Radboud University Nijmegen

Faculty of Science

Product Price Prediction

Using Image and Metadata Analysis

Thesis

MSc Computer Science

Author:
Aghil Karadathodi Prasad

Supervisor:
Prof. M.A. Larson

Second reader:
Prof. Djoerd Hiemstra

24th October, 2021

Acknowledgements

It is with immense gratitude that I acknowledge the support and help given by my supervisor, Professor M.A. Larson, during my thesis. Your willingness to devote time for our meetings generously and to give consistent feedback and motivation is very much appreciated. Thank you for your belief, patience, and enthusiasm.

I would like to thank my advisor at REV'IT!, Mridul Vishal for his help and support. I would also like to thank Zhengyu Zhao for the discussions and suggestions which helped me a lot during this thesis. Thanks to Gowtham Indiran Deekshit and Yongrui Zhao for the discussions. I would also like to thank Kasper Brink, Efraim Stevens, and Niels Talma for their IT hardware and software support. I would also like to thank Rob van de Ven from Rev'IT! sport international for our discussions on the challenges of product design and Randy Vos and Peter Kuijoers for the discussions on sales and price management.

Contents

Abstract

This work focuses on building a price prediction system by combining product images and product properties. The intent is to find the performance of these models on single and multi-category environments and to study their effect on price prediction. We also try to find the different product features contributing to the price and see the importance of metadata in the model. We used a novel dataset that contains details about the products, their price, and product images pulled from an e-commerce website. A basic model built using the VGG16 architecture was further enhanced by an InceptionResNetv2 and then proposed a two-stage prediction system with image classification to group images based on category and a price prediction unit using gradient boosting on decision tree ensembles. We also implemented a time-efficient alternative to this proposed model by the feature fusion method. The proposed methods gave good results compared to the models that only considered images for price prediction. In addition, we have analyzed the properties of the product that contribute to the price. In environments with one or more categories, the proposed methods excelled against other models.

1 Introduction

1.1 Background

One of the secrets to running a successful business is pricing the products correctly. The correct pricing of products will increase sales volume and lay the foundation for a thriving business. Pricing strategies work differently for different types of companies. In most cases, these strategies are based on a formula that works for various products and markets. There is always the risk of underpricing or overpricing the products, and this will eventually affect the company's brand equity and profits [23].

A discussion with the stakeholders of REV'IT! Sports International, a premium European motorcycle clothing brand based in the Netherlands, raised a similar kind of gap in their process of pricing their products. REV'IT! makes innovative products that help people ride better, keeping safety as the highest priority positioned for the sport, touring, and adventure rider's needs.

REV'IT! every year launch 50-60 new products in different categories (such as gloves, boots, jackets, etc.). It is done successfully by the combined effort of the product design team, product development team, and marketing team. The pricing of the new products has always been challenging for all the teams. The design and development team has a price range for new products, but it has been challenging to find the right price. The change in the proportion of materials affects the cost of the final product. It is also difficult for the sales team to check whether the price they have set is reasonable. These should also be compared to competing products in the market so that their new products fit in the acceptable price range and respect their brand equity and customer base.

1.2 Objectives

In this thesis, we build a price prediction model that will support companies in determining the price of a new product by considering the product design images and product composition details. This will help companies develop products that fit into the correct price ranges that are comparable to their competitors. The model can be trained with the details and images of old products, as well as other relevant products in the market. When we introduce a new product, we can use the model to predict the price that meets market standards.

1.3 Research Question

The research question defined for this thesis was as follows:

> ***"Given a picture or a set of pictures (plus the metadata), find the products that are similar such that it would be possible to sell them in the same price range"***

The sub-questions formalized to answer the main question were as follows:

1. What are the important product properties that contribute to the price?
2. What will be the difference in model performance against a single category versus a multi-category dataset?
3. Will the model that uses metadata give better results than the model that only uses images to predict the prices of newer products?

The first sub-question focuses on the metadata. Which essential properties of the product determine the price? It usually depends on the different product properties. By product properties, we mean the composition of the product, color, the primary material used, product type, special features, etc. For a motorcycle apparel industry like REV'IT!, which focuses on protection and design, the most important properties of the product are the level of protection, waterproofing, number of layers, insulation, ventilation, etc. The primary purpose behind this task is to find out which of these properties matter the most for pricing and how they affect them. It can be the case that for some industries, color is not a factor that affects the price. But for some businesses like fashion industries, color may be an essential factor. In addition, this research focuses on identifying the difference in model performance between a single category dataset and a multi-category dataset. There is also an emphasis on finding out whether the model using product images along with their properties performs better than the model built with only product images. We assume that adding multiple categories bring more complexity to the problem, but this could be mitigated by creating a model which considers both product images and product properties. The answers to these sub-questions will help finalize the answer to the central question of my thesis.

2 Related Work

Here we will discuss the past work which has influenced our approaches and techniques. Supervised machine learning and computer vision are used extensively for price prediction and related tasks these days. Developments in computer vision started in the 1950s where neurophysiologists experimented on animals to understand how they respond to images [7]. With the emergence of AI as an academic field of study in the 1960s, more developments in the field of machine learning were witnessed [13]. In the 1980s, algorithms were developed which could detect edges, corners, curves, and similar shapes. This millennium witnessed improvements in object recognition algorithms and the emergence of labeled public image datasets such as ImageNet.

2.1 Price Prediction

We also saw steady developments in the fields of price prediction and forecasting. Recently machine learning has been used to predict stock prices [14], house price prediction using satellite images [38] and predicting sales of retail products using machine learning [34]. With advancements in fields of deep learning, computer vision, object detection, and tracking and natural language processing tasks have better results.

PriceNet [3] use linear regression on the histogram of oriented gradients, convolutional neural network features, and multiclass SVM to predict vehicle prices using visual features. They work on two different datasets having product images and prices for automobiles and motorbikes separately. Several existing models, along with the newly proposed architecture called the PriceNet, were implemented and obtained significant improvements in results for the newly proposed model. Another work by A. Fathalla and A. Salah [10] proposed a deep model architecture using long short-term memory(LSTM) and convolutional neural architecture are used to predict the price of second-hand items. This work tried to forecast the minimum and maximum prices of second-hand items. They also try to predict an item quality score using LSTM that showed better performance than the baseline model built on support vector machines. This item quality score, along with maximum and minimum prices, is combined to predict the final price of second-hand items. Their approach of finding the item quality score sounds like a good start but creating a price cap doesn't suit our use case as we try to predict the price of brand new products.

A vision-based price estimation technique proposed by O. Poursaeed, T. Matera, and S. Belongie was considering the interior and exterior visual characteristics for price prediction[27]. They mainly focused on estimating the luxury level of houses, given the interior and exterior images of the house. First, the images were classified into different room categories and labeled to different luxury levels using crowdsourcing. The idea of labeling houses based on the luxury levels was exciting and sounded like an excellent differentiating factor for brand selection. So we decided to have different labels based on brands and see their importance on the price prediction. Another exciting research by Z. Kostic and A. Jevremovic proposes techniques to extract effective visual features for boosting the predictive algorithms for housing market predictions [19]. This work used boosting models like XGBoost and LightGBM to predict prices using the extracted features collected from the dataset and google images. Here they claim to outperform some of the strongest metadata predictors utilizing the set of 40 image features.

The research of product price proposals for used products on online marketplaces based on product details [9] yielded a lot of exciting options for our work. They tested 13 different models with regression and classification outputs for their research and included techniques like a bag of words, TFIDF and word2vec. They achieved the best results for an LSTM network with word2vec text encoding.

Our project aims to create a better price prediction solution that focuses on apparel categories, including clothing, footwear, accessories, etc., given the product images and the product properties. We started with a classic price prediction model comparable to the PriceNet [3] because it solves a problem closer to ours, and we improved that using the techniques mentioned above.

2.2 Image Classification

Image classification is a fundamental problem in computer vision which sets as the base for many others. The main objective is to categorize and label the images on particular rules. With the availability of a wide range of datasets and advancements in the fields of machine learning, image classification techniques have shown significant improvements. LeNet [22] was one of the first revolutionary convolution neural networks which showed the applicability of CNN and backpropagation algorithm to practical applications [1]. AlexNet [20] and VGGNet [33] brought the convolutional networks to fame by showing that deeper networks with data augmentation and application of concepts like dropout, ReLU activation, small filters can achieve substantial performance on datasets like ImageNet. Performance comparison of VGGNet and InceptionResNetv2 on classifying skin diseases from skin lesion images by S. Guha and S. M. Haque found better results for inception network [11]. Research investigating the classification of normal and abnormal radiographic images also compared multiple CNN networks, and InceptionResNetv2 gave the best results [2]. These inspired us to use the InceptionResNetv2 model to classify products in our price two-stage prediction system.

2.3 Feature fusion

Another important aspect of this thesis is to perceive strategies to combine different input categories for price-prediction models. Generally, we configure the regression model to take either series of text as input or a multi-dimensional array for images. The research by J. Sill, G. Takacs, L. Mackey, and D. Lin proposed a feature-weighted linear stacking technique that enables concatenating different features together to enhance regression performance retaining the speed and stability [32]. This research showed the possibility of stacking multiple independent feature vectors horizontally to create a final feature vector that could function as the input for our regression models.

3 Approach

This section describes our approach for solving the research question. The baseline for this thesis was built on the PriceNet [3] where a convolutional neural network was trained on the product images and prices. Here we consider only product images to predict the price. We then used a better model to improve the prediction results.

Then we proposed two methods for price prediction using both images and metadata. The first one is a two-step model as depicted in Figure 1 for a large uncategorized dataset. This method creates a subset of training data to build the model that is categorically similar to the test data.

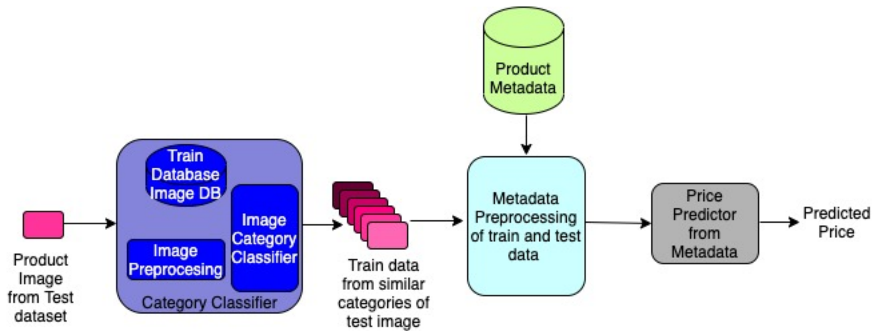
Product Metadata
Train Database Image DB
Image Category Classifier
Image Preprocesing
Category Classifier
Product Image from Test dataset
Train data from similar categories of test image
Metadata Preprocessing of train and test data
Price Predictor from Metadata
Predicted Price

Figure 1: Proposed Model of two stage prediction system

The newly proposed pipeline can be divided into two sub-modules: a classifier module and a price prediction module. The classifier module will be pre-trained to create an image category database using the product training images. This image database will be used to create a subset of similar products given a new product. So the first phase finds the products belonging to a similar category. We feed this data to the next module for price prediction.

The price prediction module gets the metadata of the subset created and performs pre-processing. This metadata will be used to predict the price of the new product using different regression models. We will find which of these product details from the metadata are essential in finalizing the price of the new product. This will also help remove some metadata information that has a more negligible effect on the model performance. Then hyperparameters were tuned to get the best results out of the price prediction regression models. We assume that this approach will help to train the regression model with data that are more relevant to the new product and will tend to predict better results. A sample use-case for this method is an enormous online retailer which allows sellers to sell multiple products through their portal. The data given by the sellers may not be accurate (about the category it belongs to). So our model can first find the categories to which it belongs and then predict price comparing other metadata to existing products in that categories.

The other approach we proposed was to extract feature vectors from images as well as the metadata and to combine them to create a final feature vector. This final feature vector was given as the input to the price regression models to predict the prices of products. This approach is more beneficial for smaller datasets where we need faster predictions. We will discuss all these in detail in section 5.

4 Data

In this section, we describe the distinguishing features of the dataset used to answer our research question. We found many public datasets with fashion products, but most of the datasets did not have images and metadata together. Also, some of the datasets had a lot of blank price columns. Lots of product review datasets were available, but those lacked the product features and characteristics which we wanted to analyze. The data for this thesis project was taken from data.world. This dataset consists of 15000 product details from an e-commerce website over five different categories, as plotted in Fig 2.

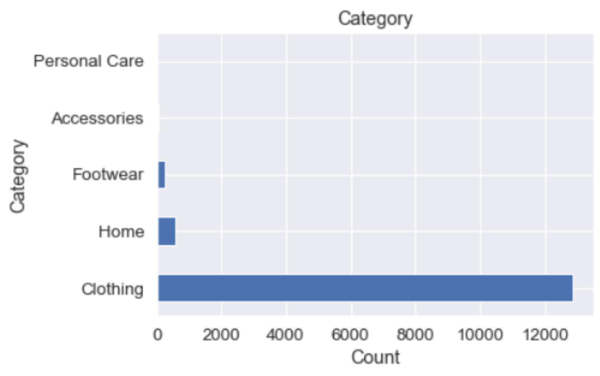

Figure 2: Categories vs Count

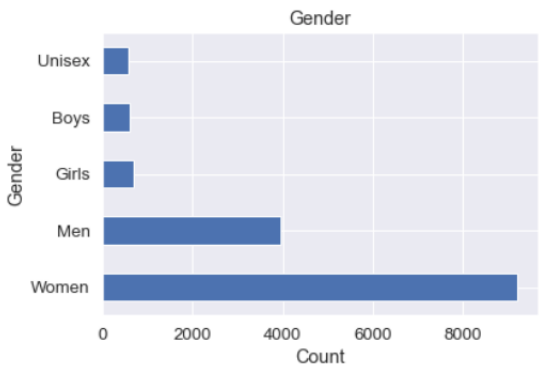

Figure 3: Plot of Gender vs Count

Looking at the count of products per category and its relevance to the problem we are trying to address, we narrowed down the categories to clothing and footwear. On plotting the bar chart(Figure 3) for gender, we could see five different values. As most of the products are contained within the Men and Women category, we removed other classes and duplicate records.

The products with blank columns for prices and images were not helpful as our main aim is to train the machine learning model to accept images and metadata as input to predict prices. The plot in Figure 4 shows the count of images for products. So the products with blank columns were removed from the dataset.

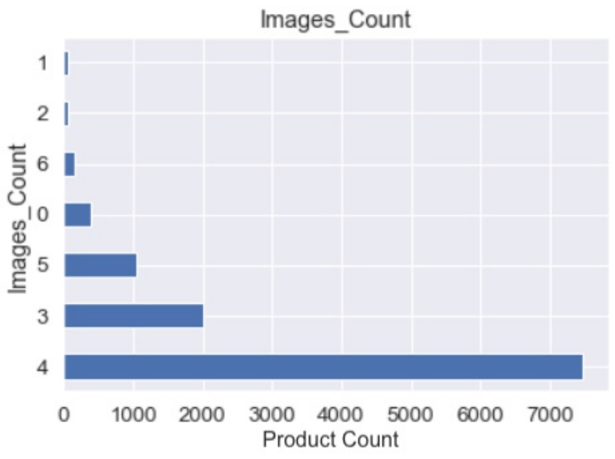

Figure 4: Image count

The columns with text data with the product title, product description, and product specifications seemed important for analyzing the metadata. To get more idea about the text fields, a word cloud was plotted with unigrams and bigrams for product title, product descriptions, and product specifications in Figures 5, 6, and 7. Product title mainly specifies the brand name, gender, color, and category of the products. The product description has more details like type of collar, design, color, etc. The specification column focuses on the patterns, sleeve details, size, etc.

Also, the category column featured five levels of categories separated by '/' like 'Clothing/Women/Top/IMARA/Tops By IMARA'. Here the first and third levels are only helpful for us, which specifies the main and subcategories for products. The second category, which specifies the gender, is already captured by the gender column. And the fourth and fifth level mentions the brand names. So we divided the category into category_0 and category_1 from the first and third levels, respectively, and used them as metadata features.

After all the data cleaning, 10759 products were remaining for building the model. This data plotted to find the price density by gender is shown in Figure 8. There were more products for the women, and the price density was more between 399-5000 INR. The maximum price for a product in our dataset was 25500 INR,

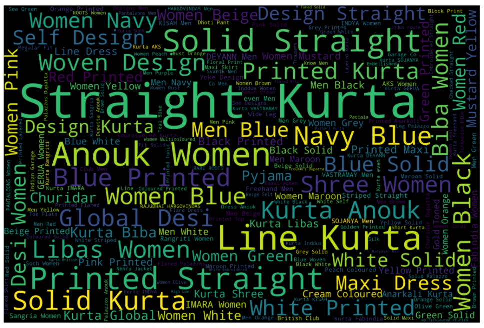

Figure 5: Product titles

quarter sleeve
side slit
mandarin collar
straight hem
hem side
three quarter
round neck
sleeves straight
straight kurta
long sleeve

Figure 6: Product descriptions

Pattern Type
Styling Regular
Wash Care
Garment Size
Sleeve Length
Sleeve Styling
Weave Pattern
Design Styling
Regular Fabric Pattern
Weave Type
Regular Sleeve
Machine Weave
Sleeves Occasion
Type Machine
Slit Detail
Size Garment

Figure 7: Product specifications

Wordcloud plots of textual metadata.

and the minimum was 399 INR. The mean price was 2362.86 INR. The median was 1799 INR, where 25% of the product lies below 1399 INR and 75% were below 2699 INR.

A study by W. Dodds and K. Monroe on the effect of price, brand, and store information on buyers' product evaluations shows the importance of brand value on price [8]. This is pretty clear if we compare two mobile phones with the exact specifications having different prices depending on the brand. So we considered categorizing brands based on the mean price. The brands whose mean price was above 15000 INR were categorized into 'luxurious_brands'. The brands whose mean price was between 5000 and 15000 were categorized as 'expensive_brands', and those whose mean price was less than 5000 were categorized as 'cheap_brands'.

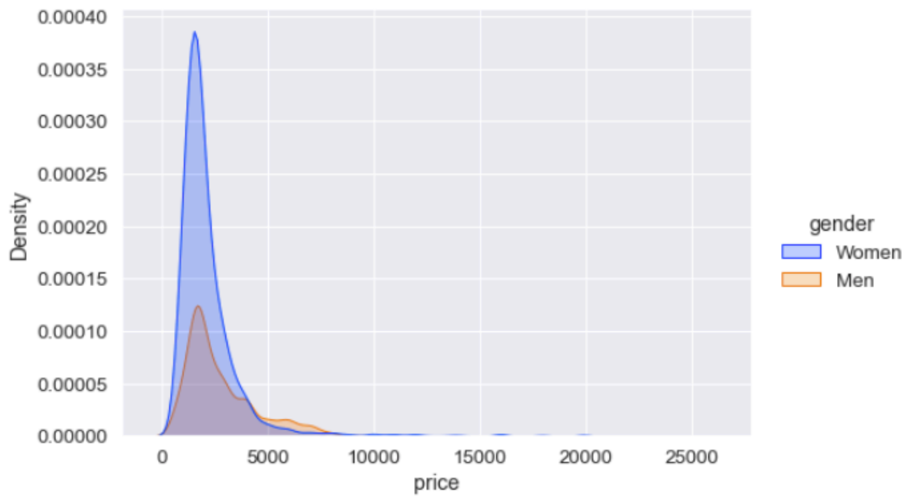

Figure 8: Price density by gender

We assume these columns will help improve the predictions.

The whole dataset was then divided and saved into three segments of 60%, 20%, and 20% into training, validation, and test data, respectively, so that the data is consistent across every model. This saved data used across different models make sure that the results are comparable between different approaches. Training, validation and test dataset was also created with single and multi-category to test the model's performance in each case. For the single category dataset, we used the clothing category products as this had the majority of data.

5 Setup

This section explains the implementation of all models in detail.

5.1 Price Prediction using Images only

A fundamental CNN-based model was built, which takes an image as an input and gives the price as the output. This model was built based on the solution mentioned on paper [3], and its general architecture is given in Figure 9. We have used VGG16 and InceptionResNetv2 model to build the base price prediction pipelines. These models take product images as input and predict the price of the products. We trained these models with the training set having product images and product prices, tuning the parameters using validation set, and evaluate on the test data set. The hidden layers with convolutional modules change according to the model. After the feature extraction from the last convolutional layer, we flatten the output. For the VGG16 model, we get an output with dimensions (1, 25088). This output is fed into a fully connected layer with ReLU as an activation function. This layer converts the output vector into size (1, 256) and is finally fed into a fully connected layer with a linear activation output unit. This final layer gives the price of the new product.

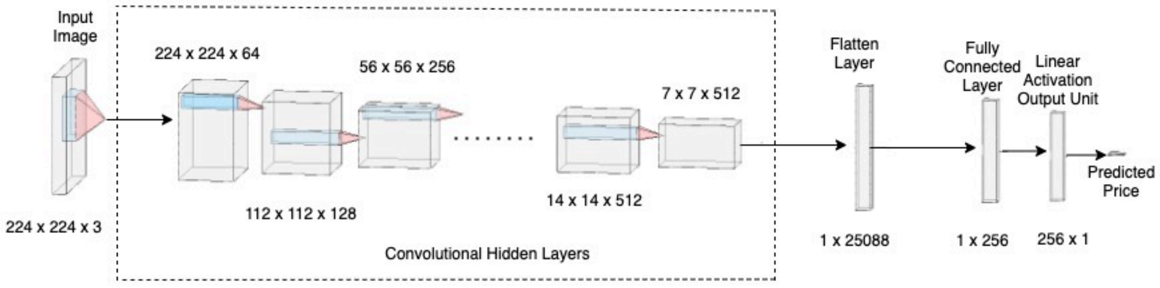

Figure 9: Price Prediction Model using Images

5.1.1 VGG16

VGG16 was proposed by A.Zisserman and K. Simonyan from Oxford University in 2015 [33]. It is a 16 layered convolutional neural network which was pre-trained on ImageNet where it achieved 92.7% top-5 test accuracy for over 14 million images which belong to 1000 categories [25].

The input image for VGG16 expects 224, 244, 3 format where the 3 represents the RGB colors. This input was pre-processed, and the model was created using Keras library [17]. The model architecture and weights were loaded, and then the softmax and dense layers were removed. A fully connected layer and a single linear activation output unit are added for the price regression task at the end [18]. Also, we make the first 19 layers non-trainable so that the weights are not altered. These layers are pre-trained with the ImageNet dataset.

5.1.2 InceptionResNetv2

To improve the base model, we used the Keras InceptionResNetv2, which has shown better performance with the ImageNet dataset in the studies before [16]. We assume this model will demonstrate this improvement in performance with our dataset. The inception network was introduced to solve problems like variations in the size of important features in the images, computational speed, vanishing gradient problem, etc. Traditionally a CNNs performance varied drastically when the size and position of salient features changed between images [29]. Ideally, a larger kernel will be preferred for an image where information is distributed globally and a smaller kernel for information present locally.

The Inception network proposed to use multiple filter sizes on the same levels, thereby making the model wider rather than deeper [36]. Also, to reduce the computational expenses of deep neural networks, InceptionNet has a max-pooling layer and 1×1 convolution before 3×3 and 5×5 convolutions as given in Figure

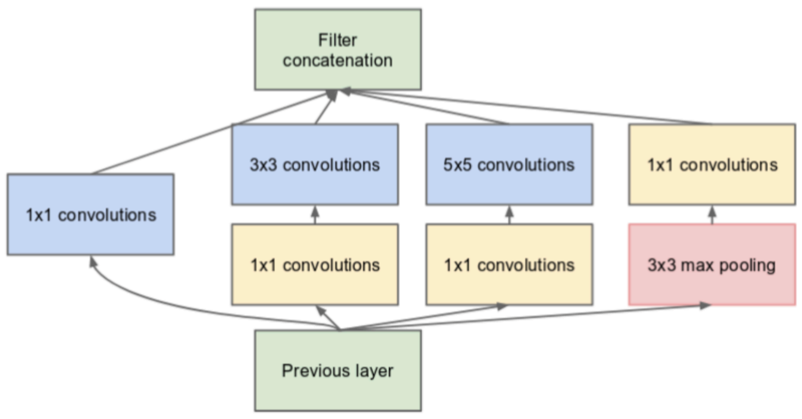

Figure 10: Inception module with dimension reduction [36]

10. In different versions of InceptionNet, a 5×5 convolution is replaced by two 3×3 convolutions to improve computational speed.

Researchers found that as the number of hidden layers increases, the training error also increases, degrading the model performance. So in 2015, Xiangyu Zhang, Shaoqing Ren, and Jian Sun introduced ResNet, a new type of neural network that had some skip blocks in the internal hidden layers [12]. The core of this network was a series of residual blocks which have a series of skip connections that help solve the vanishing gradient problem by taking the alternate skip path [24] as shown in Figure 11. A hybrid module featuring the improvements of the inception network and the residual network was used to build the Inception ResNet [35]. The preliminary steps of operations(called the stem) were modified to reduce the complexity of the model and added 1×1 convolutions after original convolutions to match the dimensions of input and output of residual blocks. Also, to solve the cause of the network to "die" when the filters of the residual unit exceed 1000, the residual activations were scaled by a factor of 0.1 to 0.3 [29].

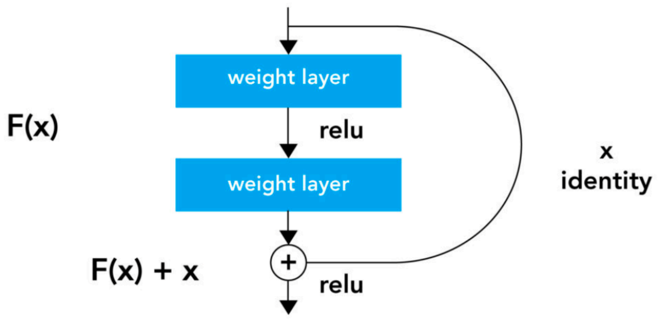

Figure 11: Residual learning block [24]

The Keras implementation of InceptionResNetv2 was used for building the price regression model [16]. The model expects an input dimension of [299,299,3] and was pre-processed by the Keras library function. As we did for the VGG16 model, we removed the softmax and dense layers and added a fully connected layer and a linear activation unit to get the predicted price. Also, we made the hidden layers non-trainable so that the weights pre-trained on the ImageNet dataset are not changed.

5.2 Price Prediction using Images and Metadata combined

5.2.1 Metadata Text Processing

The metadata we have is not fit for feature extraction as it is. Most of the text data were entered into the e-commerce website by different resellers. So they varied in structure and style. A series of data processing was applied to the textual data before extracting the features.

5.2.1.1 Data Preprocessing

The first step was to handle missing values in the columns. The missing values were returning errors while performing feature extraction. So to make it standard, the missing values were replaced with the text "other". The next step was to replace contractions from the text. Here the shortened words like 'don't' were replaced by 'do not', 'I've' with 'I have', etc. Then regex was used to remove emoji from the text, which was added into the product description by the sellers. This was followed by removing punctuations and stopwords. Negative stopwords were not removed as the paper [30] suggests that it give better performance. Then the characters other than alphanumeric and white spaces were removed. Finally, the whole text is converted to lower case.

5.2.1.2 Text Encoding

Text encoding is the process of converting text into numbers/vectors so that the machines can generate patterns by preserving the contextual meanings. The words or sentences need to be encoded into numbers as computers do not understand natural languages. For the categorical data like gender, brand, color, and categories, we performed one-hot encoding where the words were represented as binary vectors. CountVectorizer from scikit-learn was used to get the one-hot encoding for these categories [26]. The rest of the text columns are transformed into vectors using embeddings. Embeddings convert high-dimensional data into vectors while preserving the essential information contained in the original data. We tried using bag-of-words, TFIDF and word2vec and found that TFIDF performed better for my use case. So TFIDF was used for creating feature vectors for non-categorical text. It is the product of term frequency(TF) to inverse document frequency(IDF) [21] where:

$$TF(t) = \frac{\text{Number of times term t appears in a document}}{\text{Total number of terms in the document}} \tag{1}$$

$$IDF(t) = \log_e \frac{\text{Total number of documents}}{\text{Number of documents with term t in it}} \tag{2}$$

5.2.2 Two stage price prediction system

This new proposed pipeline given in Figure 1 can be divided into two stages. The first is the image classification part, and the second is the price prediction part. This approach works better for large uncategorized datasets.

5.2.2.1 Image Classification

The objective of this stage is to find products that are similar to the input image and then help us predict the price more accurately. In this stage, the images are classified into different product categories using the classification model. The Keras InceptionResNetV2 model, which is pre-trained on ImageNet, is used to build the classifier. The training set results are saved in a database so that this step is not performed every time for different test sets. Then the classification of the test image is performed, which gives the categories it belongs to. The training data that relate to these categories are selected and passed to the second stage for price regression.

5.2.2.2 Price Prediction

A training dataset is created from the subset of the product metadata that belongs to the same class of categories as the test image. Then, this text metadata is processed as mentioned in the section 5.2.1 which creates the feature vectors for all the columns in the metadata. Finally, these feature vectors are stacked together horizontally to form a final feature vector [6], and this final vector is used to train the models like XGBoost [4] and LightGBM [15] to predict the price of the product.

5.2.3 Price Prediction by feature fusion

In this approach, instead of creating a subset of the dataset using image classification, we use the entire dataset to train the regression model. we extracted features from the images using InceptionResNetv2 and *stacked* them horizontally with the text features extracted from metadata, as mentioned in the section 5.2.1 using SciPy's *hstack* [37]. This combined feature vector is provided as the input for training the XGBoost [4] and LightBGM [15] regression models. To predict the price of a new product, we will create a feature vector similar to one created for training by stacking image and text features. Then we run the prediction function of the trained model to find the price of the new product.

Implementation for all the models mentioned in this section is available on Github[1].

6 Experimental Results

6.1 Feature Selection

One of the primary goals of this thesis is to find the properties of the product that contribute to the price, as indicated in the subquestion 1. We used various feature selection methods associated with the XGBoost algorithm to determine the feature importance and combined them to complete our final analysis.

The XGBoost built-in feature importance method was used first. For this, we used all attributes which were having either integer, boolean or categorical values. Categorical values were converted into numerical labels using LabelEncoder [26]. This data was used to fit the XGBoost model and call the *feature_importance_* attribute to get the feature importance. This result was sorted and plotted as shown in Figure 9 [28].

The second method to identify the feature importance was to use the scikit-learn's permutation-based methods in XGBoost. This method generates different permutations of all input features by random shuffling and computes the difference in model performance. The plot of permutation importance and the heatmap based on correlation is shown in Figure 10 and Figure 11.

The last method which was implemented uses the Shapley values. This method is a model-independent feature and is widely used in game theory to allocate credit to the output of the model based on various input features [28]. The feature importance is plotted in Figure 12. By comparing all the feature importance results mentioned above, we could conclude that the features 'expensive_brands', 'luxury_brands' and 'cheap_brands' are crucial for price prediction. The category_1 that exclusively lists subcategories of products was more influential than the category_0, which broadly classified the products. Also, the features like dominant_material, care_instructions, and color showed some significance on price prediction in that order. On the other hand, size, is_in_stock, variant_sku, images_count, and inventory had relatively no influence on model output. So these columns were removed for the price prediction.

[1]https://github.com/aghilkp91/Price_prediction_using_image_and_metadata_analysis.git

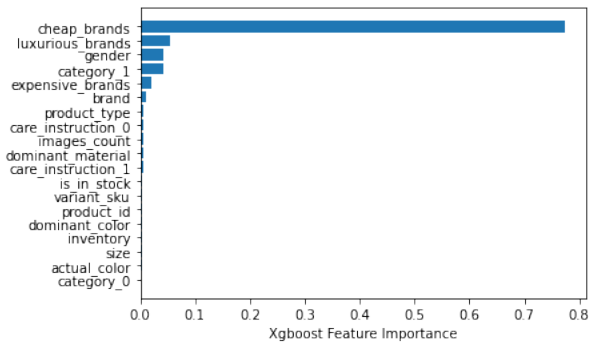

Figure 12: XGBoost Feature Importance

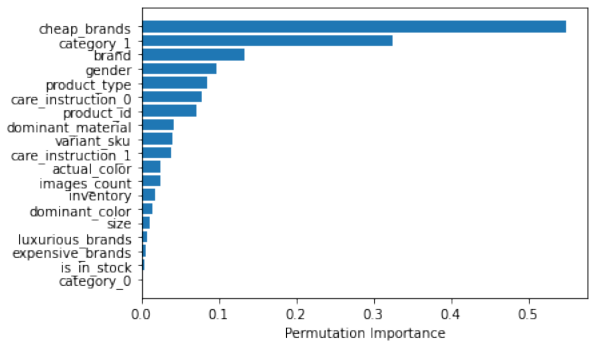

Figure 13: Permutation Feature Importance

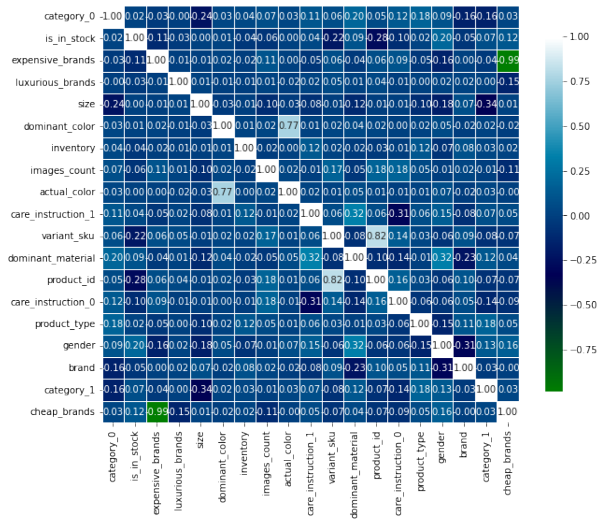

Figure 14: Product specifications

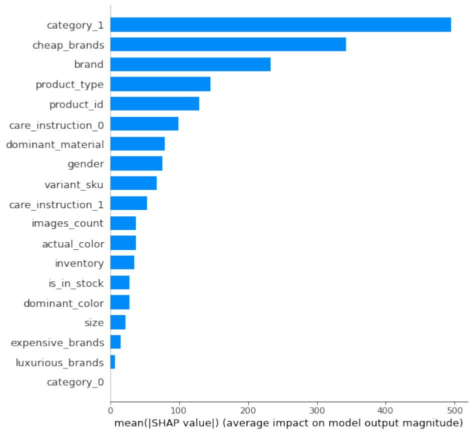

Figure 15: SHAP Values Feature Importance

6.2 Model Tuning and Parameters

Several techniques were used to tune the parameters for the price prediction models using VGG16 and InceptionResNetv2. By testing out stochastic gradient descent, adam and RMSprop, we got the best results with the RMSprop optimizer. The effects of random hidden units during training were reduced by adding a dropout in the output layer that helped in reducing overfitting [3]. The parameters which were tuned were minibatch size, trainable hidden layers, learning rate, discounting factor, and epochs. For image classification using InceptionResNetv2, most of the parameters were not altered. GridSearchCV from sklearn was used to tune the parameters for XGBoost and LigthGBM. Different values of 'eta(learning_rate)', 'gamma', 'subsample', 'reg_alpha', 'max_depth' and 'n_estimators' were produced to find the best performing parameters for XGBoost. The parameters tuned for LightGBM was 'learning_rate', 'n_estimators' and 'num_leaves'.

6.3 Evaluation

To have a fair comparison between the different models, we divided the dataset into three separate segments having 60%(training), 20%(validation), and 20%(test) of data. These segments were saved and used among all four models we have discussed above. The validation set was used in the case of feature selection and hyperparameter tuning.

Three different metrics were used to evaluate the performance of all models: mean absolute error(MAE), root mean squared error(RMSE), and coefficient of determination(R^2). MAE is the average difference between the predicted and actual value across the whole test set as represented in equation 3. MAE looks at the absolute value, and the lower value indicates better accuracy. Generally, MAE is considered when we are not worried about the outliers [31].

$$MAE = \frac{1}{M}\sum_{i=1}^{M} \mid y_i - \hat{y}_i \mid \tag{3}$$

RMSE measures the square root of the average square of predicted to the actual value as stated in equation 4. RMSE adds more weight to big errors and sums them, and then gets the average of them all. This measure is a better criterion for evaluating a dataset with outliers. The lower values of RMSE indicate better model performance.

$$RMSE = \sqrt{\frac{1}{M}\sum_{i=1}^{M}(y_i - \hat{y}_i)^2} \tag{4}$$

R^2 represents the proportion of variability in the dependent values that are exhibited by the regression model as quoted in the equation 5. The calculated value comes between 0 and 1, and the model performance is directly proportional to the value[5]. $\hat{y}_i$ represents the predicted value, and $\bar{y}$ represents the mean value of y. R^2 helps us to understand how well our selected independent variables explain the dependent variables. The value of R^2 increases with the addition of independent variables that may result in redundant variables in the model.

$$R^2 = 1 - \frac{\sum (y_i - \hat{y}_i)^2}{\sum (y_i - \bar{y})^2} \tag{5}$$

Model	RMSE	MAE	R^2
VGG16	1054.81	692.92	0.23
InceptionResNetv2	1116.36	735.74	0.40
Two Stage XGBoost	748.99	445.80	0.74
Two Stage LightBGM	**709.96**	**437.88**	**0.75**
Feature Fusion XGBoost	759.16	452.08	0.73
Feature Fusion LightBGM	743.01	463.76	0.74

Table 1: Results of single category dataset

Model	RMSE	MAE	R^2
VGG16	1127.27	719.11	0.18
InceptionResNetv2	1095.19	716.69	0.22
Two stage XGBoost	**691.94**	**416.87**	**0.78**
Two stage LightBGM	704.57	426.58	0.76
Feature Fusion XGBoost	702.65	430.44	0.77
Feature Fusion LightBGM	741.57	459.97	0.74

Table 2: Results of multi-category dataset

The results of the single category price prediction regression models are given in Table 1 and the results for the multiple category price prediction regression model are given in Table 2. For models which did not consider the metadata into account, the base VGG16 model performed best with an MAE of INR 692.92 for a single category dataset. The XGBoost with tuned parameters for the proposed model gave the overall best score with an MAE of INR 416.87 on the prices ranging from INR 399 to INR 25500 and supports our idea of using metadata along with images for price prediction. The R^2 score was better for models with metadata. This score was understandable since R^2 values were highly dependable on the more number of independent variables in the model.

It is interesting to note that the base models of VGG16 and InceptionResNetv2, which did not consider the metadata, performed better in single category datasets than the multiple category datasets. It also came to our surprise that the VGG16 outperformed the more complex InceptionResNetv2 in a single category for our dataset. A big leap in performance was achieved for the models that considered the metadata. The proposed method of two-stage price prediction obtained stronger results than the price predictions using feature fusion in both single and multiple category datasets.

Both the models predicting price by taking images and product data into consideration performed better on multiple category environments than single-category environments. This result proves that the idea of combining both images and metadata for price prediction is a solid technique for attaining better performance and producing strong results on datasets having multiple categories and product specifications. The two-stage price prediction technique takes more time to run than all other methods for our dataset. But we believe that for a huge uncategorized dataset, this proposed method will perform better since the training data set is less compared to other models. If our use-case is time sensitive, we can use the price prediction using feature fusion as it performs almost as well as the two-stage price prediction method with significantly less time to complete for smaller datasets.

7 Discussions and Future Work

In this thesis, we evaluated a fashion dataset with images and product properties and created several models for price prediction. We proposed a pipeline with a two-stage price prediction system that first finds similar images using a classification method and then analyzes the metadata to suggest a price. This proposed method works well with datasets containing products from multiple categories. We also found that the models that take both images and metadata into consideration perform better than those that only use images to predict the price. The product properties like brand, product type, product category, gender, and materials were the crucial characteristics that contributed to the price. We created categories based on brand and price to divide the products into luxury, expensive and cheap brands. These features turned to be extremely helpful for model performance.

The idea of using product images and properties to predict prices worked well with multiple categories, as expected, and even performed better than single-category datasets. While analyzing the time required to run the models, we found that the models considering both images and metadata ran the longest. But it gave a significant improvement in the results compared to the time difference. If the application is time-sensitive (like helping the user predict the bid price for a secondhand item), we would prefer the model using feature fusion which stacks image features and metadata features together to predict the price. And for applications like predicting the price of diamonds, where accuracy is more important than time, we can use the two-stage price predictor.

7.1 Future Work

7.1.1 To improve performance

For this work, we used InceptionResNetv2 for creating image features. We can also include other features to improve efficiency. Multiple feature extraction methods can be used to extract different distinct image features, which can be stacked together and run a dimensionality reduction technique to enhance the network's speed. This technique should improve the efficiency of the regression models. The downside to this is as we add more features, the model becomes more complex and takes more time to run. Also, in the two-stage prediction method, we can find similar images using a feature vector and generate a soft rank according to the similarity, which could be added to the metadata as a feature vector. Another area to improve is to use multiple images for image feature extraction. But for this approach, we need to have consistent images throughout the dataset. If we have three images per product, every image across the dataset should be representing the same features. For example, throughout the dataset, image_1 represents the front image, image_2 represents the side profile, and image_3 represents the closeup product details. If we can create a dataset with consistent images across, this could bring huge improvements to the model performance.

7.1.2 To improve time complexity

To improve the time complexity of the two-stage price prediction model, we can first find all the combinations of image categories that can come together. Then for all the valid combinations of categories, we can train the regression models and save them in our database. When the user needs to predict the price of the new product, we can find the categories of that new product using the classifier. Then we retrieve the pre-trained model from our database with those predicted categories and predict the price of the new product. This technique will save the time taken to train the subset of metadata during the price prediction process.

7.2 Applications

Real worlds applications of this work include predicting prices for a newer product compared to the competing products in the market. For example, suppose a company introduces a new product to market. In that case, it can be compared with similar products by other companies given its unique advantages/improvements and try to predict the right price it can fit into. Also, we can use these models to predict the price of a product introduced with new improvements from its older version, comparing with its old products as well as products from the market with similar features. There are plenty of applications on secondhand marketplaces like eBay, Marketplaats for suggesting the sellers the right price bracket for the products given its conditions and status. This application of the predicted model can also help users indicate a correct bid price for a product listed in the marketplace by comparing its features and age along with the look and feel given the images. Similar applications can be developed for vehicle reseller websites and comparison sites.

References

[1] Afshine Amidi and Shervine Amidi. The evolution of image classification explained. https://stanford.edu/~shervine/blog/evolution-image-classification-explained, Accessed: '20-10-2021'.

[2] Ananda, Cefa Karabağ, Aram Ter-Sarkisov, Eduardo Alonso, and Constantino Carlos Reyes-Aldasoro. Radiography classification: A comparison between eleven convolutional neural networks. In *2020 Fourth International Conference on Multimedia Computing, Networking and Applications (MCNA)*, pages 119–125, 2020.

[3] Steven Chen, Edward Chou, and Richard R. Yang. The price is right: Predicting prices with product images. *CoRR*, abs/1803.11227, 2018.

[4] Tianqi Chen and Carlos Guestrin. XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, pages 785–794. ACM, 2016.

[5] Akshita Chugh. Mae, mse, rmse, coefficient of determination, adjusted r squared — which metric is better? https://medium.com/analytics-vidhya/mae-mse-rmse-coefficient-of-determination-adjusted-r-squared-which-metric-is-better-cd0326a5697e, Accessed: 2021-10-14, December 2020.

[6] Victor Coscrato, Marco Henrique de Almeida Inácio, and Rafael Izbicki. The nn-stacking: Feature weighted linear stacking through neural networks. *Neurocomputing*, 399:141–152, 2020.

[7] Rostyslav Demush. A brief history of computer vision (and convolutional neural networks). https://hackernoon.com/a-brief-history-of-computer-vision-and-convolutional-neural-networks-8fe8aacc79f3, February 2019.

[8] William Dodds, Kent Monroe, and Dhruv Grewal. Effects of price, brand, and store information on buyers' product evaluations. *Journal of Marketing Research*, 28, 08 1991.

[9] Ian A. Naccarella Emmie E. Kohoe and Javier Raygada. Product price suggestions for online marketplaces, 2018.

[10] Ahmed Fathalla, Ahmad Salah, Kenli Li, Keqin Li, and Piccialli Francesco. Deep end-to-end learning for price prediction of second-hand items. *Knowledge and Information Systems*, pages 1 – 28, 2020.

[11] Shetu Guha and S M Rafizul Haque. *Performance Comparison of Machine Learning-Based Classification of Skin Diseases from Skin Lesion Images*, pages 15–25. 03 2020.

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015.

[13] IBM. What is computer vision? OCtober 2018. https://www.ibm.com/topics/computer-vision.

[14] B Jeevan, E Naresh, B P Vijaya kumar, and Prashanth Kambli. Share price prediction using machine learning technique. In *2018 3rd International Conference on Circuits, Control, Communication and Computing (I4C)*, pages 1–4, 2018.

[15] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. Lightgbm: A highly efficient gradient boosting decision tree. *Advances in neural information processing systems*, 30:3146–3154, 2017.

[16] Keras. Inceptionresnetv2. https : / / keras.io / api / applications / inceptionresnetv2/, Accessed: 2021-09-10, 2019.

[17] Keras. Vgg16 and vgg19. https://keras.io/api/applications/vgg/, Accessed: 2021-08-30, 2019.

[18] Huijae (Jay) Kim. Deep learning - image processing with vgg16 and keras for house price prediction. https://jaykimhuijae.wixsite.com/data-science/single-post/2017/12/19/deep-learning-image-processing-with-vgg16-and-keras, December 2017.

[19] Zona Kostic and Aleksandar Jevremovic. What image features boost housing market predictions? *IEEE Transactions on Multimedia*, 22(7):1904–1916, 2020.

[20] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. *Commun. ACM*, 60(6):84–90, May 2017.

[21] K. Latha. *Experiment and evaluation in information retrieval models*. 01 2017.

[22] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Backpropagation applied to handwritten zip code recognition. *Neural Computation*, 1(4):541–551, 1989.

[23] Catherine Lovering. How does pricing variables affect a business? September 2018. https://smallbusiness.chron.com/cup-coffee-pricing-strategies-48911.html.

[24] Hussain Mujtaba. Introduction to resnet or residual network). https://www.mygreatlearning.com/blog/resnet/, Accessed: '15:09:2021', September 2020.

[25] Nutan. Deep convolutional networks vgg16 for image recognition in keras. https://medium.com/@nutanbhogendrasharma/deep-convolutional-networks-vgg16-for-image-recognition-in-keras-a4beb59f80a7, August 2020.

[26] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[27] Omid Poursaeed, Tomáš Matera, and Serge Belongie. Vision-based real estate price estimation. *Machine Vision and Applications*, 29(4):667–676, April 2018.

[28] Piotr Płoński. Xgboost feature importance computed in 3 ways with python. https://mljar.com/blog/feature-importance-xgboost/, Accessed: 2021-10-02, August 2020.

[29] Bharath Raj. A simple guide to the versions of the inception network. https://towardsdatascience.com/a-simple-guide-to-the-versions-of-the-inception-network-7fc52b863202, Accessed: 2021-09-08, May 2018.

[30] Johan Reitan, Jørgen Faret, Björn Gambäck, and Lars Bungum. Negation scope detection for twitter sentiment analysis. pages 99–108, 01 2015.

[31] Sab. Mae vs mse vs rmse. http://zerospectrum.com/2019/06/02/mae-vs-mse-vs-rmse/, Accessed: 2021-10-12, June 2019.

[32] Joseph Sill, Gabor Takacs, Lester Mackey, and David Lin. Feature-weighted linear stacking, 2009.

[33] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition, 2015.

[34] Devendra Swami, Alay Dilipbhai Shah, and Subhrajeet K B Ray. Predicting future sales of retail products using machine learning, 2020.

[35] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alex Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning, 2016.

[36] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions, 2014.

[37] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, 2020.

[38] Quanzeng You, Ran Pang, Liangliang Cao, and Jiebo Luo. Image-based appraisal of real estate properties. *IEEE Transactions on Multimedia*, 19(12):2751–2759, 2017.