

1. Research Problem:

Accurate rent projections for the properties are required by real estate agents, homeowners, and investors. Therefore, we want a reliable price prediction algorithm that produces accurate property prices. These data analysis methods can provide realistic prices to online real estate websites, which they may then show to their customers. This makes the rental industry simple.

2. We require a variety of factors for determining the rent of house. Some of them are as follows: a. The location of the property b. Its name, c. The number of bedrooms, d. The number of bathrooms, e. Community amenities, f. pet friendliness d. rating from past residents and many more;

So we must extract more weighted characteristics for the model in order to perform better. So some of the weighted features that should be taken into account when collecting data are as follows: a. House location b. House name c. number of bedrooms d. number of bathrooms e. rating f. area

3. I intended to use web scraping techniques to get data. For making http requests, I utilized the request library. This library will simplify the difficulties involved with sending requests to websites. It functions as an API so that we may focus on communicating with services and consuming data. After using the request library to extract HTML pages, I turned a tree of Python objects from complicated HTML pages using BeautifulSoup. I have parsed HTML pages with the aid of BeautifulSoup and extracted necessary elements based on tags and classes. The extracted data is then stored in lists, and a data frame with all the necessary properties is produced from these lists. Additionally, a high number of samples are required for the model to function well. I have thus taken 972 records of data out from apartment.com website.

4. We need a huge amount of records to train the model, which will improve performance. As a result, so I downloaded 972 examples and 7 attributes from the website. I have gathered the data, transformed it into a data frame, and then saved it to a CSV file.

5. Requests, pandas, logging, and the BeautifulSoup library all need to be imported. Some websites won't allow to scrap data. I thus constructed a header with Mozilla Firefox as the user agent, sent the url and header to the request library together to receive the webpage's response. logging will help us to debug the response. I have used BeautifulSoup to properly display the web content, and obtained the apartment name and its corresponding urls from the website. and once again I scraped each apartments url for collecting information such as rent, the number of beds, the number of bathrooms, the area in square feet, the names of the flats, and the corresponding rating.

In [3]: `#python code for webscraping https://www.apartments.com to get details of apartme`

In [1]: *#Importing Libraries and setting up Logger for debugging resposes*

```
import requests
import logging
import pandas as pd
from bs4 import BeautifulSoup
logging.basicConfig()
logging.getLogger().setLevel(logging.DEBUG)
requests_log = logging.getLogger("requests.packages.urllib3")
requests_log.setLevel(logging.DEBUG)
requests_log.propagate = True
```

In [*]: *#understanding website apartments.com*

```
headers = { 'User-Agent': 'Mozilla/5.0 (Windows NT 6.0; WOW64; rv:24.0) Gecko/20100101 Firefox/24.0' }
webpage_response= requests.get('https://www.apartments.com/denton-tx/?bb=ug1x5mvs')
print(webpage_response.status_code)
webpage=webpage_response.content
soup = BeautifulSoup(webpage, 'html.parser')
print(soup.prettify())
```

DEBUG:urllib3.connectionpool:Starting new HTTPS connection (1): www.apartments.com:443

```

In [*]: #Function to get data from website
#inputs required:City, Url,Rent,Bed,Bath,Area,Names,Rating
def getdatas(city,i,url,rent,bed,bath,area,names,rating):
    a=url.split("?")
    for i in range(1,i):
        if(i==1):

            scrapu=url
        else:
            if(city=="denton"):
                scrapu=a[0]+str(i)+"/?"+a[1]
            else:
                scrapu=a[0]+str(i)+"/"
        #print(scrapu)
        webpage_response= requests.get(scrapu, headers=headers)
        #print(webpage_response.status_code)
        webpage=webpage_response.content
        soup = BeautifulSoup(webpage, 'html.parser')
        urlss=soup.find_all('article')
        urlss
        l=[]

        for i in urlss:
            dataurl= i.get('data-url')
            if dataurl != None:
                #print(dataurl)
                webpage_response= requests.get(dataurl, headers=headers)
                #print(webpage_response.status_code)
                webpage=webpage_response.content
                soup = BeautifulSoup(webpage, 'html.parser')
                mydivs = soup.find_all("p", {"class": "rentInfoDetail"})
                #print(mydivs)
                for n in soup.find_all("h1", {"id": "propertyName"}):
                    names.append(n.getText().strip())
                for p in soup.find_all("p", {"class": "rentInfoDetail"}):
                    l.append(str(p.getText()))
                rent.append(l[0])
                bed.append(l[1])
                bath.append(l[2])
                area.append(l[3])
                for k in soup.find_all("div", {"class": "profileProperty"}):
                    b=list(k.find_all("span", {"class": "reviewRating"}))
                    if(len(b)>0):
                        rating.append( b[0].getText())
                    else:
                        rating.append(0)

```

```

In [*]: #defining input variables and headers
#function call
headers = { 'User-Agent': 'Mozilla/5.0 (Windows NT 6.0; WOW64; rv:24.0) Gecko/20100101 Firefox/24.0' }
scrapurldenton='https://www.apartments.com/denton-tx/?bb=ug1x5mvs3Jrly27uN'
scrapurldallas='https://www.apartments.com/dallas-tx/'
#print(a)
rent=[]
bed=[]
bath=[]
area=[]
names=[]
rating=[]
getdatas("denton",12,scrapurldenton,rent,bed,bath,area,names,rating)
print(len(rent),len(bed),len(bath),len(area),len(names),len(rating))
getdatas("dallas",29,scrapurldallas,rent,bed,bath,area,names,rating)
print(rent)
print(bed)
print(bath)
print(area)
print(names)
print(rating)
print(len(rent),len(bed),len(bath),len(area),len(names),len(rating))

```

```

In [*]: #converting lists to data frames

city=["denton"]*231
b=["dallas"]*741
city.extend(b)
df = pd.DataFrame(list(zip(city,names,bed,bath,area,rating,rent)),columns =['City','Names','Bed','Bath','Area','Rating','Rent'])
df

```

```

In [*]: # storing data frame to csv file
df.to_csv('DedeePyDataset.csv')

```

```

In [ ]:

```