Data Refinery

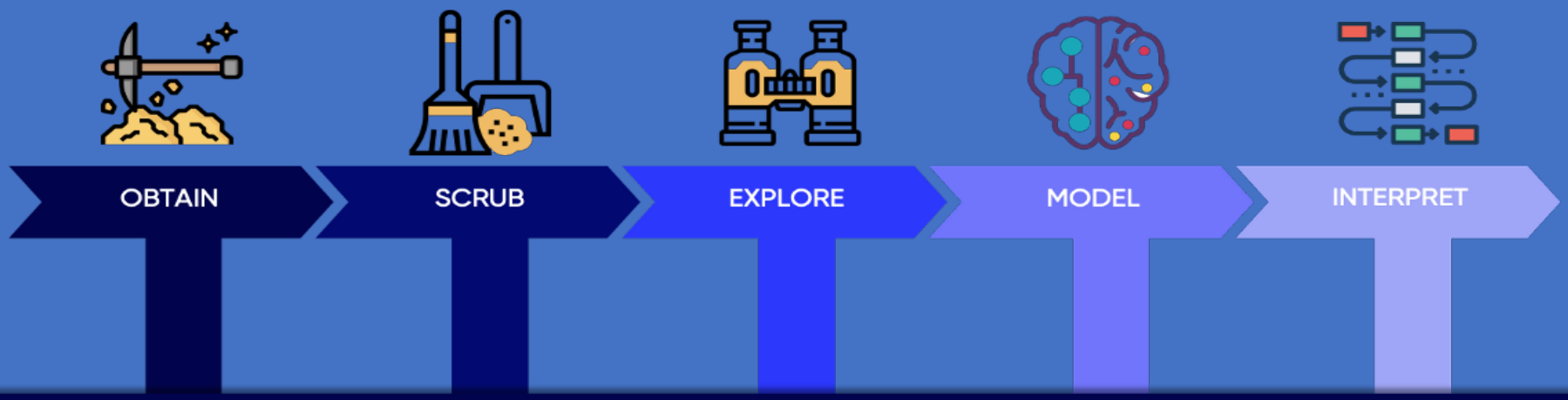
Naresh olladapu
Developer Advocate

IBM **Developer**

What is Data Science?

- Data science is the study of data.
- It involves developing methods of recording, storing, and analysing data to effectively extract useful information.
- The goal of data science is to gain insights and knowledge from any type of data (Structured or Unstructured)

Data Science Process



O
Gather data from relevant sources

S
Clean data to formats that machine understands

E
Find significant patterns and trends using statistical methods

M
Construct models to predict and forecast

N
Put the results into good use

Originally by Hilary Mason and Chris Wiggins

What is Data Refinery?

- Consists of –
  - Cleansing Data - fix or remove data that is incorrect, incomplete, improperly formatted, or duplicated
  - Shape Data - customize it by filtering, sorting, combining or removing columns, and performing operations

Outcome is creation of customized Data Refinery flow

Why is Data Refinery Important?

- Dataset needs to be structured and combined from multiple sources.
- Dataset might contain discrepancies in the names or the codes.
- Dataset might contains outliers or errors.
- Dataset lacks your attributes of interest for analysis.
- Noisy Datasets can lead to incorrect Insights.

Challenges in Data Refinery?

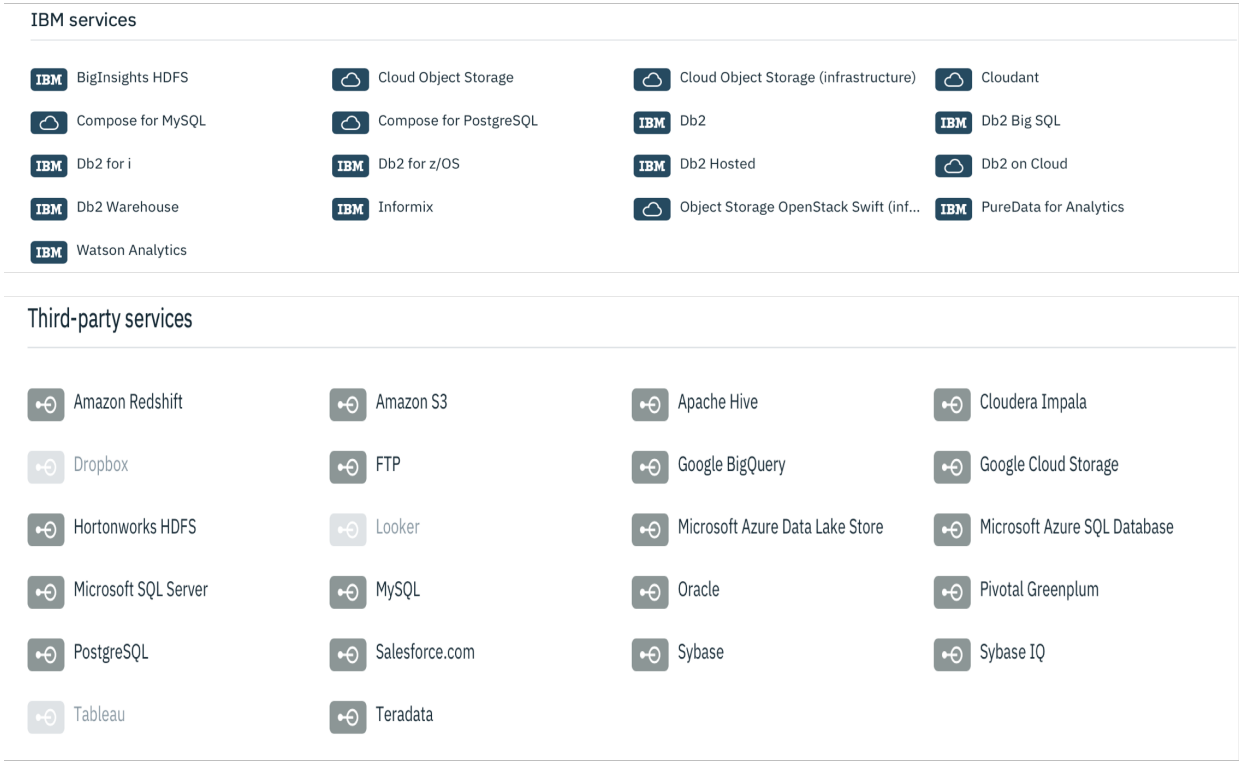
- Handle data from Multiple sources (NoSQL, Hadoop, RDMS, Dropbox etc.)
- Increase the Compute and storage based on our requirement very easily.
- Data refinery process should provide diagnostic and cognitive Analytics.
- Scheduling Data refinery process on a real time data streaming from sources like social media , IoT devices etc.
- Age old ETL process is slow, time consuming , limits output and performance.

Exploring Data Refinery Tool

- Adding data to Data Refinery
- View your Sample Data set
- Specifying the format of the data
- Validating the data
- Visualizing the data
- Scheduling Data Refinery flows
- Managing Data Refinery flows
- GUI operations

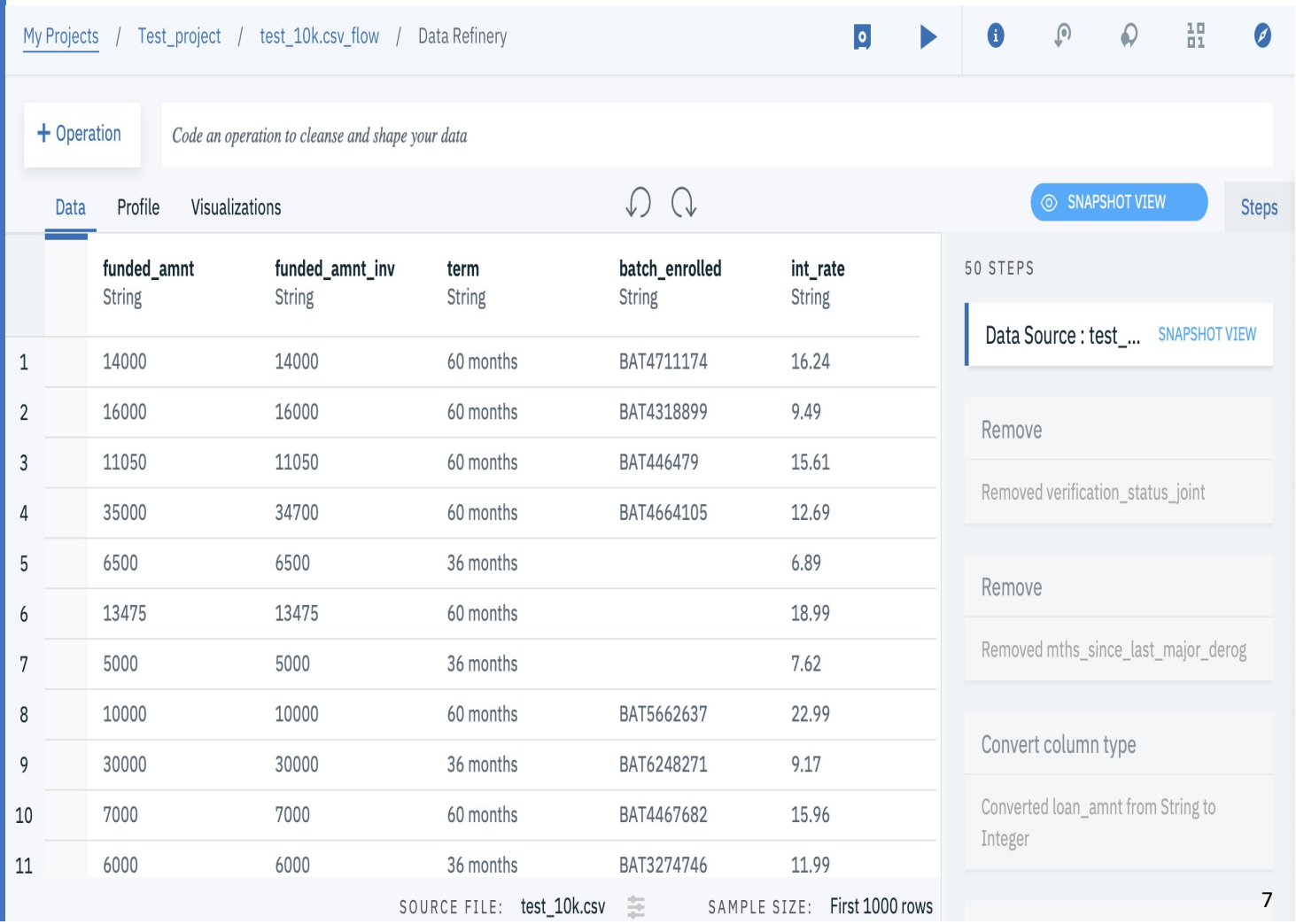
Adding data to Data Refinery

- Push Files to Assets Directory (Drag & Drop)
- Create connections with IBM or Third Party Storage Services



View Sample Data Set

- This helps to view the subset of data for exploring the columns and their data formats



	funded_amnt String	funded_amnt_inv String	term String	batch_enrolled String	int_rate String
1	14000	14000	60 months	BAT4711174	16.24
2	16000	16000	60 months	BAT4318899	9.49
3	11050	11050	60 months	BAT446479	15.61
4	35000	34700	60 months	BAT4664105	12.69
5	6500	6500	36 months		6.89
6	13475	13475	60 months		18.99
7	5000	5000	36 months		7.62
8	10000	10000	60 months	BAT5662637	22.99
9	30000	30000	36 months	BAT6248271	9.17
10	7000	7000	60 months	BAT4467682	15.96
11	6000	6000	36 months	BAT3274746	11.99

Validating the Data

- Remove unwanted rows for your problem Case
- Check the Datatypes of columns and change them according to the data
- If required change the column names according to your convenience
- Remove the rows of a dataset based on emptiness of columns(based on a threshold of column emptiness)
- Mask the Confidential data which cannot be exposed to Data Scientist/Software Engineers
- Remove/Replace Unnecessary Characters from the String Columns
- Create new columns by applying conditions on two or more columns

Validating the Data

Combine two or more columns which can be represented as single feature for better understanding and readability.

Split columns based on the separators(alphanumeric) or patterns or based on position.

Apply Math functions on columns to derive new data/columns from the existing dataset/columns.

Fill the missing values with constant ,mean,min,max,median for integer/decimal columns.

Join the two or more datasets based on the common key columns.

Arrange the dataset in required order based on columns [Ascending/Descending order]

Sample Python Code Snippet:

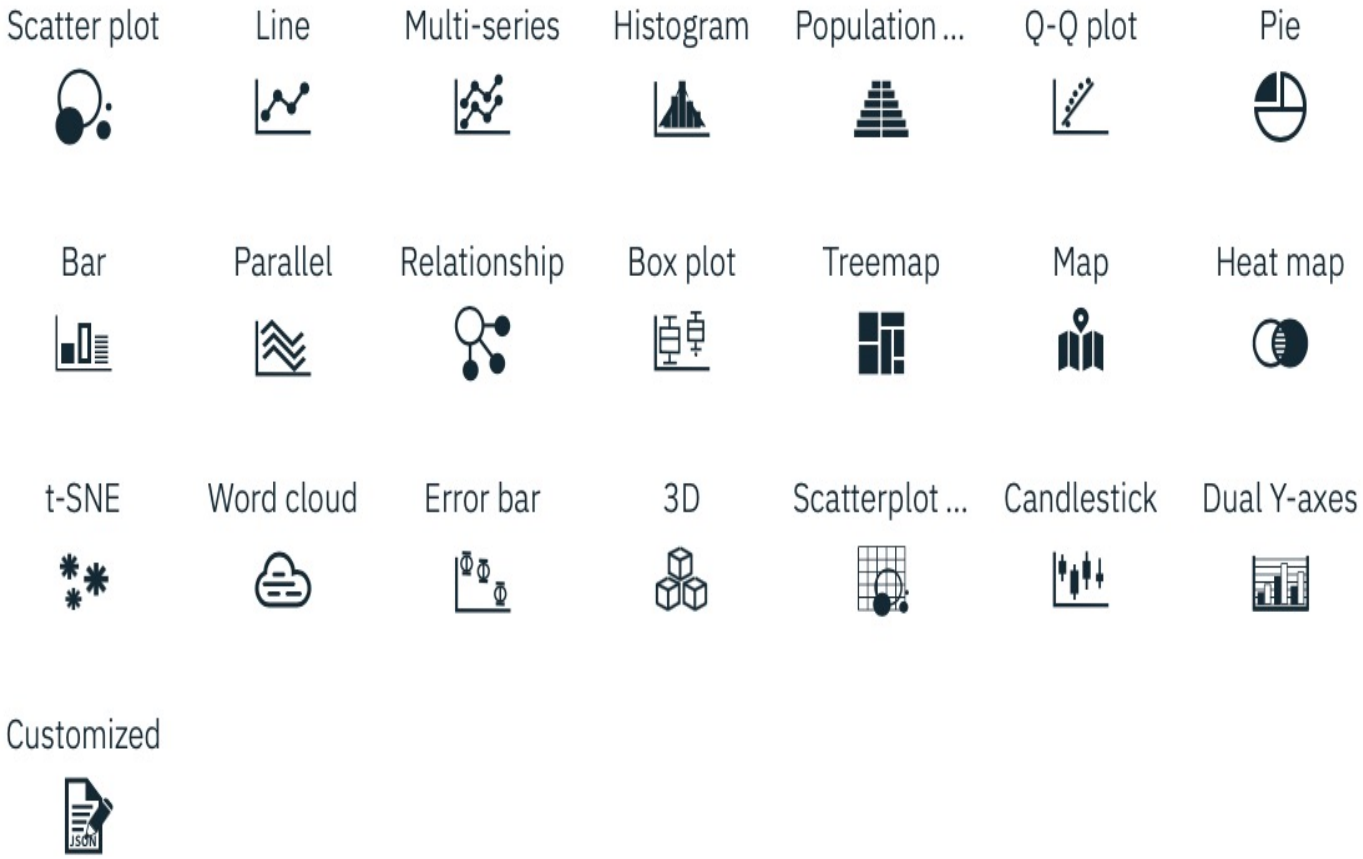
```python
import numpy as np
import pandas as pd
import uuid
#read the data
df=pd.read_csv('Users/Desktop/test.csv',sep=',')
#remove unwanted columns
del_column_list=['col_1','col_2']
for i in del_column_list:
    del df[i]

#Change the column data types
df['col_3']=df['col_3'].astype(np.int64)
#rename the column names for better redability and understanding the features
df.rename(index=str, columns={"col_3": "sales_data", "col_4": "accounts_data"},copy=True)
#check the number of empty cells in a columns and if it is more than a threshold
if (float(df['batch_enrolled'].isnull().sum())/len(df.index))*100 >10:
    df = df[df['batch_enrolled'].isnull() == False]

#mask the confidential data columns
for name in df['batch_enrolled'].unique():
    df.loc[df['batch_enrolled'] == name, 'masked_batch_enrolled'] = uuid.uuid4()
#remove/replace unnecessary characters from columns
df['sample_col']=[x.replace(' ','_') for x in df['sample_col']]
#apply conditions on two or more columns to create new columns
df['only_funded_amount']=df['loan_amount']==df['funded_amount']
#Combine two or more columns which can be represented as single feature for better understanding
# and readability
df['address']=[str(x)+","+str(y) for x,y in zip(df['zip_code'],df['addr_state'])]
#Split columns based on the separators(alphanumeric) or patterns or based on position
df[['First Name','Last Name']] = df.Name.str.split(",",expand=True)
#Apply Math functions on columns to derive new data/columns from the existing dataset/columns
df['loan_amnt_sqrt']=[np.sqrt(x) for x in df['loan_amnt']]
#Fill the missing values  with constant ,mean,min,max,median  for integer/decimal columns
df['tot_amount'].fillna(np.mean(df['tot_amount']),inplace=True)
#Join the two or more datasets based on the common key columns
df_right=pd.read_csv('Users/Desktop/test_v2.csv',sep=',')
result= pd.merge(df,df_right,how='left',on=['key1', 'key2'])
#Arrange the dataset  in required order based on columns [Ascending/Descending order]
result.sort(['key1', 'key2'], ascending=[1, 0],inplace=True)
```

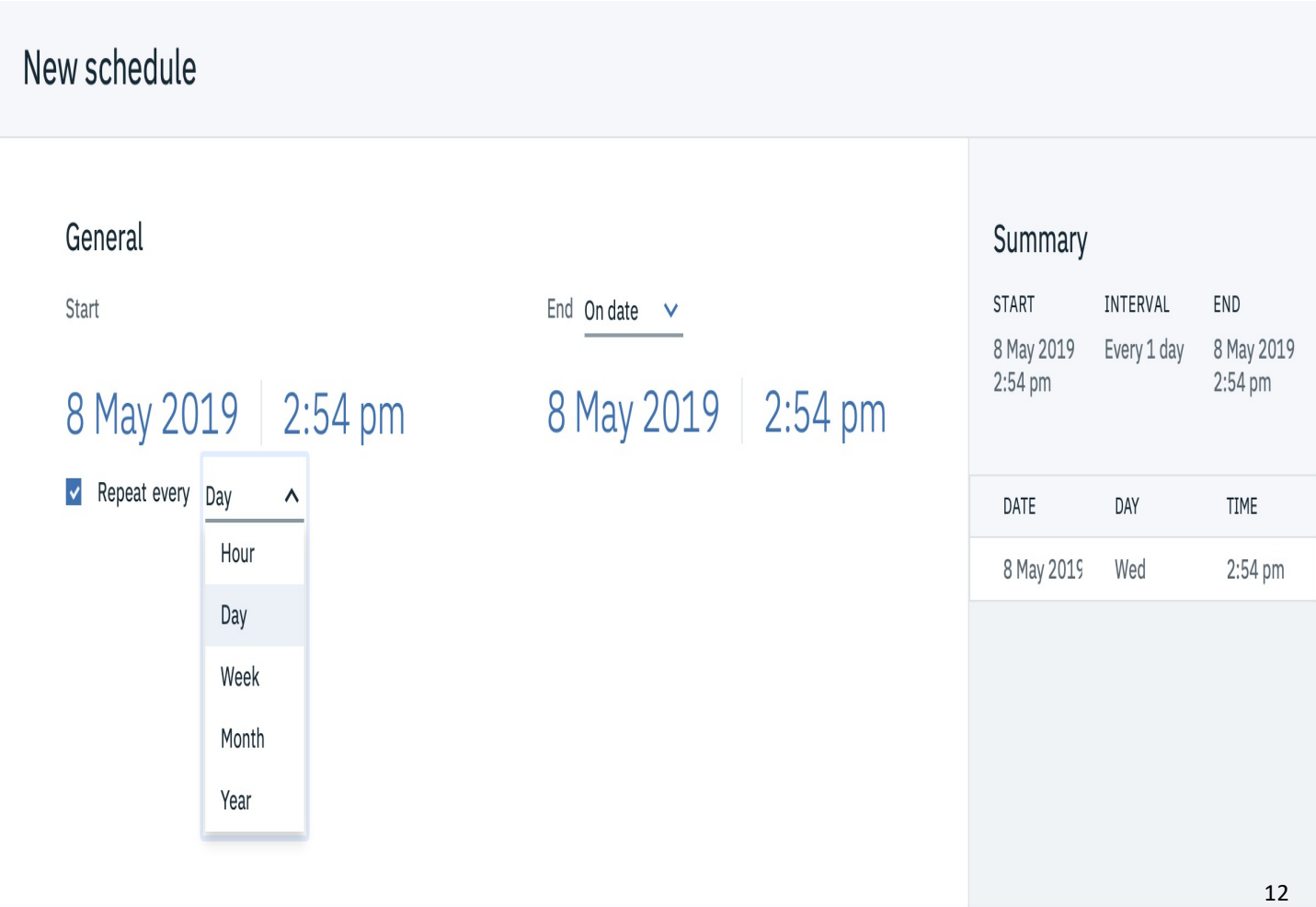
Visualizing the data

- Different types of visualization graphs are available in order to view ,analyze and understand the data better



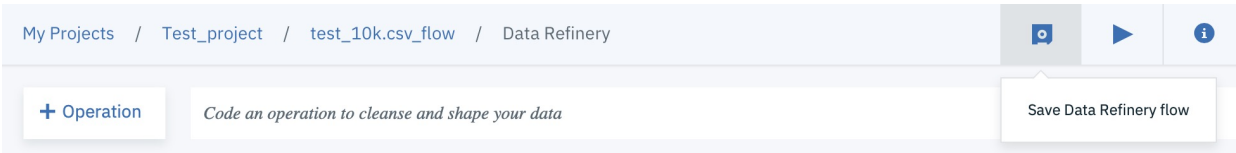
Scheduling Data Refinery Flow

- Schedule data refinery flow as jobs(recurring) on the datasets



Manage Data Refinery Flow

- Save the data refinery flow in Watson studio with a series of cleaning steps
- Add/Remove the refinery steps from the data refinery flow
- Snapshot view of Data refinery steps



Thank you

Naresh Olladapu

Developer Advocate

nolladap@in.ibm.com

+91-8132097165

IBM