

Academic year 2023-24

PROJECT-VA

NAME: Aman Singh Bhogal and Mausmi Sinha

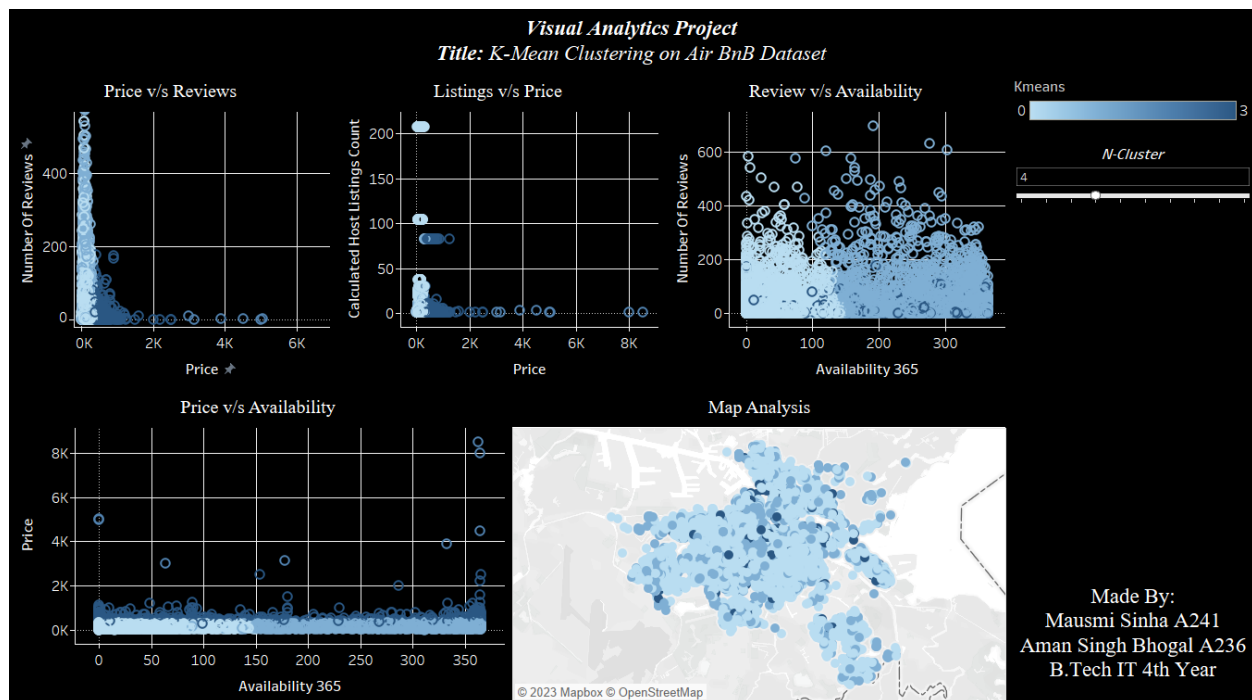
ROLL-NO: A236 & A241

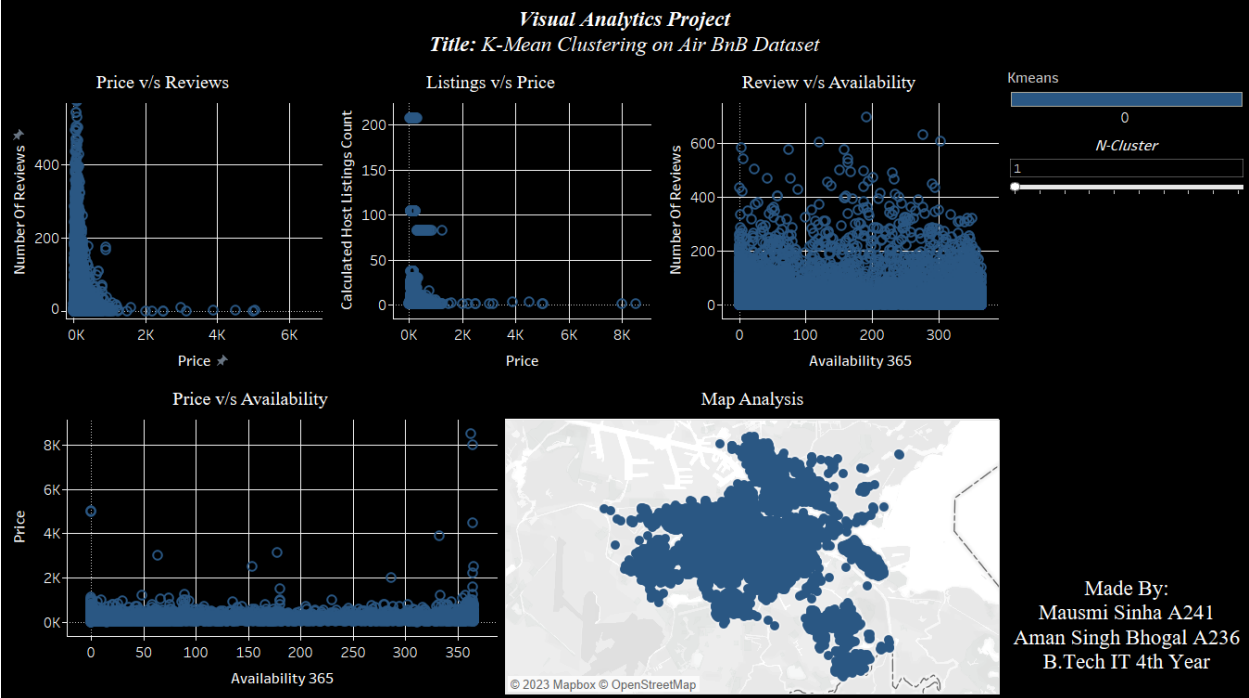
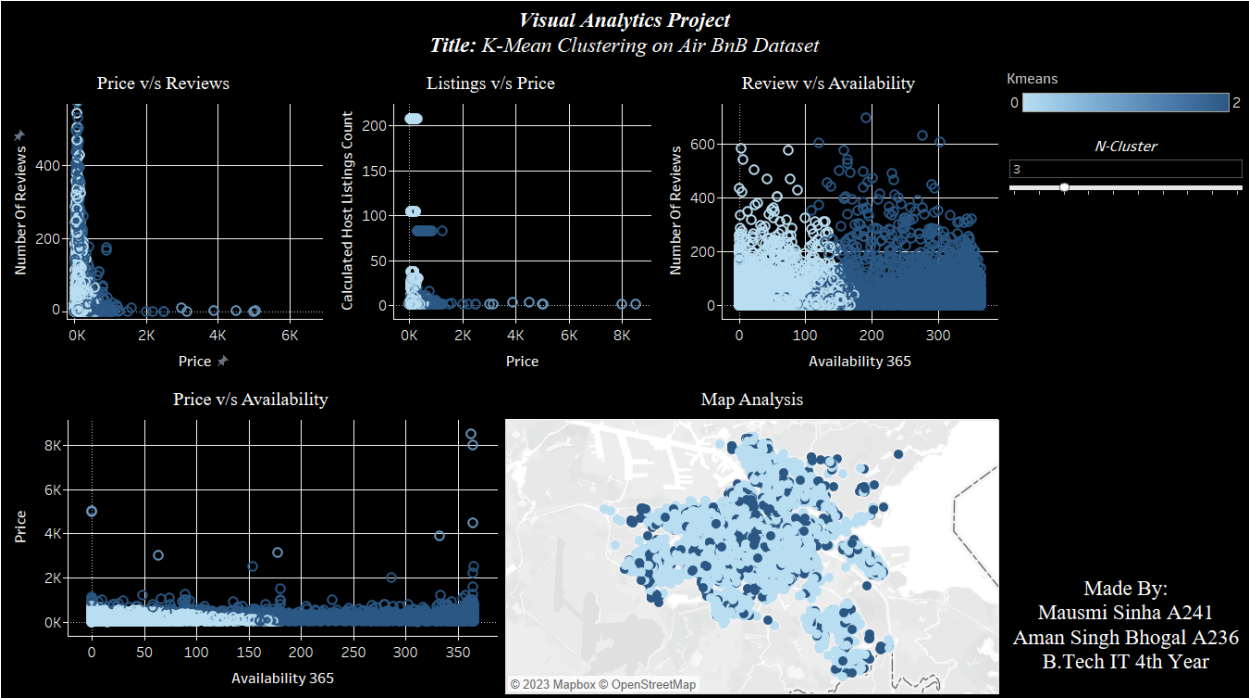
B.TECH IT-Final year.

Apply any Statistical Analysis Technique for any real-world problem with respect to any dataset in platforms like R, PYTHON etc Implement a ML Model. Provide an Interface of ML model to any BI tool to show the visualizations of the Project you have implemented.

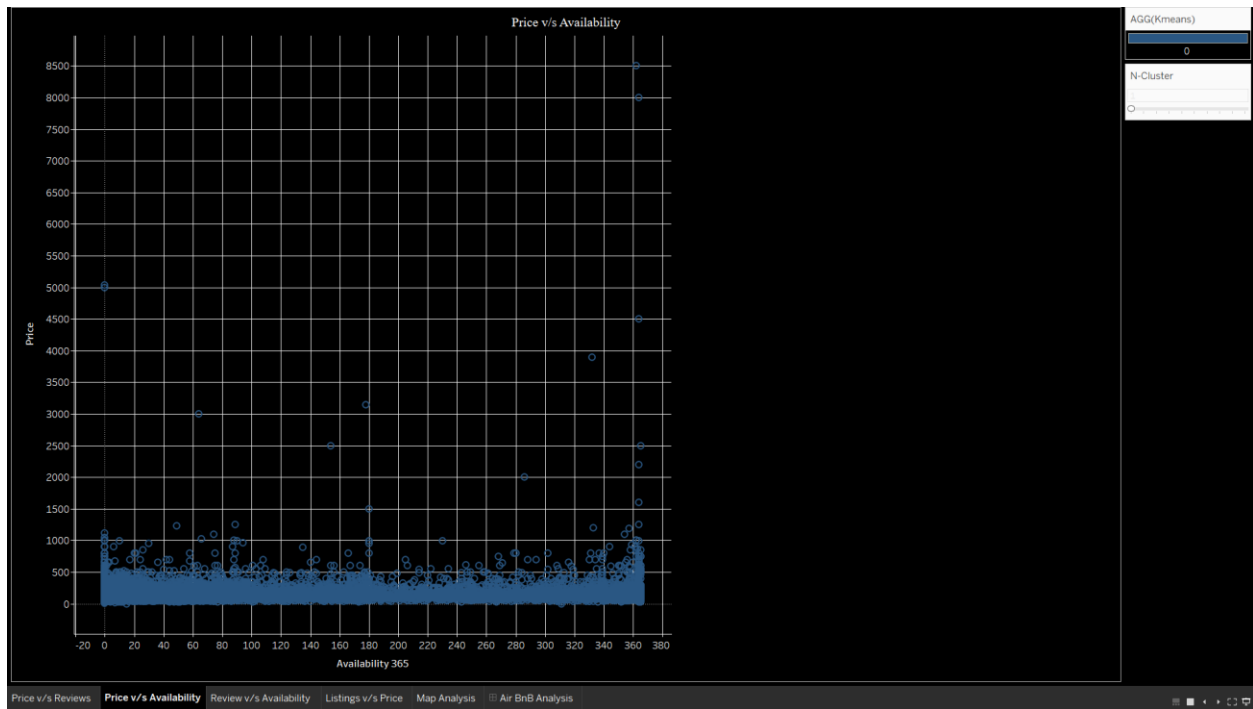
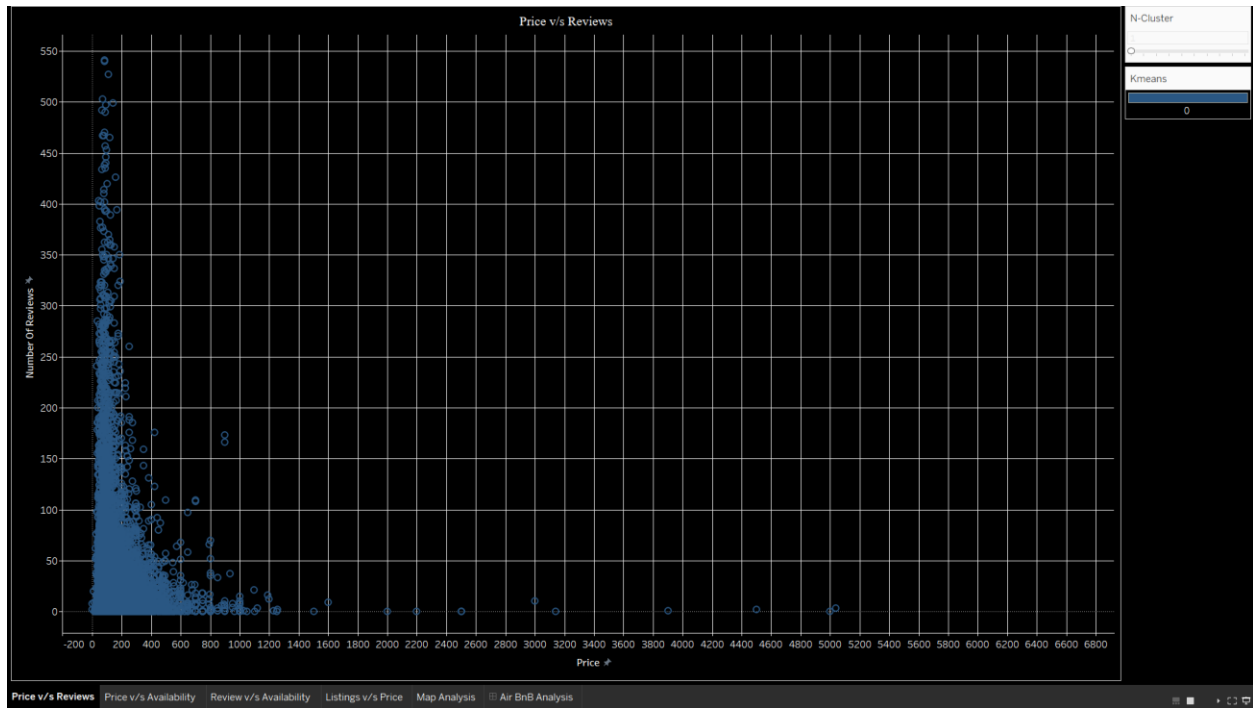
Title: *K-Mean Clustering on Air BnB Dataset*

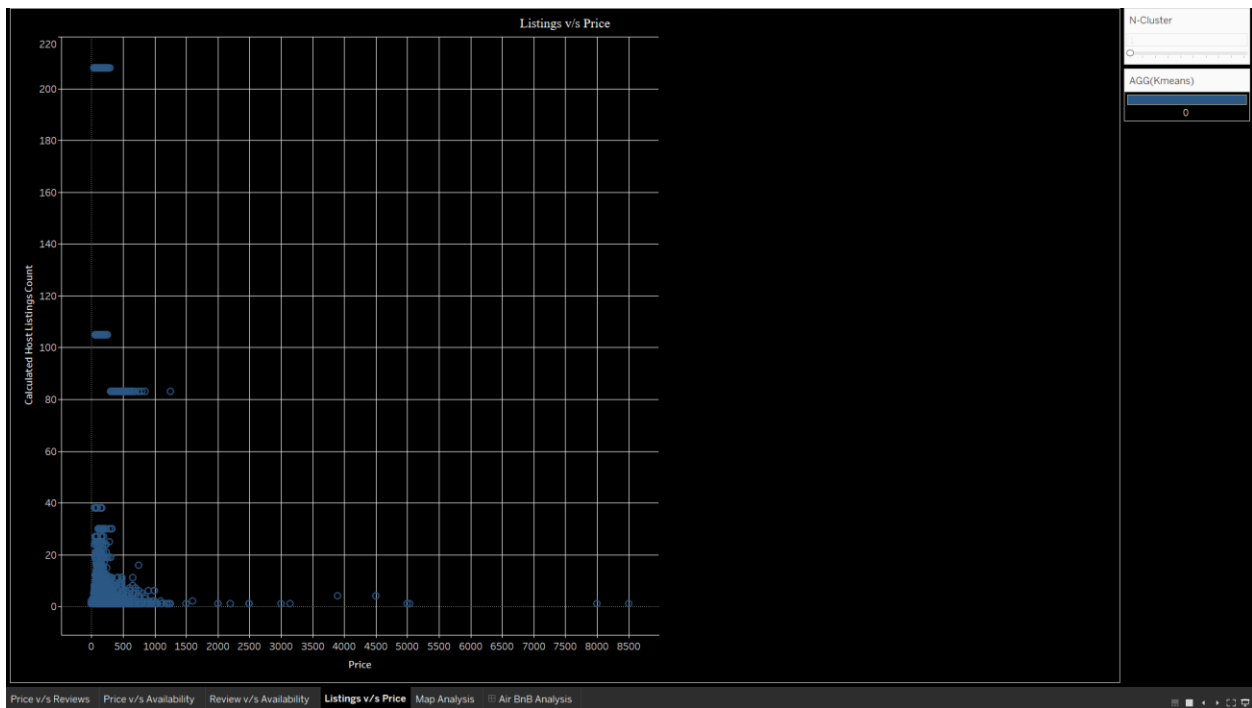
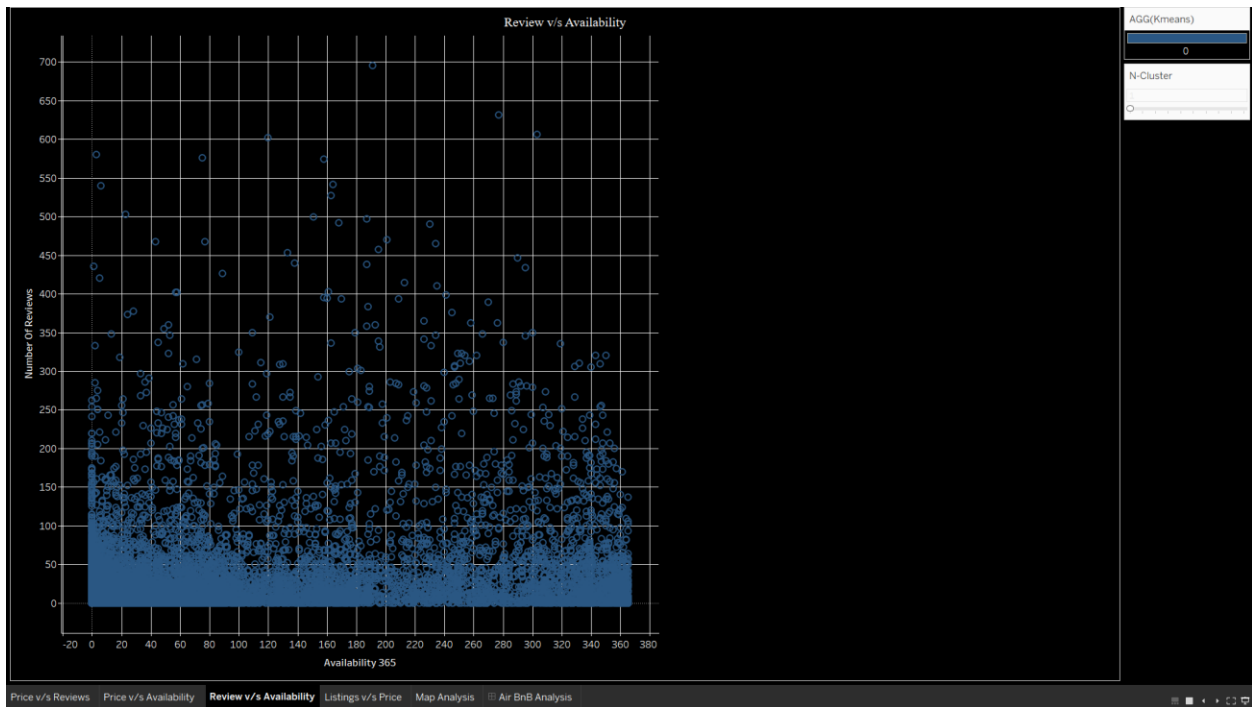
Output Screenshots:

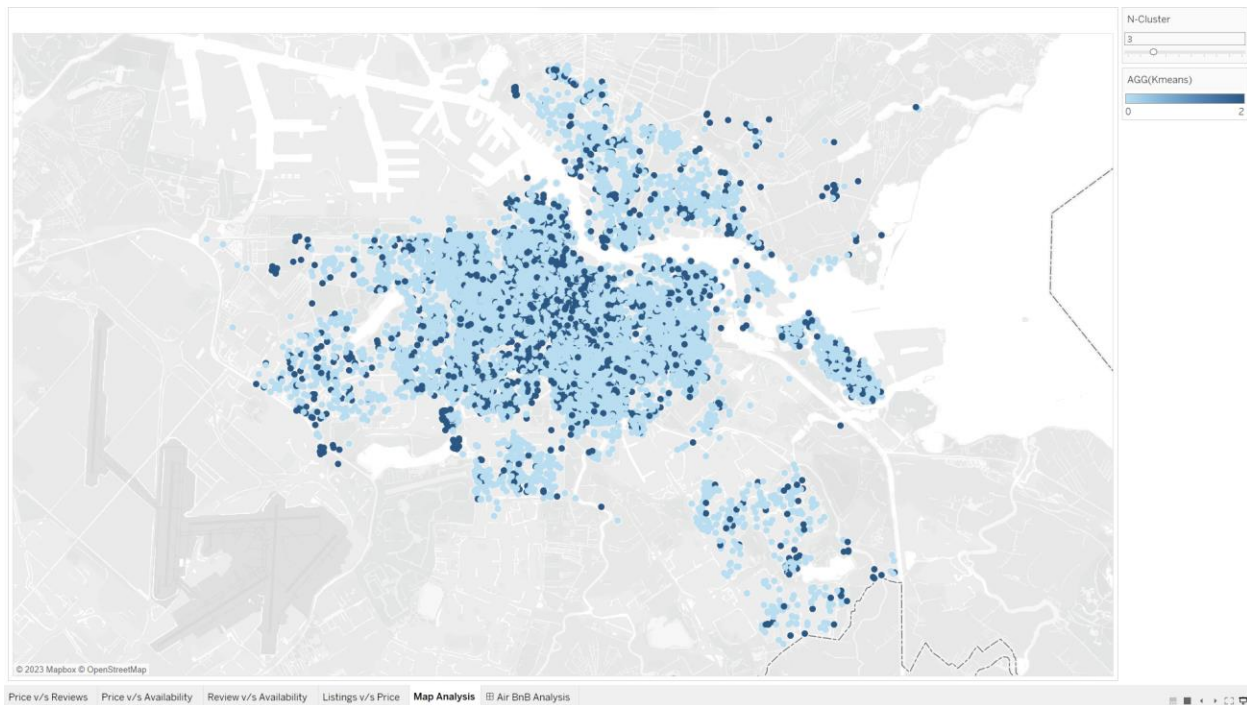




Individual Charts:







Method Used:

In this project we have used TabPy library that allows us to connect our tableau environment with Python Server.

Starting TabPy server:

```

C:\Windows\system32\cmd.e: X + v
(base) C:\Users\bhoga>tabpy
2023-10-25,11:36:02 [INFO] (app.py:app:300): Parsing config file C:\Users\bhoga\anaconda3\Lib\site-packages\tabpy\tabpy_server\app\...\common\default.conf
2023-10-25,11:36:02 [INFO] (app.py:app:363): Setting max request size to 104857600 bytes
2023-10-25,11:36:02 [INFO] (app.py:app:527): Loading state from state file C:\Users\bhoga\anaconda3\Lib\site-packages\tabpy\tabpy_server\state.ini

WARNING: This TabPy server is not currently configured for username/password authentication. This means that, because the TABPY_EVALUATE_ENABLE feature is enabled, there is the potential that unauthenticated individuals may be able to remotely execute code on this machine. We strongly advise against proceeding without authentication as it poses a significant security risk.

Do you wish to proceed without authentication? (y/N): y
2023-10-25,11:36:04 [INFO] (app.py:app:492): Password file is not specified: Authentication is not enabled
2023-10-25,11:36:04 [INFO] (app.py:app:411): Call context logging is disabled
2023-10-25,11:36:04 [INFO] (app.py:app:181): Initializing TabPy...
2023-10-25,11:36:04 [INFO] (callbacks.py:callbacks:43): Initializing TabPy Server...
2023-10-25,11:36:04 [INFO] (app.py:app:185): Done initializing TabPy.
2023-10-25,11:36:04 [INFO] (callbacks.py:callbacks:64): Initializing models...
2023-10-25,11:36:04 [INFO] (callbacks.py:callbacks:75): Load endpoint: prophet, version: 5, type: model
2023-10-25,11:36:04 [INFO] (python_service.py:python_service:89): Loading object: URI=prophet, URL=C:\Users\bhoga\anaconda3\Lib\site-packages\tabpy\tabpy_server\query_objects\prophet\5, version=5, is_updated=False
2023-10-25,11:36:04 [INFO] (callbacks.py:callbacks:75): Load endpoint: KMeans, version: 4, type: model
2023-10-25,11:36:04 [INFO] (app.py:app:151): Web service listening on port 9004
2023-10-25,11:36:05 [INFO] (query_object.py:query_object:80): Loaded query object "CustomQueryObject" successfully
2023-10-25,11:36:05 [INFO] (python_service.py:python_service:89): Loading object: URI=KMeans, URL=C:\Users\bhoga\anaconda3\Lib\site-packages\tabpy\tabpy_server\query_objects\KMeans\4, version=4, is_updated=False
2023-10-25,11:36:05 [INFO] (query_object.py:query_object:80): Loaded query object "CustomQueryObject" successfully
  
```

Tableau Python Server x +


localhost:9004

TabPy Server Info:

```
{
  "description": "",
  "creation_time": 0,
  "state_path": "C:\\Users\\bhoga\\anaconda3\\Lib\\site-packages\\tabpy\\tabpy_server",
  "server_version": "2.9.0",
  "name": "TabPy Server",
  "versions": {
    "v1": {
      "features": {
        "evaluate_enabled": true,
        "grip_enabled": true,
        "arrow_enabled": false
      }
    }
  }
}
```

Deployed Models:

```
{
  "prophet": {
    "description": "forecast time series data using prophet",
    "type": "model",
    "version": 5,
    "dependencies": [],
    "target": null,
    "creation_time": 1698165928,
    "last_modified_time": 1698169111,
    "schema": null,
    "docstring": "-- no docstring found in query function --"
  },
  "KMeans": {
    "description": "Air BnB K Means",
    "type": "model",
    "version": 4,
    "dependencies": [],
    "target": null,
    "creation_time": 1698205676,
    "last_modified_time": 1698206800,
    "schema": null,
    "docstring": "-- no docstring found in query function --"
  }
}
```



Windows taskbar: Search, 11:36 AM, 25-10-2023

Connecting Tableau to our TabPy Server:

Tableau - Air BnB

File Data Worksheet Dashboard Story Analysis Map Format Server Window Help

Entire View Show Me

Data Analytics Pages Columns Rows

Price
Number Of Reviews

Search

Tables

- # Host Id
- Alt Host Name
- # Id
- # Last Review
- Alt Name
- Alt Neighbourhood
- Alt Neighbourhood Group
- # Reviews Per Month
- Alt Room Type
- Alt Measure Names
- # Availability 365
- # Calculated Host Listings Co...
- # Kmeans
- # Latitude
- # Longitude
- # Minimum Nights
- # Number Of Reviews
- # Price
- # listings.csv (Count)
- # Measure Values

Parameters

- # N-Cluster

Filters

Marks

Automatic

Color Size Label

Detail Tooltip Shape

Kmeans Host Id

Manage Analytics Extensions Connection

Edit TabPy Connection

☐ Require SSL

Hostname localhost Port 9004

☐ Sign in with username and password

Username Password

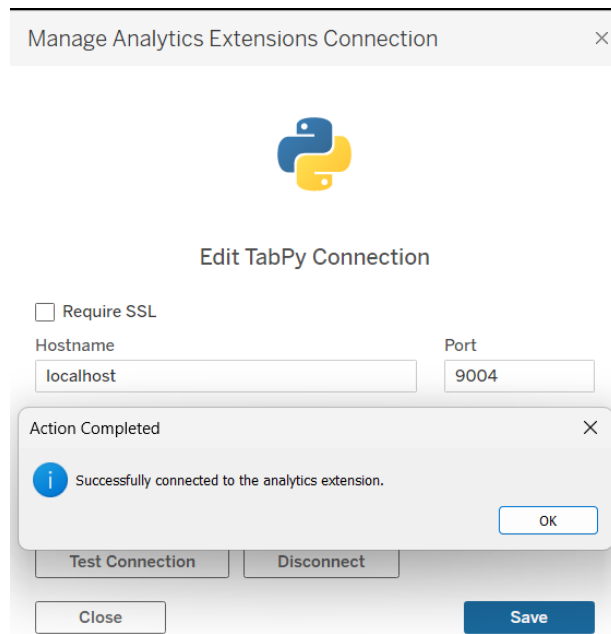
Test Connection Disconnect

Close Save

Number Of Reviews

Price

Windows taskbar: Search, 11:39 AM, 25-10-2023



Writing Script in Tableau using SCRIPT_INT:

We created a calculated field using SCRIPT_INT() that takes data attributes and N-Cluster Parameter as an input and passes it to our python code for calculating the clusters. We used K Means Clustering algorithm provided by sklearn library.

Code:

```
SCRIPT_INT("from sklearn.preprocessing import LabelEncoder
from sklearn.cluster import KMeans
import numpy as np
```

```
LE = LabelEncoder()
```

```
neighbourhood = LE.fit_transform(_arg1)
room_type = LE.fit_transform(_arg2)
```

```
price = _arg3
minimum_nights = _arg4
number_of_reviews = _arg5
availability_365 = _arg6
calculated_host_listings_count = _arg7
```

```
N = _arg8[0]
```

```
X = np.column_stack(
    [
```

```

    neighbourhood,
    room_type,
    price,
    minimum_nights,
    number_of_reviews,
    availability_365,
    calculated_host_listings_count,
]
)

kmeans = KMeans(n_clusters=N, random_state=35)
return kmeans.fit_predict(X).tolist()

",
ATTR([Neighbourhood]),
ATTR([Room Type]),
AVG([Price]),
MEDIAN([Minimum Nights]),
SUM([Number Of Reviews]),
AVG([Availability 365]),
AVG([Calculated Host Listings Count]),
[N-Cluster]
)

```

Kmeans

Results are computed along Host Id.

```

SCRIPT_INT("from sklearn.preprocessing import LabelEncoder
from sklearn.cluster import KMeans
import numpy as np

LE = LabelEncoder()

neighbourhood = LE.fit_transform(_arg1)
room_type = LE.fit_transform(_arg2)

price = _arg3
minimum_nights = _arg4
number_of_reviews = _arg5
availability_365 = _arg6
calculated_host_listings_count = _arg7

N = _arg8[0]

X = np.column_stack(
    [
        neighbourhood,
        room_type,
        price,
        minimum_nights,
        number_of_reviews,
        availability_365,
        calculated_host_listings_count,
    ]
)

kmeans = KMeans(n_clusters=N, random_state=35)
return kmeans.fit_predict(X).tolist()

",
ATTR([Neighbourhood]),
ATTR([Room Type]),
AVG([Price]),
MEDIAN([Minimum Nights]),
SUM([Number Of Reviews]),
AVG([Availability 365]),
AVG([Calculated Host Listings Count]),
[N-Cluster]
)

```

Default Table Calculation

The calculation is valid.
6 Dependencies
Apply
OK

All

Search

ABS
ACOS
AND
AREA
ASCII
ASIN
ATAN
ATAN2
ATTR
AVG
BUFFER
CASE
CEILING
CHAR
COLLECT
CONTAINS
CORR
COS
COT
COUNT
COUNTD
COVAR
COVARP
DATE
DATEADD
DATEDIFF
DATENAME

ABS (number)

Returns the absolute value of the given number.

Example: ABS (-7) = 7

Dialog box titled "Edit Parameter [N-Cluster]".

Name: N-Cluster

Properties:

- Data type: Integer
- Display format: 3
- Current value: 3
- Value when workbook opens: Current value

Allowable values:

- ☐ All
- ☐ List
- ☒ Range

Range of values:

- ☒ Minimum: 1
- ☒ Maximum: 10
- ☐ Step size: 1
- ☒ Fixed
- ☐ When workbook opens
- Add values from

Buttons: Cancel, OK

Conclusion:

In summary, our lab experiment showcased the seamless integration of a K-means prediction model into Tableau using the TabPy library within the Anaconda environment. This integration resulted in the creation of dynamic dashboards that adapt their visualizations based on user input, significantly enhancing the clarity and accessibility of the information presented. The collaborative effort between data scientists and visualization experts proved instrumental in not only developing a robust prediction model but also translating its output into actionable insights. This interdisciplinary approach highlights the potential of combining advanced machine learning techniques with interactive data visualization, demonstrating how such integration can revolutionize the way we analyze and comprehend complex datasets, paving the way for informed decision-making in various domains.