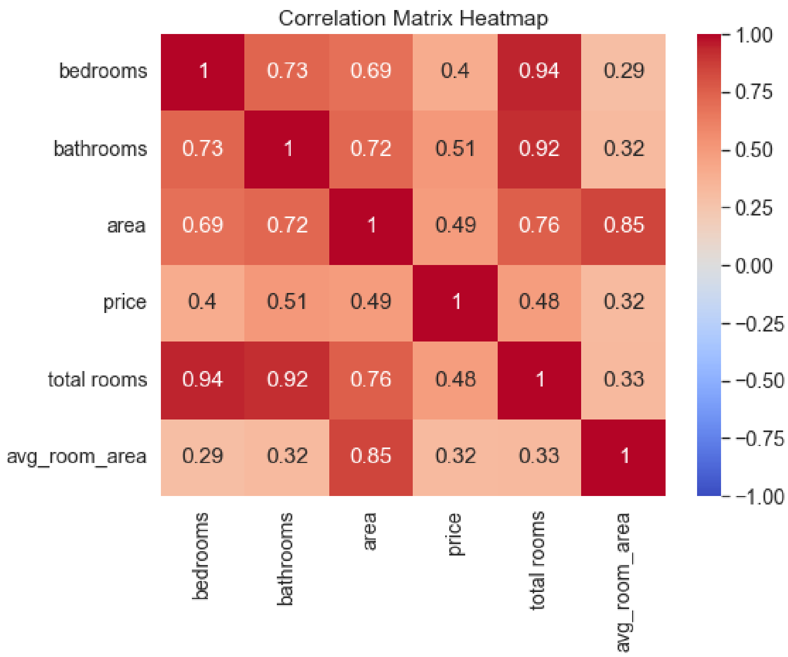
It is evident that each feature shares a strong correlation with the target especially total rooms as a feature we added. We will discuss it statistically later.



The two charts provided above, show the relationship between total area and average room area with the target. We can see that there is a relatively strong correlation between them.

6.2 Correlation Analysis

In this section we prove what we saw before statistically. The following chart shows the correlation among all features.



It is clear that the target shares a good correlation with other features. Also we can see correlation among other features such as number of bedrooms and bathrooms.

7 Conclusion

Through experiments, it was shown that the results were not as high as we expected for the classic algorithms. However, advanced methods have achieved a high score among all even neural network we designed.

References

https://www.sciencedirect.com/science/article/pii/S2667305322000217

https://www.researchgate.net/figure/The-architecture-of-our-proposed-mixing-deep-visual-and-textual-features-for-house-price_fig3_343856615

https://arxiv.org/abs/1609.08399

https://github.com/emanhamed/Houses-dataset