**User Guide for Implementation of Analytical Model in Python by Team 4**

## Executive Summary:

This document primarily intended to provide information about the Python code implementation and details about various steps of our analysis which include uploading and cleaning data, filtering, clustering, scoring and calculating final scores (based on relative weights of variables) and finally exporting output files with desired list of properties as comma separated files (CSV). This guide explains what each code block or cell in the iPython Notebook specifically does so that function of the code is explained and modifications can be made wherever required.

## Scope:

The scope of the document pertains to analyzing data from CoStar website along with some additional fields obtained from other data sources ( NPA data) which is then formatted into a CSV file (kindly refer the input file) to contain all the variables of interest stored as a CSV file on the Google drive or in a folder in a local computer.

The weights of the variables, the number of clusters, scoring levels for various quantiles used in assigning scores can be changed/modified as per requirement before running the code.

## Pre-requisites:

1. Python and iPython notebooks installed on the computer or Availability of Google drive with Google Collaborate (to open and run the.ipynb code file)

2. Input File location should be specified correctly in the code.

3. Input files should be in CSV format.

3. All mandatory variables should be present in the input file.

4. Basic understanding of python coding, data analytics (to determine and select the best clusters in clustering model) is highly recommended

## Guide:

**Note: \*Please follow additional notes written in Python code to gain full understanding of our coding techniques.**

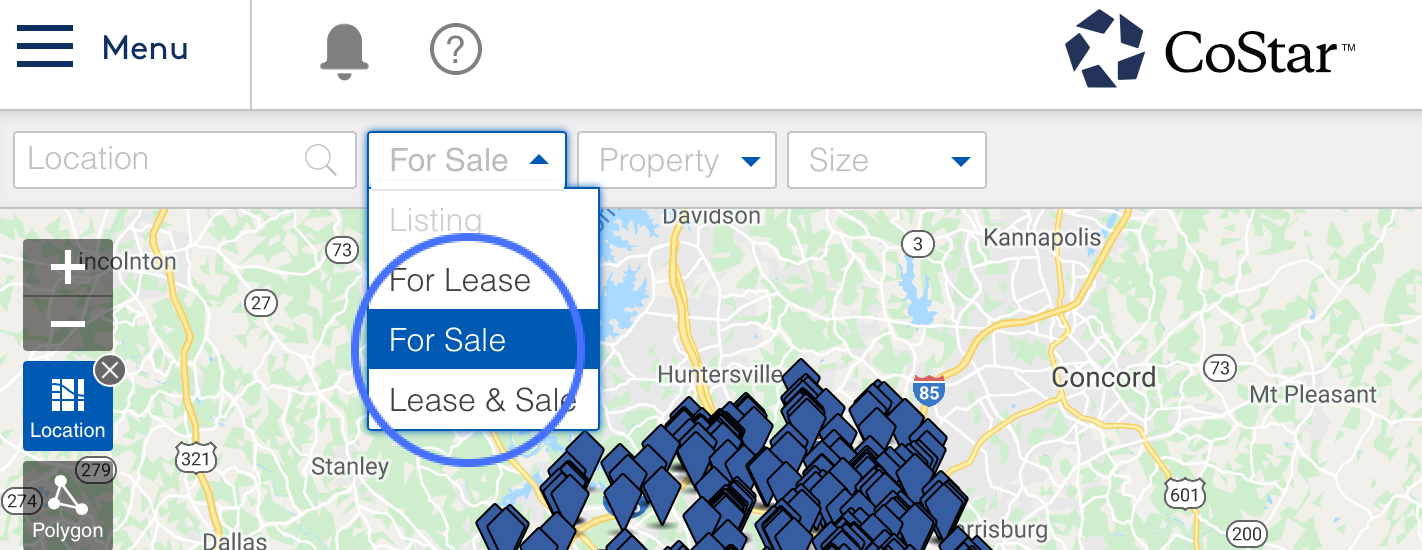
### Preliminary Step: Extracting Data from CoStar

Login into CoStar using your username and password. In the Lookup field choose desired location where you would like to search properties, such as Charlotte, NC.

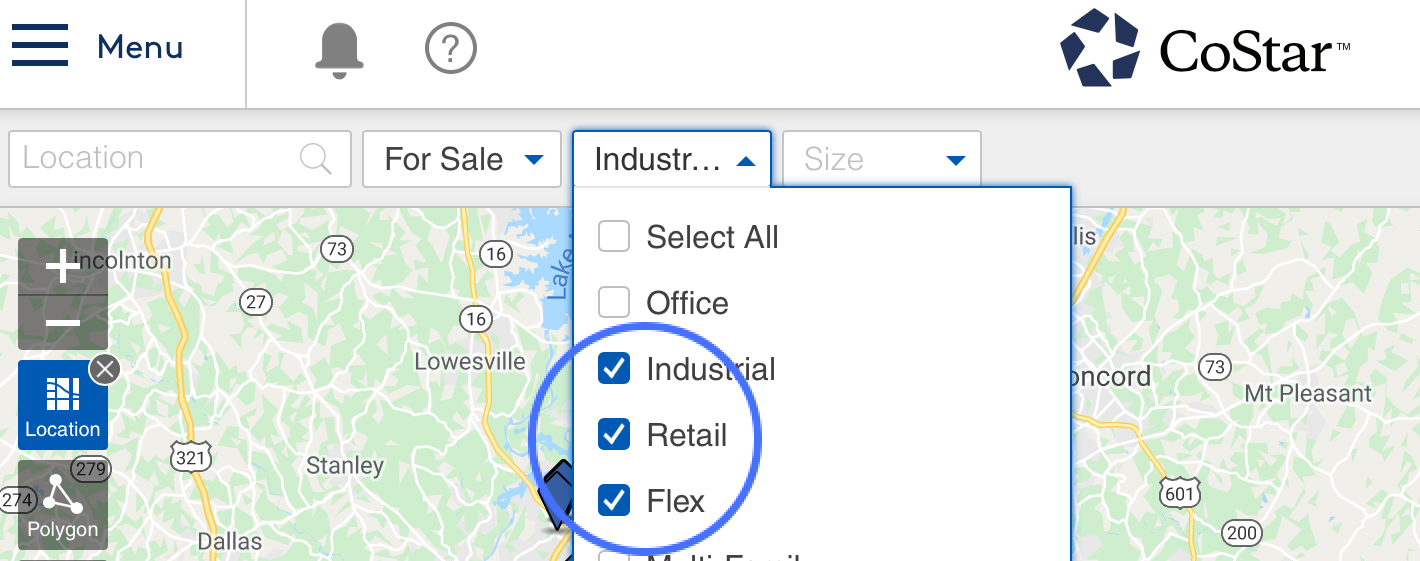
A picture containing screenshot

Description automatically generated

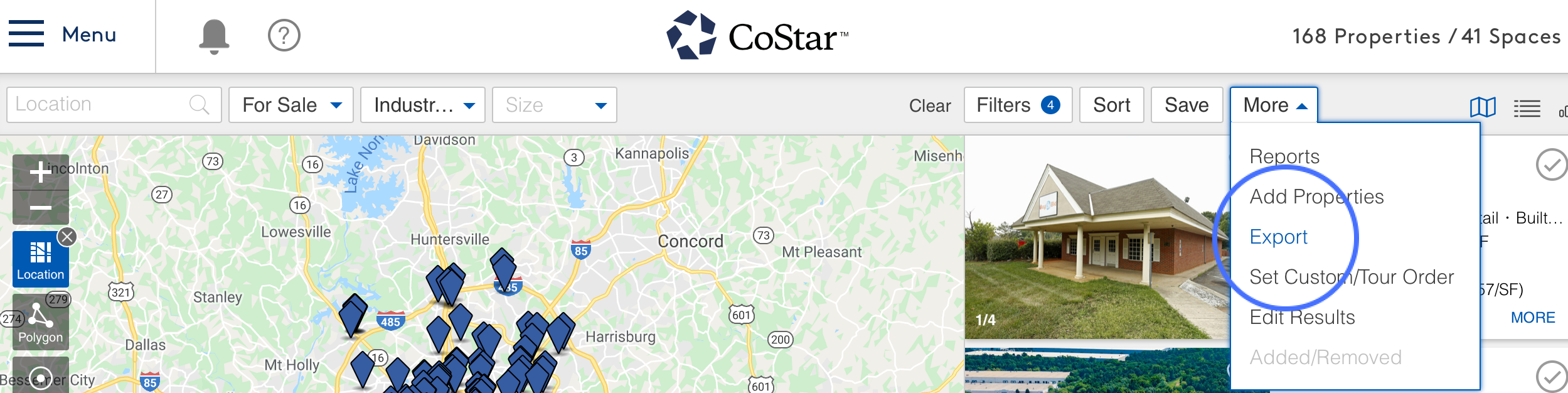
Next, in the Sale type dropdown menu choose for Sale type to be For Sale.

****

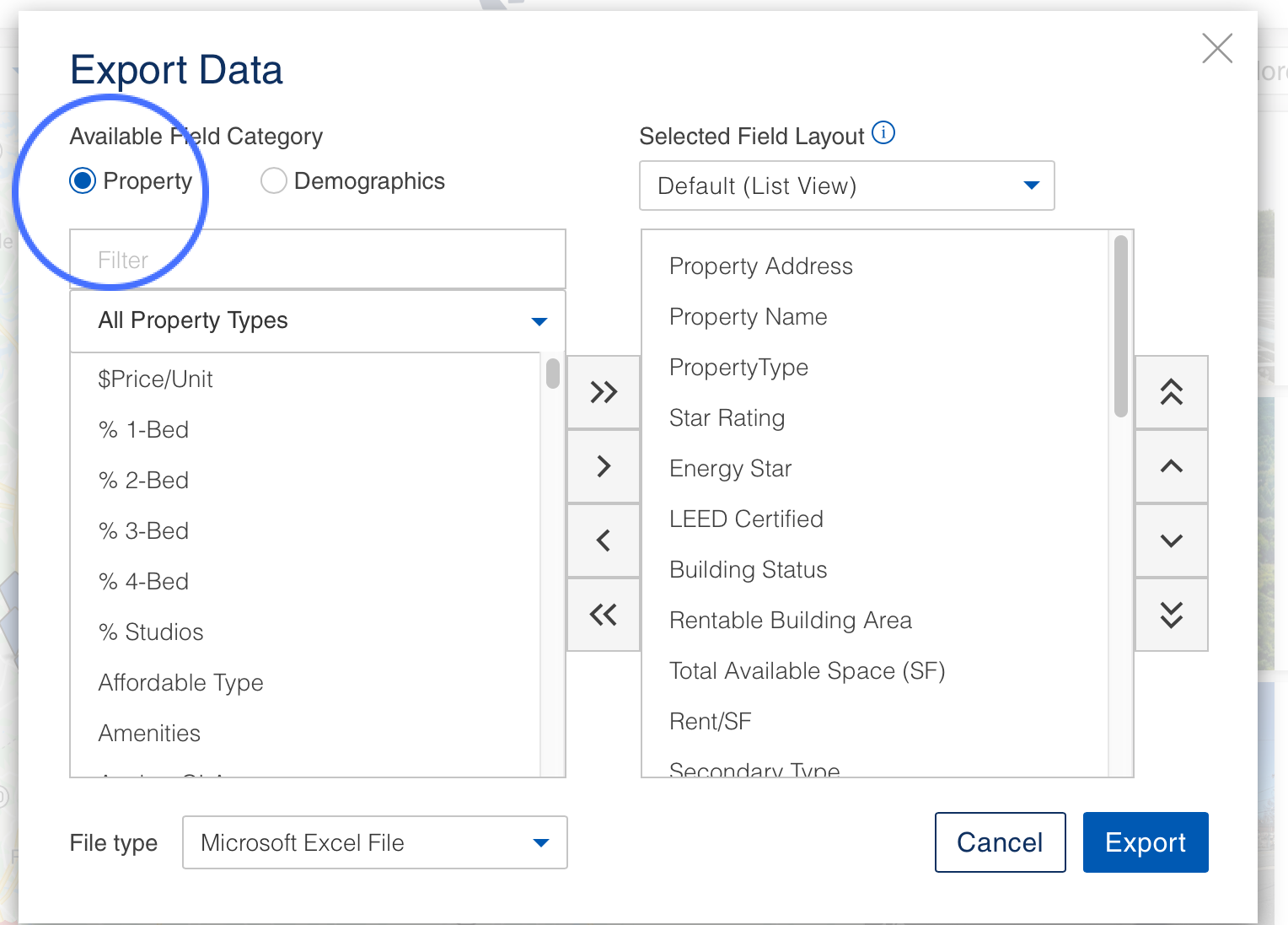
Following, in the property type dropdown menu choose desired types of properties you want to analyze, such as industrial, retail, and flex properties. Choose other filters if desired.

****

To export data for analysis, click on More option dropdown menu and choose Export.

****

The variables to be considered for analysis can be **Property** variables or **Demographics** variables. Choose option for the type of variables desired to be analyzed.

****

Pick desired variables, such as Vacancy % in the right table, and click on the right arrow to add the desired variable to the list of variables to be exported. The variables in the right table are variables to be exported, these can be added and removed by utilizing right and left arrows and shifting variables between left table - available variable category, and right table - to be exported variables. For analysis, variables such as property ID, property address, 2019 Median Household Income (3 miles),2019 Population (3 miles), 2024 Population (3 miles), percent leased and etc. can be exported by utilizing this method for further analysis.

**A screenshot of a cell phone

Description automatically generated**

For the file export choose file type to be Comma Separated Values (CSV).

**A screenshot of a cell phone

Description automatically generated**

Export data by clicking Export button.

**A screenshot of a cell phone

Description automatically generated**

The data containing properties with desired variables will be saved in your Downloads folder on your local computer.

Moreover, this data can also be saved to a Google Drive by simply dragging and dropping the exported file to Google Drive storage folder. Google Drive is a file storage and synchronization service by Google that allows users to store files on their Google storage and synchronize files across devices. To use Google Drive, go to <https://www.google.com/drive/> and create an account with your google email.

### Python Clustering and Scoring:

**Python Essential Notes**

-To run a code press run button  or press Shift + Enter on your keyboard.

-Columns and rows start with a count of 0 up to a number, excluding the number. For example, to select columns 10 through 20: df2 = df.iloc [:,9:21].

-# Hashtags are used to blackout a code that you don’t want to interpret and to add comments to a code. If you want to run a code remove a # at the beginning of the code.

-To view a table type head(). For example, to view table df2: df2.head().

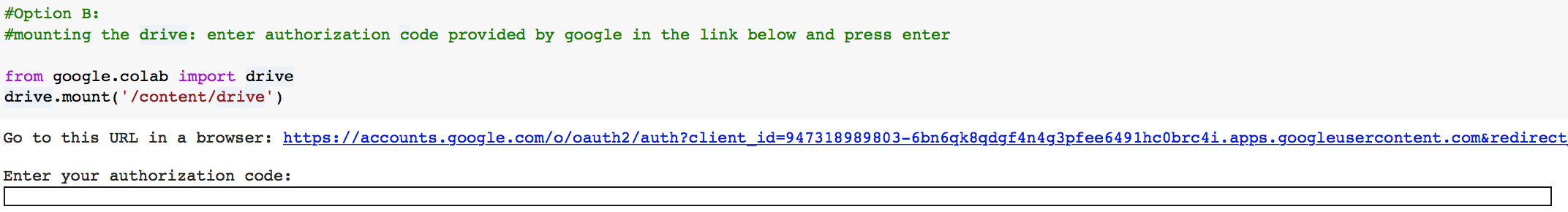
**Step1:** Load data set into Google Collaboratory. There are two ways to do this.

1st option is to load data from local computer. When loading data from local computer specify folder where the data file is located (your Downloads folder). For example, here we are specifying the Downloads folder where exported CoStar data was downloaded: "C:/Users/Agastya Bhavana/Downloads"

A close up of a mans face

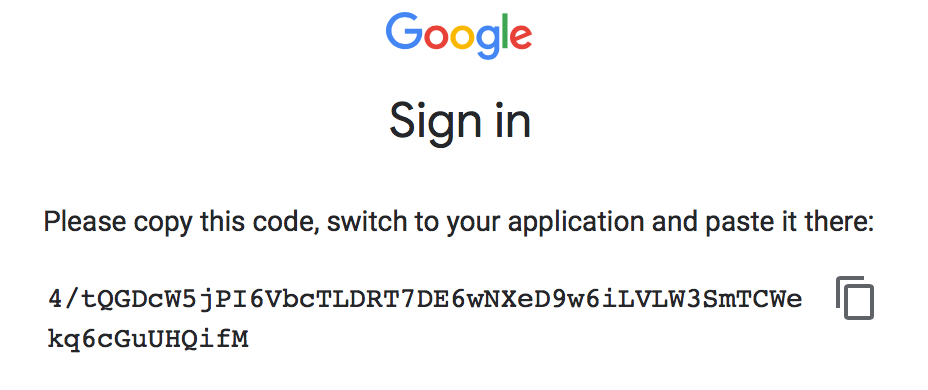
Description automatically generated

2nd option is to load data into Google Drive and read data file from your google drive. First step of 2nd option is to mount the drive. Run the following code (by pressing run button  or pressing Shift + Enter):



Sign in into your google account and **Allow** Google Drive File Stream to access your google account.

Copy the provided code (see below) and pasted it into provided “Enter your authorization code” space. Press Enter to run the code.



Second step of 2nd option is to load your downloaded CoStar data into google collaborator by specifying the path where your file is located. You file path will begin with '/content/drive/My Drive/’ followed by the folder name where you dragged and dropped the CoStar file and followed by file name. If you directly dropped and dropped your file into My Drive specify the file name such as this:

'/content/drive/My Drive/examplefilename.csv’

**Step2:** Filter rows to desired sales price. Specify the column name in double quotes, “For Sale Price” here, and desired amount to be filtered, >=2,000,000 is specified here. For multiple filters use code bellow, specify parameters and remove the hashtag before running the code.

A screenshot of a cell phone

Description automatically generated

**Step3:** Display statistical information from each column by typing dataset name (df in this example) followed by describe.

A screenshot of a social media post

Description automatically generated

**Step4:** Collect desired variables for clustering and ensure that null values are not present. Here we collected desired columns 10 through 20, with variables such as 2019 Median Household Income, 2019 Population (3miles), 2024 Population (3miles), House Density, Transit Proximity, and etc. When specifying columns, it helps to ensure that desired columns are located next to each other in the original file and also, it’s important to remember that columns and rows start with a count of 0 up to a number, excluding the number.

A screenshot of a cell phone

Description automatically generated

A close up of a logo

Description automatically generated

**Step5:** Make a list of property IDs by saving them in a variable propid. This propid variable will be used to for next step to construct and display a cluster tree.

A screenshot of a cell phone

Description automatically generated

**Step6:** Plot the cluster dendrogram. Specify labels = propid, as your list containing property IDs. Follow provided python code without any changes. Export it to desired location. Specify location to be either local drive or google drive by identifying folder where to save the data and csv file name. Here we saved data into our Shared drive, your file should be saved into

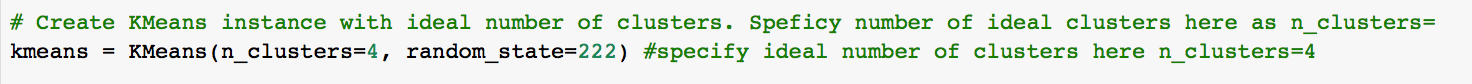
'/content/drive/My Drive/examplefilename.csv’.

**Step7:** Identifying ideal number of clusters: we select the number of clusters where the change in WCSS begins to level off (elbow method). In our example the WCSS begins to level off at approximately 4 clusters. Range of clusters can also be specified here. For example: for i in range(1, 11), choose range for clusters here.

A close up of a map

Description automatically generated

**Step8:** Perform KMeans clustering with identified ideal number of clusters.

****

**Step9:** Add a new column named “Clusters” (as shown) of identified cluster for each record into to original df data table. To view added clusters for each record type df.head()

A screenshot of a cell phone

Description automatically generated

**Step10:** Prepare data for Parallel Coordinate Plot by standardizing data. Here we use df2 data that contains variables for clusters. Also, the name of the columns can be changed here by specifying each column name in single quotation marks.

A screenshot of a cell phone

Description automatically generated

**Step11:** Prepare data for Parallel Coordinate Plot by saving clusters in df2 data as a variable label, grouping df2 data by “Clusters”, finding their mean and specifying cluster names.

**A screenshot of a cell phone

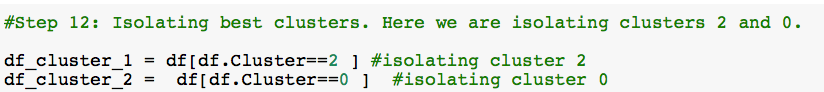
Description automatically generated**

Plot theParallel Coordinate Plot with provided python code without any changes and identify best clusters by comparing their parameters.

A close up of a map

Description automatically generated

**Step12:** Isolate identified best clusters in variables df\_cluster\_1 and df\_cluster\_2. Name of the variables can be changed to your desire.

****

To view the number of records in each cluster type cluster variable name followed by .shape. For example,df\_cluster\_1, a variable containing records within cluster 2 has a shape of (16,27), meaning that it has 16 rows and 27 columns.

**A close up of a logo

Description automatically generated**

**Step13:** Combine best identified clusters by concatenating (pd.concat)best clusters variables df\_cluster\_1 and df\_cluster\_2.

**A screenshot of a cell phone

Description automatically generated**

**Step14:** Save data table variable with best clusters into your local computer or your google drive by specifying folder where to save the data and csv file name. Here we saved data into our Shared drive, your file should be saved into

'/content/drive/My Drive/examplefilename.csv’.

A close up of a logo

Description automatically generated

**Step15:** Separate your new best clusters data into different types of properties by specifying a column name (“PropertyType”) which has different types of properties and type of property (“Flex”, “Retail” or “Industrial’).Separate types of properties can be saved into variables such as df\_flex, df\_retail, df\_industrial.

**A screenshot of a cell phone

Description automatically generated**

**Step16:** For each property type isolate variables that will be used for quantile scoring.For example, for each property type we isolated variables that we wanted to consider for quantile scoring: 'PropertyID','PropertyType','Property Address','2019 Med HH Inc(3m)','2019 Population(3m)','2024 Population(3m)','Transit Proximity','Income Growth ','% Pop Growth 2010-2019(3m)','% Pop Growth 2019-2024(3m)','Crime']].

**Step17:** For each property type assign quantile score for each variable. Please follow detailed python code. It must be noticed here that “Crime” variable receives the lowest score of 1 for having a value in 75th quantile and higher, however all other variables receive the highest score of 4 whenever the value is in 75th quantile and higher.

**A screenshot of a cell phone

Description automatically generated**

**Step18:** For each property type multiply each variable by a weight score. This score is to be determined by the user’s suggestion on the importance of the variable, thus determining the weight of the variable. The final score is saved into a new column, i.e. “final\_score” for each property type data.

**Step19:** Sort Industrial properties in descending order by their final score to get top scored properties. Further, for all desired variables from the initial CoStar data import to be present for industrial properties, merge the original data, df, with the industrial data, df\_industrial, on Property ID.

A screenshot of a cell phone

Description automatically generated

**Step20:** Sort Retail properties in descending order by their final score to get top scored properties. Further, for all desired variables from the initial CoStar data import to be present for retail properties, merge the original data, df, with the retail data, df\_retail, on Property ID.

**A picture containing screenshot

Description automatically generated**

**Step21:** Sort Flex properties in descending order by their final score to get top scored properties. Further, for all desired variables from the initial CoStar data import to be present for flex properties, merge the original data, df, with the flex data, df\_flex, on Property ID.

A screenshot of a cell phone

Description automatically generated

**Step22:** Export files for each type of property with best properties and their corresponding score. Specify path where to save files as well as the name of the files. These files can be saved to local computer or google drive.Here we saved data into our Shared drive, your file should be saved into

'/content/drive/My Drive/examplefilename.csv’.

**A screenshot of a cell phone

Description automatically generated**

**Step23:** To choose suitable properties for purchase further analysis can be conducted on these properties by looking up their price/sf., rent/sf./year, number of parking spaces, traffic counts, occupancy/percent leased and the year that the property was last renovated. It is important to note that these variables cannot be exported as variables from CoStar data, or the exported variables contain plethora of missing data, thus each high scored property has to be further analyzed for suitability on individual basis. This can be done by opening each scored property type data set in Excel file and manually adding and looking up desired variables that cannot be exported from CoStar. For example, after clustering and identifying highest scored industrial properties we additionally looked up criteria on variables that were not exportable from CoStar data: price/sf. (price/sf CoStar exportable variable includes land size and not building size), market rent/sf., rent/sf./year, traffic counts, percent leased, renovation year; and/or variables that had many missing values, such as: number of parking spaces. After comparing high scored properties on these variables final decision can be made on properties are suitable for purchase.

**A screenshot of a cell phone

Description automatically generated**