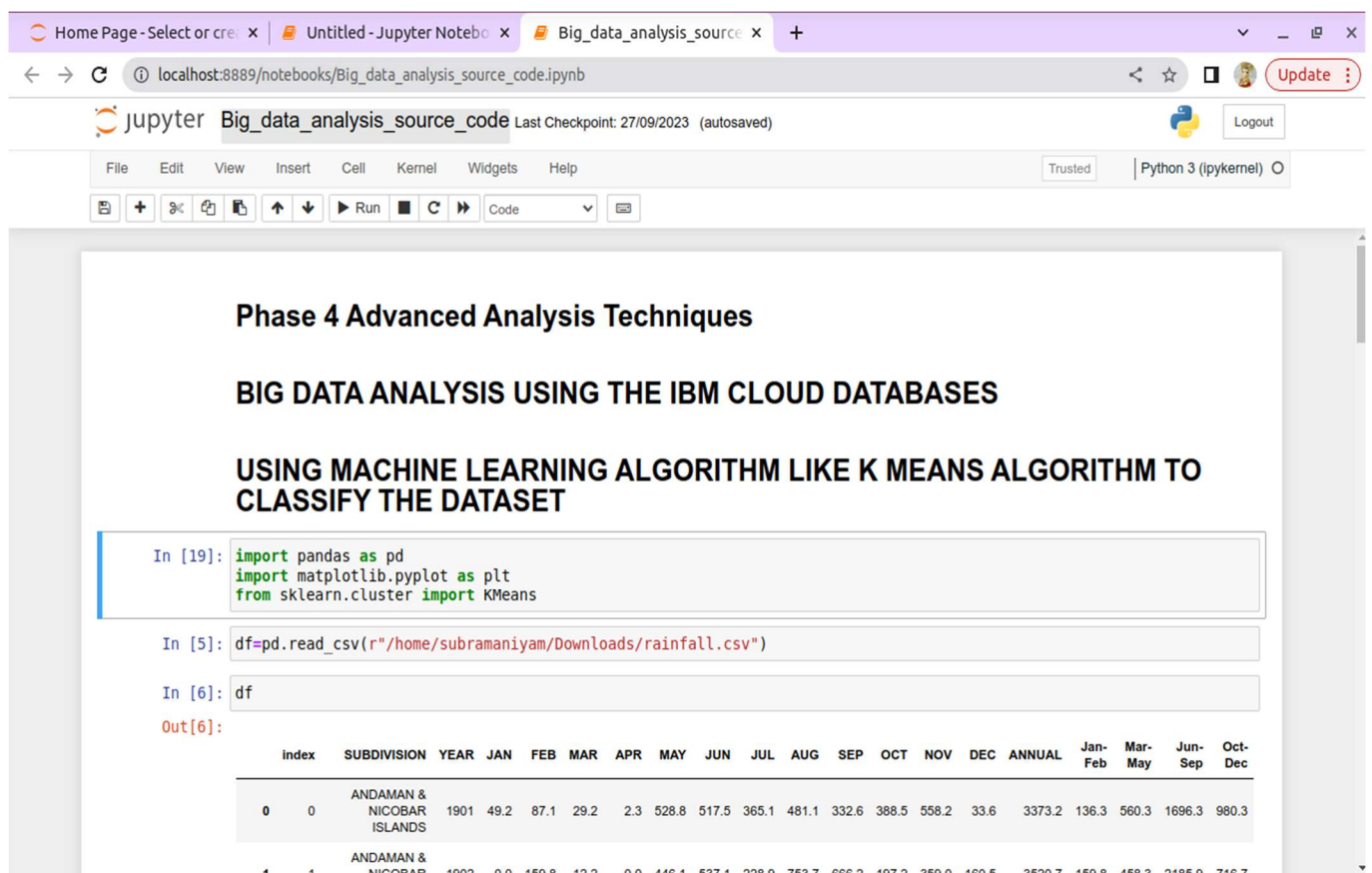


**Advanced Analytics Techniques:** Apply more complex analysis techniques, such as machine learning algorithms, time series analysis, or sentiment analysis, depending on the dataset and objectives.

**Follow the below steps for Advanced Analytics Techniques:**

**NOTE:** We are going to use the Machine Learning Algorithm like K Means Clustering Algorithm for analysis.

**Step 1 – Import the necessary libraries and the dataset in Jupyter Notebook.**



The screenshot shows a Jupyter Notebook titled "Big\_data\_analysis\_source\_code" running on a local host. The notebook contains the following code cells:

```
In [19]: import pandas as pd
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans
```

```
In [5]: df=pd.read_csv(r"/home/subramaniyam/Downloads/rainfall.csv")
```

```
In [6]: df
```

The output of the code cell shows a preview of the dataset:

	index	SUBDIVISION	YEAR	JAN	FEB	MAR	APR	MAY	JUN	JUL	AUG	SEP	OCT	NOV	DEC	ANNUAL	Jan- Feb	Mar- May	Jun- Sep	Oct- Dec
0	0	ANDAMAN & NICOBAR ISLANDS	1901	49.2	87.1	29.2	2.3	528.8	517.5	365.1	481.1	332.6	388.5	558.2	33.6	3373.2	136.3	560.3	1696.3	980.3
1	1	ANDAMAN & NICOBAR	1902	0.0	159.8	12.2	0.0	446.1	537.1	228.9	753.7	666.2	197.2	359.0	160.5	3520.7	159.8	458.3	2185.9	716.7

## Step 2 – Clean and remove the noisy data in the dataset using python script.

Home Page - Select or create a new notebook | Untitled - Jupyter Notebook | Big\_data\_analysis\_source | +

localhost:8889/notebooks/Big\_data\_analysis\_source\_code.ipynb

Jupyter Big\_data\_analysis\_source\_code Last Checkpoint: 27/09/2023 (autosaved)

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 (ipykernel)

```
In [11]: df2=df[df['YEAR']==2015]
In [12]: df2
Out[12]:
```

	Index	SUBDIVISION	YEAR	JAN	FEB	MAR	APR	MAY	JUN	JUL	AUG	SEP	OCT	NOV	DEC	ANNUAL	Jan-Feb	Mar-May	Jun-Sep	Oct-Dec
109	109	ANDAMAN & NICOBAR ISLANDS	2015	126.8	7.6	3.1	138.2	331.9	346.4	328.9	480.0	523.3	252.1	236.3	129.9	2904.6	134.4	473.2	1678.6	618.4
206	206	ARUNACHAL PRADESH	2015	30.8	47.5	97.5	287.1	238.9	637.9	329.3	595.5	374.2	65.2	33.8	29.8	2767.5	78.3	623.5	1936.9	128.5
321	321	ASSAM & MEGHALAYA	2015	13.4	15.5	37.5	250.9	332.5	558.5	300.1	590.9	279.9	62.6	14.0	15.2	2470.9	28.9	620.9	1729.3	91.9
436	436	NAGA MANI MIZO TRIPURA	2015	14.4	14.2	21.6	253.5	198.3	283.9	413.6	334.2	255.9	118.7	3.9	10.0	1922.4	28.7	473.4	1287.7	132.6
551	551	SUB HIMALAYAN WEST BENGAL & SIKKIM	2015	15.7	15.0	64.8	149.0	304.6	508.2	393.3	626.6	354.9	53.6	23.8	9.0	2518.6	30.7	518.5	1883.0	86.3
666	666	GANGETIC WEST BENGAL	2015	12.0	5.5	10.3	88.7	57.6	247.3	622.1	260.6	164.0	32.7	2.3	6.3	1530.3	18.4	165.6	1304.9	41.9

```
In [13]: df3=df2[['SUBDIVISION', 'ANNUAL']]
In [14]: df3
Out[14]:
```

	SUBDIVISION	ANNUAL
109	ANDAMAN & NICOBAR ISLANDS	2904.6
206	ARUNACHAL PRADESH	2767.5
321	ASSAM & MEGHALAYA	2470.9
436	NAGA MANI MIZO TRIPURA	1922.4
551	SUB HIMALAYAN WEST BENGAL & SIKKIM	2518.6
666	GANGETIC WEST BENGAL	1530.3

Home Page - Select or create a new notebook | Untitled - Jupyter Notebook | Big\_data\_analysis\_source | +

localhost:8889/notebooks/Big\_data\_analysis\_source\_code.ipynb

Jupyter Big\_data\_analysis\_source\_code Last Checkpoint: 27/09/2023 (autosaved)

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 (ipykernel)

```
In [13]: df3=df2[['SUBDIVISION', 'ANNUAL']]
In [14]: df3
Out[14]:
```

	SUBDIVISION	ANNUAL
109	ANDAMAN & NICOBAR ISLANDS	2904.6
206	ARUNACHAL PRADESH	2767.5
321	ASSAM & MEGHALAYA	2470.9
436	NAGA MANI MIZO TRIPURA	1922.4
551	SUB HIMALAYAN WEST BENGAL & SIKKIM	2518.6
666	GANGETIC WEST BENGAL	1530.3
781	ORISSA	1210.1
896	JHARKHAND	1081.8
1011	BIHAR	872.7
1126	EAST UTTAR PRADESH	603.3
1241	WEST UTTAR PRADESH	582.7
1356	UTTARAKHAND	1247.6
1471	HARYANA DELHI & CHANDIGARH	435.3
1586	PUNJAB	510.8
1701	HIMACHAL PRADESH	1210.5
1816	JAMMU & KASHMIR	1572.8
1931	WEST RAJASTHAN	458.4

**Step 3** – Store the necessary values into the empty array for plotting purpose.

```
Home Page - Select or cre x | Untitled - Jupyter Notebo x | Big_data_analysis_source x +
localhost:8889/notebooks/Big_data_analysis_source_code.ipynb
jupyter Big_data_analysis_source_code Last Checkpoint: 27/09/2023 (autosaved)
Python 3 (ipykernel)

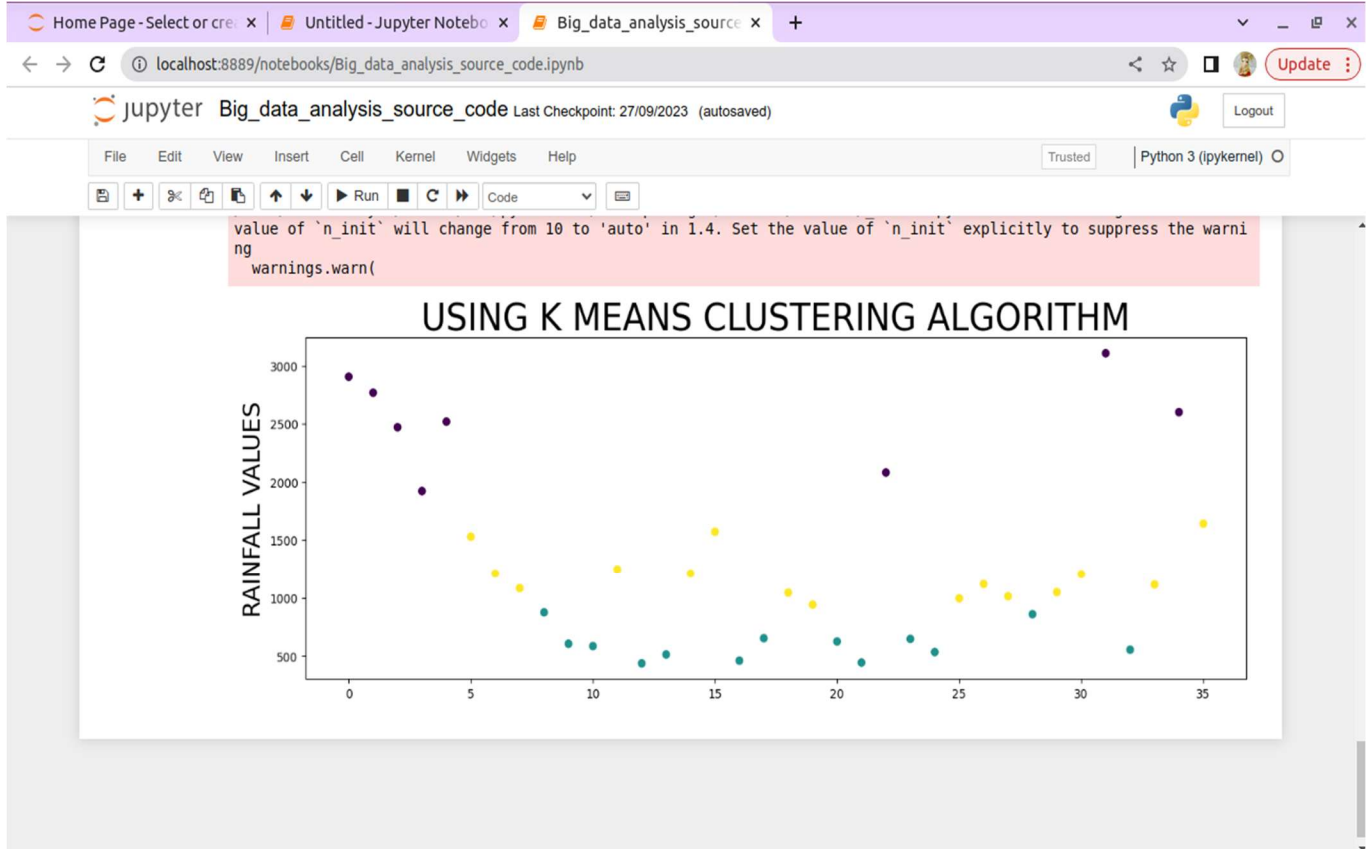
In [61]: df4=df3['SUBDIVISION']
df5=df3['ANNUAL']
states=[]
annual_rainfall_values=[]
for i in range(0,len(df4),1):
    states.append(i)
    annual_rainfall_values.append(df5.iloc[i])
print(states)
print(annual_rainfall_values)

[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30,
31, 32, 33, 34, 35]
[2904.6, 2767.5, 2470.9, 1922.4, 2518.6, 1530.3, 1210.1, 1081.8, 872.7, 603.3, 582.7, 1247.6, 435.3, 510.8, 1210.
5, 1572.8, 458.4, 650.7, 1042.3, 939.2, 622.9, 441.7, 2082.0, 644.5, 532.2, 993.8, 1117.6, 1010.9, 857.3, 1047.1,
1204.6, 3106.0, 551.9, 1112.5, 2600.6, 1642.9]

In [67]: whole_data=list(zip(states,annual_rainfall_values))
kmeans = KMeans(n_clusters=3)
kmeans.fit(whole_data)
width1 = 15.
height1 = 5.
width,height1 = (width1,height1)
plt.figure(figsize=width,height1)
plt.title("USING K MEANS CLUSTERING ALGORITHM",fontsize=30)
plt.ylabel("RAINFALL VALUES",fontsize=20)
plt.scatter(states,annual_rainfall_values , c=kmeans.labels_)
plt.show()

/home/subramaniyam/.local/lib/python3.10/site-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default
value of 'n_init' will change from 10 to 'auto' in 1.4. Set the value of 'n_init' explicitly to suppress the warni
ng
```

**Step 4** – After storing the values in array using K Means Clustering Algorithm to plot the graph and analyze the results.



**Visualization:** Create visualizations to showcase the analysis results. Use tools like Matplotlib, Plotly, or IBM Watson Studio for creating graphs and charts.

## Follow the below steps for Advanced Analytics Techniques:

**NOTE:** We are going to use the IBM Watson Studio for creating graphs and charts.

Step 1 – Open our cloud account and create the IBM Watson Studio then go to the Resource Pool and click Artificial Intelligence and Machine Learning and choose IBM Watson Studio.

The screenshot displays the IBM Watson Studio interface within a Google Chrome browser. The browser's address bar shows the URL: `eu-gb.dataplatform.cloud.ibm.com/shaper?project_id=64804e62-d699-401f-bf9e-24b7e951c21b&dataset_id=a58c1ec8-e...`. The page title is "IBM Cloud Pak for Data". The breadcrumb navigation indicates the current location: `Projects / rainfall_visualization / rainfall-1.csv / Data Refinery`.

The interface is divided into three main sections:

- Steps (3):** A list of steps for the data refinery process:
  - 1. Convert column type:** Automatically converted one or more columns to inferred data types. Strings that are converted to decimal use a dot (.) for the decimal symbol. (Auto-generated)
  - 2. Custom code:** `select('SUBDIVISION','YEAR','ANNUAL')`
  - 3. Custom code:** `filter('YEAR' == 2015)` (Just added)
- Data Table:** A table with 3 columns: `SUBDIVISI...` (String), `YEAR` (Integer), and `ANNUAL` (Decimal). The table displays 11 rows of data for the year 2015, representing various Indian states and union territories.
- Visualizations:** A section for creating visualizations, currently empty.

At the bottom of the interface, it shows "Viewing: 36 rows, 3 columns" and "Full data set: 4116 rows, 20 columns".

	SUBDIVISI...	YEAR	ANNUAL
	String	Integer	Decimal
1	ANDAMAN & NICO...	2015	2904.6
2	ARUNACHAL PRAD...	2015	2767.5
3	ASSAM & MEGHALA...	2015	2470.9
4	NAGA MANI MIZO T...	2015	1922.4
5	SUB HIMALAYAN W...	2015	2518.6
6	GANGETIC WEST B...	2015	1530.3
7	ORISSA	2015	1210.1
8	JHARKHAND	2015	1081.8
9	BIHAR	2015	872.7
10	EAST UTTAR PRAD...	2015	603.3
11	WEST UTTAR PRAD...	2015	582.7
	UTTARAKHAND	2015	1247.6

Step 2 – Load the Dataset and put some queries to refine the data for our visualization.

The screenshot shows the IBM Cloud 'Resource list' page. The table lists resources grouped by category. The 'Storage' group contains one resource: 'Cloud Object Storage-ar' (Default group, Global location, Cloud Object Storage product, Active status). The 'AI / Machine Learning' group contains one resource: 'Watson Studio-ut' (Default group, London location, Watson Studio product, Active status). Other groups like Networking, Converged infrastructure, Enterprise applications, Analytics, Blockchain, Databases, and Developer tools are listed but empty.

Name	Group	Location	Product	Status	Tags
Cloud Object Storage-ar	Default	Global	Cloud Object Storage	Active	1
Watson Studio-ut	Default	London	Watson Studio	Active	—

Step 3 – Finally, using the refine script to perform the visualization.

