AIRBNB RESEARCH: AN ANALYSIS IN NEXUS BETWEEN VISUAL DESCRIPTION AND PRODUCT RATING

Chun Kit Fu[1], Yu Sun[2]

[1]7 Lakes high school, 9251 S Fry Rd, Katy, TX 77494
[2]Computer Science Department, California State Polytechnic University,
Pomona, CA 91768

ABSTRACT

Hosts are often desperate to find ways to rent their house, However, most of them do not have possess the knowledge of knowing what type of image cover would grasp the attention of their customer. Gilded by these needs, I have designed an application that uses machine learning to find the relationship between the images and their rating [1]. I first used JSON to convert the HTML file resource to a format where we can use in python for web scraping [2]. This paper designs an application tool to find all the object or characters inside images by web scraping and changes it into a model for machine learning [3]. Applied our application to predict the rating and conducted a qualitative evaluation of the approach. In order to prove our result, I imported an image from Airbnb and found its rating. It turns out that the predicted rating is extremely close to the real rating, Proving The system's usability.

KEYWORDS

Web scraping, Machine learning, Airbnb

1. INTRODUCTION

Rental service, or homestay service, is a common sight these days. Its history are so old that the origin is probably lost in prehistory. In medieval times, most land was owned either by the King or by lords, and almost all farmers were tenant farmers who paid a rent - usually a percentage or portion of crops grown - in return for living on and farming the land with. Nowadays , rental service is a popular form of hospitality and lodging, whereby visitors share a residence with a local of the area to which they are traveling.

Airbnb is a vacation rental company that operates an online marketplace focused on short-term homestays and experiences [4]. Its features compared with its competitor include Reservation screening,$3M damage protection, Pet damage, 24-hour safety line and much more. On the app, the host could offer to rent their housing by uploading information of their house in terms of images, price, location, etc. However, there are still many hosts who are still having a hard time determining what they need to grasp the attention of their customers. According to RMA, Rental Market Analysis, images play a major role in what customers think of the housing [5]. In the hopes of helping hosts in Airbnb find the best image, we have created a program called hbnb to find the relationship between the attributes inside the images and its rating by using web scraping

David C. Wyld et al. (Eds): CCSEA, AIFU, EMSA, NLCAI, NCOM, SIPRO, SEA, DKMP, BDML, BIoT, CLOUD - 2023
pp. 229-238, 2023. CS & IT - CSCP 2023 DOI: 10.5121/csit.2023.130520

and machine learning [6]. This program was designed to help many real estate agents and hosts to find the images that will produce the highest rating, so it would save many host's time.

Some techniques and systems proposed by previous methods such as regression lines, hedonic models, or time-series methods all allow users to predict rich attributes in real estate images by using their own unique way [7]. However, they all have their own limitations in certain field and are inconvenient in some ways. Regression lines assume that the price is a weighted sum of property characteristics, and are unable to address non-linearity or detect outliers. Hedonic models, or hedonic regression, is a revealed preference method for estimating demand or value [8]. However, it's rather restrictive, imposing uniformity of coefficients across both space and time. They both a lack of datasets for predicting rich attributes such as landscaping, restroom, ceilings, hardwood floors, fireplaces, etc., Time series method on the other hand is a really handy tool for forecasting purposes since it's high in accuracy and simplicity. However, it also requires more skill than regression analysis since model needs to be adapted according to the historical database. It has been proven that it's only efficient when inaccessible, meaning that if a model has been built on historical data, it cannot be used to predict future values or trends because no one can guarantee that the historical data will remain the same as time passes. However, in our program, we used a more variety of what's inside the picture to find the connection between image and rating, but not as heavy(overfit) as time-series, so it would take less time. The JSON value we extracted for web scraping includes the column labels that finds the furniture, and then we save the module into a file named 'aribnb_model.sav' by defining a function called save_module [9]. This process that save the model for future use save much more time than most method and are less likely to overfit. In addition, the image can change adjust base on the need of the host or real estate, so it won't be so reliant on historical data like time-series method in order to construct the models.

In this paper, we follow the same line of research by the use of machine learning and web scraping. Our goal is to successfully predict the outcome and identify the similarity in the images of high rating. Therefore, the feature included in this project are web scrapping and prediction of rating. In the Web scrapping feature, the user just needs to provide a URL to the software, and it will automatically extract the JSON value that contains the image attribute. With those resources, the program will be able to download the corresponding image from the internet. In this case, the user does not need to load the image into the software manually, so it would reserve much time for the user. The rating prediction feature on the other side can be used to find the rating of a house given the provided image URL and ID. Our methods are inspired by the IRIS flower method, so we used lots of computational methods and class methods to operate our project. There are many good methods and feature that build up the project. For the Method, we First used a method like Univariate Selection by using Scikit-learn [10]. In addition, we used the pandas method to build up the data frame.

In two application scenarios, we demonstrate how the above combination of techniques increases the accuracy of rating by inputting another ID along with its image to compare the output with its rating. First, we show the usefulness of our approach by a comprehensive case study on the data and rating of airbnb. Second, We analyze the relationship between the attribute inside the image and the rating of the airbnb. Then we applied machine learning to produce a table containing an x variable from the input data frame to count the picture attribute as the input data and a y variable as the rating for the output variable. Lastly, to prove that our project worked successfully, We used photos from Airbnb to forecast each listing's rating, and then compared those predictions to the listings' actual ratings to demonstrate our accuracy. We repeat this process a total of 5 times for analysis purpose and proved that the percent error between the predicted rating from our model and the actual rating from airbnb are extremely close. This result shed light on the conclusion that the project has worked well and proved our result to be accurate.

The paper is organized as the followed, we put the abstract the in front of all sections. Section 1 gives the introduction to what we are doing (background, benefits, etc.), what previous method exist, what feature is within our application and how we proved our results. Section 2 gives the details on the challenges that we met during the experiment and designing the sample along with how we conquered them; Section 3 focuses on the details of our solutions corresponding to the challenges that we mentioned in Section 2 and what is the component of our application; Section 4 presents the relevant details about the experiment we did, following by presenting the related work in Section 5. Finally, Section 6 gives the conclusion remarks, as well as pointing out the future work of this project.

2. CHALLENGES

In order to build the project, a few challenges have been identified as follows.

2.1. Libraries

A Python library is a reusable chunk of code that you may want to include in your programs in order to save time. Unlike other coding languages like C++, Python libraries do not pertain to any specific context in Python. A 'library' is just a loose description of a collection of core modules. In this project we used many libraries like BeautifulSoup, JSON, Image from PIL, torch, glob, pandas, requests, OS, wget, and pickle to save time and perform the function we needed. To find all the Libraries, it took quite a bit of time to find all the features of the libraries and what we needed each library to perform. In addition, when importing is not enough, we took a while to understand that we needed to install some libraries before importing it like wget, which was not usable until we installed it.

2.2. Web Scraping

JSON is an open standard file format and data interchange format that uses human-readable text to store and transmit data objects consisting of attribute–value pairs and arrays. In order to use machine learning to achieve our goal, we had to first use web scraping to extract the right JSON value that contains the information of the items in the webpage to make into readable text [11]. It was a difficult task since we had quite a bit of trouble understanding what sections in the JSON value are the sections that need to be extracted and what's the best pre-build model (python library) to use. JSON value contains a huge data set and are The two primary, made up of keys and values. Together they make a key/value pair, and finding the right information is quite difficult. In addition, converting the value into a table is also quite troublesome as it involves the use of libraries and changing it into a readable data file and organization. in the end, we decided to used BeautifulSoup to get the content (the whole HTML file) in airbnb_url in the form of parser(html.parser) to locate the image resource in the JSON.

2.3. Machine Learn

Machine Learning is the instrumental tool that we use to achieve our goal, it can be used to analyze large datasets and conduct model selection in the context of causal inference. After using the beautiful soup method to extract the JSON value to complete web scraping and creating an input data using, we tried many ways to fit the input and output. For machine learning, our input data is the data frame created for the count item function, while the output data is the rating of the images. The job of machine learning is to connect the variety of items inside the image that we downloaded from the web scraping with its rating and predict what is the rating if we input an image inside. It was a difficult task since it requires meticulous attention to optimize an algorithm and debugging machine learning algorithms is difficult because the code includes multiple dimensions where information can be incorrect.

3. SOLUTION

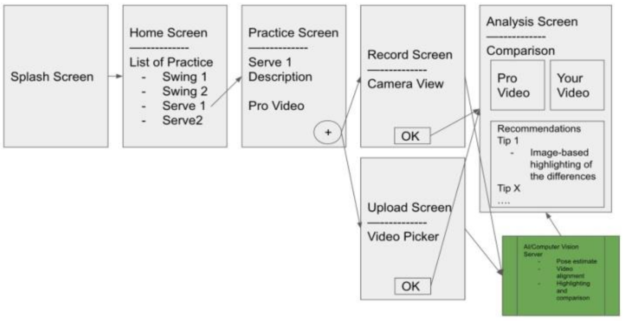
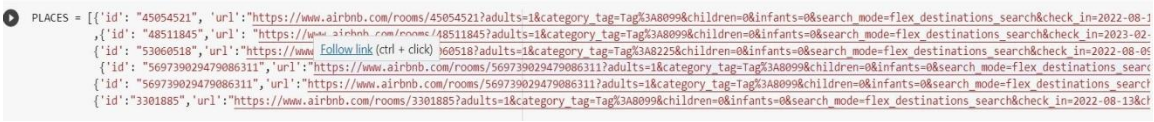


Figure 1. Overview of the solution

For this project, its application can be separated the application into 5 components, the raw data, web scraping, counting items, creating input dataset and machine learning

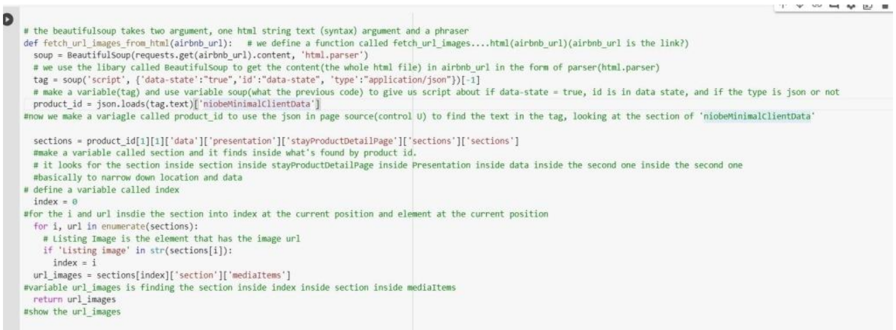
The raw data is the most fundamental building block of our program. We got these data from a rental company name airbnb directly. We used airbnb's housing information to create a big list of all places for future machine learning by adding the ID, URL and rating of the image. The idea was to use machine learning to connect the attribute inside the URL of the picture along with it's rating.



```
PLACES = [{'id': "45054521", 'url':"https://www.airbnb.com/rooms/45054521?adults=1&category_tag=Tag%3A8099&children=0&infants=0&search_mode=flex_destinations_search&check_in=2022-08-1
          ,{'id': "48511845",'url': "https://www.airbnb.com/rooms/48511845?adults=1&category_tag=Tag%3A8099&children=0&infants=0&search_mode=flex_destinations_search&check_in=2023-02-
          {'id': "53060518",'url':"https://www  Follow link (ctrl + click) 060518?adults=1&category_tag=Tag%3A8225&children=0&infants=0&search_mode=flex_destinations_search&check_in=2022-08-0
           {'id': "569739029479086311",'url':"https://www.airbnb.com/rooms/569739029479086311?adults=1&category_tag=Tag%3A8099&children=0&infants=0&search_mode=flex_destinations_searc
          {'id': "569739029479086311",'url':"https://www.airbnb.com/rooms/569739029479086311?adults=1&category_tag=Tag%3A8099&children=0&infants=0&search_mode=flex_destinations_searc
          {'id':"3301885",'url':"https://www.airbnb.com/rooms/3301885?adults=1&category_tag=Tag%3A8099&children=0&infants=0&search_mode=flex_destinations_search&check_in=2022-08-13&c
```

Figure 2. URL

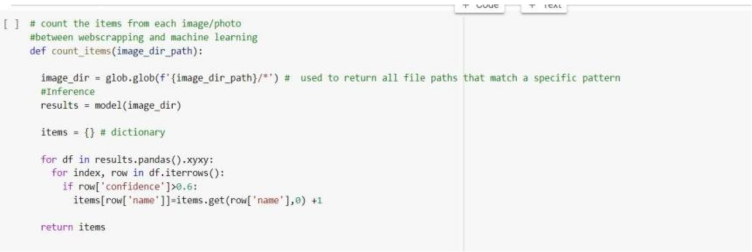
For the web scraping portion, it is also very important and well connect to the raw data section. This part of the code is used to find the attribute inside a URL of the raw data set. To do web scraping, we fetched the URL from the webpages inside the raw dataset (PLACES) by using a python library called BeautifulSoup to get the content (the whole HTML file) in airbnb_url in the form of parser(html.parser) to locate the image resource in the JSON value inside the page source to extract that specific section of JSON to return the URL of the images. Next, we download the image from the URL that we got from the previous step. Each place (web page) will have its own folder for images, and the folder name will be the 'id' of these places. Lastly, we will fetch all Places by repeating the previous 2 steps in a loop for every element in our raw dataset to ensure that we have the folder for each of the webpage.



```
# the beautifulsoup takes two argument, one html string text (syntax) argument and a phraser
def fetch_url_images_from_html(airbnb_url):   # we define a function called fetch_url_images....html(airbnb_url)(airbnb_url is the link?)
  soup = BeautifulSoup(requests.get(airbnb_url).content, 'html.parser')
  # we use the libary called BeautifulSoup to get the content(the whole html file) in airbnb_url in the form of parser(html.parser)
  tag = soup('script', {'data-state':"true",'id':"data-state", 'type':"application/json"})[-1]
  # make a variable(tag) and use variable soup(what the previous code) to give us script about if data-state = true, id is in data state, and if the type is json or not
  product_id = json.loads(tag.text)['niobeMinimalClientData']
#now we make a variagle called product_id to use the json in page source(control U) to find the text in the tag, looking at the section of 'niobeMinimalClientData'

  sections = product_id[1][1]['data']['presentation']['stayProductDetailPage']['sections']['sections']
  #make a variable called section and it finds inside what's found by product id.
  # it looks for the section inside section inside stayProductDetailPage inside Presentation inside data inside the second one inside the second one
  #basically to narrow down location and data
# define a variable called index
  index = 0
#for the i and url insdie the section into index at the current position and element at the current position
  for i, url in enumerate(sections):
    # Listing Image is the element that has the image url
    if 'Listing image' in str(sections[i]):
      index = i
  url_images = sections[index]['section']['mediaItems']
#variable url_images is finding the section inside index inside section inside mediaItems
  return url_images
#show the url_images
```

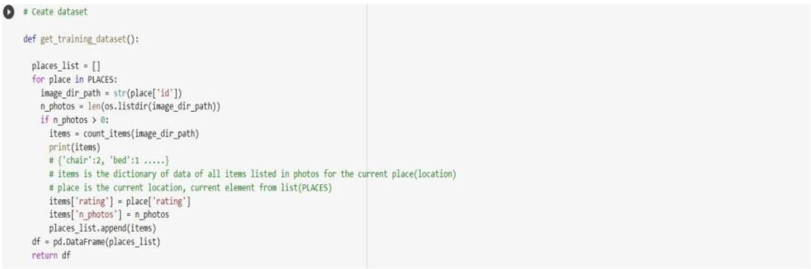
Figure 3. Webpage 1

The count item section is rather simple compared with the rest, we imported some pre-build model (python libraries) like Yolov5 and glob, to complete our objective. We used yolov5 to count each different items in an image by dividing images into a grid system to count the input dataset and glob to return all file paths that match a specific pattern.



```
[ ] # count the items from each image/photo
    #between webscrapping and machine learning
    def count_items(image_dir_path):

      image_dir = glob.glob(f'{image_dir_path}/*') #  used to return all file paths that match a specific pattern
      #Inference
      results = model(image_dir)

      items = {} # dictionary

      for df in results.pandas().xyxy:
        for index, row in df.iterrows():
          if row['confidence']>0.6:
            items[row['name']]=items.get(row['name'],0) +1

      return items
```

Figure 4. Webpage 2

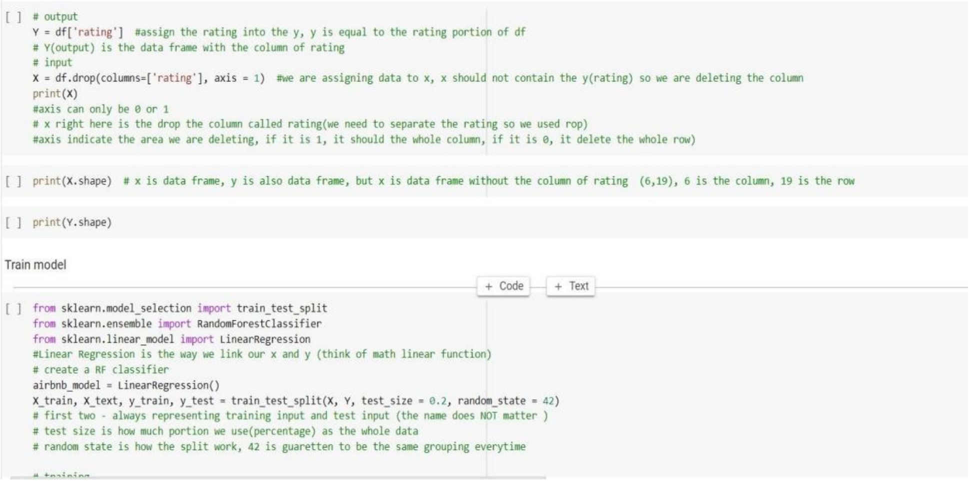


```
# Ceate dataset

def get_training_dataset():

  places_list = []
  for place in PLACES:
    image_dir_path = str(place['id'])
    n_photos = len(os.listdir(image_dir_path))
    if n_photos > 0:
      items = count_items(image_dir_path)
      print(items)
      # {'chair':2, 'bed':1 .....}
      # items is the dictionary of data of all items listed in photos for the current place(location)
      # place is the current location, current element from list(PLACES)
      items['rating'] = place['rating']
      items['n_photos'] = n_photos
      places_list.append(items)
  df = pd.DataFrame(places_list)
  return df
```

Figure 5. Webpage 3

However, before we start counting, we would have to create an input dataset in order for the program to know what to look for. We created a function name get_training_dataset specifically for this by using a python libraries name pandas (short for pd).The Pandas library is an open source Python package that is most widely used for data science/data analysis and machine learning tasks, in this scenario, it made a perfect model to get the items and quantities we need into a data frame to let the application know the items it needs to count

Lastly, after completing all the previous steps, the final and most important part is to connect them using machine learning. We created a training dataset by using the input x value (data frame from the previous step) and the output y value (the rating value from the raw dataset (Places) to build a model to save from future use and use the model to predict rating from new PLACES that were not included in the raw dataset.



```
[ ] # output
    Y = df['rating']  #assign the rating into the y, y is equal to the rating portion of df
    # Y(output) is the data frame with the column of rating
    # input
    X = df.drop(columns=['rating'], axis = 1)  #we are assigning data to x, x should not contain the y(rating) so we are deleting the column
    print(X)
    #axis can only be 0 or 1
    # x right here is the drop the column called rating(we need to separate the rating so we used rop)
    #axis indicate the area we are deleting, if it is 1, it should the whole column, if it is 0, it delete the whole row)
```

```
[ ] print(X.shape)  # x is data frame, y is also data frame, but x is data frame without the column of rating  (6,19), 6 is the column, 19 is the row
```

```
[ ] print(Y.shape)
```

Train model

```
[ ] from sklearn.model_selection import train_test_split
    from sklearn.ensemble import RandomForestClassifier
    from sklearn.linear_model import LinearRegression
    #Linear Regression is the way we link our x and y (think of math linear function)
    # create a RF classifier
    airbnb_model = LinearRegression()
    X_train, X_text, y_train, y_test = train_test_split(X, Y, test_size = 0.2, random_state = 42)
    # first two - always representing training input and test input (the name does NOT matter )
    # test size is how much portion we use(percentage) as the whole data
    # random state is how the split work, 42 is guaretten to be the same grouping everytime

    # training
```

Figure 6. Webpage 4

4. EXPERIMENT

4.1. Experiment 1

In this project, we utilized machine learning to create a table with a y variable as the rating for the output variable and an x variable from the input data frame to count the picture attribute as the input data. We used photos from Airbnb to forecast each listing's rating, and then compared those predictions to the listings' actual ratings to demonstrate our accuracy. We forecast the rating of the existing airbnb using photographs to support our claim. Then, we repeat the process four more times with different datasets (image). For analysis, we take the predicted values and their corresponding actual values from Airbnb. To compare them and determine the correctness of our mode, we determined the percent inaccuracy. Some Accuracy problem that sill may occur in these predictions because the data sample for machine learning is rather small.

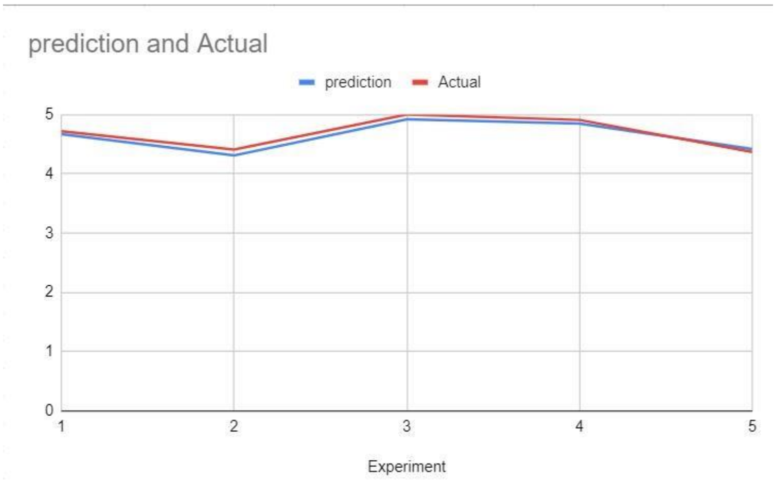
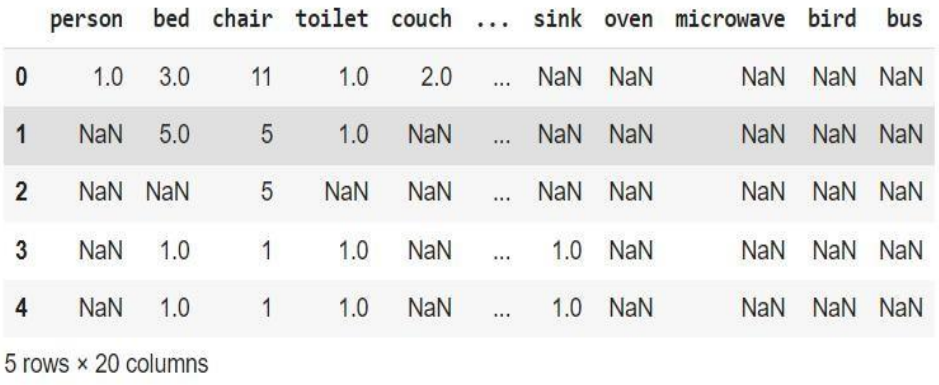


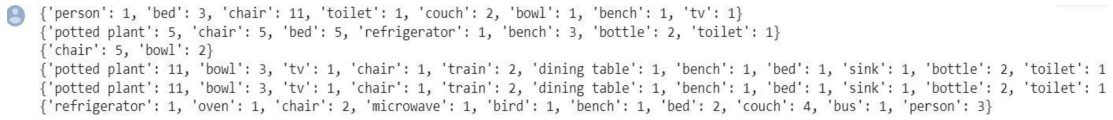
Figure 7. Prediction and Actual of experiment 1



	person	bed	chair	toilet	couch	...	sink	oven	microwave	bird	bus
0	1.0	3.0	11	1.0	2.0	...	NaN	NaN	NaN	NaN	NaN
1	NaN	5.0	5	1.0	NaN	...	NaN	NaN	NaN	NaN	NaN
2	NaN	NaN	5	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN
3	NaN	1.0	1	1.0	NaN	...	1.0	NaN	NaN	NaN	NaN
4	NaN	1.0	1	1.0	NaN	...	1.0	NaN	NaN	NaN	NaN

5 rows × 20 columns

Figure 8. Table of amounts



```
{'person': 1, 'bed': 3, 'chair': 11, 'toilet': 1, 'couch': 2, 'bowl': 1, 'bench': 1, 'tv': 1}
{'potted plant': 5, 'chair': 5, 'bed': 5, 'refrigerator': 1, 'bench': 3, 'bottle': 2, 'toilet': 1}
{'chair': 5, 'bowl': 2}
{'potted plant': 11, 'bowl': 3, 'tv': 1, 'chair': 1, 'train': 2, 'dining table': 1, 'bench': 1, 'bed': 1, 'sink': 1, 'bottle': 2, 'toilet': 1
{'potted plant': 11, 'bowl': 3, 'tv': 1, 'chair': 1, 'train': 2, 'dining table': 1, 'bench': 1, 'bed': 1, 'sink': 1, 'bottle': 2, 'toilet': 1
{'refrigerator': 1, 'oven': 1, 'chair': 2, 'microwave': 1, 'bird': 1, 'bench': 1, 'bed': 2, 'couch': 4, 'bus': 1, 'person': 3}
```

Figure 9. Code of amounts

Our first hypothesis is that the percentage error of the predicted value from our model and actual value from the Airbnb page will within 5 percent. To prove our hypothesis, we use images from the existing airbnb and predict its rating. Next, We perform the same procedure 4 more times using different datasets (image). We take the prediction values, and it's corresponding actual value in airbnb for analysis. We calculated the percent error to contrast them to see the accuracy of our model. This process is completed by using 5 PLACES (experiment image set) from airbnb using its ID and URL. The average percent error turns out to be 1.456 percent, which indicate that the project has worked well.

4.2. Experiment 2

For this experiment, we used machine learning to creating a table with x variable from the input data frame to count the image attribute as the input data and y variable as rating for output variable. To prove our accuracy, we used images from airbnb to predict it's rating to see if it's similar to its actual rating in airbnb, However Accuracy maybe a major issue since the dataset is so small and the result proves it quite well.

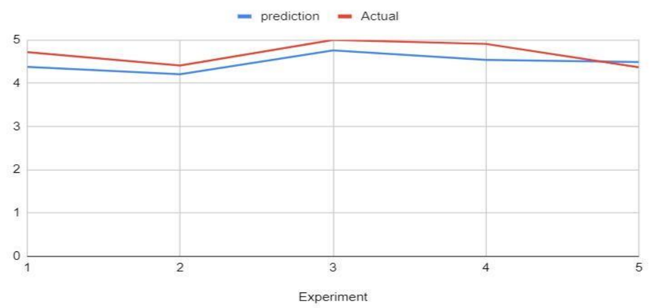


Figure 10. Prediction and Actual of experiment 2

My prediction is that there will be a sizable percentage difference between the anticipated number from our algorithm and the real figure from the Airbnb page. We forecast the rating of the existing airbnb using photographs to support our claim. Then, we repeat the process four more times with different datasets (image). For analysis, we take the predicted values and their corresponding actual values from Airbnb. To compare them and assess the quality of our model, we determined the percent error. The 5 PLACES (experiment image collection) from airbnb is used to finish this process using its ID and URL. The project did not work too well because the average percent error is above 15%.

The experiment proved that all the project's components, including the challenges like python libraries, web scraping, and machine learning, worked successfully, therefore it answered all the issues. The percent error between the projected rating and the actual rating from Airbnb in the examination of 50 sample machine learning data states that the percent error between the predicted rating using the model and the actual rating from airbnb are less than 1.5 percent. The initial objective, or assumption, was a 5% error rate between these two. These experiment findings thus demonstrate that all problems have been resolved and go above and beyond my expectations.

5. Related Work

David Koch, Miroslav Despotovic, Sascha Leiber, Muntaha Sakeena, Mario Döller and Matthias Zeppelzauer of the University of applied science presents Real Estate Image Analysis by the use of real estate applications [12]. Compare to our work, these people used different method and contains feature like classification and others. Although we share some similar feature like object detection, they have a bigger data set and many more feature like 3D reconstruction, classification, and image registration. However, since they have so many features and other consideration, their model run rather slowly compare to ours.

Nick Desmond presents the Predicting Airbnb Review Scores, in his work, he tries to ascertain an adequate customer satisfaction metric and understand what Airbnb hosts can do to improve the customer experience [13]. He used methods like numpy to make function convert Airbnb listing data to integers and Running OLS regression on a 70/30 train/test split of the data yields a very impressive R2 score of 0.95. Regression analysis like these performs exceptionally well for linearly separable data and Easier to implement, interpret and efficient to train. However, these types of methods are usually prone to overfitting and have problems with assumption of linearity between dependent and independent variables. Our project on the other hand are much more defined to find the linearity between dependent and independent variables.

Shunyuan Zhang,Dokyun Lee, Param Vir Singh, And Kannan Srinivasan presents the work to find What Makes a good Image [14]? They studied how Airbnb property demand changed after the acquisition of verified images (taken by Airbnb’s photographers) and explore what makes a good image for an Airbnb property. They used method like deep learning and difference-indifference analyses on an Airbnb panel for their work and has a dataset spanning 7423 properties. Furthermore, they have a total of 12 human-interpretable image attributes that pertain to three artistic aspects—composition, color, and the figure-ground relationship as they find systematic differences between the verified and unverified images. Their result are applicable to any photographers who wish to optimize their image. Compare to our work, they have many more dataset and aspect to put into consider for optimization but are comparatively slow to give the result, which also indicate they have a presumably higher accuracy compare to us.

6. CONCLUSIONS

What we have done is proposing an application to predict the rating of a house by the attribute inside it's image by first collecting the raw data set from a house rental company name airbnb that contains the ID, URL, and the rating of the house. Then we Fetch the URL from the webpage by locating the image resource in the JSON and extracting that section of the JSON value out to return the URL of the image. Afterward, we downloaded the image from the URL that we extracted from the previous step to created a folder so that each place(web page) will have its own folder for image. To ensure that we got all the places, we keep repeating the last 2 steps in a loop for every element in our raw dataset. Next up, we imported the pre-build model Yolov5 to count different items of the image in the imputed data set that we have made into a data frame [15]. Lastly, use machine learning to train data of the input value x(data frame) and the output value y which is the "rating" value from the raw data set. We build a model for future use and apply the application to experiment by predicting rating from a new "place" other than the raw data set. The experiment results indicate its effectiveness and solve challenges because it successfully found all the input value and the rating was extremely close to the actual rating in the airbnb.

There are still quite a few limitations in our application, first of all, the data is considerably small, so the accuracy might not be enough to cover all, secondly all data used for machine learning are from airbnb, although Airbnb is a considerably large website, it might still affect the accuracy, in addition, the current version of our project does not consider location and outdoor view, this might also affect practicality of some user. In the future optimization, we plan on adding more variables like how many customers selected the price rate, how far away is the house, or how many people rated and percent of how many people like it.

In the future, I plan on adding a lot more database to improve the accuracy of the project, another limitation I will solve is the renter's location, I plan in the future to extract the renter's location JSON out of the page source so that it could also be placed in the machine learning.

REFERENCES

[1] Jordan, Michael I., and Tom M. Mitchell. "Machine learning: Trends, perspectives, and prospects." Science 349.6245 (2015): 255-260.

[2] Pezoa, Felipe, et al. "Foundations of JSON schema." Proceedings of the 25th international conference on World Wide Web. 2016.

[3] Glez-Peña, Daniel, et al. "Web scraping technologies in an API world." Briefings in bioinformatics 15.5 (2014): 788-797.

[4] Guttentag, Daniel. "Progress on Airbnb: a literature review." Journal of Hospitality and Tourism Technology 10.4 (2019): 814-844.

[5] Ayouba, Kassoum, et al. "Does Airbnb disrupt the private rental market? An empirical analysis for French cities." International Regional Science Review 43.1-2 (2020): 76-104.

[6] Mahesh, Batta. "Machine learning algorithms-a review." International Journal of Science and Research (IJSR).[Internet] 9 (2020): 381-386.

[7] Ekeland, Ivar, James J. Heckman, and Lars Nesheim. "Identifying hedonic models." American Economic Review 92.2 (2002): 304-309.

[8] Ekeland, Ivar, James J. Heckman, and Lars Nesheim. "Identification and estimation of hedonic models." Journal of political economy 112.S1 (2004): S60-S109.

[9] Nurseitov, Nurzhan, et al. "Comparison of JSON and XML data interchange formats: a case study." Caine 9 (2009): 157-162.

[10] Pedregosa, Fabian, et al. "Scikit-learn: Machine learning in Python." the Journal of machine Learning research 12 (2011): 2825-2830.

[11] Stevenson, Julie S., Gordon C. Bruner, and Anand Kumar. "Webpage background and viewer attitudes." Journal of advertising research 40.1-2 (2000): 29-34.

[12] Koch, David, et al. "Real estate image analysis: a literature review." Journal of Real Estate Literature 27.2 (2019): 269-300.

[13] Luo, Yuanhang, Xuanyu Zhou, and Yulian Zhou. "Predicting airbnb listing price across different cities." (2019).

[14] Zhang, Shunyuan, et al. "What makes a good image? Airbnb demand analytics leveraging interpretable image features." Management Science 68.8 (2022): 5644-5666.

[15] Zhu, Xingkui, et al. "TPH-YOLOv5: Improved YOLOv5 based on transformer prediction head for object detection on drone-captured scenarios." Proceedings of the IEEE/CVF international conference on computer vision. 2021.