Spring 2021

Image-based Real Estate Appraisal using CNNs and Ensemble Learning

Prathamesh Dnyanesh Kumkar
San Jose State University

Recommended Citation

Kumkar, Prathamesh Dnyanesh, "Image-based Real Estate Appraisal using CNNs and Ensemble Learning" (2021). *Master's Projects*. 1015.
DOI: https://doi.org/10.31979/etd.km4q-65hg
https://scholarworks.sjsu.edu/etd_projects/1015

Image-based Real Estate Appraisal using CNNs and Ensemble Learning

A Project Report

Presented to

The Faculty of the Department of Computer Science

San José State University

In Partial Fulfillment

Of the Requirements of the Class

CS 298

By

Prathamesh Dnyanesh Kumkar

May 2021

The Designated Project Committee Approves the Project Titled

Image-based Real Estate Appraisal using CNNs and Ensemble Learning

by

Prathamesh Dnyanesh Kumkar

APPROVED FOR THE DEPARTMENT OF COMPUTER SCIENCE

SAN JOSÉ STATE UNIVERSITY

May 2021

Dr. Ching-seh Wu, Department of Computer Science

Dr. Robert Chun, Department of Computer Science

Dr. Fabio Di Troia, Department of Computer Science

ACKNOWLEDGEMENT

The past year has been extremely exciting as well as arduous with unprecedented challenges and situations along the way. Although it took great effort and strength to overcome the obstacles along the way, it was a great learning experience for me. I take this opportunity to express my gratitude towards Dr. Mike Wu, who not only supported me through these tough times but also guided me throughout the journey and ensured the success of my project.

I also want to thank Dr. Robert Chun and Dr. Fabio Di Troia for agreeing to be on the committee for this project and providing their valuable feedback, which helped me greatly during the course of this project.

I would also like to thank my parents and my brother, Siddhesh, for believing in me and encouraging me throughout my graduate studies.

Abstract

Real Estate Appraisal is performed to evaluate properties during a range of activities like buying, selling, mortgaging, or insuring. Traditionally, this process is done by real estate brokers who consider factors like the location of a house, its area, the number of bedrooms and bathrooms, along with other amenities to assess the property. This approach is quite subjective since different brokers may arrive at a different quote for the same property depending on their analysis. The development in machine learning algorithms has given rise to several Automated Valuation Models (AVMs) to estimate real estate prices. Real estate websites use such AVMs to provide an estimated price to potential buyers and sellers of properties. However, these models do not consider the impact that the appearance of a house has on its price.

Recent advancements in Convolutional Neural Networks (CNNs) have achieved state-of-the-art performance for several computer vision tasks. This project uses a CNN to evaluate and score the image data associated with a house. This information was then combined with other property-related data and three ensemble learning models, namely, Random Forest, Gradient Boosting, and Extreme Gradient Boosting (XGBoost), were then trained to estimate real estate prices. The performance of these models was compared with each other and it was found that the XGBoost model achieved the best performance with a MAPE score of 9.86%. Although the image-based model performed better, the XGBoost model which did not consider image data also achieved comparable results with a MAPE score of 10.33%.

Index terms – Real Estate Appraisal, Machine Learning, Convolutional Neural Network, Ensemble Learning, Image Evaluation

Table of Contents

List of Figures

List of Tables

I. Introduction And Research Objective

A. Introduction

Real estate appraisal is the process of evaluating the market value of a property by considering several factors that could potentially impact its price. Banks and other financial institutions rely on this process to assess the amount they can lend or insure to their customers. This process is also a vital part of several endeavors like buying and selling a property, rental price valuation, tax assessment, and mortgage risk control. Therefore, an appropriate evaluation of real estate prices is crucial for the stakeholders to make an informed decision. The price of a property depends on several factors such as its location, living area, construction quality, available amenities, and so on. Hence, the appraisal of properties is a challenging task and is typically done by professional appraisers who have a lot of expertise in the field of real estate. However, different appraisers may quote different prices for the same property depending on their knowledge and experience. Thus, an Automated Valuation Model (AVM) that takes a scientific approach can be useful for evaluating property prices by comprehensively analyzing the related factors.

In today's world, real estate websites like Zillow [1], Redfin [2], and Trulia [3] are seeing increased popularity. Property owners post their listings on these websites which include the characteristics of the property like location, area, number of bedrooms and bathrooms, age, and other facilities, along with interior and exterior photos. While listing their properties on such websites, owners also need to provide a selling price for their property. For this purpose, these websites offer a prediction for the price of the property based on the user-provided information using their proprietary AVM. The Zestimate valuation model [4] by Zillow provides a reference price to buyers and sellers, by taking into account the location and other characteristics of the property along with the market conditions. The median error rate for Zestimate is 1.9% nationwide for on-market homes, meaning that Zillow's prediction is within 2% of the final sale price for half of the homes. Redfin also provides a Redfin Estimate [5] with a median error rate of 1.82% for properties on sale.

Traditionally, the AVMs developed by the real estate websites do not consider the impact of the visual features of a property while providing an estimate for the price. However, the appearance of the house plays a major role in its appraisal. This factor is used by the sellers and home staging companies to make a house visually appealing. Using the visual attributes of a house, customers can get a good idea about the overall infrastructure, construction, and neighborhood of the house. These factors can be difficult to quantify and use to determine the price of a house computationally. However, using today's computing power and resources, it is much easier to analyze these high-level attributes. The advancement in deep learning techniques enables us to interpret and evaluate this pictorial representation in the same way as human beings.

Considering the recent developments in deep learning, this project aims at using the visual data associated with a property to arrive at an appropriate market value. Convolutional Neural Networks (CNNs) have achieved good results in several image-related tasks like image classification, aesthetic estimation, digit recognition, etc. Using CNNs, it is possible to quantify the image data related to a property and use this information along with other property-related details to estimate its price.

B. Research Objective

The objective of this project is to develop a model for estimating the prices of real estate listings using images and other property-related metadata. The data for the project will be obtained from Zillow by considering the property listings corresponding to recent sales in the city of San Jose. The images related to a given property will first be aesthetically scored using CNNs. Following this, the aesthetic score will be used along with other property-related data (total area, number of bedrooms and bathrooms, location, etc.) to train a regression model to estimate property prices. During the training process, the selling price of the houses will serve as the price labels. The final estimated price of a property will be based on an ensemble of several regression models. Interested buyers or sellers of a property can use this model to arrive at a quote for the property’s price.

II. Background And Related Work

The purpose of this section is to evaluate and compare the performance of several approaches to incorporate image data in the process of assessing the prices of real estate properties. The aim is to answer two questions: Does the accuracy of prediction increase when considering the visual attributes of properties? Which methods can be employed to achieve the best results for evaluating property prices?

The use of AVMs for estimating real estate prices has been a topic of academic and industrial interest. Ensemble Methods [24] have been widely employed for estimating real estate prices [26, 27]. Most of the research involves the use of the textual or numeric attributes of the house like its location, area, age, and so on. However, there is very little literature focusing on the use of the visual attributes of a house in determining its price. This review considers several research papers that use visual data about the properties to estimate a price for a given property. It consists of the following sections: Section A describes the approaches which are used to quantify the visual characteristics of a property. Section B describes an approach in which the image data is used to cluster similar houses together before estimating their prices. Section C describes how the image data is used to filter duplicates from the dataset. Figure 1 illustrates the organization of this review.

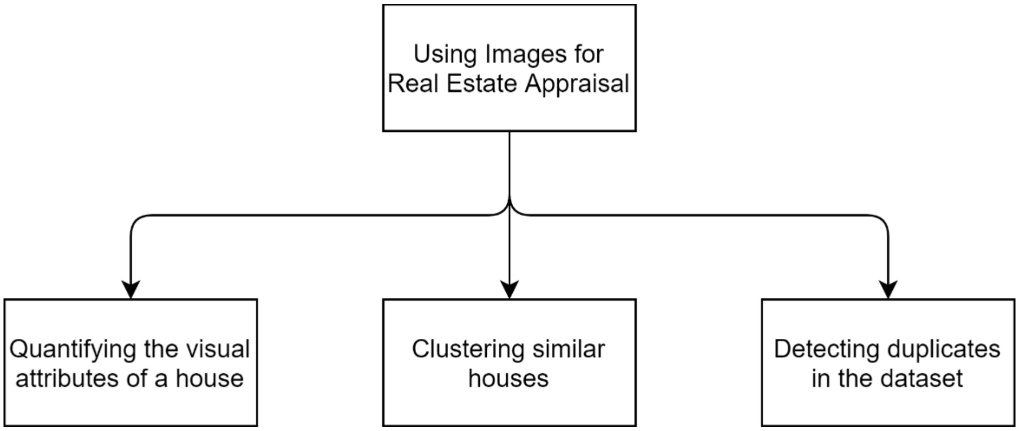

Using Images for Real Estate Appraisal
Quantifying the visual attributes of a house
Clustering similar houses
Detecting duplicates in the dataset

Figure 1: Different approaches to using images for real estate appraisal

A. Using Image Data To Quantify The Visual Attributes Of A Property

Ahmed and Moustafa [6] proposed a system to analyze the effect of image data in the overall prediction model. For their experiment, the authors considered four images for each house: the frontal view, the kitchen, the bedroom, and the bathroom. They designed an image processing module that performed two functions: increase the contrast of the images in the dataset and extract visual features from the images. For the first part, they used the histogram equalization technique described in [7] and the resulting images had a uniform intensity of colors throughout by boosting the areas having lower contrast. The latter part used the SURF extractor described in [8] to extract features from the images.

The authors extracted the n strongest features from each of the four images of the house and then concatenated them into a single vector with other numeric attributes. The experiments were performed by considering different values for n each time, to find the optimal number of features to be extracted. The combined features for each house were then normalized and passed to one of the two modules: Support Vector Regression (SVR) or Neural Network (NN). Figure 2 describes the processing stages involved in the experiment in [6].

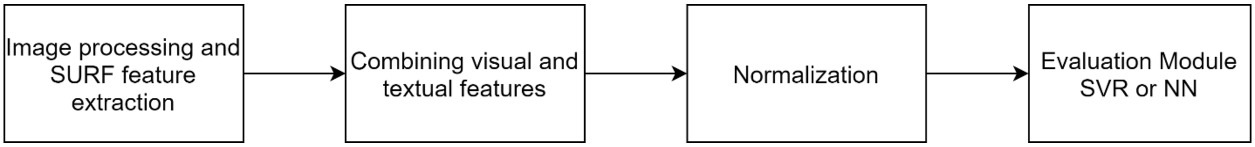


Image processing and SURF feature extraction
Combining visual and textual features
Normalization
Evaluation Module SVR or NN

Figure 2: Processing stages in [6]

This research concludes that the addition of visual data increased the accuracy of prediction by comparing their model with another Neural Network model [25] which only used numeric data. Their model achieved a higher R-value by 6.8% on the training set and 6.97% on the testing set. In summary, the accuracy of the overall prediction is improved by using image data along with the other characteristics of the house.

Poursaeed, Matera, and Belongie [9] proposed a method to evaluate the luxury level of a property from its images. For this experiment, the authors used a crowdsourcing approach to create a dataset of real estate images with a luxury level assigned to each of the images. Thus, each image in the dataset had a score assigned from 1 to 8 (8 being the highest). The number of luxury levels was determined by obtaining a low-dimensional embedding of the images in the dataset using the t-STE algorithm [10], such that similar images were in the same cluster.

The authors trained a DenseNet [11] for classifying the images into one of the following 7 categories: bedroom, bathroom, living room, dining room, kitchen, interior, and exterior. Following this, they trained a DenseNet for classifying the images into one of the 8 luxury levels. Thus, each house had 7 different luxury values corresponding to each of the 7 categories described above. In case the luxury value was missing for any category, they used the average luxury value for the rest of the categories. This luxury information was then concatenated with the other attributes of the house to create a vector that captured the impact of both the visual and numeric data. The metadata for each house also included its Zestimate value [4] as shown on Zillow. The aggregate vector was then passed to a Support Vector Regressor to obtain the estimated price for the property. The loss during training was computed using the actual purchase price of the property as the ground truth. Figure 3 shows the price estimation network used for the experiment.

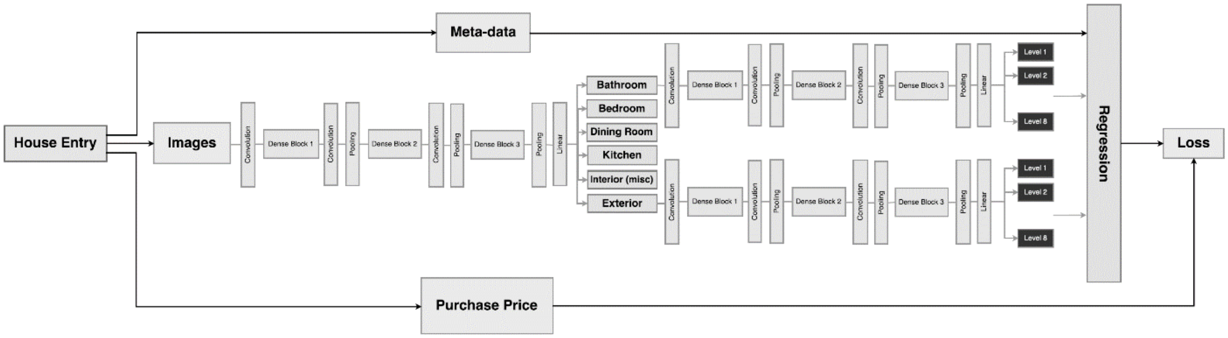


Figure 3: The price estimation network used in [9]

The authors also conducted this experiment without considering the image data for predicting property prices. The median error rate without taking the visual data into account was 8% which was reduced to 5.6% when the visual characteristics of the house were included along with other features. Thus, this experiment concludes that the appearance of a house plays an important role in determining its price. Furthermore, it gives a novel way to quantify the visual information included within the images associated with each house by assigning a score to each of the images.

Zhao, Chetty, and Tran [12] proposed a framework to aesthetically evaluate the images associated with each property. For this task, they used the AVA dataset [13] which consists of images from different categories, each labeled with an aesthetic score from 1 to 10. The scores are obtained via a crowdsourcing approach by averaging the scores received from the participants. They randomly selected 4 images for each of the houses, cropped them to 122 x 122 pixels, and then stitched these images together to form a resultant image of size 244 x 244 pixels. A MobileNet [14] (Softmax activation function) with pre-trained weights based trained on the AVA dataset was used to evaluate the 4 property images and assign a score to each of them. Following this, a Convolutional Neural Network (CNN) [23] model with ReLU activation function was used to extract visual features from the stitched image. The model takes in the normalized dataset as input where the values are scale to be in the range 0 to 1. The numeric data is analyzed using a Multilayer Perceptron (MLP) model with ReLU activation function. It is then combined with the outputs of the two modules mentioned above and then passed to an XGBoost [15] Regressor to arrive at an estimate for the price of the property. Figure 4 shows the model used in [12] for the appraisal of properties.

The resulting model obtained a MAPE score of 8.70% using the proposed model. The authors also tested the performance of the model without considering the visual attributes of the properties. The MAPE score was 10.09% in this case indicating that considering aesthetic scores in estimating real estate prices leads to an improvement in performance. Additionally, using CNNs for extracting features from images can eliminate the need to perform manual feature engineering for modeling visual data.

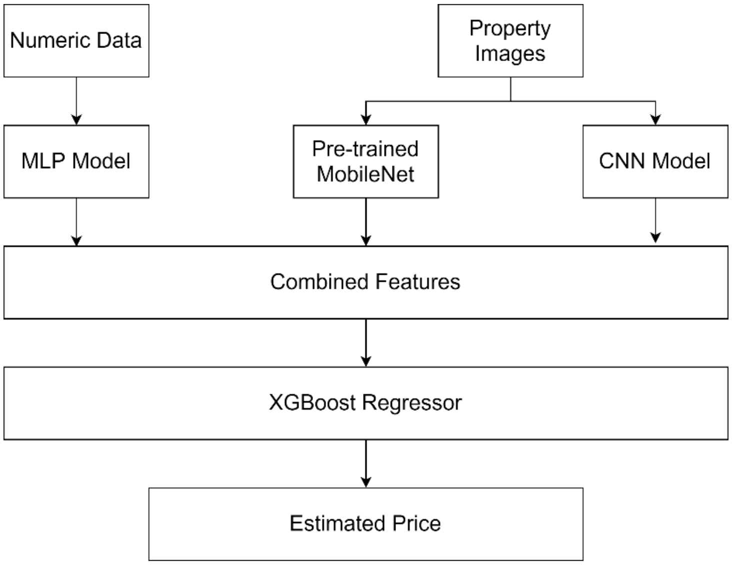


Figure 4: Model proposed in [12] for estimating real estate prices

B. Using Image Data To Cluster Similar Houses

You et al. [16] proposed a novel method to estimate prices of properties by considering only the visual data without considering the numeric features associated with the properties in two cities: San Jose and Rochester. To begin with, they recorded the coordinate of each house in the database using the Bing Map API. These coordinates were then used to compute the distance between each pair of houses using Vincenty's formulae [17]. This information was used to build an undirected graph for all the houses, where each node v_i represents the ith house. The weight of the edge e_{ij} between any two houses h_i and h_j is equal to the similarity score s_{ij} calculated as:

$$s_{ij} = \exp\left(\frac{dist(h_i, h_j)}{2\sigma^2}\right)$$

where $dist(h_i, h_j)$ is the geodesic distance between houses h_i and h_j. The parameter σ controls the decrease in similarity with an increase in distance.

Once the directed graph was constructed, a random walk (inspired by DeepWalk [18]) was employed through the graph to generate a sequence of houses. To do this, a node was selected at random and added to the sequence. The next node to be added to the sequence was selected from the neighboring nodes v_j according to the formula:

$$p_j = \frac{e_{ji}}{\sum_{k \in N(i)} e_{ki}}$$

where *N(i)* is the set of neighbor nodes of v_i. This process was continued until a sequence of desired length L is generated. Since the sequence is constructed by considering the locations of the houses, it can be said that the prices of houses in the same sequence are close to each other.

The authors proposed a Bidirectional Recurrent Neural Network [19], particularly Bidirectional-LSTM (B-LSTM), to estimate the property prices. They used two B-LSTM layers in the network and the output of the second layer was passed as input to the output layer which gave the predicted price of the house. For testing, each house was added as a new node in the previously built undirected graph. Using the new graph, several sequences were constructed which contained only one testing house. The final price of the house was determined to be the average of the outputs from all the sequences. Figure 5 shows how a testing sequence was constructed for each house.

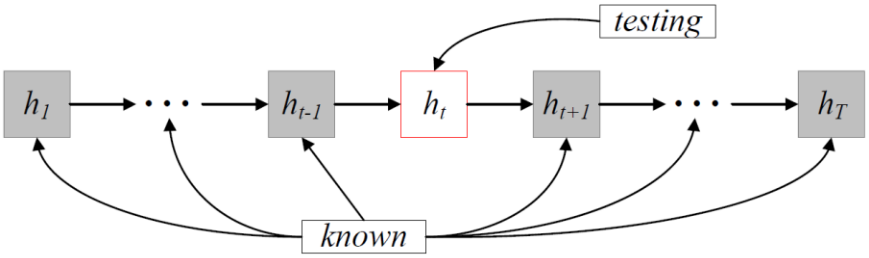


Figure 5: Building a testing sequence for a house

Upon evaluating the performance of the model, the MAPE score for the average predicted price was 16.11% for San Jose and 22.69% for Rochester. The MAPE scores are comparatively worse than the other models because the authors only use image data and do not consider any other information related to the house. However, this experiment shows that the location and visual features associated with the properties are important factors for estimating their prices.

C. Using Image Data To Remove Duplicates From The Dataset

Niu et al. [20] utilized the images associated with the properties to improve the data quality by eliminating duplicates from the dataset. The authors collected the following data for each house: caption of the listing, house address, its description, area, floor, price, house type, housing pictures, and listing time. They proposed a Repeated House Recognition Module which can identify repeated listings within the data. For this task, they trained a CNN to extract features from each of the images for a given house. Thus, any redundant listings can be detected and removed from the data. Following this, they proposed a Feature Extraction and Qualification Module where the original data was classified into three types of features: community-level, building-level, and house-level. Once the data was preprocessed, it was fed to each of the following three models: Random Forest (RF) [21], Gradient Boosted Decision Tree (GBDT), and Backpropagation Neural Network (BPNN) [22]. The final price was computed by weighted voting, an ensemble learning [24] technique to enhance the accuracy of the overall prediction. Figure 6 shows the overall architecture of the estimation technique proposed in [20].

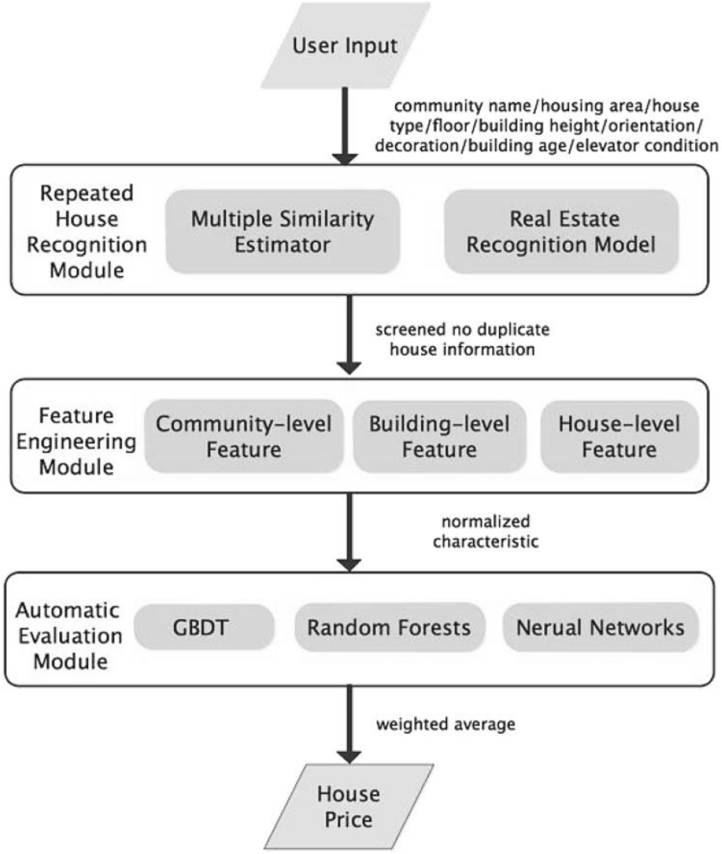

User Input
community name/housing area/house type/floor/building height/orientation/ decoration/building age/elevator condition
Repeated House Recognition Module
Multiple Similarity Estimator
Real Estate Recognition Model
screened no duplicate house information
Feature Engineering Module
Community-level Feature
Building-level Feature
House-level Feature
normalized characteristic
Automatic Evaluation Module
GBDT
Random Forests
Nerual Networks
weighted average
House Price

Figure 6: A proposed architecture for real estate appraisal in [20]

The experimental results showed that the Mean Squared Error scores for the GBDT, RF, and BPNN models were 7.58, 8.36, and 6.72, respectively. However, the score significantly reduced to 5.42 when an ensemble of the three models was used to estimate the price. It can be concluded that the accuracy of the prediction is greatly improved when an ensemble is used instead of the individual models. Also, this experiment employed a novel way to clean the data by using the visual data for finding repeated entries in the dataset thereby enhancing the data.

In conclusion, the recent developments in estimating real estate prices can be summarized as follows:

Table 1: Summary of literature review

Name	**Dataset**	**Algorithms Employed**	**Metric and Performance**
Ahmed et al. [6]	Set of 4 images (frontal view, bedroom, bathroom, and kitchen) for each house	SURF for feature extraction Support Vector Regression (SVR) and Neural Network (NN) for estimating prices	R-value SVR: 0.78602 NN: 0.95053
Poursaeed et al. [9]	Crowdsourced data with a luxury level assigned to each image	DenseNet for image classification and evaluation Support Vector Regression for estimating prices	Median error: 5.6%
Zhao et al. [12]	Property images from Allhomes (www.allhomes.com.au)	MobileNet for image evaluation CNN for feature extraction XGBoost for estimating prices	Mean Absolute Percentage error: 8.70%
You et al. [16]	Property images (San Jose and Rochester) from Realtor (www.realtor.com)	Bidirectional-LSTM (B-LSTM)	Mean Absolute Percentage Error San Jose: 16.11% Rochester: 22.69%
Niu et al. [20]	Property data of houses in Xihu district, Hangzhou, China	CNN for feature extraction An ensemble of Random Forest (RF), Gradient Boosted Decision Tree (GBDT), and Backpropagation Neural Network (BPNN) for estimating prices	Mean Squared Error GBDT: 7.58 RF: 8.36 BPNN: 6.72 Ensemble: 5.42

III. TOOLS AND TECHNOLOGIES USED

A. Development Environment

Python 3 was used as the programming language for the implementation of the entire project. It is an open-source language and has an extensive set of libraries and developmental tools to ease the process of processing the data. It also has a wide variety of visualization tools that can be used for exploring and analyzing the input data. Additionally, it is platform-independent which allows users to run the code written on one machine across different environments and operating systems.

The code was primarily written using the Integrated Development Environment, PyCharm. It offers a wide range of features like code completion, error highlighting, debugging support, and an interactive Python console making it easier to write programs as compared to a standard text editor.

B. Libraries Used

Python has a rich and growing set of libraries that eliminate the need for writing code for commonly performed tasks, thus reducing the time to implement a project. The libraries used for this project are Scrapy, NumPy, Pandas, Scikit-learn, TensorFlow, Matplotlib, and Seaborn [47]. Scrapy is an open-source and portable framework used for web-crawling that can be used to extract data of interest from a specific domain. NumPy provides support for fast calculations on multidimensional arrays and matrices, the most common data structures used in machine learning computations. Pandas is another widely used library built on NumPy that supports data analysis and processing by providing easy-to-use tabular structures. Scikit-learn is an open-source library that provides tools for generating end-to-end pipelines for various machine learning tasks including regression, classification, and clustering. It also includes support for various preprocessing tasks such as dimensionality reduction, feature extraction, and normalization. TensorFlow is an end-to-end library mainly used for training multilayer deep learning models while leveraging the computing power of CPUs and GPUs. Matplotlib and Seaborn are visualization libraries used to plot data and create statistical graphics using Python and NumPy objects.

C. Cloud Technologies Used

a) Amazon Elastic Compute Cloud (EC2)

Amazon EC2 is a service provided as a part of Amazon Web Services (AWS). It provides computing resources to developers using Amazon's environment allowing them to obtain access to computing power and configure it according to the use case. These virtual instances have the advantage of automatic load balancing, fault tolerance, and high availability. More information about Amazon EC2 can be found at *https://aws.amazon.com/ec2/*.

EC2 machines were used for the process of data collection to create the dataset for the project. Each instance of EC2 had a web crawler running that extracted and stored the desired property listings from Zillow. This facilitated extracting multiple listings simultaneously using three EC2 instances as there was a limit on the rate at which data could be extracted using a single machine.

b) Docker

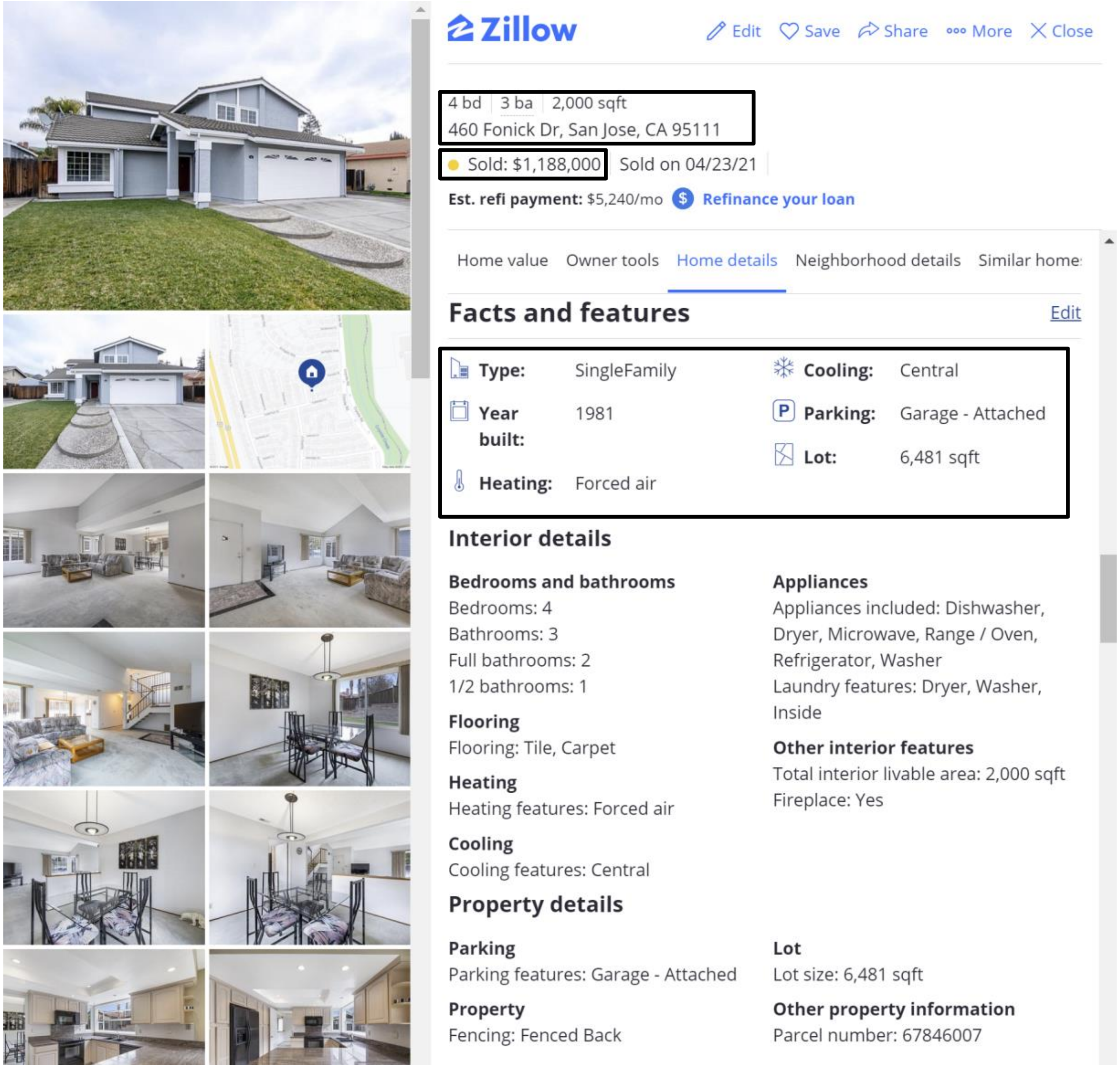
Docker is an open-source platform that simplifies the process of developing, deploying, and running applications. Docker allows users to bundle and run applications in a lightweight environment called a container which has everything needed to run the application. Developers can also have different copies of their applications for testing and debugging in an isolated setting. After making necessary changes to the application, it is required to only push a new image to reflect them into production.

In this project, the weights of the CNN after training were stored in a separate file. A docker image was then created to evaluate and score property images by using this weights file. Thus, the pictures associated with a particular property could be evaluated in a single operation using Docker which greatly simplified the process of assessing image data.

IV. Dataset

A. Housing Data

Nowadays, a lot of people rely on online real estate database websites like Zillow [1], Redfin [2], and Trulia [3] to browse properties according to their needs. Along with the numerous property listings for sale and rent, these websites also contain the data for recently sold properties. This project uses the listings corresponding to recently sold properties from Zillow as the source of data.



4 bd | 3 ba | 2,000 sqft
460 Fonick Dr, San Jose, CA 95111

● Sold: $1,188,000 | Sold on 04/23/21

Est. refi payment: $5,240/mo

Facts and features

Type:	SingleFamily	**Cooling:**	Central
Year built:	1981	**Parking:**	Garage - Attached
		Lot:	6,481 sqft
Heating:	Forced air		

Interior details

Bedrooms and bathrooms
Bedrooms: 4
Bathrooms: 3
Full bathrooms: 2
1/2 bathrooms: 1

Flooring
Flooring: Tile, Carpet

Heating
Heating features: Forced air

Cooling
Cooling features: Central

Appliances
Appliances included: Dishwasher, Dryer, Microwave, Range / Oven, Refrigerator, Washer
Laundry features: Dryer, Washer, Inside

Other interior features
Total interior livable area: 2,000 sqft
Fireplace: Yes

Property details

Parking
Parking features: Garage - Attached

Property
Fencing: Fenced Back

Lot
Lot size: 6,481 sqft

Other property information
Parcel number: 67846007

Figure 7: An example of a property listing on Zillow [43]

Figure 7 shows an example of a real estate listing on Zillow. A typical property listing includes the details about the house such as the number of bedrooms and bathrooms, living area, address, and other features of the house like its age, parking, type of heating and cooling, and the lot size. Along with this, it also includes the pictures of the property which are of interest to potential buyers to get a general idea about the look and infrastructure of the house.

The dataset used for this project consists of 728 properties listed on Zillow that were sold in San Jose. A web crawler was used to fetch each of these property listings and extract desired data from the response. Here is an example data obtained from scraping a property listing [44]:

```
{
    "date":"03/19/21",
    "bedrooms":3,
    "bathrooms":2,
    "area":1197,
    "address":"3331 Senter Rd, San Jose, CA 95111",
    "facts":{
        "Type":"Townhouse",
        "Year built":"1971",
        "Heating":"Forced air, Gas",
        "Cooling":"None",
        "Parking":"Carport, Garage - Attached",
        "HOA":"$363 monthly",
        "Lot":"1,494 sqft"
    },
    "nearby_schools":{
        "9-12":"0.1 mi",
        "4-8":"0.5 mi",
        "K-3":"0.3 mi"
    },
    "price":560000,
    "images":[
        "https://photos.zillowstatic.com/fp/86ab5f62c163cb9db98086958790ea19-p_h.jpg",
        "https://photos.zillowstatic.com/fp/01af1b5b419f09c5b07351fd8752835c-p_h.jpg",
        "https://photos.zillowstatic.com/fp/b31d587f023ba2f8f0a6c63fe53e1a4f-p_h.jpg",
        "https://photos.zillowstatic.com/fp/685306eb8c38a4c24f17e9dbf10a52ba-p_h.jpg",
        "https://photos.zillowstatic.com/fp/cdea9669a9e2419ae7e480c9fc23c1a9-p_h.jpg",
        "https://photos.zillowstatic.com/fp/2bf4005e869aefab2e76efdd6f432c13-p_h.jpg",
        "https://photos.zillowstatic.com/fp/e488b2f3a446673572d2c2fc4e604d92-p_h.jpg",
        "https://photos.zillowstatic.com/fp/cf48b4e1c5049564dcd6956d134dafac-p_h.jpg",
        "https://photos.zillowstatic.com/fp/ca6b2447b4185f48f4a43493186930e2-p_h.jpg",
        "https://photos.zillowstatic.com/fp/110c73105c3a1956447ea78ea8caf3fb-p_h.jpg",
        "https://photos.zillowstatic.com/fp/82dd960c7562eca1c3f9180b9ab1d1e2-p_h.jpg",
        "https://photos.zillowstatic.com/fp/5efe4592255bacc44319b133804c362a-p_h.jpg",
        "https://photos.zillowstatic.com/fp/d2aebf1dcee28dc9acf9afb361f2e44b-p_h.jpg",
        "https://photos.zillowstatic.com/fp/41639dd60061be240bb2c1b3a462d1ec-p_h.jpg",
        "https://photos.zillowstatic.com/fp/005b6c96cbf5472bf7c76de6eeecaa2b-p_h.jpg",
        "https://photos.zillowstatic.com/fp/93eb08daccf9592f99f7830a47e9413f-p_h.jpg",
        "https://photos.zillowstatic.com/fp/41234dfdfe8d8a6769c1bb435e141f9f-p_h.jpg",
        "https://photos.zillowstatic.com/fp/b52231fbd21736f53a0de244835a0399-p_h.jpg",
        "https://photos.zillowstatic.com/fp/5d6998cb12ecd987d15a7d6b61012e1d-p_h.jpg",
        "https://photos.zillowstatic.com/fp/d749ebcb3775aaee9139cff8ce3484c6-p_h.jpg"
    ]
}
```

The above data in JSON format contains the information about a house along with a list of images for each house. To predict the final price of a house, the images need to be evaluated and scored so that they can be used in a machine learning model for estimating property prices. This process was done using a CNN for image evaluation and the input data to the Price Estimation Model consisted of only numerical and categorical values. Here is an example data after evaluating the images and processing the data:

```
 {
   "bedrooms":3,
   "bathrooms":2,
   "area":1197,
   "score":4.732561728126017,
   "zip_code":95111,
   "house_type":"Townhouse",
   "age":50,
   "lot_size":1494,
   "heating":{
      "Forced air":1,
      "Gas":1,
      "Wall":0,
      "Wood / Pellet":0,
      "Electric":0,
      "Solar":0,
      "Heat pump":0,
      "Radiant":0,
      "Baseboard":0,
      "Other":0
   },
   "cooling":{
      "Central":0,
      "Wall":0,
      "Refrigerator":0,
      "Solar":0,
      "Other":0
   },
   "parking":{
      "Garage":1,
      "Garage - Attached":1,
      "Garage - Detached":0,
      "Carport":1,
      "Covered":0,
      "On-street":0,
      "Off-street":0
   },
   "school":{
      "high":0.1,
      "middle":0.5,
      "elementary":0.3
   },
   "price":560000
}
```

The score attribute consists of the image evaluation score and the non-numeric attributes of the property were converted into categorical ones. This data can be employed to estimate property prices using a regression model.

B. Data For Training The Image Evaluation Model

The Image Evaluation Model was trained using the AVA dataset [13] which is a large-scale dataset used for aesthetically analyzing images. This dataset consists of 255,000 images spanning across various categories like architecture, cityscape, and landscape. These images are rated by various photographers on a scale of 1 to 10 with an average of 210 votes per image. The score of an image is directly proportional to its aesthetic judgment by the photographers. These ratings are given in the AVA dataset as labels for each of the images as follows:

```
27859 8873 24 23 40 64 46 28 9 7 2 1 0 0 44
```

Here, the first field represents the serial number, and the second field represents the image ID. Fields 3 through 12 represent the frequency distribution of the aesthetic ratings from 1 to 10. Field 3 represents the number of ratings corresponding to a score of 1 and field 12 represents the number of ratings corresponding to a score of 10.

To use these ratings for aesthetically evaluating images, the fields were converted to a JSON format, that mapped the image ID to the corresponding aesthetic ratings. Here is an example of such a conversion:

```
{
    "image_id":"8873",
    "label":[
        24,
        23,
        40,
        64,
        46,
        28,
        9,
        7,
        2,
        1
    ]
}
```

V. Data Analysis And Preprocessing

A. Exploratory Data Analysis

To get an insight into the data, each variable was inspected to understand its relationship and relevance with respect to the price attribute. To begin with, some statistical details related to the attributes of the properties were observed as seen in Tables 2 and 3.

Table 2: Statistical observations about the data

Statistic	Mean	Standard Deviation	Minimum	Maximum
House Price (USD)	1221733.324	634559.021	2650.000	4150000.000
House area (sq. ft.)	1596.503	659.242	543.000	4862.000
Number of bedrooms	3.088	1.043	1.000	6.000
Number of bathrooms	2.305	0.822	1.000	6.000
Image Score	4.940	0.256	3.386	5.437

Table 3: Statistical observations about the data (continued)

Statistic	25%	50%	75%
House Price (USD)	780000.000	1123500.000	1511500.000
House area (sq. ft.)	1194.000	1468.000	1818.000
Number of bedrooms	2.000	3.000	4.000
Number of bathrooms	2.000	2.000	3.000
Image Score	4.845	4.990	5.091

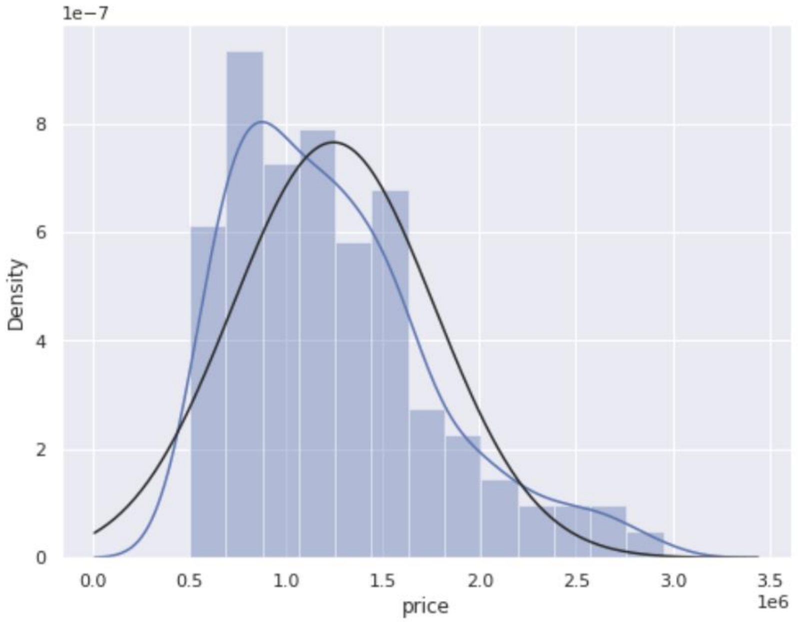


Figure 8: Distribution of selling prices of properties

Figure 8 shows the univariate distribution of the selling prices of the properties in the dataset. It can be seen that the distribution is positively skewed with skewness of 1.267802.

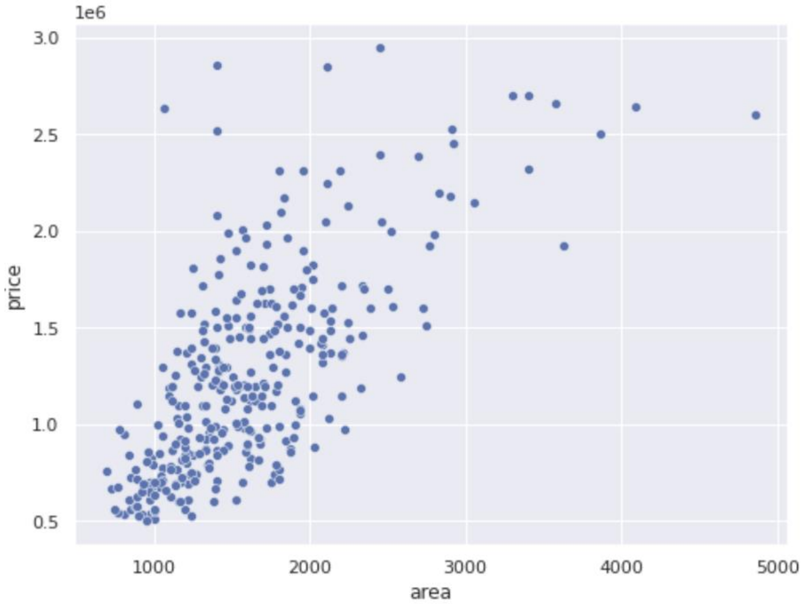


Figure 9: Relationship between area and price of a property

Figure 9 shows the relationship between the area and price of the properties in the dataset. It can be seen that the relationship is linear which is intuitive and confirms the reliability of the data.

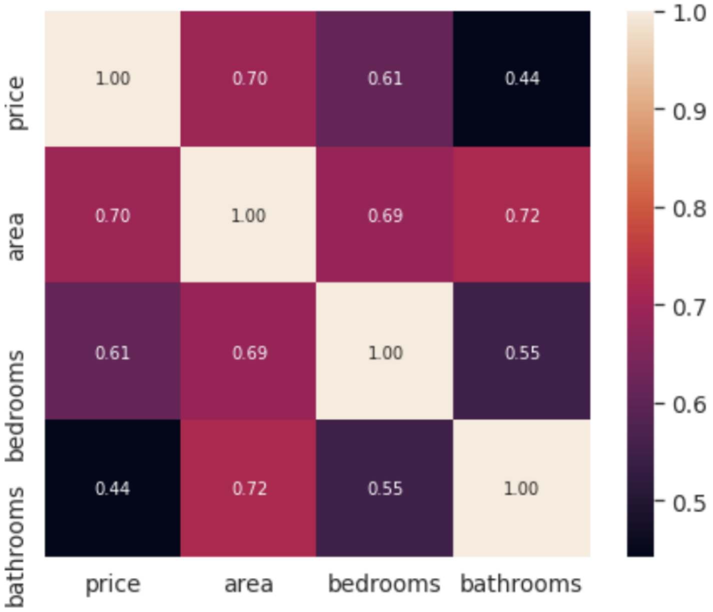


Figure 10: Correlation between price and other attributes

Figure 10 shows the correlation between the price of a property and the three attributes having the largest correlation with the price. It is observed that the area of a property is the most strongly correlated variable with its price followed by the number of bedrooms and bathrooms.

B. Handling Missing Data

Some of the observations in the data may have missing values for some attributes. Missing data can lead to incorrect predictions and thus needs to be handled before proceeding ahead. Two common approaches for handling missing data are imputation of data and removal of data [29]. The imputation method is generally preferred when the number of missing values for a particular attribute is low and involves making a reasonable guess to fill in the missing values. If the number of missing values is high, the imputation process can lead to an ineffective model since the observations would lack the inherent variation in the data. The approach of deleting the corresponding attribute is preferred in such cases.

Table 4: Percentage of missing values

Attributes	Percentage of Missing Values
lot_size	25.824176
school.elementary	0.549451
school.middle	0.549451
age	0.274725
school.high	0.274725

Table 4 shows the attributes with missing values and the corresponding percentages. Since the number of missing values for the attribute 'lot_size' is too high, filling in values for the missing observations would lead to inaccurate results. Hence, this attribute was dropped and was not considered in estimating the property prices. The percentage of missing values for the other attributes is very small. Hence, the corresponding observations were completely deleted without highly impacting the cardinality of the dataset.

C. Data Preprocessing

To apply multivariate techniques, the attributes of the dataset must comply with some statistical assumptions. Hair et al. [28] defines such assumptions that should be tested:

- Normality: It means that the data should be present as a normal distribution. Ensuring normality in the data can help prevent other problems such as heteroscedasticity.
- Homoscedasticity: It refers to having equal levels of variance within dependent variables as present in the independent variables.

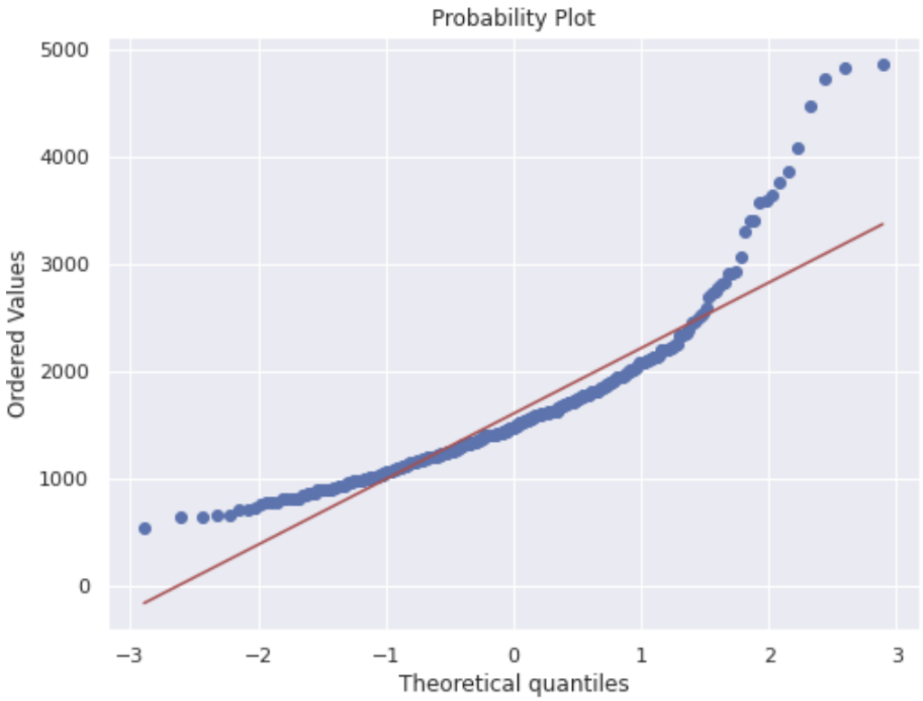


Figure 11: Normal probability plot for price

It was seen in Fig. 8 that the price attribute had positive skewness. In a probability plot, the data should follow the diagonal representing the normal distribution. It can be seen from Figure 11 that the data varies significantly from the diagonal line. According to [28], this can be addressed using a log transformation in case of positive skewness.

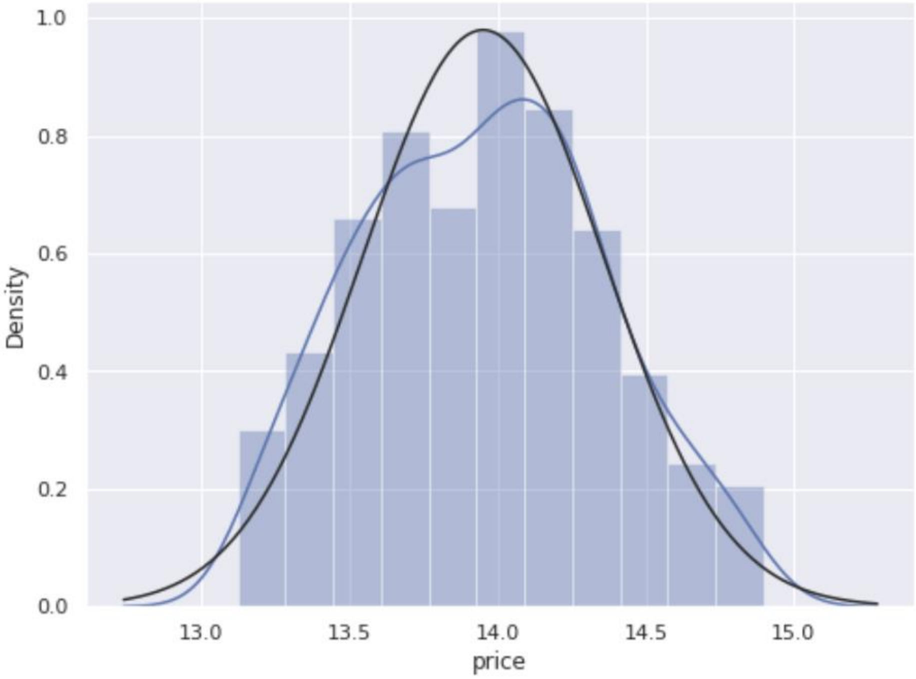


Figure 12: Distribution of selling prices of properties (after transformation)

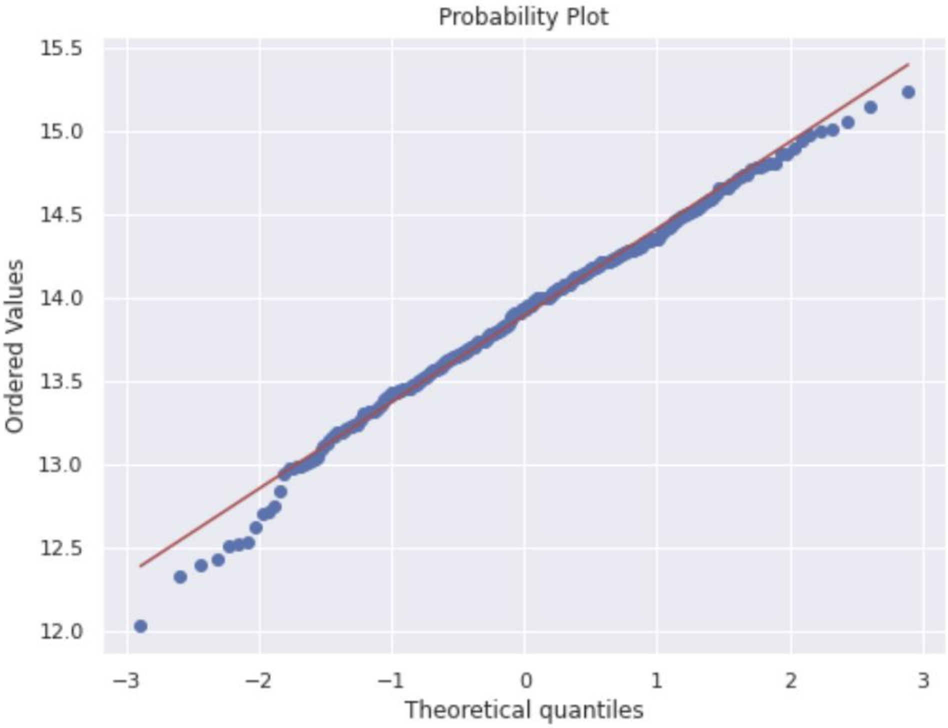


Figure 13: Normal probability plot for the price (after transformation)

It can be seen from Figure 12 and Figure 13 that the price attribute after transformation shows a greater degree of normality than before.

A similar observation was made in the case of the area attribute of the properties in the dataset.

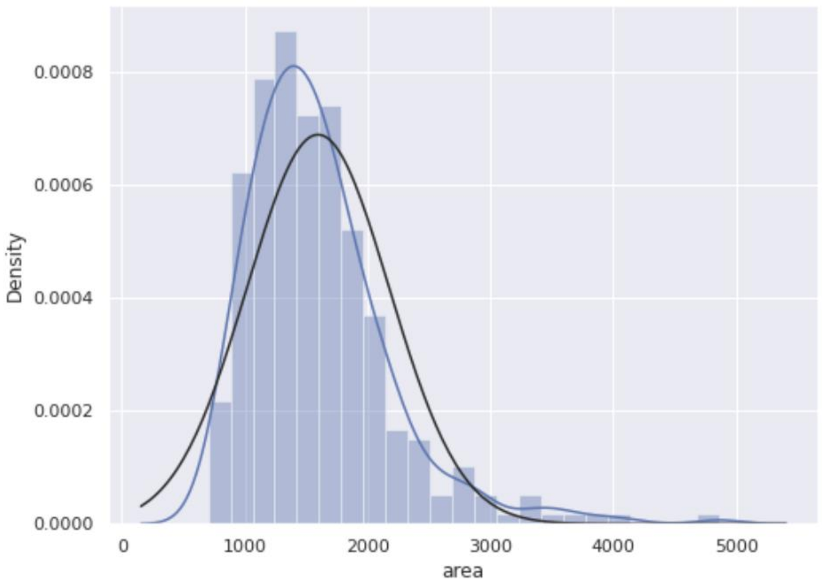


Figure 14: Distribution of areas of properties

Figure 14 shows the univariate distribution of the areas of the properties in the dataset. It can be seen that the distribution is positively skewed with skewness of 1.964933.

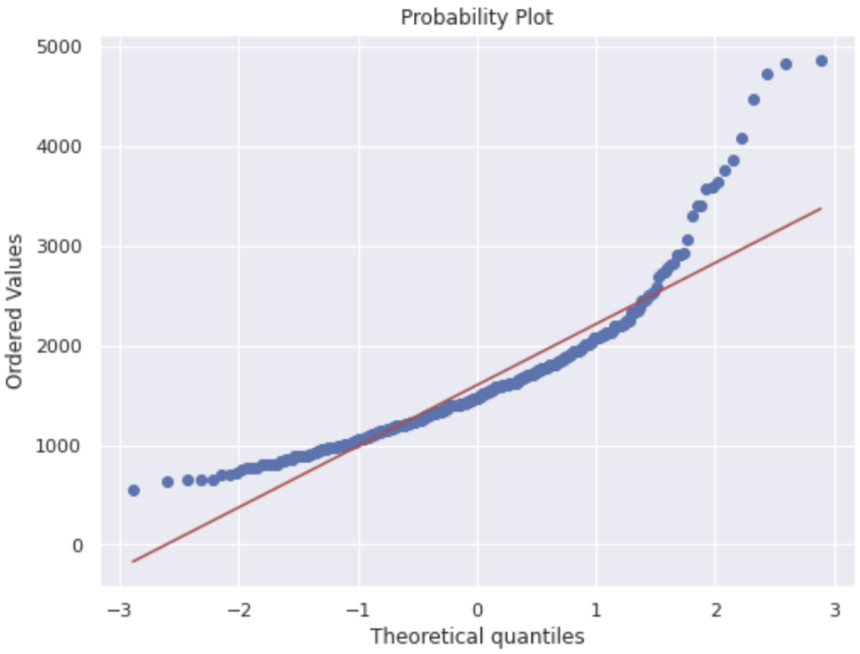


Figure 15: Normal probability plot for area

Also, the data points do not follow the diagonal representing the normal distribution as shown in Figure 15.

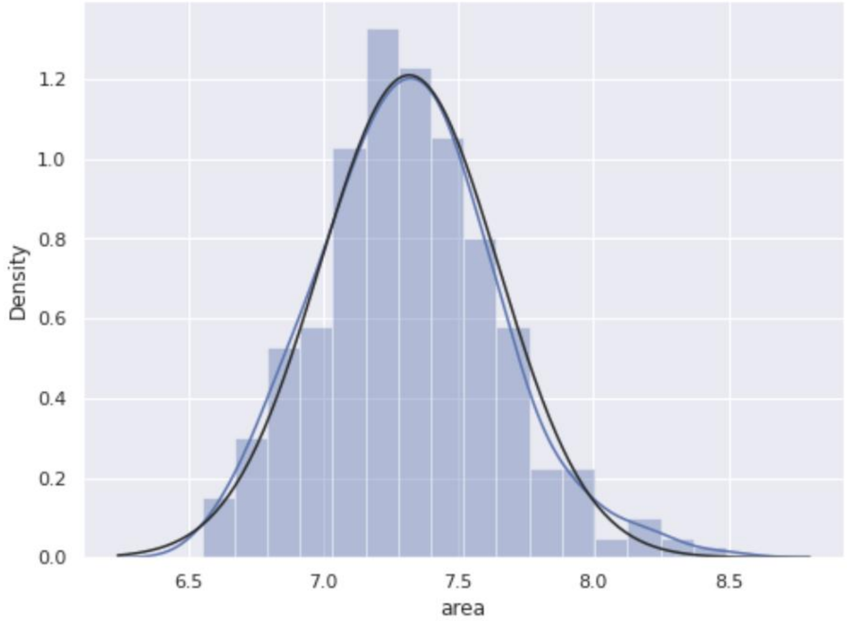


Figure 16: Distribution of areas of properties (after transformation)

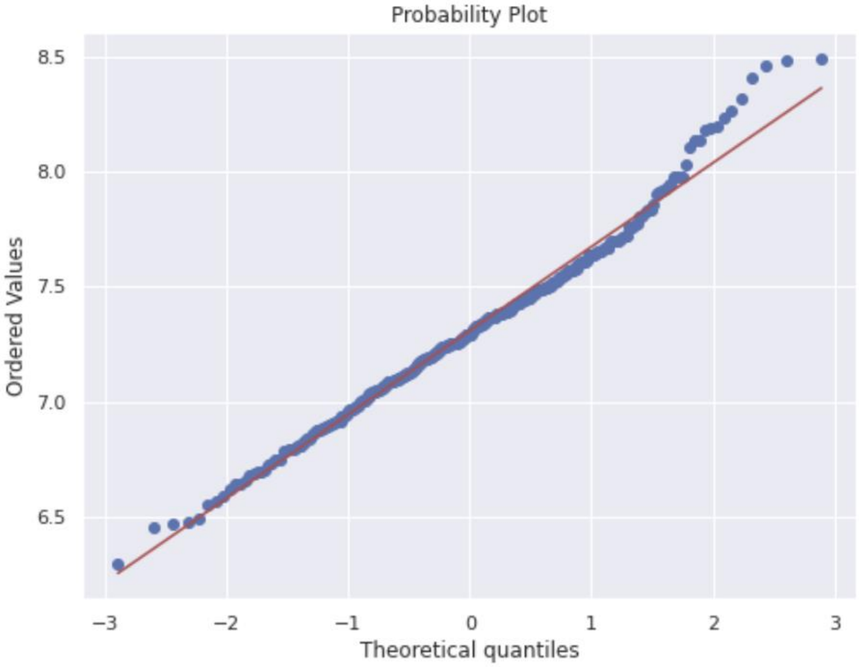


Figure 17: Normal probability plot for the area (after transformation)

Similar to the price attribute, this was also addressed by applying a log transformation to the area attribute. It can be seen from Figure 16 and Figure 17 that the price attribute after transformation shows a greater degree of normality than before.

As seen in Table 2, the standard deviation for the score attribute is low. It means that the feature would not contribute much to the estimation model. In order to handle this, the score feature was scaled to a range of 100 from 10 to increase the variance. To do this, the attribute was first normalized using min-max feature scaling which uses the formula:

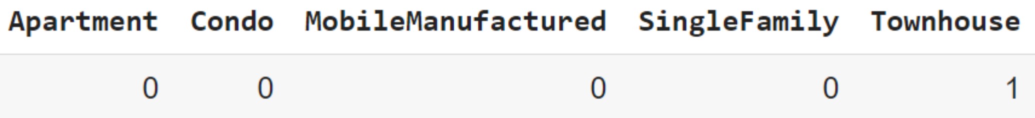
$$x' = \frac{x - \min(x)}{\max(x) - \min(x)} \quad [46]$$

After this transformation, each value in the score attribute was multiplied by 100 to scale it and in turn, increase the variance.

D. Handling Categorical Attributes

The dataset has categorical features like house type, parking type, and heating and cooling type. Categorical features usually have a limited number of values possible for each feature. These features need to be converted into numerical values so that they can be processed by a mathematical model. To perform this task, a technique called one-hot encoding was used. In this technique, the categorical feature is represented as a vector whose length is equal to the number of categories within that feature. The vector has a value of 1 corresponding to the category represented in the observation and a value of 0 for other categories [30].

For example, in the dataset, a property can be of one of the following types: Apartment, Condo, MobileManufactured, SingleFamily, or Townhouse. A house of type Townhouse would have the following vector representation as shown in Figure 18.



Apartment	Condo	MobileManufactured	SingleFamily	Townhouse
0	0	0	0	1

Figure 18: Example of one-hot encoding for the ‘house type’ feature

VI. IMAGE EVALUATION MODEL

A. Convolutional Neural Network (CNN)

Convolutional Neural Networks (CNNs) are a type of neural networks that work well when the input data has a grid-like structure [39]. These networks have been successful in various tasks in the field of computer vision. A CNN is essentially a neural network that makes use of an operation called convolution in at least one of the layers of the network instead of the typical matrix multiplication operation [39].

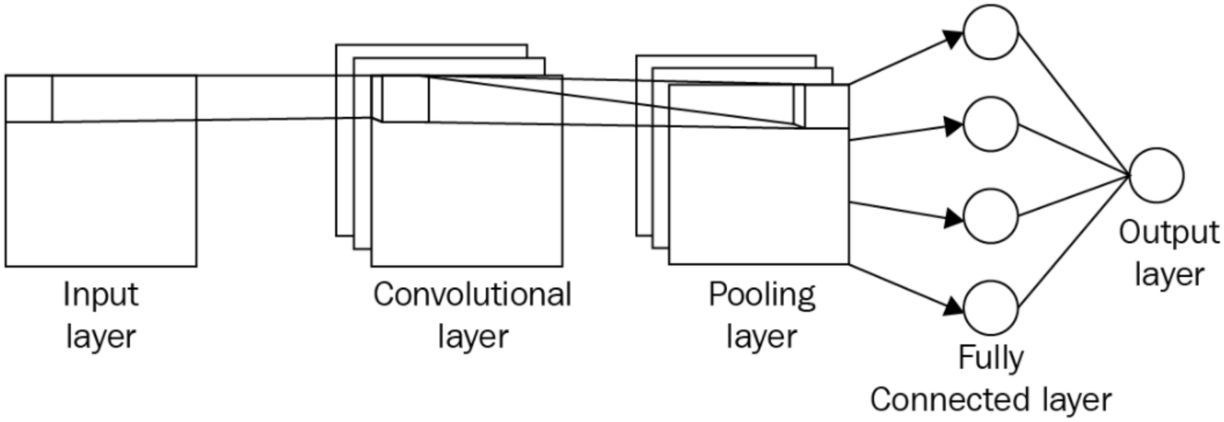


Figure 19: Architecture of a CNN [40]

Figure 19 shows the architecture of a typical CNN. It consists of three types of layers: convolution layer, pooling layer, and fully connected layer [40].

A convolution operation is performed between two parameters: the input data and the set of parameters to be learned, called the kernel. For 2-D images, the operation is usually a dot product between the two matrices where only a restricted part of the input data is considered. The result of the convolution operation is referred to as the feature map or the activation map. Figure 20 shows an example of a 2-D convolution operation performed on the given input and kernel.

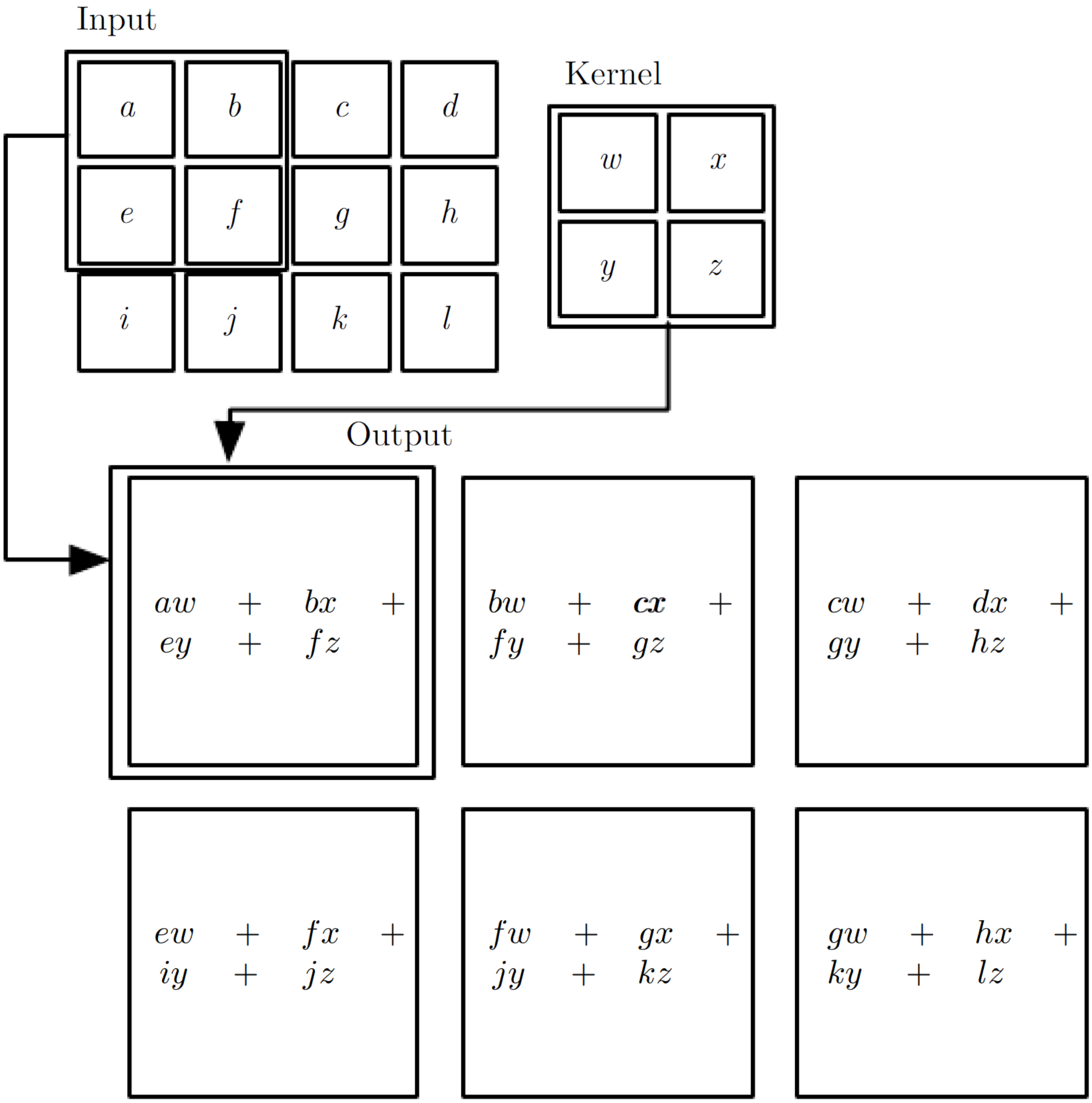


Figure 20: An example of a 2-D convolution [39]

A pooling layer reduces the size of the input image by employing a technique called sampling. It selects the maximum value in the window (in case of max sampling) or takes an average of all the values in the window (in case of average sampling). This has the advantage of avoiding overfitting since the sampling operation can be seen as a regularization technique [41].

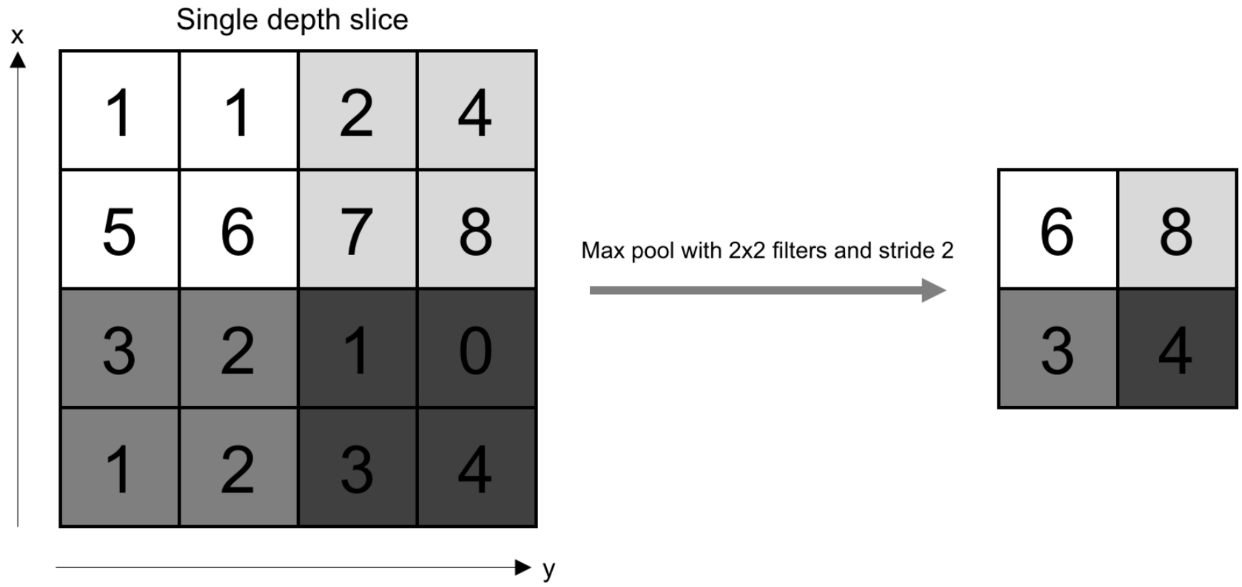


Figure 21: Pooling operation [41]

Figure 21 shows an example of max pooling with 2x2 filters and a stride of 2. The stride is the amount by which the filter moves along the input to produce the result.

The neurons in the fully connected layer have links or edges to each neuron in the layer preceding it. The operation performed in this layer is a matrix multiplication to establish a mapping between the input and the output.

B. MobileNet

According to Denil et al. [42], the computation for finding the parameters of a CNN has a lot of redundancy and this overhead only contributes to a slight improvement in the accuracy. MobileNet [14] is a network that allows for faster training as compared to a CNN by reducing the number of parameters to be estimated and eliminating redundant computations.

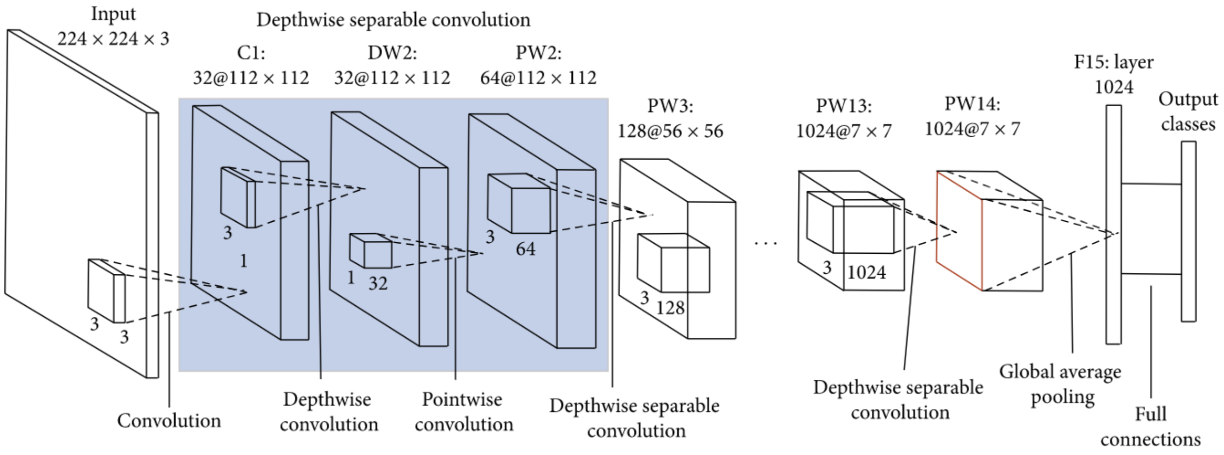


Figure 22: Architecture of MobileNet [38]

As seen in Figure 22, the architecture of MobileNet is based on the idea of Depthwise separable convolution [38]. It consists of using two layers: a depthwise convolution layer and a pointwise convolution layer [38]. For each channel in the input (depth of the input), the depthwise convolution layer applies a single convolution (filter). The output of this operation is then combined by the pointwise convolution layer using a simple 1 x 1 convolution [14].

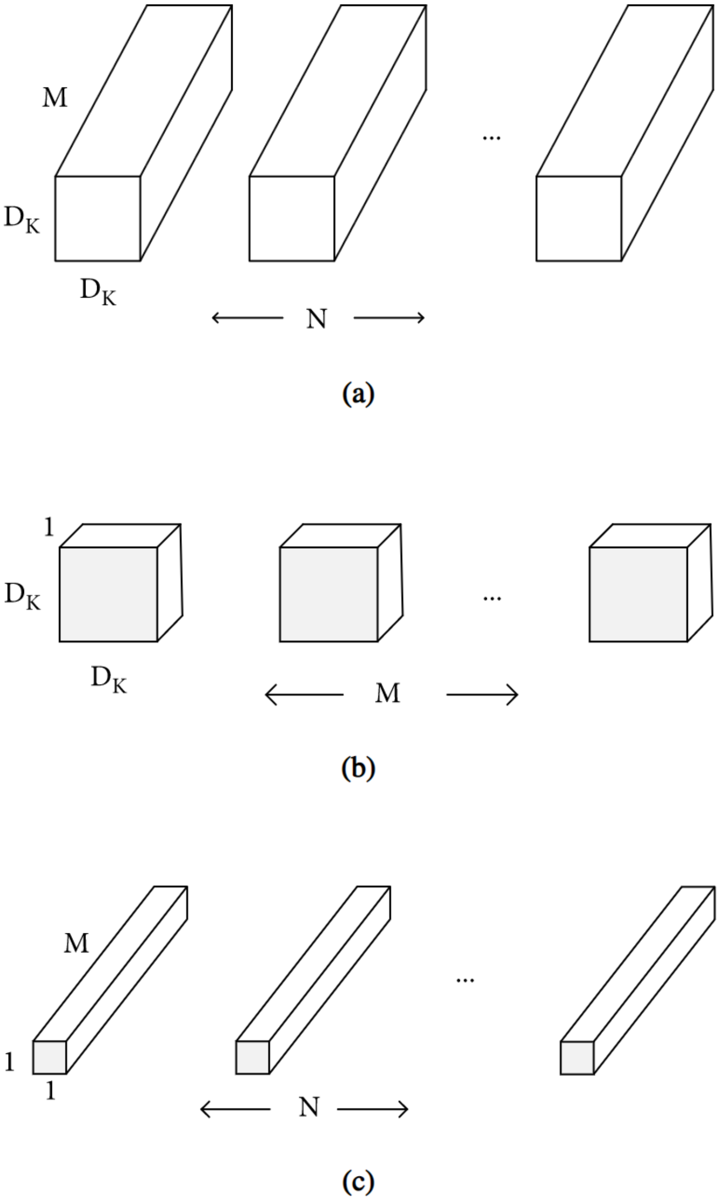


Figure 23: (a) standard convolution, (b) depthwise convolution, and (c) pointwise convolution [38]

Figure 23 shows the working of standard, depthwise, and pointwise convolution. A depthwise separable convolution is the output obtained after applying a depthwise and a pointwise convolution consecutively [14].

C. Training The Image Evaluation Model

For evaluating property images, a classifier based on NIMA [32] was used to estimate the aesthetic quality of images. NIMA uses the earth mover's distance [33] as the loss function which measures the distance between two probability distributions. Intuitively, it can be thought of as the amount of dirt (earth) that needs to be added on top of another pile of dirt to make both of them equal. Using the earth mover's distance as the loss function has the advantage that it can capture the ordering between classes. Specifically, if an observation has a predicted score of 3 when the true score 10, it needs to be punished more than when the true score is 4 [31].

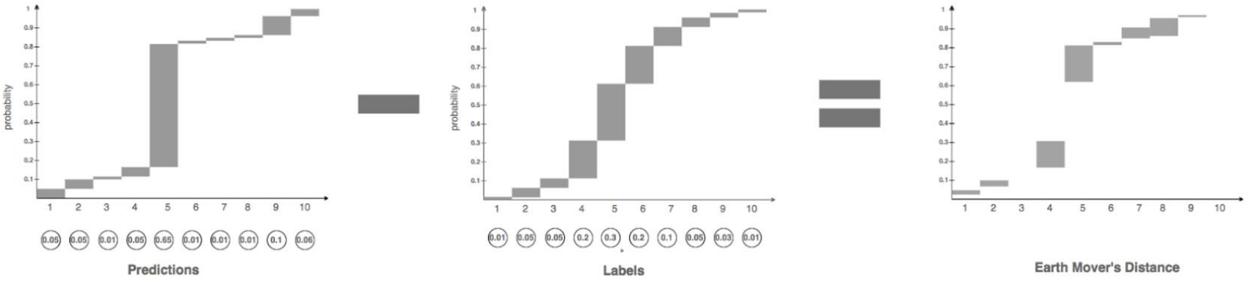


Figure 24: An example of Earth mover's distance [31]

Figure 24 shows an example calculation of the earth mover's distance. The NIMA image classifier uses a MobileNet [14] architecture with pre-trained ImageNet [37] weights for predicting image quality. The last layer of the MobileNet architecture was replaced by a fully connected dense layer that had 10 classes as output representing scores from 1 to 10. To use the earth mover's distance, NIMA requires a frequency distribution of image scores across each of the values in the range. This distribution was already available within the AVA dataset [13] as labels for each of the images. These labels corresponding to the ratings were then converted into a probability distribution that was used as the labels for training as shown in Figure 25.

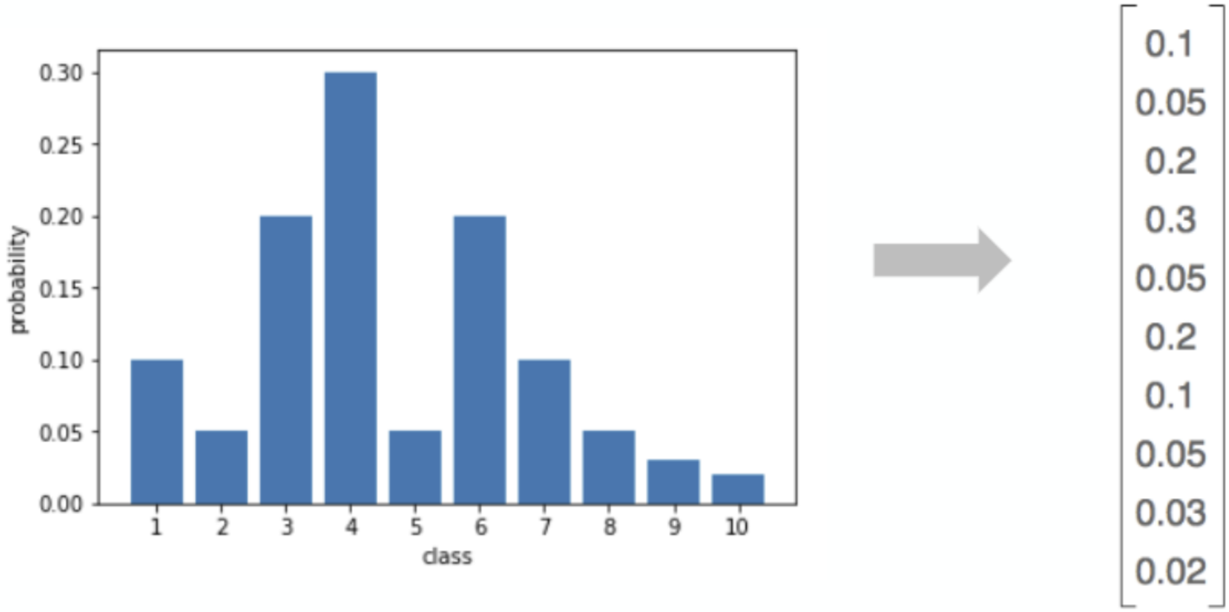


Figure 25: Conversion of ratings into a probability distribution [31]

The training of the model was completed in two stages [31]:

1. In the first stage, a high learning rate was used to train the last fully connected layer. This was done to ensure that the randomly added new weights adjust to the weights of the pre-trained ImageNet.
2. In the second stage, a low learning rate was used to train the entire CNN.

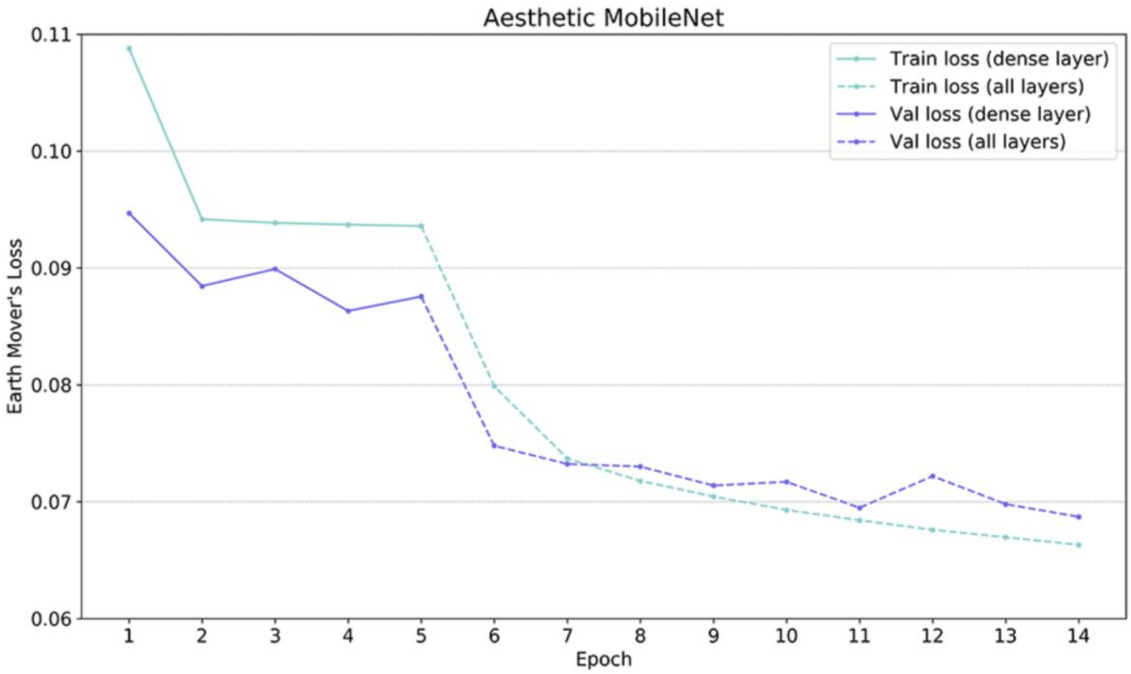


Figure 26: Training and Validation loss [31]

Figure 26 shows the training and validation loss for the training process over 14 epochs. The solid lines indicate the respective losses while training the last dense layer while the dashed lines indicate the losses while training the entire convolutional layers. The loss decreased significantly once the entire network weights were trained, which indicates that the pre-trained ImageNet weights needed to be adjusted significantly for the image classification task [31]. Upon training the model, the logs and the best model weights were then stored for testing the model and using it for aesthetic image evaluation.

VII. Price Estimation Model

A. Ensemble Learning

Ensemble learning aims to improve the performance of the prediction model by combining the estimation outputs from multiple base estimators that use a particular learning algorithm. The ensemble methods are broadly classified into two categories [45]:

- Bagging: These methods average the prediction outputs of several estimators which are built independently. The result is better than each of the individual estimators since the overall variance is reduced.
- Boosting: The idea behind boosting is to build several weak estimators sequentially rather than independently. Thus, each subsequent estimator tries to reduce the bias of the previous one and, as a result, it improves the prediction of the combined estimator.

B. Random Forests

Using this algorithm, each tree in the forest (ensemble of estimators) is constructed using a randomly selected subset of the data during the training process. In addition to this, each decision tree consists of a random subset of attributes [21], thereby reducing the overall variance of the ensemble model. Usually, the individual trees in the forest have high variance and, as a result, these trees have a trend to overfit. However, by averaging the predictions from the individual trees, the ensemble of the trees has a reduced variance thus giving a better overall prediction.

Additionally, this algorithm can also provide relative importance of the features in the dataset. The features that are at the top of the tree contribute to classifying a greater number of samples than those at the bottom. Thus, the importance of a feature is directly proportional to the fraction of the total samples that it contributes to classify.

C. Gradient Boosting

This algorithm works by generating a sequence of estimators and aims to reduce the error with each subsequent estimator that is produced. A new weak estimator is added at every step and the previous estimators are also kept untouched. This algorithm has three main elements [36]:

1. Loss function: This function is problem-dependent, and it should be a differentiable mathematical function. This project uses the least squared error function described by the following equation:

$$\text{Loss} = \sum_{i=0}^{n_{\text{samples}}-1} (y_i - \hat{y}_i)^2.$$

   where y_i is the actual value and $\hat{y}_i$ is the predicted value of the target variable.

2. Weak Learners: Gradient Boosting uses decision trees (regression trees) as weak learners. The construction of each tree is done in a greedy way such that the nodes with the highest importance are at the top. The trees are restricted by adding constraints on the maximum number of nodes, layers, or leaf nodes to ensure that they remain weak.

3. Additive model: The decision trees are added sequentially one by one and the loss during training is minimized using gradient descent. The gradient descent

approach tries to minimize the loss between subsequent trees by modifying the parameters of the tree and adding a modified tree to the model that follows the gradient.

D. Extreme Gradient Boosting (XGBoost)

This algorithm built upon Gradient Boosting aims to improve its performance and computing speed. It makes use of all the available cores in the CPU to provide parallelization. It also provides support for large datasets that cannot be accommodated into memory via out-of-core computing. It can also leverage a connected cluster of computing resources for training in a distributed manner. The main optimization behind XGBoost is the inclusion of a regularization term in the objective function along with the loss function which allows for a larger number of hyperparameters to be tuned.

E. Training The Price Estimation Model

Using the Image Evaluation Model described in section VI, the images corresponding to each property in the dataset were scored. A particular property listing may have several images associated with it. Each of these images of a given house was evaluated and the final score for that house was calculated as the average of the scores of the individual images. The tabular data includes the attributes of the house like area, number of bedrooms and bathrooms, zip code, age, parking type, heating and cooling type, and distances from nearby schools. This data was cleaned and processed as described in section V. Then, the score for each house was concatenated to the other attributes and this data was the combined input that was passed to the regression models, namely, Random Forest, Gradient Boosting, and XGBoost. To perform the training, the selling prices of the properties in the dataset were used as the labels. Thus, the input to the Price Estimation Model was the score obtained after aesthetically evaluating property images combined with the tabular data and its output was the estimated price of the house.

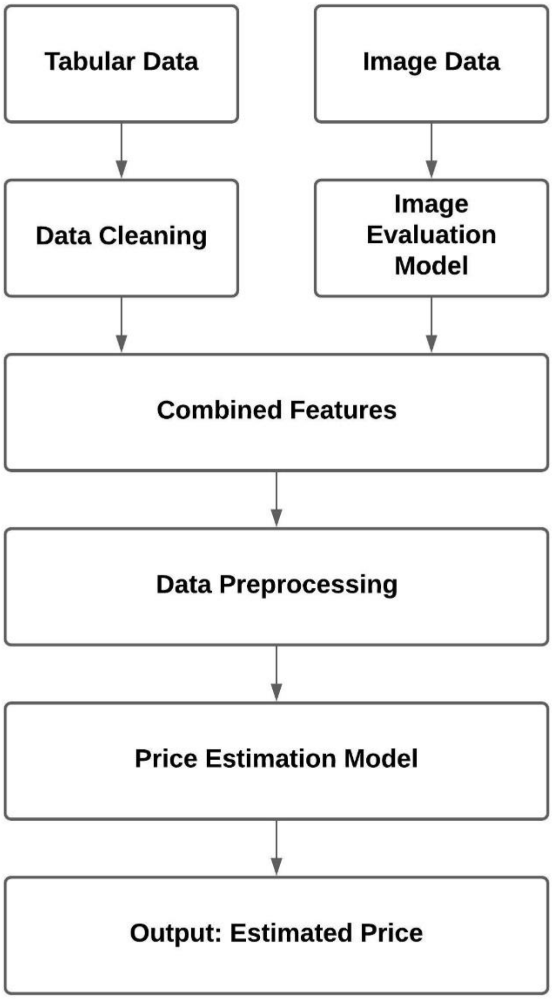


Figure 27: Overall architecture of the proposed system

Figure 27 shows the overall architecture of the proposed system. The loss during training was calculated by comparing the output estimated price with the actual selling price of a property and this loss was used to adjust the parameters of the regression model.

VIII. Experimental Results

A. Evaluation Metrics

The following metrics were used to compare the performance of the models:

1. Mean Absolute Error (MAE)

This metric is popular since it has the same unit of measurement as the value being predicted. It is calculated by taking the average of the absolute error values as shown in the following formula [34]:

$$\mathrm{MAE}(y, \hat{y}) = \frac{1}{n_{\text{samples}}} \sum_{i=0}^{n_{\text{samples}}-1} |y_i - \hat{y}_i|$$

where y_i is the actual value and $\hat{y}_i$ is the predicted value of the target variable.

2. Root Mean Squared Error (RMSE)

This metric is calculated by taking the square root of the mean of the squared error values as shown in the following formula [34]:

$$\mathrm{RMSE}(y, \hat{y}) = \sqrt{\frac{1}{n_{\text{samples}}} \sum_{i=0}^{n_{\text{samples}}-1} (y_i - \hat{y}_i)^2}$$

where y_i is the actual value and $\hat{y}_i$ is the predicted value of the target variable.

This metric has a greater penalty for larger error values since the difference between the actual and predicted value is squared resulting in a larger squared positive error [35].

3. Mean Absolute Percentage Error (MAPE)

This metric captures the relative errors in the predictions. It is not affected when the target variable is scaled globally. It is calculated using the following formula [34]:

$$\text{MAPE}(y, \hat{y}) = \frac{1}{n_{\text{samples}}} \sum_{i=0}^{n_{\text{samples}}-1} \frac{|y_i - \hat{y}_i|}{max(\epsilon, |y_i|)}$$

where y_i is the actual value and $\hat{y}_i$ is the predicted value of the target variable.

This metric captures a smaller magnitude of errors as opposed to metrics like Mean absolute error since it considers the relative error for each observation.

B. Results

For this experiment, out of the 728 properties listings in the dataset, 80% of the listings were used for training the Price Estimation Model and the remaining 20% were used for evaluating the performance. The model was trained using the architecture depicted in Figure 27 and its performance was evaluated using the metrics described earlier.

To begin with, the images corresponding to the properties in the dataset were evaluated using the Image Evaluation Model described in section VI. This process was done for each of the listings in the database and the image score was then appended to the other attributes of the house. This data was then used to train and evaluate the Price Estimation Model described in section VII.

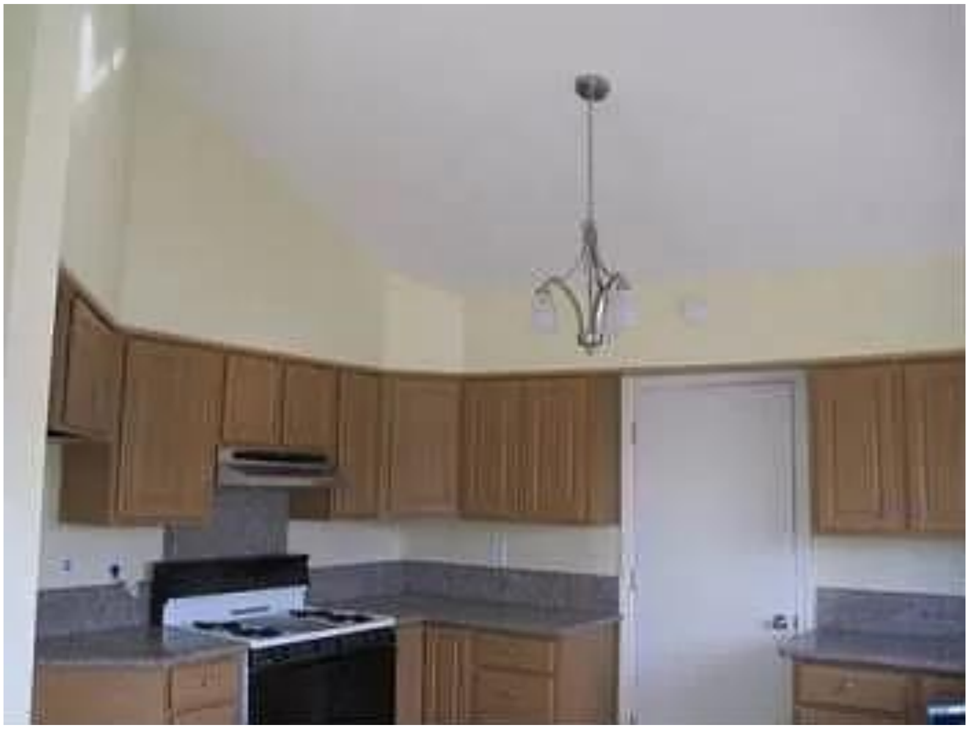


Figure 28: An example of a poor picture (score: 3.621)

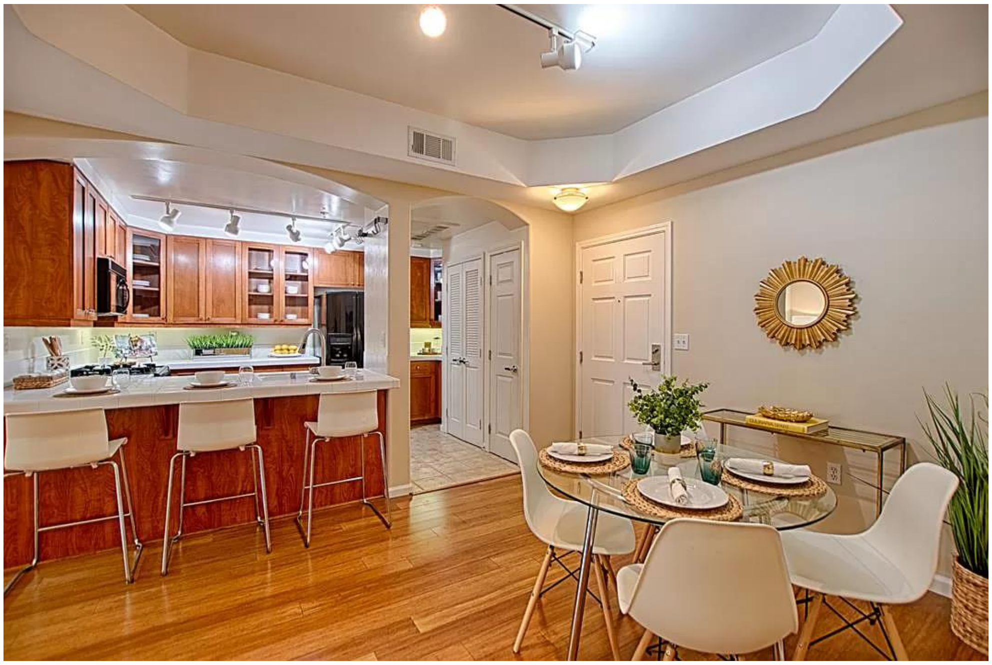


Figure 29: An example of a good picture (score: 5.428)

Figures 28 and 29 show the evaluation score for two sample images corresponding to two separate properties in the dataset. Figure 29 has a greater evaluation score than Figure 28 since it has a higher aesthetic value.

During the training process of the Price Estimation Model, the parameters of the individual models were tuned exhaustively to find the set of values that give the best results. The best values for parameters were then used to test the performance of the model. The error during training was calculated by taking the difference between the estimated price and the actual price for the property as seen in the testing set.

Table 5: Performance of the Price Estimation Model (using image scores)

Algorithm	**Mean Absolute Percentage Error (MAPE)**	**Root Mean Squared Error (RMSE)**	**Mean Absolute Error (MAE)**
Random Forest	10.16 %	5.831	2.063
Gradient Boosting	11.19 %	5.385	2.107
XGBoost	9.86 %	5.774	2.045

Table 5 shows the best results obtained for each of the algorithms when used for the Price Estimation Model. The XGBoost model achieved the lowest MAPE of 9.86 % as compared to the rest of the models. This can be attributed to the fact that the objective function in XGBoost has a regularization term that is not present in the other models. The model proposed in this project achieved a better performance than the model used in [16] which had a MAPE of 16.11%. Since the B-LSTM model used in [16] only considered image and location information of properties for estimating prices, it is expected that the approach proposed in this project, that considers several additional factors related to a property, would achieve better performance.

Table 6: Performance of the XGBoost Model (without image data)

Algorithm	**Mean Absolute Percentage Error (MAPE)**	**Root Mean Squared Error (RMSE)**	**Mean Absolute Error (MAE)**
XGBoost	10.33 %	5.040	2.413

Table 6 shows the performance of the XGBoost algorithm for estimating real estate prices without considering the image data. The model achieved a MAPE of 10.33% which is not significantly worse than the MAPE of 9.86 % which was obtained when image data was considered for predicting property prices. As seen in Table 2, the image scores have very low variance and hence their impact on the final estimated price is greatly reduced. Moreover, as seen in Figure 10, the attributes that have the highest correlation with the price of a property are its area and the number of bedrooms and bathrooms. Hence, the image scores did not have a great influence on the price of the properties.

Figures 30-32 show the feature importance of various features (attributes) that were used to predict the price of a property. The importance of an attribute is directly proportional to the number of samples that were classified using the attribute. Hence, attributes with higher importance are usually found at the top of a regression tree.

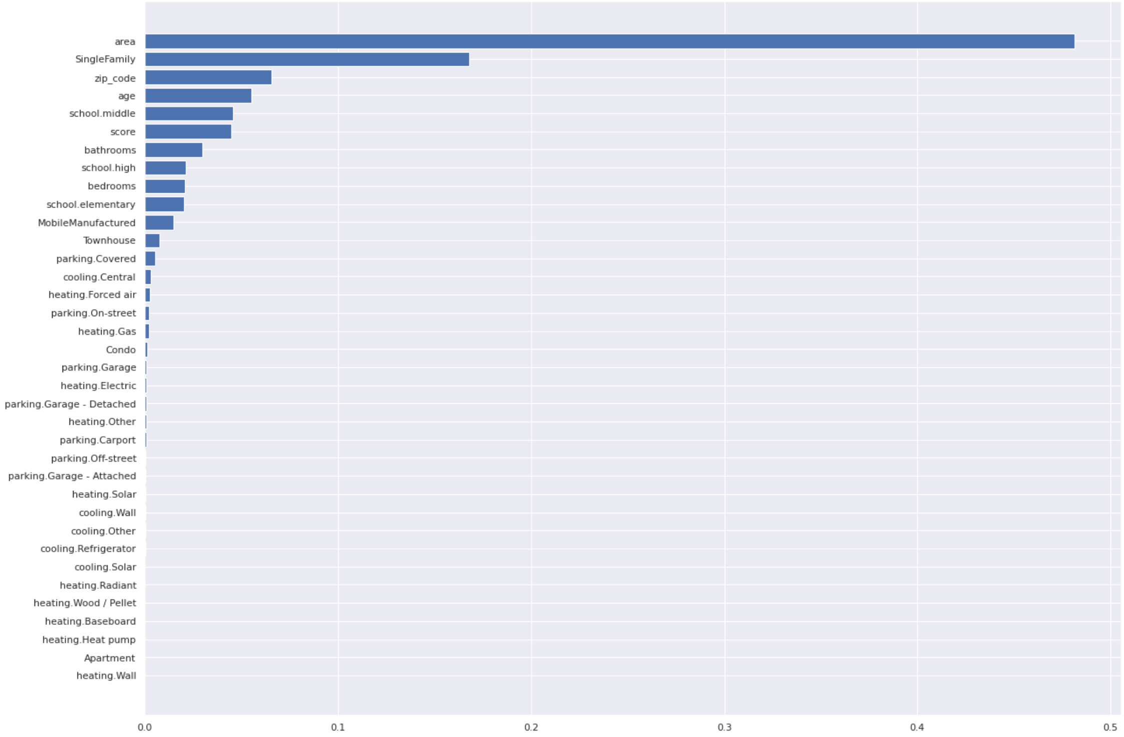


Figure 30: Feature importance using the Random Forest Algorithm

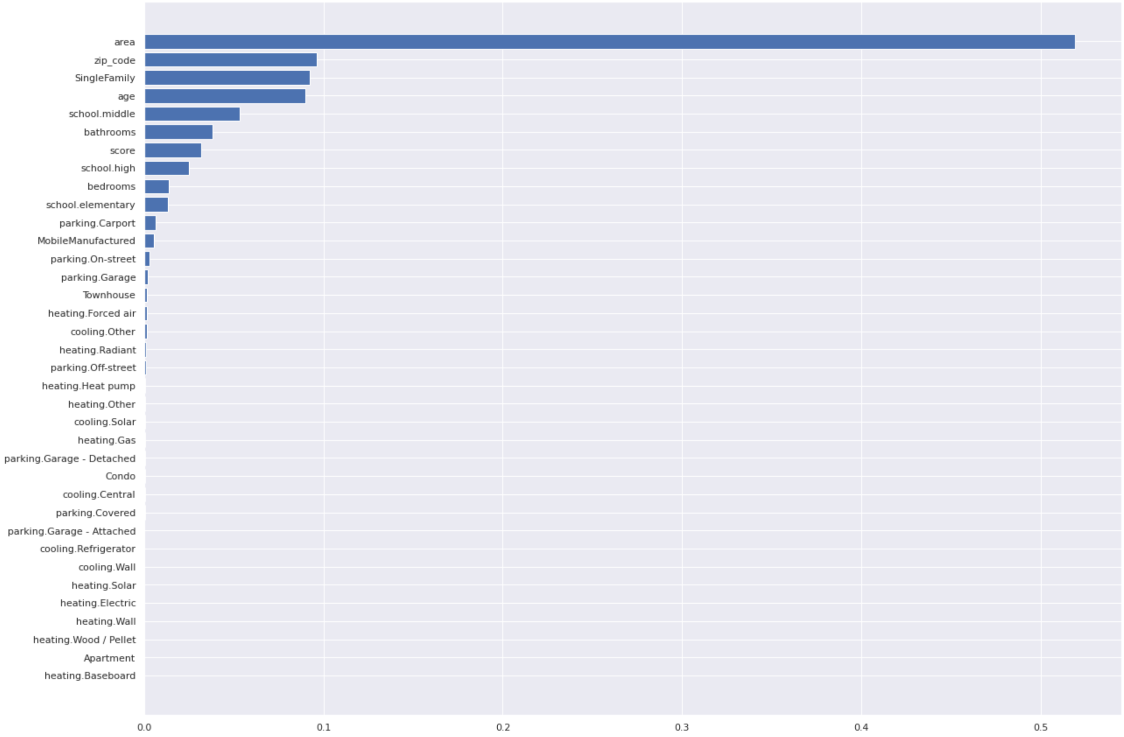


Figure 31: Feature importance using the Gradient Boosting Algorithm

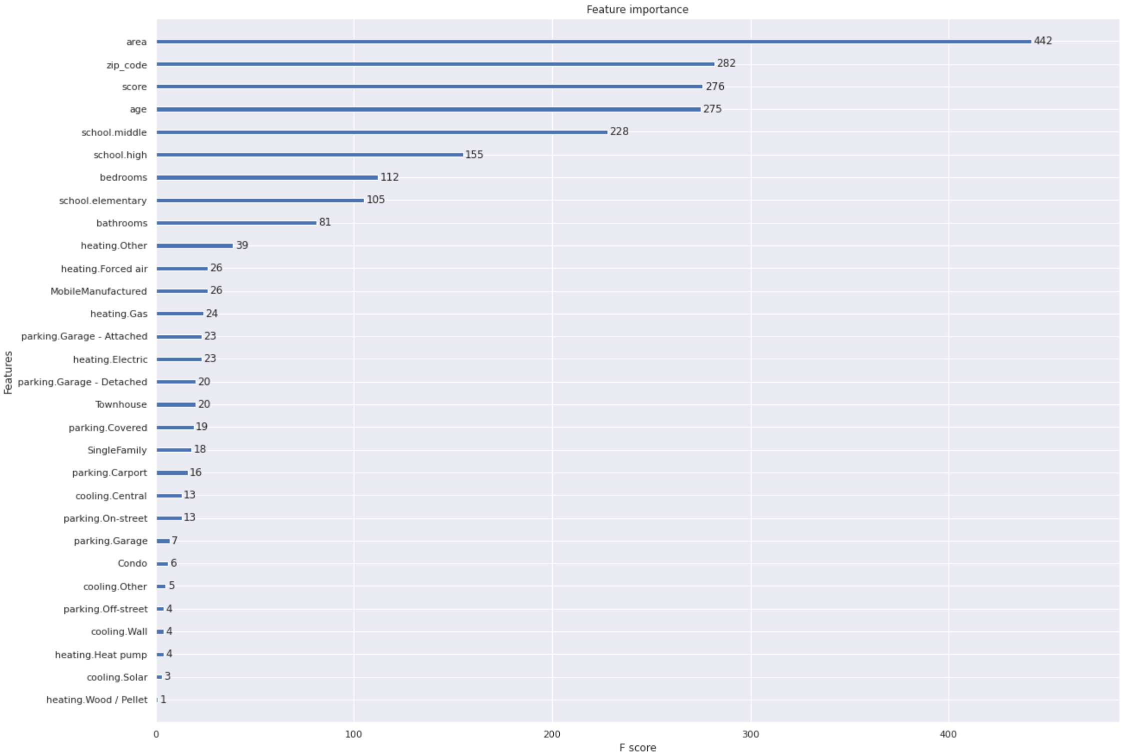


Figure 32: Feature importance using the XGBoost Algorithm

As seen from Figures 30-32, the price of the properties was greatly dominated by the living area. In case of the Random Forest and Gradient Boosting algorithms, the image score had a much lesser impact on the price as compared to the area and zip code (location). However, the XGBoost model ranked the image score to be comparatively more important (than Random Forest and Gradient Boosting) but the price was still highly influenced by the area and zip code. Hence, the image scores did not have a high impact in estimating the real estate prices when these ensemble algorithms were used for the task.

It is also worth noticing that the prices of the properties are not always proportional to their image score. For example, the property shown in Figure 28 with an image score of 3.621 has a price of $2,315,000 whereas the property shown in Figure 29 with an image score of 5.428 has a price of $745,000. Although these are two extreme examples, it must be understood that the price of a property is highly impacted by other attributes and cannot be determined only based on image data.

IX. CONCLUSION AND FUTURE WORK

This project proposes the use of images associated with properties along with other attributes for real estate appraisal. It leveraged the use of a CNN-based model for evaluating and scoring property images based on the MobileNet architecture proposed in NIMA. It was found that the Price Estimation Model trained using the image and tabular data performed well at the task of estimating real estate prices with the XGBoost model achieving the lowest MAPE of 9.86%. However, when image data was not considered for estimating prices, the model still achieved good results with the lowest MAPE of 10.33%. Thus, it can be concluded that the ensemble learning approaches are reliable for estimating property prices. Additionally, while the aesthetic condition of a house is an important aspect while making a purchase, the price of the house was greatly dominated by other attributes like the area, number of bedrooms and bathrooms, the house type, and other amenities. Although an image-based approach was expected to outperform a traditional model, the latter was still reliable for the task of real estate appraisal. This can be attributed to the fact that the AVA dataset has pictures from several different categories like landscape, people, and food, and is not limited to the context of real estate.

To increase the impact of the visual data, a new real-estate-specific dataset could be created by a crowdsourcing approach. It would be interesting to see how such a dataset can impact the performance of the models. Also, the image-based prediction of real estate prices is susceptible to scenarios where the house is beautified by professional staging companies before posting the pictures on the real estate websites. An automated model to detect such staging can greatly increase the performance of the Price Estimation Model by penalizing the image scores for such houses.

References

[1] “Zillow: Real Estate, Apartments, Mortgages & Home Values,” *Zillow.com*. [Online]. Available: https://www.zillow.com/. [Accessed: 07-May-2021].

[2] “Real Estate, Homes for Sale, MLS Listings, Agents | Redfin,” *Redfin.com*. [Online]. Available: https://www.redfin.com/. [Accessed: 07-May-2021].

[3] “Trulia: Real Estate Listings, Homes For Sale, Housing Data,” *Trulia.com*. [Online]. Available: https://www.trulia.com/. [Accessed: 07-May-2021].

[4] “What is a Zestimate? Zillow’s Zestimate Accuracy | Zillow,” *Zillow.com*. [Online]. Available: https://www.zillow.com/zestimate/. [Accessed: 07-May-2021].

[5] “About the Redfin Estimate | Home Value Estimator,” *Redfin.com*. [Online]. Available: https://www.redfin.com/redfin-estimate/. [Accessed: 07-May-2021].

[6] E. Ahmed and M. Moustafa, “House price estimation from visual and textual features,” *CoRR*, vol. abs/1609.08399, 2016.

[7] K. Kapoor and S. Arora, “Colour Image Enhancement based on Histogram Equalization,” *Electrical & Computer Engineering: An International Journal*, vol. 4, pp. 73–82, Sep. 2015.

[8] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, “Speeded-Up Robust Features (SURF),” *Computer Vision and Image Understanding*, vol. 110, no. 3, pp. 346–359, 2008.

[9] O. Poursaeed, T. Matera, and S. Belongie, “Vision-based real estate price estimation,” *Machine Vision and Applications*, vol. 29, no. 4, pp. 667–676, May 2018.

[10] L. Van Der Maaten and K. Weinberger, “Stochastic triplet embedding,” in *2012 IEEE International Workshop on Machine Learning for Signal Processing*, 2012, pp. 1–6.

[11] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, “Densely Connected Convolutional Networks,” in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 2261–2269.

[12] Y. Zhao, G. Chetty, and D. Tran, “Deep Learning with XGBoost for Real Estate Appraisal,” in *2019 IEEE Symposium Series on Computational Intelligence, SSCI 2019*, 2019, pp. 1396–1401.

[13] N. Murray, L. Marchesotti, and F. Perronnin, “AVA: A large-scale database for aesthetic visual analysis,” in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, 2012, pp. 2408–2415.

[14] A. G. Howard *et al.*, “MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications,” *CoRR*, vol. abs/1704.04861, 2017.

[15] T. Chen and C. Guestrin, “XGBoost: A Scalable Tree Boosting System,” in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016, pp. 785–794.

[16] Q. You, R. Pang, L. Cao, and J. Luo, “Image-Based Appraisal of Real Estate Properties,” *IEEE Transactions on Multimedia*, vol. 19, no. 12, pp. 2751–2759, 2017.

[17] Wikipedia contributors, “Vincenty’s formulae,” *Wikipedia, The Free Encyclopedia*, 02-May-2021. [Online]. Available: https://en.wikipedia.org/wiki/Vincenty%27s_formulae. [Accessed: 07-May-2021].

[18] B. Perozzi, R. Al-Rfou, and S. Skiena, “DeepWalk: Online Learning of Social Representations,” in *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2014, pp. 701–710.

[19] M. Schuster and K. K. Paliwal, “Bidirectional recurrent neural networks,” *IEEE Transactions on Signal Processing*, vol. 45, no. 11, pp. 2673–2681, 1997.

[20] J. Niu and P. Niu, “An Intelligent Automatic Valuation System for Real Estate Based on Machine Learning,” in *Proceedings of the International Conference on Artificial Intelligence, Information Processing and Cloud Computing*, 2019.

[21] L. Breiman, “Random Forests,” *Machine Learning*, vol. 45, no. 1, pp. 5–32, Oct. 2001.

[22] Hecht-Nielsen, “Theory of the backpropagation neural network,” in *International 1989 Joint Conference on Neural Networks*, 1989, pp. 593–605 vol.1.

[23] K. O’Shea and R. Nash, “An Introduction to Convolutional Neural Networks,” *CoRR*, vol. abs/1511.08458, 2015.

[24] O. Sagi and L. Rokach, “Ensemble learning: A survey,” *WIREs Data Mining and Knowledge Discovery*, vol. 8, no. 4, p. e1249, 2018.

[25] A. Khamis and N. Kamarudin, “Comparative Study On Estimate House Price Using Statistical And Neural Network Model,” *International Journal of Scientific & Technology Research*, vol. 3, pp. 126–131, 2014.

[26] P. Kumkar, I. Madan, A. Kale, O. Khanvilkar, and A. Khan, “Comparison of Ensemble Methods for Real Estate Appraisal,” in *2018 3rd International Conference on Inventive Computation Technologies (ICICT)*, 2018, pp. 297–300.

[27] M. Shahhosseini, G. Hu, and H. Pham, “Optimizing ensemble weights for machine learning models: a case study for housing price prediction,” in *INFORMS International Conference on Service Science*, 2019, pp. 87–97.

[28] J. F. Hair, W. C. Black, B. J. Babin, and R. E. Anderson, *Multivariate Data Analysis*. Pearson Education Limited, 2013.

[29] L. Duesbery and T. Twyman, “How do I deal with missing data?,” in *100 Questions (and Answers) About Action Research*, Thousand Oaks, California, 2020.

[30] M. Lukic, “One-hot encoding in python with pandas and scikit-learn,” *Stackabuse.com*. [Online]. Available: https://stackabuse.com/one-hot-encoding-in-python-with-pandas-and-scikit-learn/. [Accessed: 09-May-2021].

[31] C. Lennan, D. Tran, V. A. P. by Lennan, and V. A. P. by Tran, “Deep learning for classifying hotel aesthetics photos,” *Nvidia.com*, 30-Oct-2018. [Online]. Available: https://developer.nvidia.com/blog/deep-learning-hotel-aesthetics-photos/. [Accessed: 09-May-2021].

[32] H. Talebi and P. Milanfar, “NIMA: Neural Image Assessment,” *IEEE Transactions on Image Processing*, vol. 27, no. 8, pp. 3998–4011, 2018.

[33] Wikipedia contributors, “Earth mover’s distance,” *Wikipedia, The Free Encyclopedia*, 26-Feb-2021. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Earth_mover%27s_distance&oldid=1009039001. [Accessed: 09-May-2021].

[34] “3.3. Metrics and scoring: quantifying the quality of predictions — scikit-learn 0.24.2 documentation,” *Scikit-learn.org*. [Online]. Available: https://scikit-learn.org/stable/modules/model_evaluation.html. [Accessed: 09-May-2021].

[35] J. Brownlee, “Regression metrics for machine learning,” *Machinelearningmastery.com*, 19-Jan-2021. [Online]. Available:

https://machinelearningmastery.com/regression-metrics-for-machine-learning. [Accessed: 09-May-2021].

[36] J. Brownlee, “A gentle introduction to the gradient boosting algorithm for machine learning,” *Machinelearningmastery.com*, 08-Sep-2016. [Online]. Available: https://machinelearningmastery.com/gentle-introduction-gradient-boosting-algorithm-machine-learning. [Accessed: 09-May-2021].

[37] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “ImageNet: A large-scale hierarchical image database,” in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 248–255.

[38] W. Wang, Y. Li, T. Zou, X. Wang, J. You, and Y. Luo, “A Novel Image Classification Approach via Dense-MobileNet Models,” *Mobile Information Systems*, vol. 2020, p. 7602384, Jan. 2020.

[39] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.

[40] S. Ravichandiran, *Hands-On Reinforcement Learning with Python: Master reinforcement and deep reinforcement learning using OpenAI Gym and TensorFlow*. Birmingham, England: Packt Publishing, 2018.

[41] R. Shanmugamani, *Deep Learning for Computer Vision: Expert techniques to train advanced neural networks using TensorFlow and Keras*. Birmingham, England: Packt Publishing, 2017.

[42] M. Denil, B. Shakibi, L. Dinh, M. A. Ranzato, and N. de Freitas, “Predicting Parameters in Deep Learning,” in *Advances in Neural Information Processing Systems*, 2013, vol. 26.

[43] “460 Fonick Dr, San Jose, CA 95111 | Zillow,” *Zillow.com*. [Online]. Available: https://www.zillow.com/homes/460-Fonick-Dr-San-Jose,-CA,-95111_rb/19810133_zpid/. [Accessed: 09-May-2021].

[44] “3331 Senter Rd, San Jose, CA 95111 | Zillow,” *Zillow.com*. [Online]. Available: https://www.zillow.com/homes/3331-Senter-Rd-San-Jose,-CA,-95111_rb/19734437_zpid/. [Accessed: 09-May-2021].

[45] “1.11. Ensemble methods — scikit-learn 0.24.2 documentation,” *Scikit-learn.org*. [Online]. Available: https://scikit-learn.org/stable/modules/ensemble.html. [Accessed: 09-May-2021].

[46] Wikipedia contributors, “Feature scaling,” Wikipedia, The Free Encyclopedia, 21-Jan-2021. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Feature_scaling&oldid=1001781300. [Accessed: 09-May-2021].

[47] C. Custer, “15 Python libraries for data science you should know,” Dataquest.io, 05-Feb-2020. [Online]. Available: https://www.dataquest.io/blog/15-python-libraries-for-data-science/. [Accessed: 09-May-2021].