Introduction

In this project we have developed a state ranking system using the census 2011 data. Each district in a state is ranked on relevant attributes like the Assets possessed by the people and the quality of their homes. MCDA (Multiple Criteria Decision Aiding) algorithms have been used extensively in developing this ranking system.

Method

Using the pyodbc library of python we established a connection to our SQL Server database containing the enumeration records.

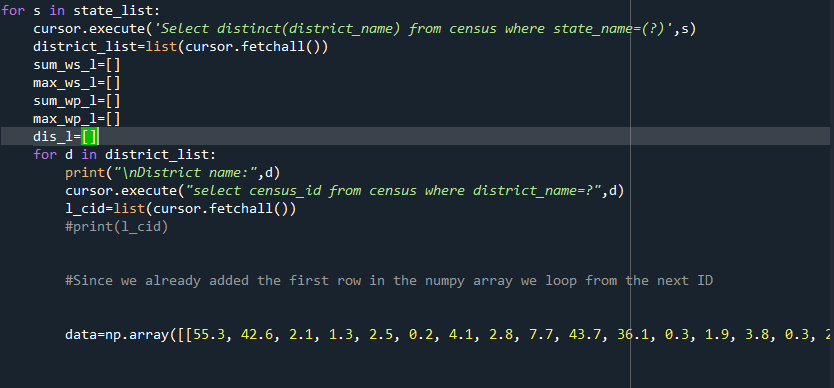


We wrote a query to retrieve distinct state names from our database and stored it in a list

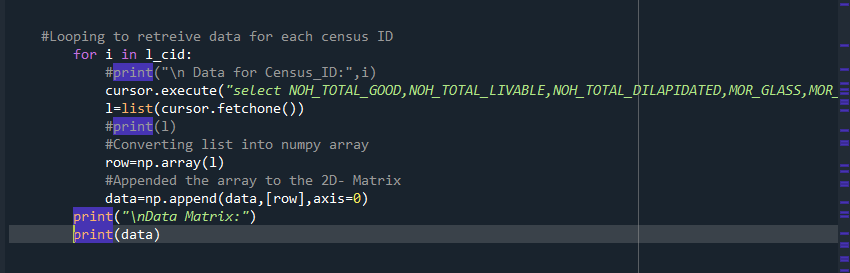




After retrieving all states we start evaluating the districts:

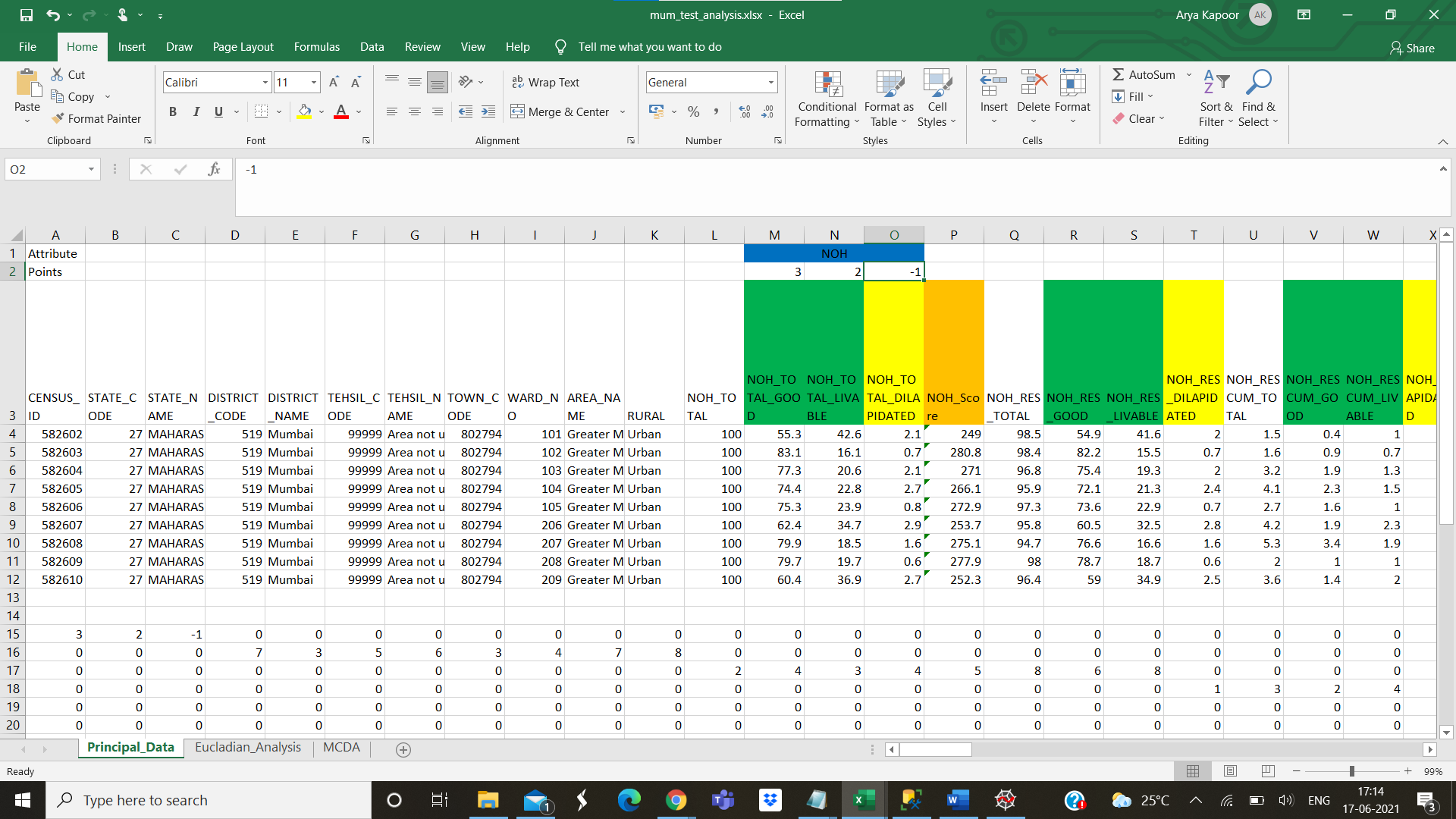


Using the sql query we are able to fetch the census id’s for each enumeration. After fetching this data for each id we append it in the data matrix containing enumeration details for each district.



This code helps us to prepare the data matrix

Data Matrix in tabular format



Our aim is to convert this data matrix into a data-frame on which the algorithm can run

If you take a closer look weights have been assigned above every attribute in the table:

For example in the attribute NOH (Number of Households) there are child attributes like (NOH\_TOTAL\_GOOD, NOH\_TOTAL\_LIVABLE, NOH\_TOTAL\_DILAPIDATED)

We multiply this percentage with the weights to get a score for the parent attribute.

**Attributes considered for ranking:**

NOH (Number of Households)

|  |  |
| --- | --- |
| Child Attribute | Points |
| NOH\_TOTAL\_GOOD | 3 |
| NOH\_TOTAL\_LIVABLE | 2 |
| NOH\_TOTAL\_DILAPIDATED | -1 |

MOR (Material of Roof)

|  |  |
| --- | --- |
| Child Attribute | Points |
| MOR\_GLASS | 7 |
| MOR\_PLASTIC\_POLYTHENE | 3 |
| MOR\_HAND\_MADE\_TILES | 5 |
| MOR\_MACHINE\_MADE\_TILES | 6 |
| MOR\_BURNT\_BRICK | 3 |
| MOR\_STONE | 3 |
| MOR\_GI\* | 7 |
| MOR\_CONCRETE | 8 |

\*GI- Galvanized Iron

MOW (Material of Wall)

|  |  |
| --- | --- |
| Child Attribute | Points |
| MOW\_GRASS | 2 |
| MOW\_PLASTIC | 4 |
| MOW\_MUD | 3 |
| MOW\_WOOD | 4 |
| MOW\_STONE\_NOT\_PACKED\_WITH\_MORTAR | 5 |
| MOW\_STONE\_PACKED\_WITH\_MORTAR | 8 |
| MOW\_GI\* | 6 |
| MOW\_BURNT\_BRICK | 8 |

\*GI- Galvanized Iron

MOF (Material of Roof)

|  |  |
| --- | --- |
| Child Attribute | Points |
| MOF\_MUD | 1 |
| MOF\_WOOD | 3 |
| MOF\_BURNT\_BRICK | 2 |
| MOF\_STONE | 4 |
| MOF\_CEMENT | 5 |
| MOF\_MOSAIC | 6 |

NOD (Number of Domestic Rooms)

|  |  |
| --- | --- |
| Child Attribute | Points |
| NOD\_NO | -1 |
| NOD\_ONE\_ROOM | 1 |
| NOD\_TWO\_ROOM | 2 |
| NOD\_THREE\_ROOM | 3 |
| NOD\_FOUR\_ROOM | 4 |
| NOD\_FIVE\_ROOM | 5 |
| NOD\_SIX\_ROOM\_AND\_ABOVE | 6 |

OWNERSHIP

|  |  |
| --- | --- |
| Child Attribute | Points |
| OWNERSHIP\_STATUS\_OWNED | 3 |
| OWNERSHIP\_STATUS\_RENTED | 2 |
| OWNERSHIP\_STATUS\_ANY\_OTHERS | 1 |

MSDW (Main Source of Drinking Water)

|  |  |
| --- | --- |
| Child Attribute | Points |
| MSDW\_TAP\_WATER\_FROM\_TREATED\_SOURCE | 9 |
| MSDW\_TAP\_WATER\_FROM\_UN\_TREATED\_SOURCE | 6 |
| MSDW\_COVERED\_WELL | 6 |
| MSDW\_UNCOVERED\_WELL | 5 |
| MSDW\_HANDPUMP | 7 |
| MSDW\_TUBEWELL | 7 |
| MSDW\_SPRING | 4 |
| MSDW\_RIVER | 2 |
| MSDW\_TANK | 8 |

LDW (Latrine Waste Depositing Facility)

|  |  |
| --- | --- |
| Child Attribute | Points |
| LDW\_WITHIN\_PREMISES | 3 |
| LDW\_NEAR\_PREMISES | 2 |
| LDW\_AWAY | 1 |

MSL (Main Source of Electricity)

|  |  |
| --- | --- |
| Child Attributes | Points |
| MSL\_ELECTRICITY | 5 |
| MSL\_KEROSENE | 3 |
| MSL\_SOLAR\_ENERGY | 5 |
| MSL\_OTHER\_OIL | 2 |
| MSL\_ANY\_OTHER | 0 |
| MSL\_NO\_LIGHTNING | -1 |

FLUSH

|  |  |
| --- | --- |
| Child Attributes | Points |
| FLUSH\_PIPED\_SEWER\_SYSTEM | 3 |
| FLUSH\_SEPTIC\_TANK | 2 |
| FLUSH\_OTHER\_SYSTEM | 1 |

BATHROOM

|  |  |
| --- | --- |
| Child Attributes | Points |
| NUMBER\_OF\_HOUSEHOLDS\_YES\_BATHROOM | 3 |
| NUMBER\_OF\_HOUSEHOLDS\_YES\_ENCLOSURE\_WITHOUT\_ROOF | 1 |
| NUMBER\_OF\_HOUSEHOLDS\_NO | -1 |

WWO (Waste Water Outlet)

|  |  |
| --- | --- |
| Child Attributes | Points |
| WWO\_CONNECTED\_TO\_CLOSED\_DRAINAGE | 3 |
| WWO\_CONNECTED\_TO\_OPEN\_DRAINAGE | 1 |
| WWO\_CONNECTED\_TO\_NO\_DRAINAGE | -1 |

TFUC (Fuel Used for Cooking)

|  |  |
| --- | --- |
| Child Attributes | Points |
| TFUC\_FIRE\_WOOD | 2 |
| TFUC\_CROP\_RESIDUE | 3 |
| TFUC\_COWDUNG\_CAKE | 1 |
| TFUC\_COAL | 4.5 |
| TFUC\_KEROSENE | 5 |
| TFUC\_LPG | 6 |
| TFUC\_ELECTRICITY | 5.5 |
| TFUC\_ELECTRICITY | 6 |
| TFUC\_ANY\_OTHER | 1 |
| TFUC\_NO\_COOKING | 0 |

KFUC (Fuel Used for Cooking)

|  |  |
| --- | --- |
| Child Attributes | Points |
| KF\_COOKING\_INSIDE\_HOUSE | 3 |
| KF\_HAS\_KITCHEN | 2 |
| KF\_HAS\_DOES\_NOT\_HAVE\_KITCHEN | 1 |
| KF\_COOKING\_OUTSIDE\_HOUSE | 1.5 |
| KF\_CO\_HAS\_KITCHEN | 0 |
| KF\_CO\_HAS\_NO\_KITCHEN | 0 |
| KF\_NO\_COOKING | 0 |

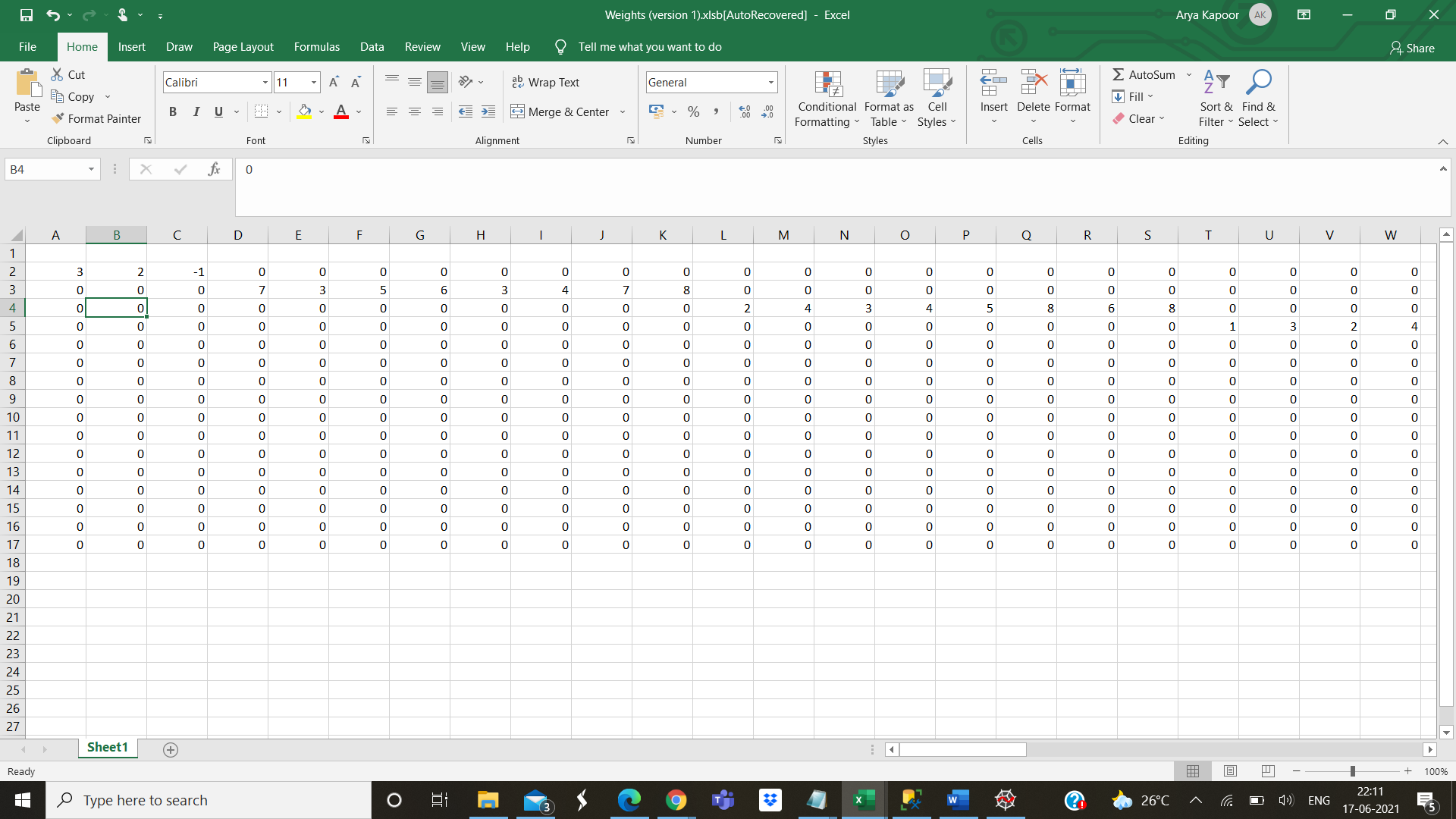
AOA (Availability of Assets)

|  |  |
| --- | --- |
| Child Attributes | Points |
| AOA\_RADIO | 1 |
| AOA\_TELEVISION | 2 |
| AOA\_COMPUTER\_HAS\_INTERNET | 3 |
| AOA\_COMPUTER\_HAS\_NO\_INTERNET | 4 |
| AOA\_TELEPHONE\_LANDLINE | 2 |
| AOA\_TELEPHONE\_MOBILE | 3 |
| AOA\_TELEPHONE\_BOTH | 4 |
| AOA\_BICYCLE | 4 |
| AOA\_SCOOTER | 5 |
| AOA\_VAN | 6 |
| AOA\_HOUSEHOLD\_WITH\_TV | 4 |

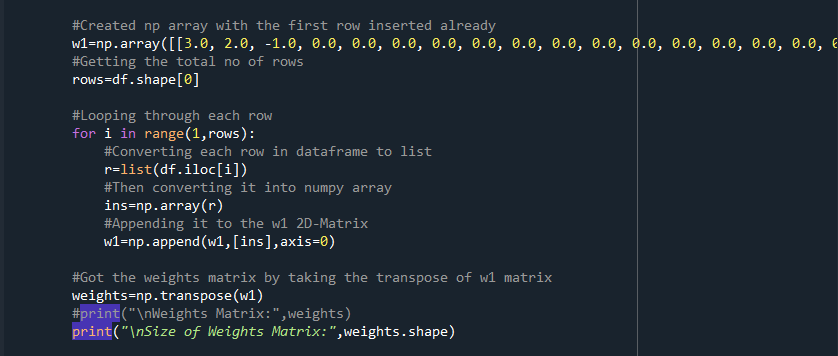
Household Classification

|  |  |
| --- | --- |
| Child Attributes | Points |
| HOUSEHOLD\_PERMANENT | 10 |
| HOUSEHOLD\_SEMI\_PERMANENT | 7.5 |
| HOUSEHOLD\_TOTAL\_TEMPORARY | 5 |
| HOUSEHOLD\_UNCLASSIFABLE | 2.5 |

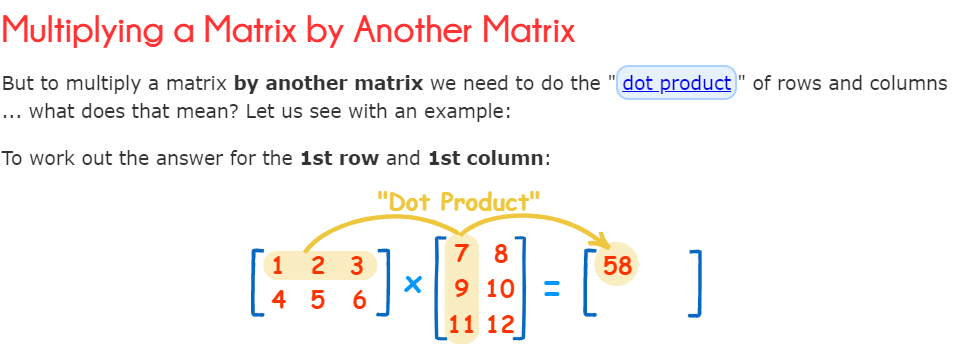
We store such weights in an excel file and then directly append it into the weights matrix.



Code to append this data into the weights matrix



After we form our Data and Weights Matrix it was time for us to get the results for the parent attributes using matrix multiplication.

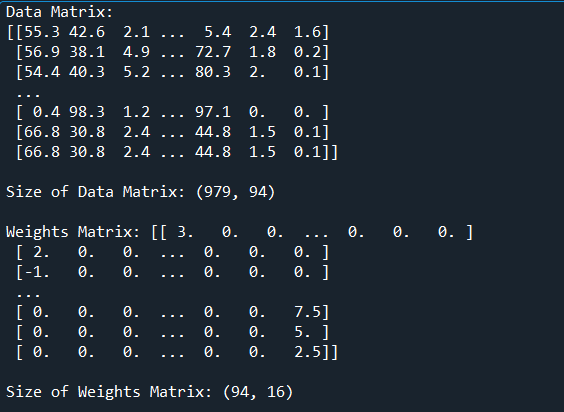


Using this same rule we multiply our Data Matrix of size (m \* n) and Weights Matrix of size (n\*p) to produce Results matrix (m\*p)

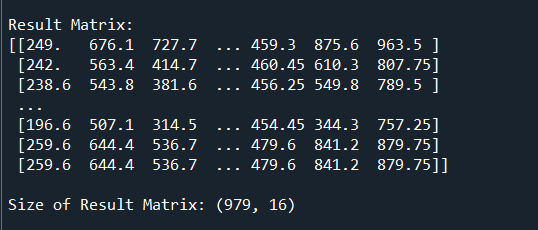
**Note:**

* **The number of rows in the Data Matrix are variable depending upon the number of rows for the district but the number of columns is fixed**
* **The size of weights matrix is constant 94 \* 16**

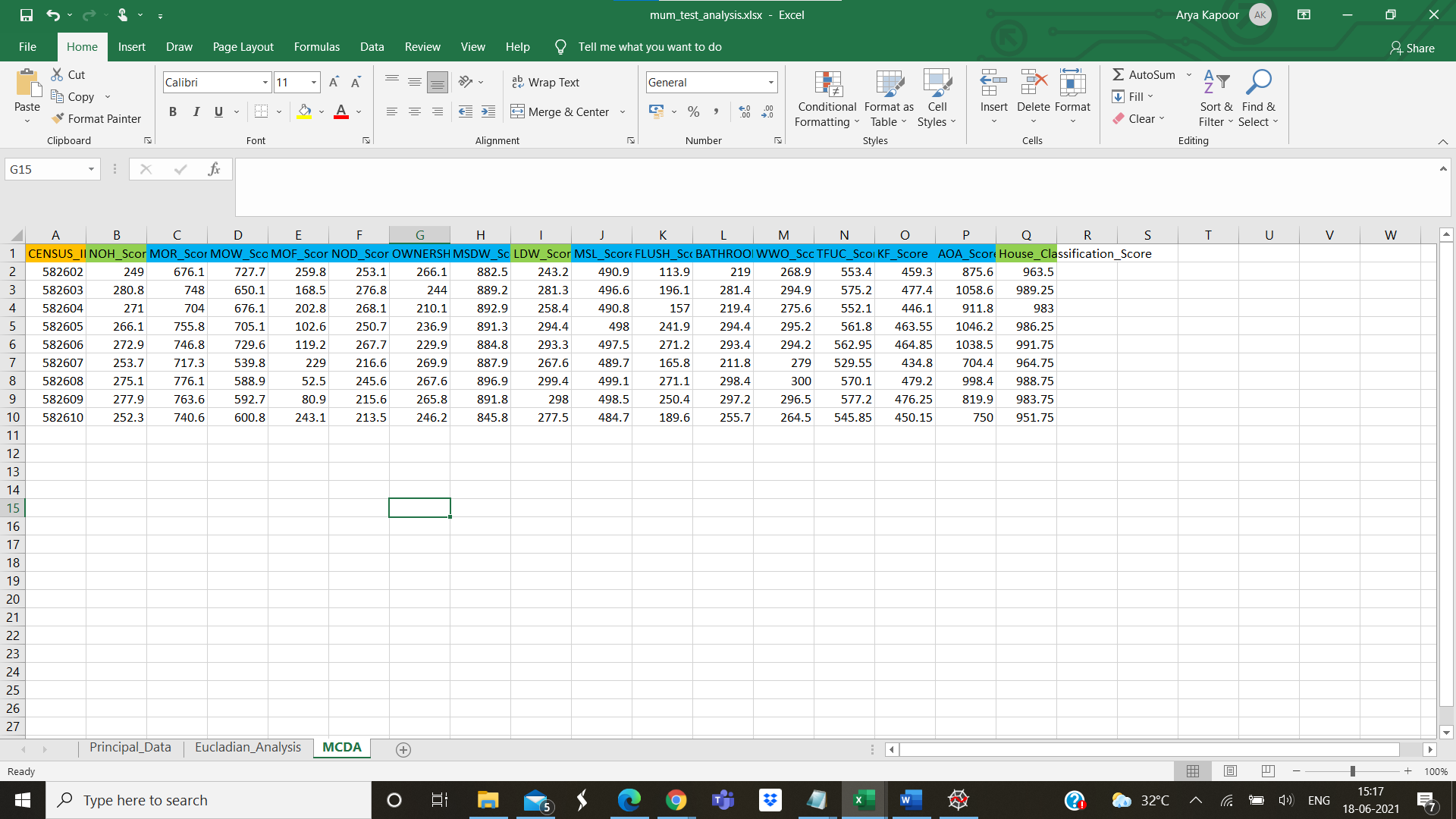
Below is the representation of the numpy array containing the Data and Weights Matrix



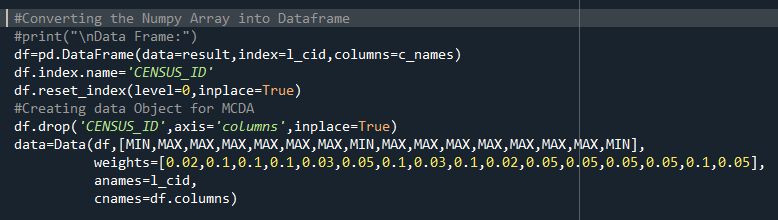
After multiplying this we get our Result Matrix

****

This is the tabular view of how does a result matrix looks like:



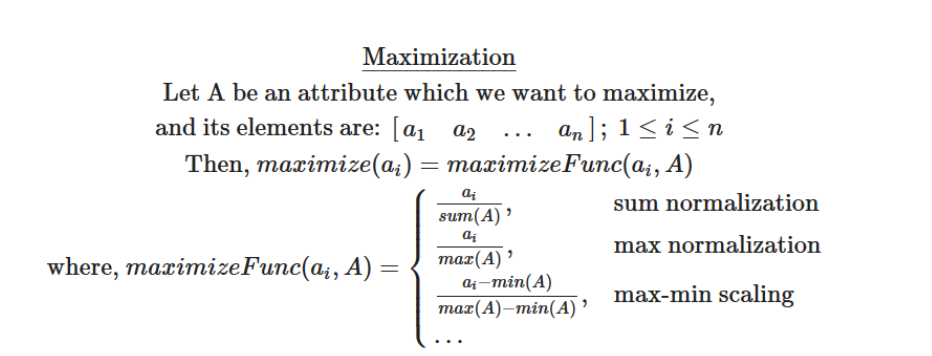
Now we must convert this Result Matrix into a Data-frame suitable for the MCDA algorithm.



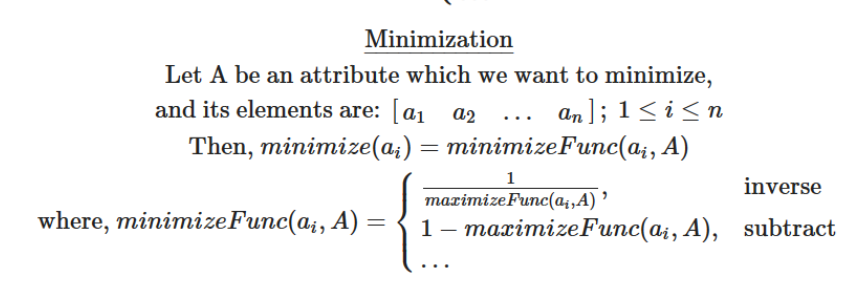
After we make the data-frame we load this into our Data Object which has weights assigned for each parent attribute (ie.NOH\_SCORE,MOR\_SCORE,MOF\_SCORE etc.).

We also take the copy of this data-frame so that the ranks and points can be appended into it after we run our algorithms.

It is essential to normalise the scores using maximization and minimization.



In the maximization we can normalise the data using the sum, maximum value and max-min scaling.



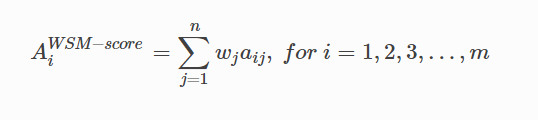
Minimization is just the inverse of maximization.

We have implemented sum and max normalisation on our data.

We have made use of two techniques to obtain ranks:

* Weighted Sum
* Weighted Product

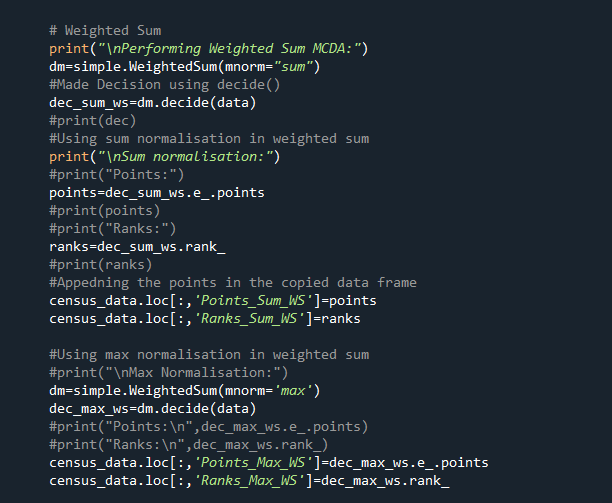
**Weighted Sum**



Here m is the number of rows in data-frame and n is the number of attributes considered for ranking.

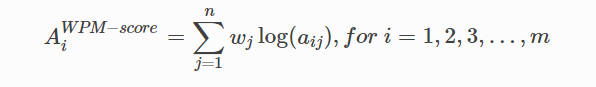
**Below is the code which performs Weighted Sum**

Here we have used two methods of normalization that is max and sum.

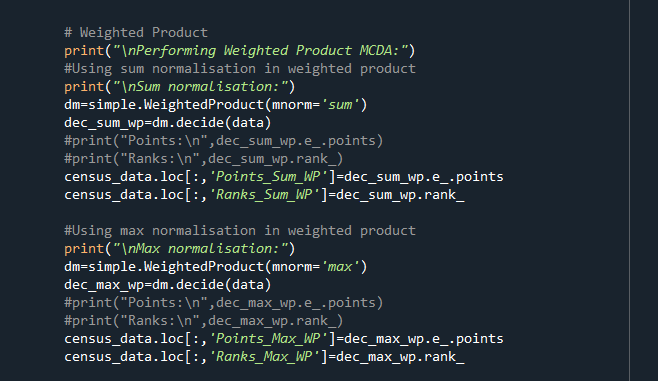


After obtaining the ranks and the points we append them in the data-frame.

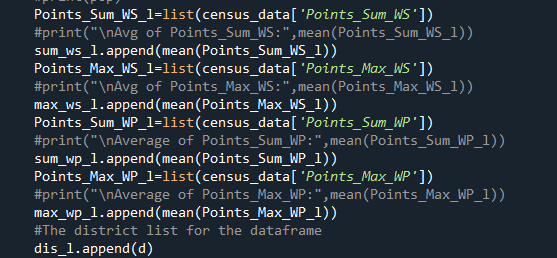
**Weighted Product**

****

Here m is the number of rows in the data-frame and n is the number of attributes considered for ranking.



Here we take the average of the points generated by each algorithm.



**Note:**

**Points\_Sum\_WS = points generated by Weighted Sum method on data normalised by sum**

**Points\_Max\_WS = points generated by Weighted Sum method on data normalised by maximum value**

**Points\_Sum\_WP = points generated by Weighted Product method on data normalised by sum**

**Points\_Max\_WP = points generated by Weighted Product method on data normalised by maxiumum value.**

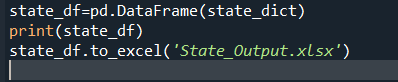
We append the maximum number of points for each district in another list and add the name of the district with maximum number of points in another list.



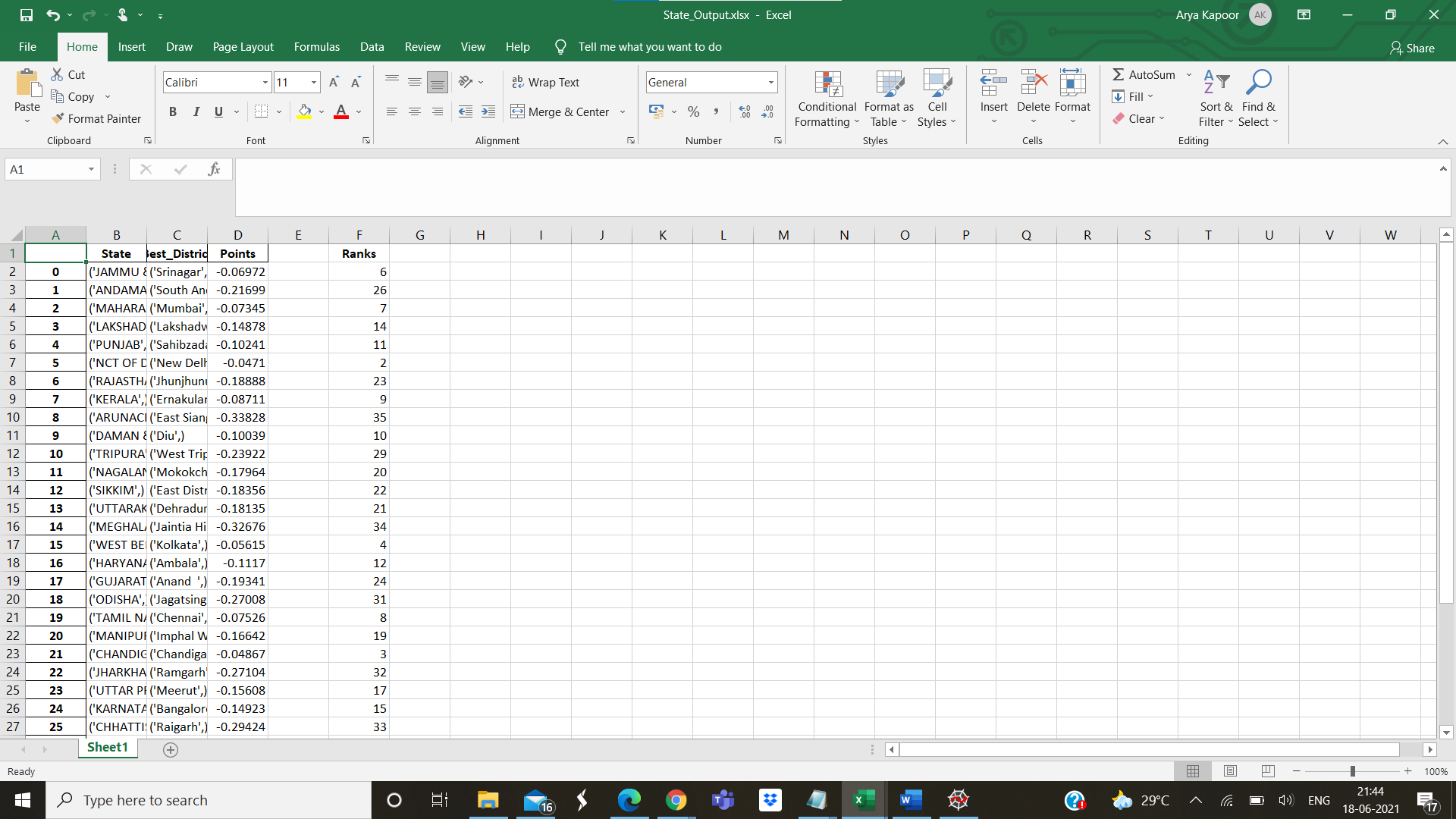
After forming these two lists we make a data-frame containing the name of state and district with the maximum points for ranking process.



We convert this data-frame into an excel file for ranking purposes.



Excel file is generated and states are ranked.



Result

After we run custom sort on this data in excel, we obtain these results.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **State** | **Best\_District** | **Points** |  | **Ranks** |
| ('ANDHRA PRADESH',) | ('Hyderabad',) | -0.04356 |  | 1 |
| ('NCT OF DELHI',) | ('New Delhi',) | -0.0471 |  | 2 |
| ('CHANDIGARH',) | ('Chandigarh',) | -0.04867 |  | 3 |
| ('WEST BENGAL',) | ('Kolkata',) | -0.05615 |  | 4 |
| ('PUDUCHERRY',) | ('Mahe',) | -0.05947 |  | 5 |
| ('JAMMU & KASHMIR',) | ('Srinagar',) | -0.06972 |  | 6 |
| ('MAHARASHTRA',) | ('Mumbai',) | -0.07345 |  | 7 |
| ('TAMIL NADU',) | ('Chennai',) | -0.07526 |  | 8 |
| ('KERALA',) | ('Ernakulam',) | -0.08711 |  | 9 |
| ('DAMAN & DIU',) | ('Diu',) | -0.10039 |  | 10 |
| ('PUNJAB',) | ('Sahibzada Ajit Singh Nagar',) | -0.10241 |  | 11 |
| ('HARYANA',) | ('Ambala',) | -0.1117 |  | 12 |
| ('GOA',) | ('North Goa',) | -0.12729 |  | 13 |
| ('LAKSHADWEEP',) | ('Lakshadweep',) | -0.14878 |  | 14 |
| ('KARNATAKA',) | ('Bangalore',) | -0.14923 |  | 15 |
| ('MIZORAM',) | ('Serchhip',) | -0.15365 |  | 16 |
| ('UTTAR PRADESH',) | ('Meerut',) | -0.15608 |  | 17 |
| ('HIMACHAL PRADESH',) | ('Sirmaur',) | -0.1655 |  | 18 |
| ('MANIPUR',) | ('Imphal West',) | -0.16642 |  | 19 |
| ('NAGALAND',) | ('Mokokchung',) | -0.17964 |  | 20 |
| ('UTTARAKHAND',) | ('Dehradun',) | -0.18135 |  | 21 |
| ('SIKKIM',) | ('East District',) | -0.18356 |  | 22 |
| ('RAJASTHAN',) | ('Jhunjhunun',) | -0.18888 |  | 23 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ('GUJARAT',) | ('Anand ',) | -0.19341 |  | 24 |
| ('BIHAR',) | ('Arwal',) | -0.20759 |  | 25 |
| ('ANDAMAN & NICOBAR ISLANDS',) | ('South Andaman',) | -0.21699 |  | 26 |
| ('MADHYA PRADESH',) | ('Indore',) | -0.22452 |  | 27 |
| ('ASSAM',) | ('Kamrup Metropolitan',) | -0.22758 |  | 28 |
| ('TRIPURA',) | ('West Tripura ',) | -0.23922 |  | 29 |
| ('DADRA & NAGAR HAVELI',) | ('Dadra & Nagar Haveli',) | -0.25475 |  | 30 |
| ('ODISHA',) | ('Jagatsinghapur ',) | -0.27008 |  | 31 |
| ('JHARKHAND',) | ('Ramgarh',) | -0.27104 |  | 32 |
| ('CHHATTISGARH',) | ('Raigarh',) | -0.29424 |  | 33 |
| ('MEGHALAYA',) | ('Jaintia Hills',) | -0.32676 |  | 34 |
| ('ARUNACHAL PRADESH',) | ('East Siang',) | -0.33828 |  | 35 |

Discussion

Hyderabad secures the first position because it has the least percentage of dilapidated houses. Having dilapidated houses has a negative impact on the points for NOH (Number of Households) Score.

Besides this attribute Hyderabad has also performed well in other aspects like only 1% of the population is not having bathrooms while in other cities like Delhi the number is as high as 14%.

A good proportion of the city has a proper waste water outlet and proper latrine facilities which also influences the points secured by the district. This also one of the primary reasons for Mumbai scoring less points despite being the financial capital.

Delhi secures the second position as it has the least percentage of houses having no lightning. Having more number of houses with no lightning has a negative impact on MSL (Main Source of Lightning) Score.

Delhi has performed well when it comes to number of households not having domestic rooms it is as low as 0.45% which helps it to secure second position.

Chandigarh secures the third position as it has less number of houses having no lightning and also lesser number of dilapidated houses. It has performed well in other attributes which fetch negative points. Hence it is able to secure third position.

In our algorithm we have associated negative points against certain attributes like (Dilapidated houses, Houses having no lightning , Houses having no drainage system etc.) .Chandigarh in this case has a poor performance in terms of number of households not having a bathroom hence secures third position.