PREDICTING PRICES OF REAL-ESTATE IN CANADA

Data Science Project

By Omri Bakal

RESEARCH QUESTION

- Can we predict the prices values of an apartment by other variables like : Number of beds,baths,name of the city?

My Research – Apartments in Canada Provinces

I gathered information and data from the following provinces in Canada:

Alberta

Newfoundland And Labrador

Ontario

British Columbia

Quebec

Prince Edward Island

Manitoba

Nova Scotia

New Brunswick

Sakatchewan

Data Source:

Crawling from: https://www.point2homes.com/CA

MAIN STEPS

- Obtaining Data
- Cleaning the Data
- EDA Visualisation
- Machine Learning

CRAWLING FROM THE WEBSITE

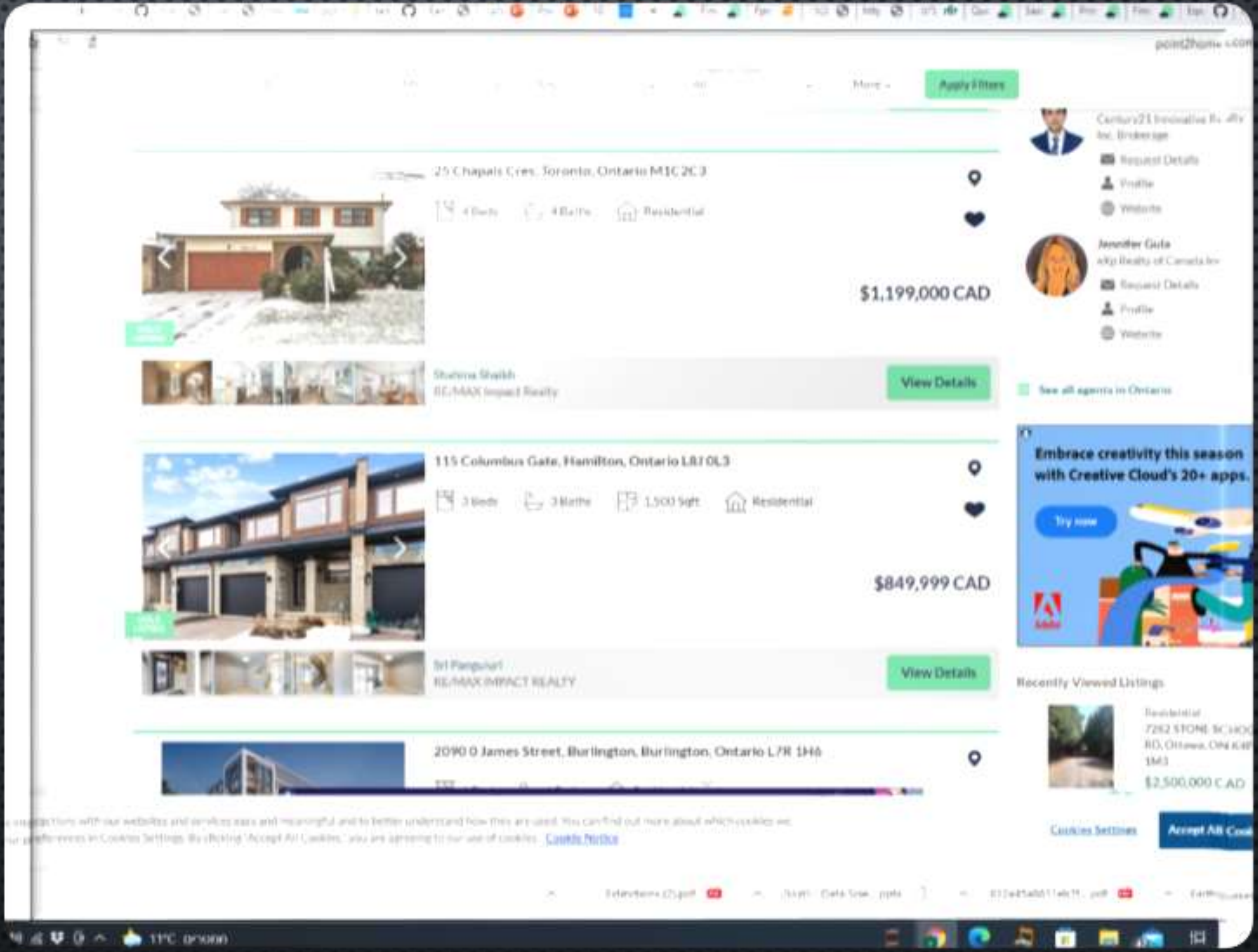
I Took the most significant

Parameters from the website such as:

Number of beds,city,address,baths,

Seller,company,type of apartment and price.

Every Category of provence contains 30 pages of apartments , so i had to scrape 30 pages of all the provences.



25 Chapais Cres, Toronto, Ontario M1C 2C3

4 Beds 4 Baths Residential

$1,199,000 CAD

Shatima Shaikh
RE/MAX Impact Realty

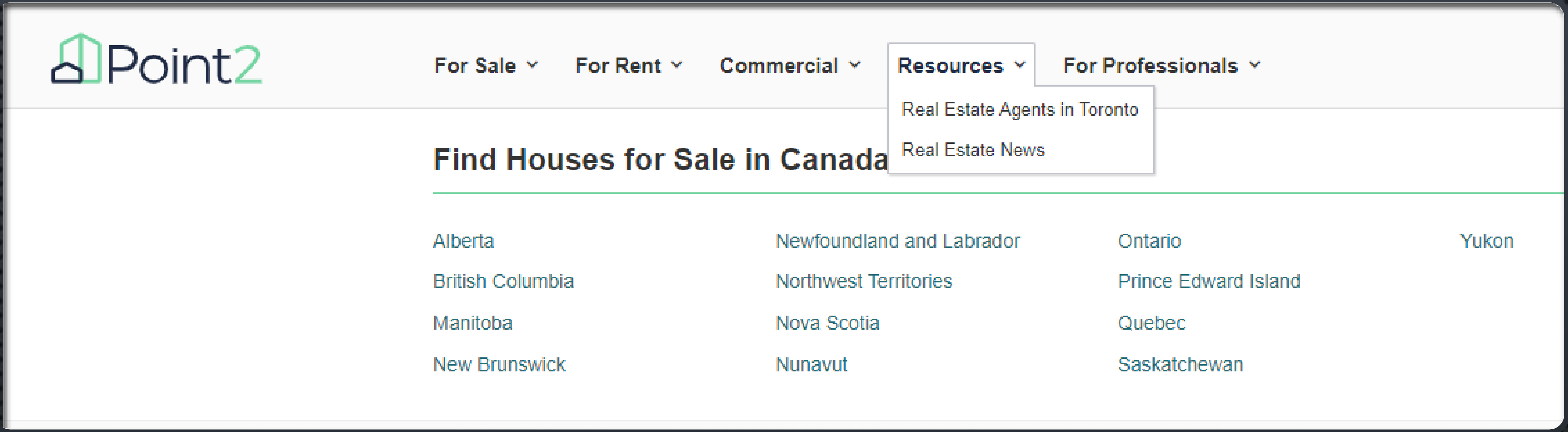
115 Columbus Gate, Hamilton, Ontario L8J 0L3

3 Beds 3 Baths 1,500 Sqft Residential

$849,999 CAD

Sri Panguluri
RE/MAX IMPACT REALTY

2090 0 James Street, Burlington, Burlington, Ontario L7R 1H6



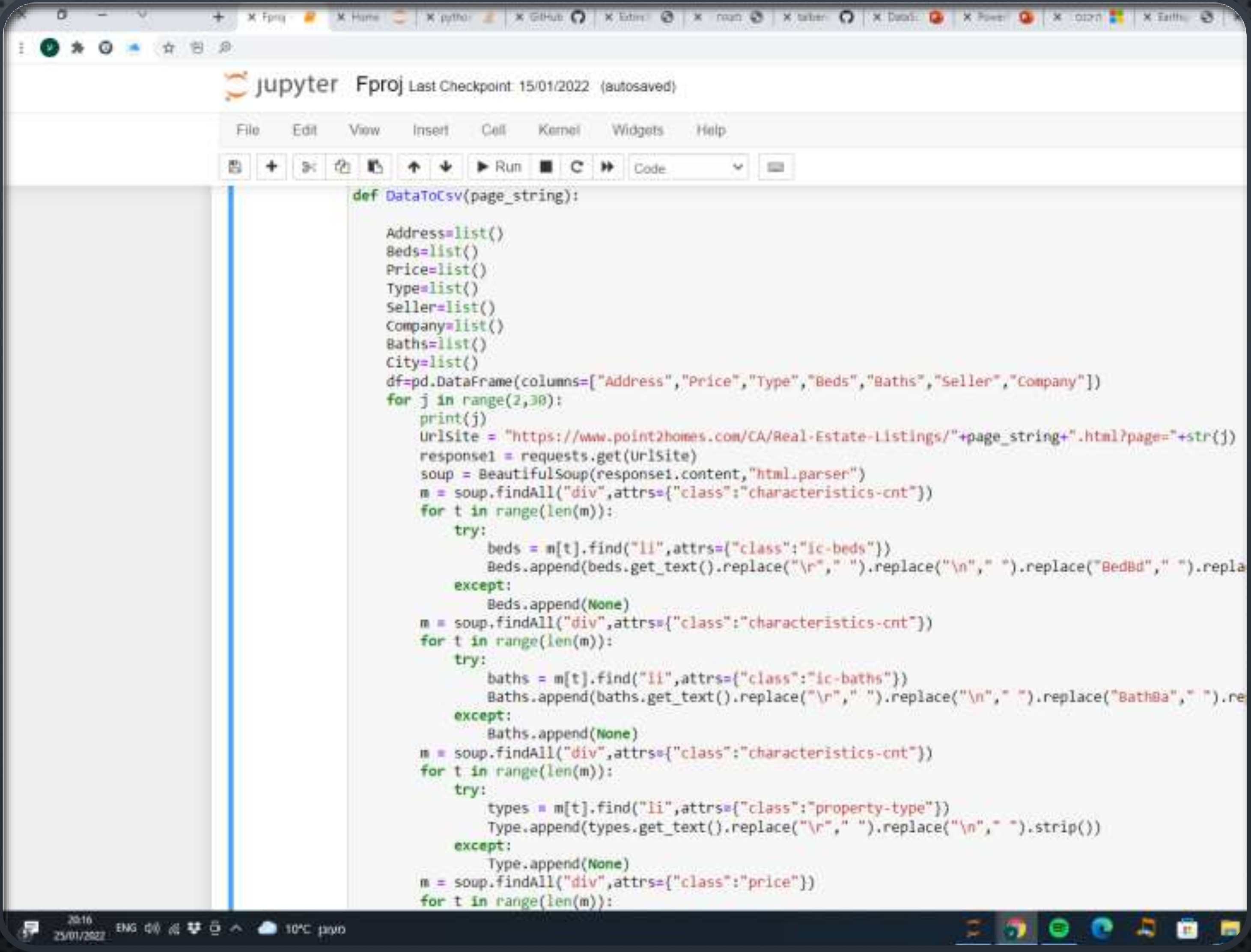
Find Houses for Sale in Canada

Alberta
British Columbia
Manitoba
New Brunswick
Newfoundland and Labrador
Northwest Territories
Nova Scotia
Nunavut
Ontario
Prince Edward Island
Quebec
Saskatchewan
Yukon

PAGE OF ALL THE PROVENCES I HAD TO SCRAPE:

THE FUNCTION THAT SCRAPING THE DATA AND PUTTING IT INTO A CSV:

- I PUT THE DATA INTO DIFFERENT LISTS BY PARAMETRS/CATEGORIES AND THEN TRANSFERRED IT INTO A CSV FILE.



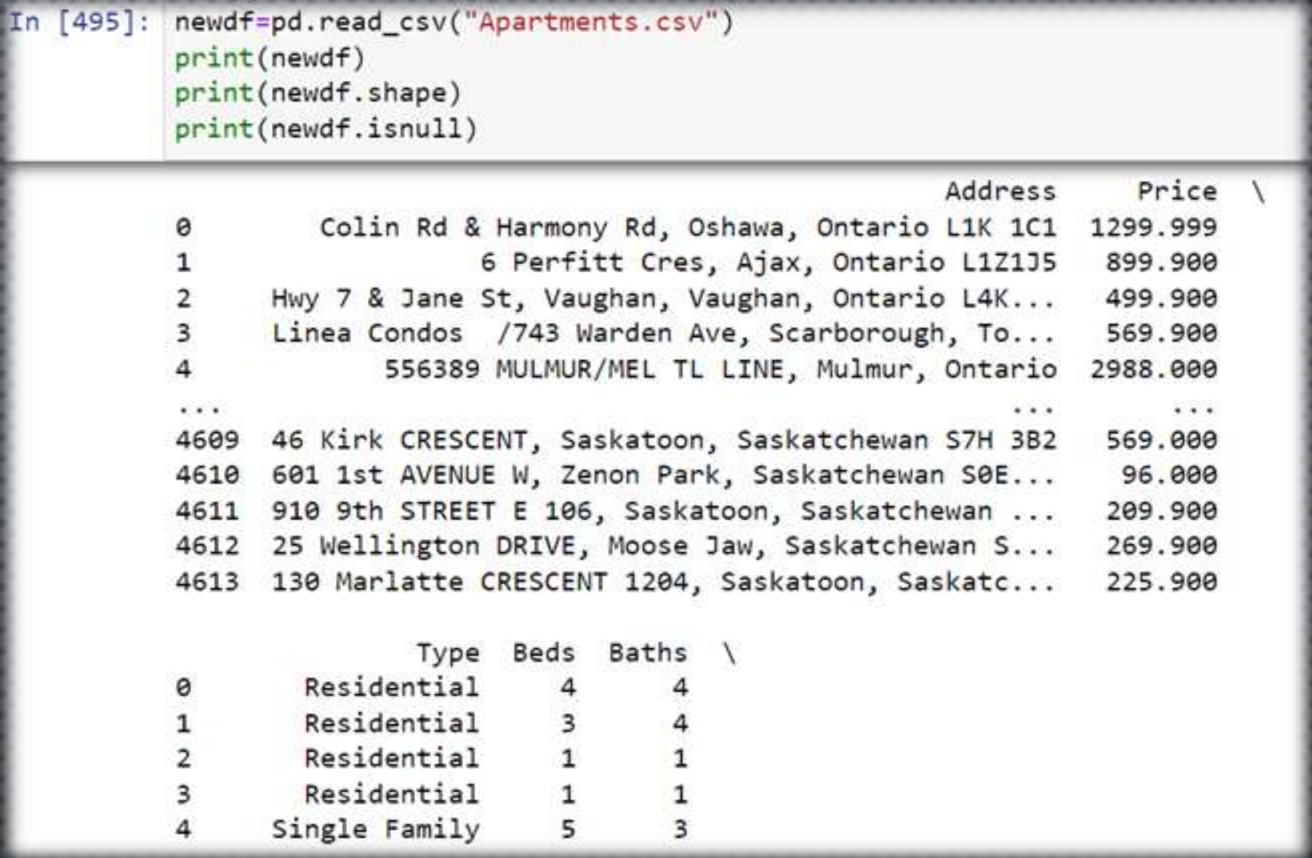
```
def DataToCsv(page_string):

    Address=list()
    Beds=list()
    Price=list()
    Type=list()
    Seller=list()
    Company=list()
    Baths=list()
    City=list()
    df=pd.DataFrame(columns=["Address","Price","Type","Beds","Baths","Seller","Company"])
    for j in range(2,30):
        print(j)
        UrlSite = "https://www.point2homes.com/CA/Real-Estate-Listings/"+page_string+".html?page="+str(j)
        response1 = requests.get(UrlSite)
        soup = BeautifulSoup(response1.content,"html.parser")
        m = soup.findAll("div",attrs={"class":"characteristics-cnt"})
        for t in range(len(m)):
            try:
                beds = m[t].find("li",attrs={"class":"ic-beds"})
                Beds.append(beds.get_text().replace("\r"," ").replace("\n"," ").replace("BedBd"," ").repla
            except:
                Beds.append(None)
        m = soup.findAll("div",attrs={"class":"characteristics-cnt"})
        for t in range(len(m)):
            try:
                baths = m[t].find("li",attrs={"class":"ic-baths"})
                Baths.append(baths.get_text().replace("\r"," ").replace("\n"," ").replace("BathBa"," ").re
            except:
                Baths.append(None)
        m = soup.findAll("div",attrs={"class":"characteristics-cnt"})
        for t in range(len(m)):
            try:
                types = m[t].find("li",attrs={"class":"property-type"})
                Type.append(types.get_text().replace("\r"," ").replace("\n"," ").strip())
            except:
                Type.append(None)
        m = soup.findAll("div",attrs={"class":"price"})
        for t in range(len(m)):
```

The Dataframe:

I Gathered the data from 10 different categories of provences in Canada.

After collecting the data I transferred it into a csv file and then to a dataframe containing the columns:

- Address
- Price
- Type
- Beds
- Baths
- Company
- Seller
- city

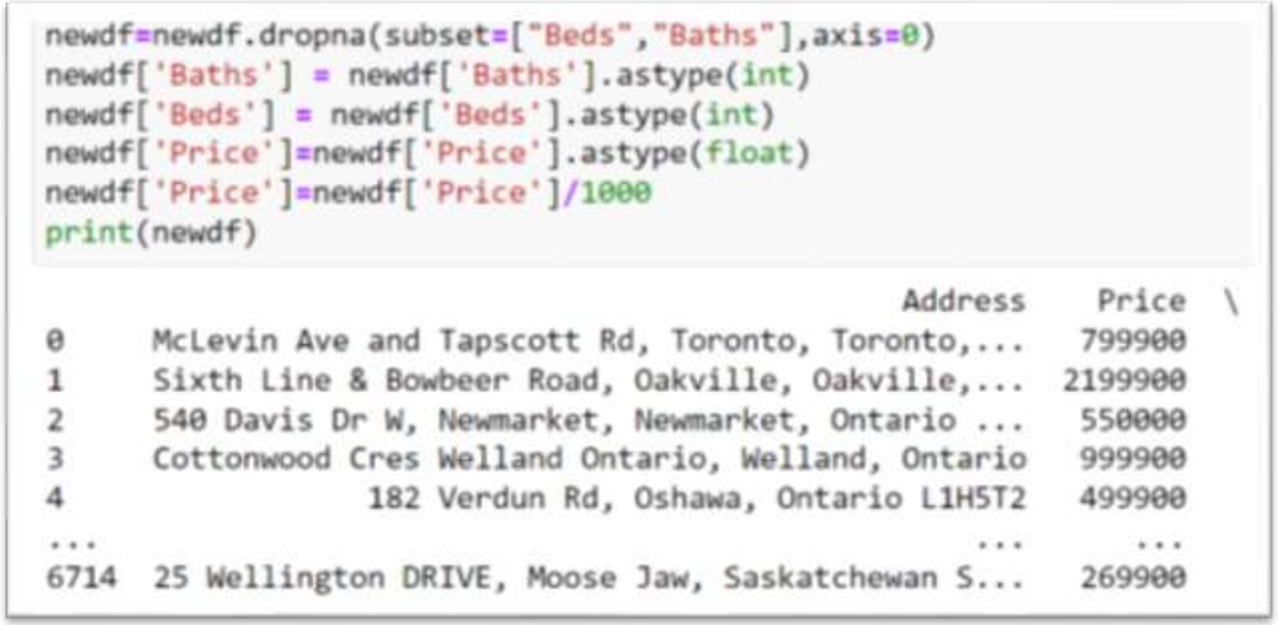


```
In [495]: newdf=pd.read_csv("Apartments.csv")
          print(newdf)
          print(newdf.shape)
          print(newdf.isnull)
```

```
                                                     Address     Price  \
0        Colin Rd & Harmony Rd, Oshawa, Ontario L1K 1C1  1299.999
1                   6 Perfitt Cres, Ajax, Ontario L1Z1J5   899.900
2     Hwy 7 & Jane St, Vaughan, Vaughan, Ontario L4K...   499.900
3     Linea Condos  /743 Warden Ave, Scarborough, To...   569.900
4             556389 MULMUR/MEL TL LINE, Mulmur, Ontario  2988.000
...                                                  ...       ...
4609  46 Kirk CRESCENT, Saskatoon, Saskatchewan S7H 3B2   569.000
4610  601 1st AVENUE W, Zenon Park, Saskatchewan S0E...    96.000
4611  910 9th STREET E 106, Saskatoon, Saskatchewan ...   209.900
4612  25 Wellington DRIVE, Moose Jaw, Saskatchewan S...   269.900
4613  130 Marlatte CRESCENT 1204, Saskatoon, Saskatc...   225.900

                Type  Beds  Baths  \
0        Residential     4      4
1        Residential     3      4
2        Residential     1      1
3        Residential     1      1
4      Single Family     5      3
```

Data Cleaning Data Type Converting

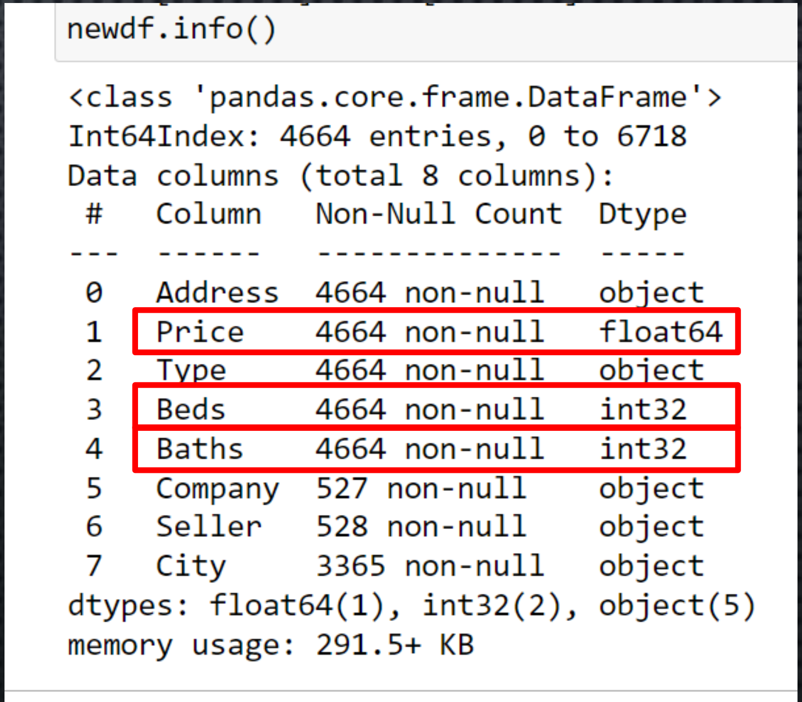
- As you can see in the following picture, I dropped all the missing values in the columns Beds and Baths , and i also converted the object categories Beds, Baths and Price into a int and float values.
- This step is going to help futher on EDA step and Machine Learning.



```
newdf=newdf.dropna(subset=["Beds","Baths"],axis=0)
newdf['Baths'] = newdf['Baths'].astype(int)
newdf['Beds'] = newdf['Beds'].astype(int)
newdf['Price']=newdf['Price'].astype(float)
newdf['Price']=newdf['Price']/1000
print(newdf)
```

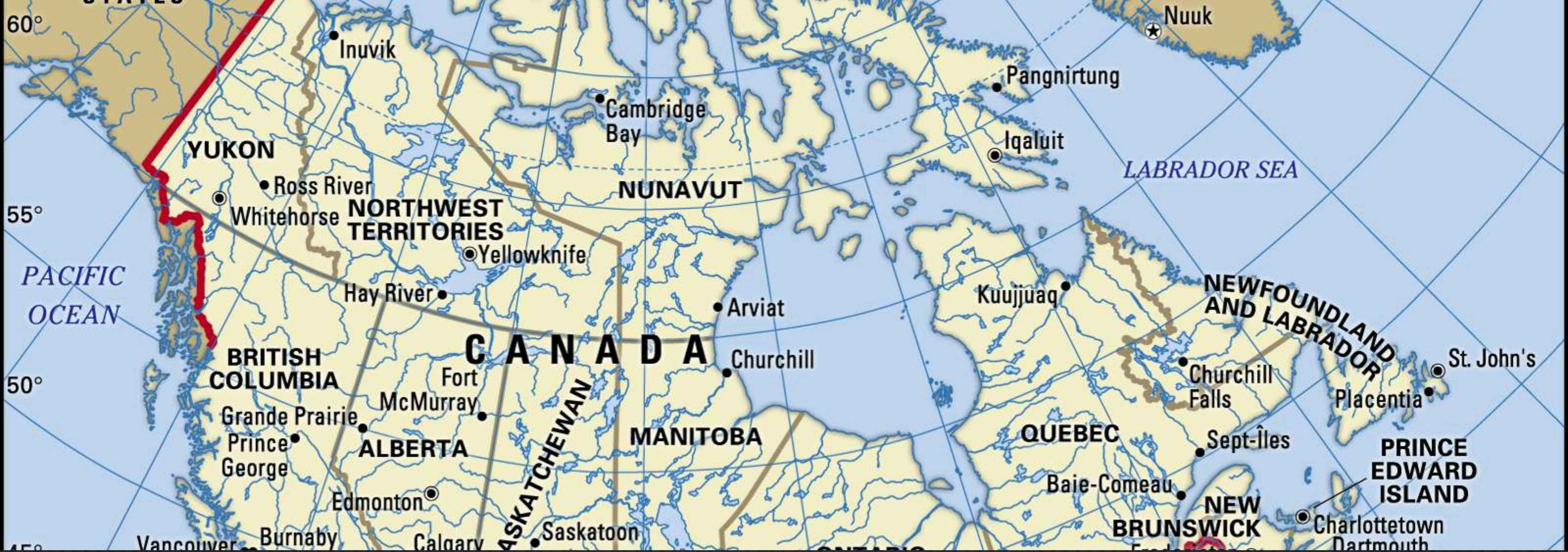
```
                                                Address    Price  \
0     McLevin Ave and Tapscott Rd, Toronto, Toronto,...   799900
1     Sixth Line & Bowbeer Road, Oakville, Oakville,...  2199900
2     540 Davis Dr W, Newmarket, Newmarket, Ontario ...   550000
3     Cottonwood Cres Welland Ontario, Welland, Ontario   999900
4                 182 Verdun Rd, Oshawa, Ontario L1H5T2   499900
...                                                 ...      ...
6714  25 Wellington DRIVE, Moose Jaw, Saskatchewan S...   269900
```

AS YOU CAN SEE IN THE DF.INFO() COMMAND, NOW THE DATA CONTAINS OBJECT COLUMNS AND ALSO INT/FLOAT COLUMNS

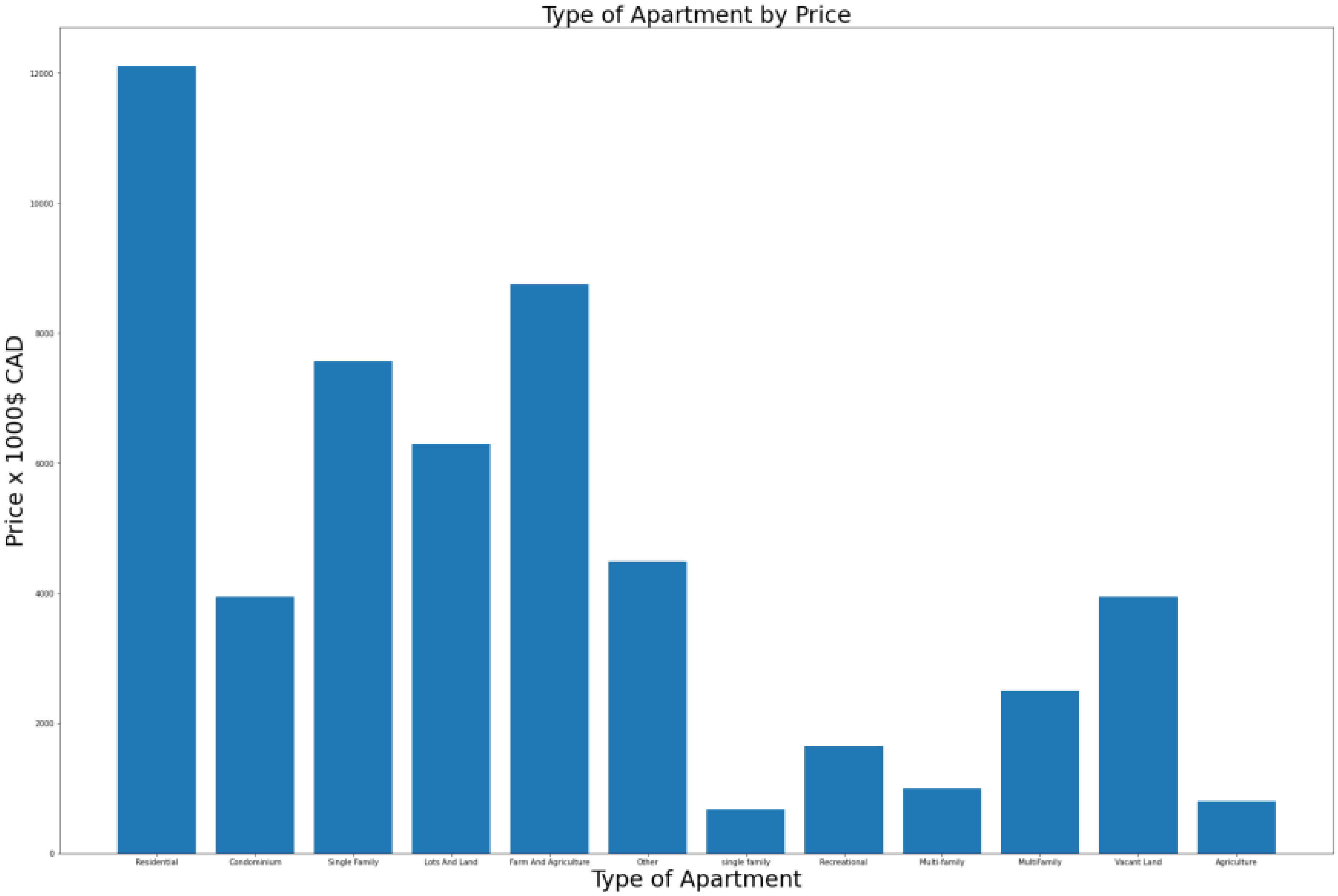


```
newdf.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 4664 entries, 0 to 6718
Data columns (total 8 columns):
 #   Column    Non-Null Count  Dtype  
---  ------    --------------  -----  
 0   Address   4664 non-null   object 
 1   Price     4664 non-null   float64
 2   Type      4664 non-null   object 
 3   Beds      4664 non-null   int32  
 4   Baths     4664 non-null   int32  
 5   Company   527 non-null    object 
 6   Seller    528 non-null    object 
 7   City      3365 non-null   object 
dtypes: float64(1), int32(2), object(5)
memory usage: 291.5+ KB
```



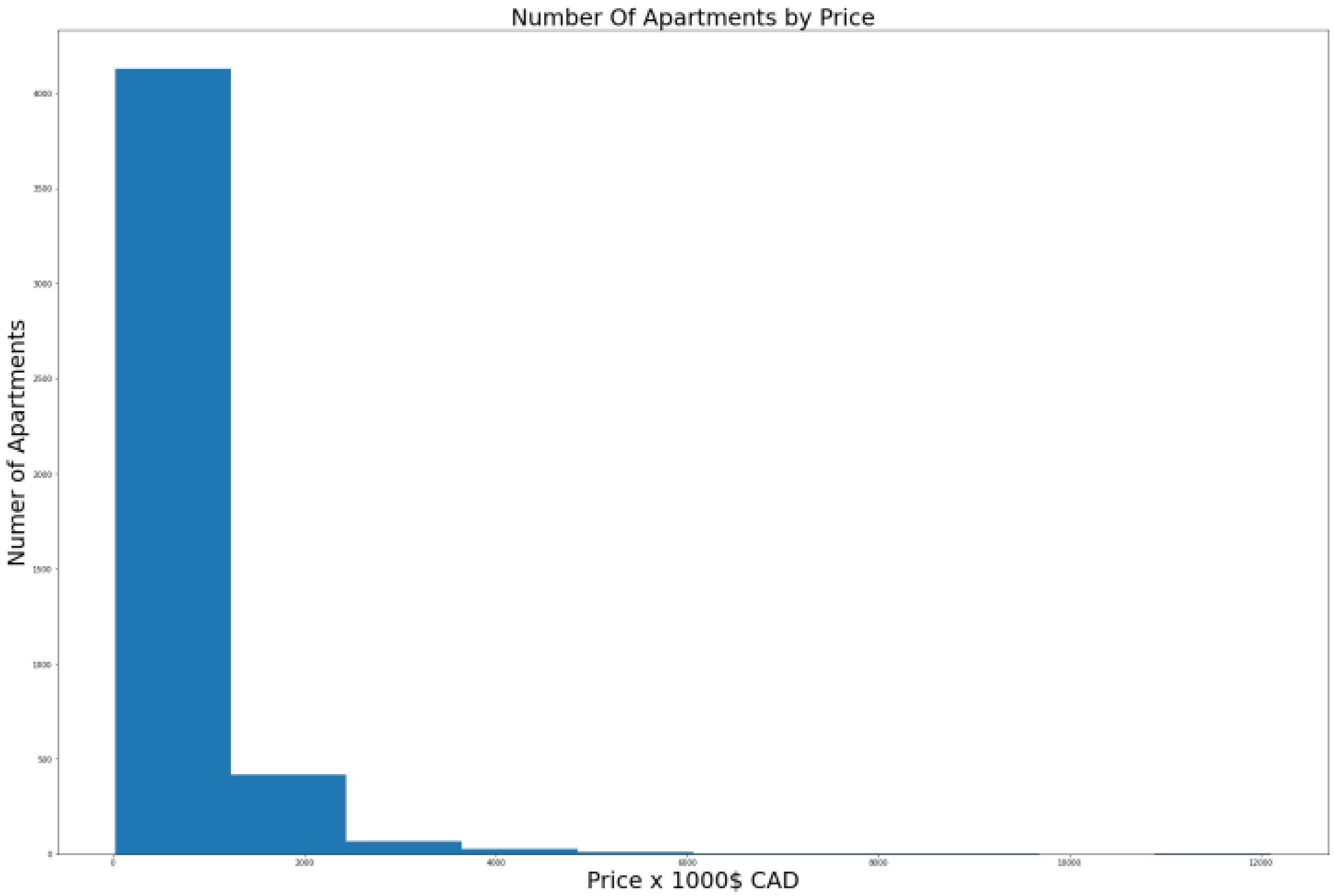
EDA VISUALISATIONS



Type of Apartment by Price
Price x 1000$ CAD
12000
10000
8000
6000
4000
2000
0
Residential
Condominium
Single Family
Lots And Land
Farm And Agriculture
Other
single family
Recreational
Multi-family
MultiFamily
Vacant Land
Agriculture
Type of Apartment

TYPE OF APARTMENT BY PRICE GRAPH

- THIS IS A **BAR** PLOT , COMPARES THE AMOUNT TYPE OF HOUSES BY THEIR PRICES.
- X BAR CONTAINS THE TYPE OF THE RESIDENCE , AND THE Y BAR CONTAINS THE PRICE
- WE CAN SEE FROM THE PLOT THAR RESIDENTIAL TYPE OF HOUSES ARE MORE WANTED AMONG THE PEOPLE IN CANADA.

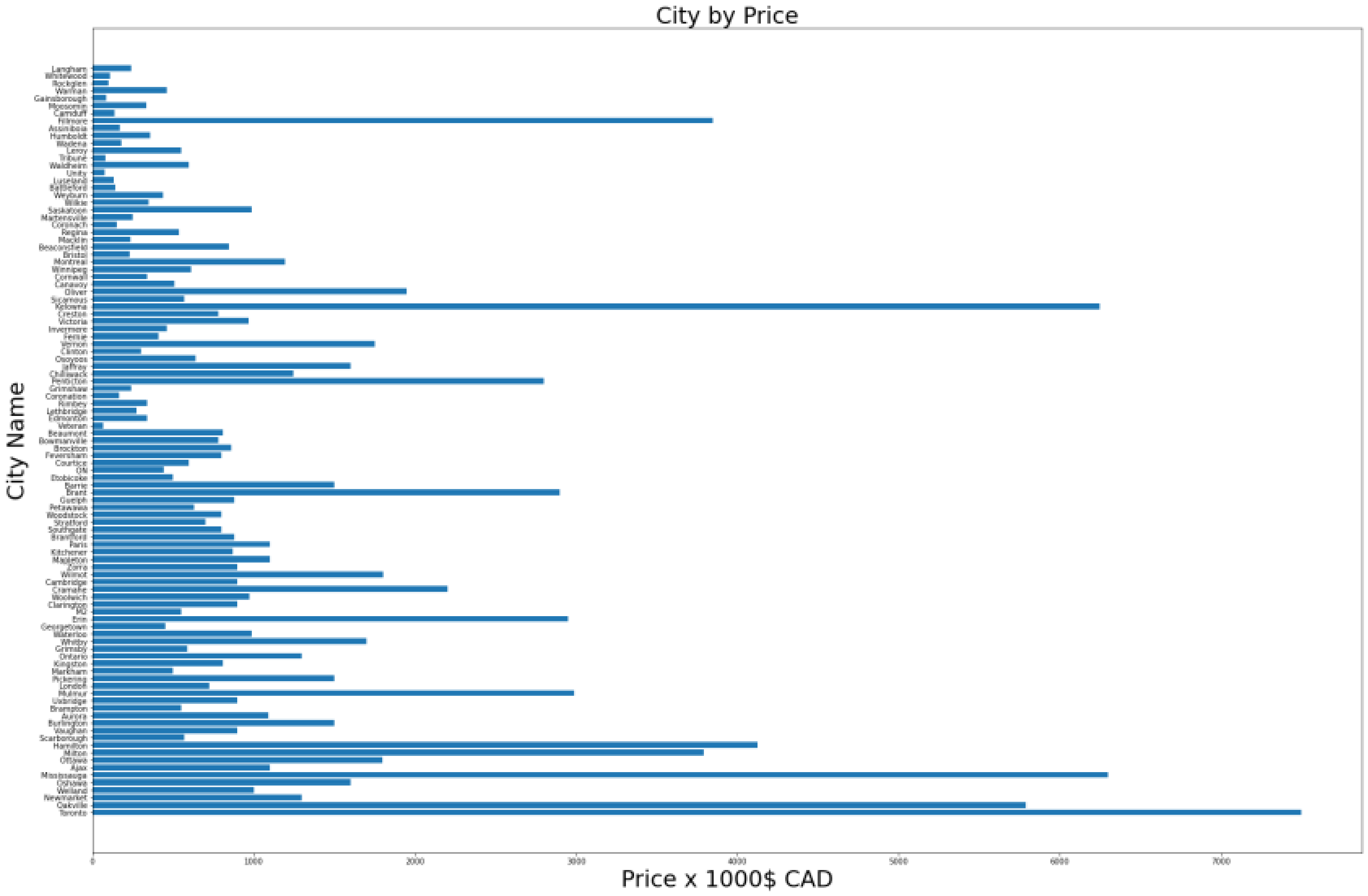

Number Of Apartments by Price
Numer of Apartments
4000
3500
3000
2500
2000
1500
1000
500
0
0
2000
4000
6000
8000
10000
12000
Price x 1000$ CAD

NUMBER OF APARTMENTS BY PRICE GRAPH:

This is **Histogram** Plot , that shows us the number of the Apartments in Canada by
.their prices

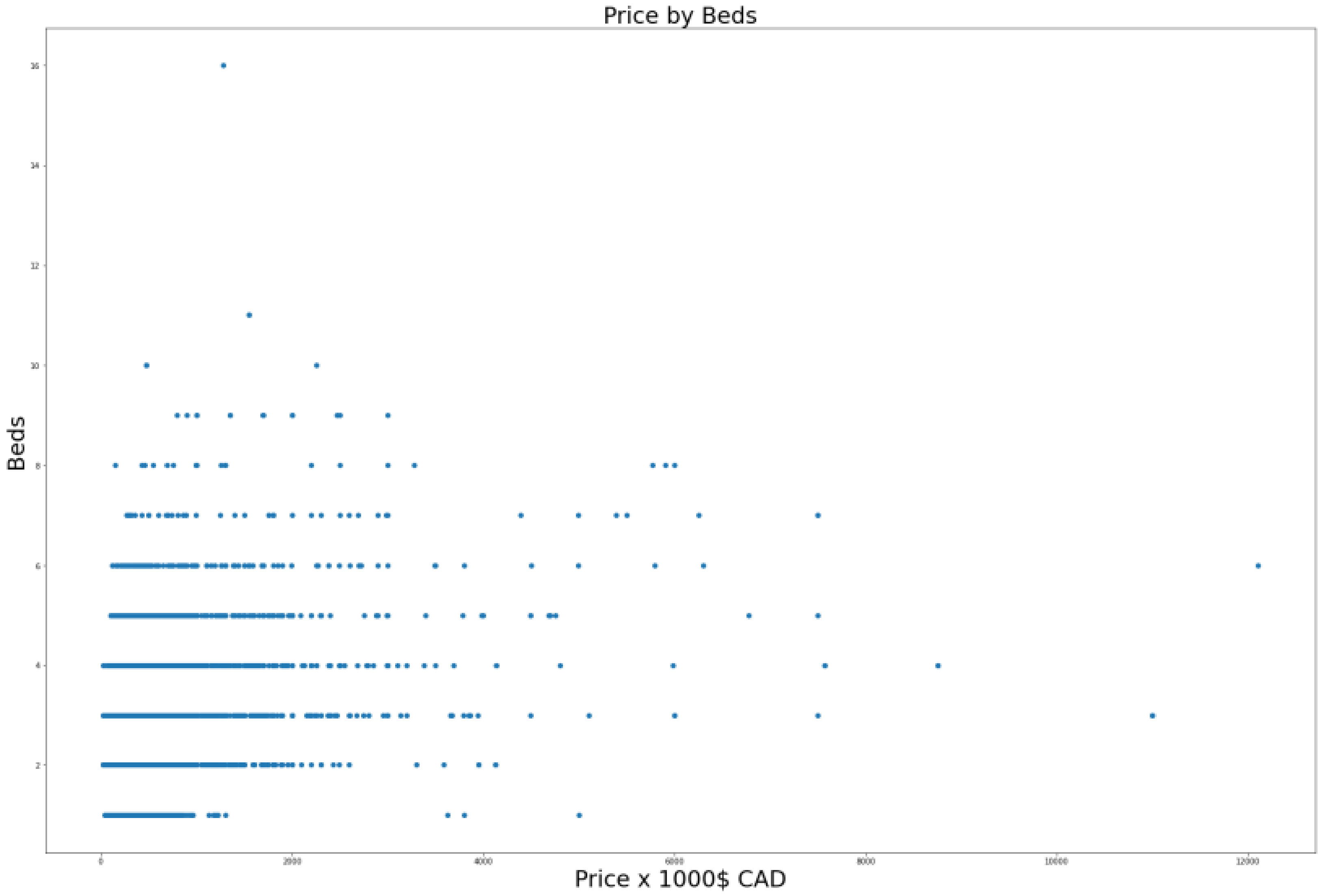
The X bar contians the Prices , and the Y bar contains the amount of the houses in
.the range of the price shown below

We can see from this plot that most of the houses in the provinces in Canada are
.million CAD dollars**1-2** sold between


City by Price
City Name
Price x 1000$ CAD
0
1000
2000
3000
4000
5000
6000
7000
Langham
Whitewood
Rockglen
Warman
Gainsborough
Moosomin
Carnduff
Fillmore
Assiniboia
Humboldt
Wadena
Leroy
Tribune
Waldheim
Unity
Luseland
Battleford
Weyburn
Wilkie
Saskatoon
Martensville
Coronach
Regina
Macklin
Beaconsfield
Bristol
Montreal
Winnipeg
Cornwall
Canaway
Oliver
Sicamous
Kelowna
Creston
Victoria
Invermere
Fernie
Vernon
Clinton
Osoyoos
Jaffray
Chilliwack
Penticton
Grimshaw
Coronation
Rimbey
Lethbridge
Edmonton
Veteran
Beaumont
Bowmanville
Brockton
Feversham
Courtice
ON
Etobicoke
Barrie
Brant
Guelph
Petawawa
Woodstock
Stratford
Southgate
Brantford
Paris
Kitchener
Mapleton
Zorra
Wilmot
Cambridge
Cramahe
Woolwich
Clarington
MQ
Erin
Georgetown
Waterloo
Whitby
Grimsby
Ontario
Kingston
Markham
Pickering
London
Mulmur
Uxbridge
Brampton
Aurora
Burlington
Vaughan
Scarborough
Hamilton
Milton
Ottawa
Ajax
Mississauga
Oshawa
Welland
Newmarket
Oakville
Toronto

City by price graph:

- This is another **bar** plot that shows us which city in Canada is the most expensive , we can see from the plot that Toronto has the most expensive houses from all other cities.
- The x bar represents the price and the y bar represnts the city.


Price by Beds
Beds
16
14
12
10
8
6
4
2
0
2000
4000
6000
8000
10000
12000
Price x 1000$ CAD

PRICE BY BEDS SCATTER PLOT:

- THIS IS A **SCATTER** EHT SU SWOHS TAHT TOLP EHT OT GNIDROCCA SECIRP EHT FO SEULAV GNIRETTACS .SDEB FO REBMUN
- THE X BAR REPRESENTS THE PRICES , AND THE Y BAR REPRESENTS THE PRICES.
- THIS IS AN IMPORTANT PLOT THAT WILL HELP US FUTHER ON IN THE MACHINE LEARNING PROCESS,WE CAN SEE FROM THIS PLOT THAT THERE IS A CORRELATION BETWEEN THE TWO VARIABLES –THE PRICE OF THE APARTMENT IS GETTING MORE EXPENSIVE BY THE NUMBER OF BEDS.

MACHINE LEARNING

MAIN MACHINE LEARNING TYPE:SUPERVISED LEARNING

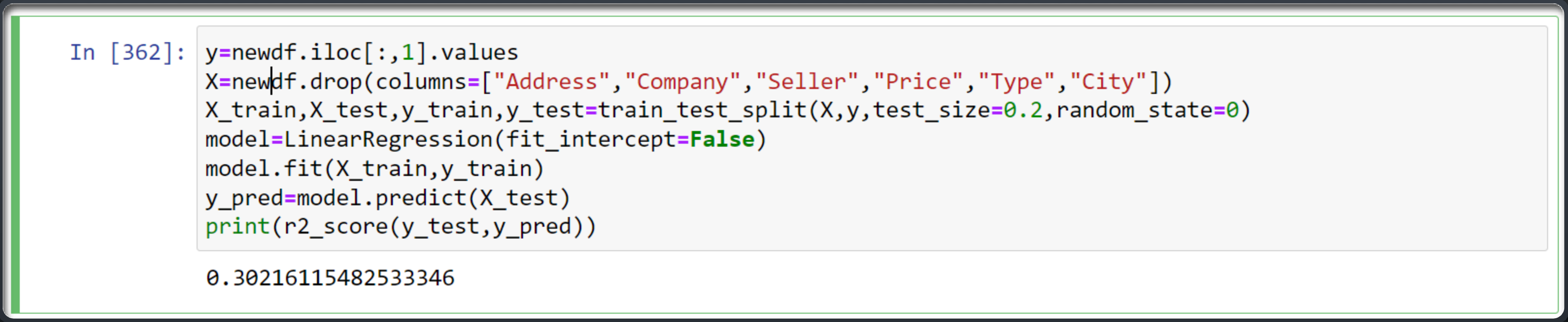
Because the prices values are continious varibales , I had to use a supervised learning method to get the best predicted values , and I had to use two models during the process:

1.Linear Regression

2.Random Forest Regressor

Linear Regression:

- As you can see in the code below , I removed the problematic string columns from the DataFrame , and then splitted the DataFrame into X,y in a ratio of 80% and 20%.
- Unfortunately, the pridected model score had a low value so I had to change to other one.



In [362]:

```
y=newdf.iloc[:,1].values
X=newdf.drop(columns=["Address","Company","Seller","Price","Type","City"])
X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.2,random_state=0)
model=LinearRegression(fit_intercept=False)
model.fit(X_train,y_train)
y_pred=model.predict(X_test)
print(r2_score(y_test,y_pred))
```

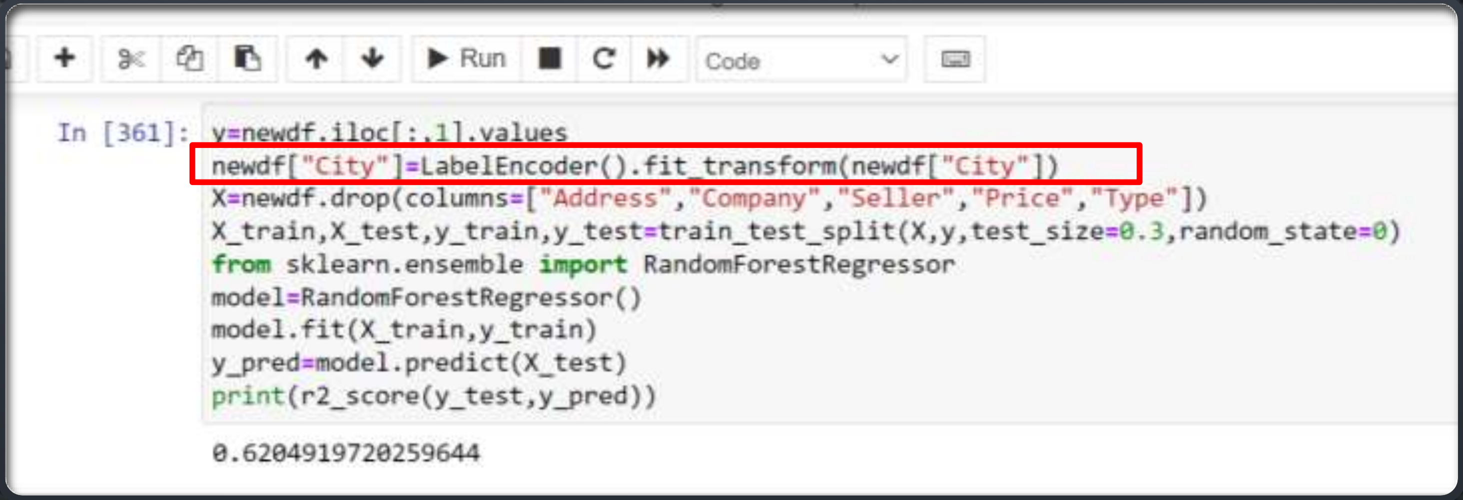
```
0.302161154825333‍46
```

RANDOM FOREST REGRESSOR:

So after getting low results on the linear regression model , I tried another supervised model and had to choose a random forest regressor.

In addition I used the Label Encoder feature of sklearn to transform the city coulmn into a numeric column , something that can help in predicting the model more precisely.

Seems like the score of the model got a better result after that.



```
In [361]: y=newdf.iloc[:,1].values
          newdf["City"]=LabelEncoder().fit_transform(newdf["City"])
          X=newdf.drop(columns=["Address","Company","Seller","Price","Type"])
          X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.3,random_state=0)
          from sklearn.ensemble import RandomForestRegressor
          model=RandomForestRegressor()
          model.fit(X_train,y_train)
          y_pred=model.predict(X_test)
          print(r2_score(y_test,y_pred))

          0.6204919720259644
```

CONCLUSIONS

Final words…

Seems like the model succeeded to predict 62% of the values , according to the values of number of beds and baths and also the city.