**Extract**

**Finding Data – Data Source:**

*Reddit|Twitter |YouTube*

* Data was formatted in JSON. The following are the libraries we use to extract data from the above three resource’s API:
* YouTube API (YOUTUBE\_API\_VERSION = "v3")
  + Libraries (YOUTUBE\_API\_VERSION)
    - from apiclient.discovery import build
    - from apiclient.errors import HttpError
    - from oauth2client.tools import argparser
    - google-api-python-client
    - import pprint
  + Data (youtube.search().list)
    - Type of data - JSON
* Twitter API
  + Libraries
    - import twitter
      * Type of data – Dictionaries
    - import datetime as dt
* Reddit API
  + Libraries
    - import praw
      * Type of data – Dictionaries
    - import datetime as dt
* Libraries (Data Analysis)
  + import pandas as pd
  + import matplotlib.pyplot as plt
  + Import numpy as np

**Transform**

**Data Cleanup & Analysis**

We write code in Python with Jupiter in three different ipynb for three resources:

* RedditData.ipynb
* TwittData.ipynb
* Youtube.ipynb

In each of above ipynb files, we exact data and transform dataset into Pandas DataFrame which is the table we are going to use. The **final Data Frames** we produced from APIs of above three resources were all named as **new\_topics\_data\_df.**

**Reddit**:

DataFrame Name: new\_topics\_data\_df

Columns: [ ‘id’, ‘title’, ‘score’, ‘comms\_num’, ‘timestamp’, ‘source’]

**Twitter:**

DataFrame Name: new\_topics\_data\_df

Columns: ['screenName', 'tweet', 'retweet', 'favorite\_count', 'created', ‘source’]

**Youtube**:

DataFrame Name: new\_topics\_data\_df

Columns: [‘channelTitle’, ‘Title’, ‘viewCount’, ‘likeCount’, ‘publishAt’, ‘source’]

**Steps Taken**

1. Added ‘Source’ column to each of the three Data Frames named as new\_topics\_data\_df, the content for this column is respectively ‘Reddit’, ‘Twitter’, ‘Youtube’.
2. Transformed Datatype of column with Date and Time into pandas Datetime64ns data type, then strip of Time and keep Data in column, then convert Datetime64ns column into String type in order to load table into database
3. Converted Integer (Int32 or Int64) data type in Data Frames into String in order to load table into database
4. Replaced NaN content if existing with value None for all the three dataframes.
5. After all the above transform and cleaning job, the final tables are ready to be loaded to database, there names are still new\_topics\_data\_df.

**Load:**

* Final database: relational MYSQL, Schema name: crypto\_db
* Reason to select relational database is mature and there are many resources and easy to use.

**Tables**:

Table Name: new\_topics\_data\_df

Columns: [ ‘id’, ‘title’, ‘score’, ‘comms\_num’, ‘timestamp’, ‘source’]

Table Name: new\_topics\_data\_df

Columns: ['screenName', 'tweet', 'retweet', 'favorite\_count', 'created', ‘source’]

Table Name: new\_topics\_data\_df

Columns: [‘channelTitle’, ‘Title’, ‘viewCount’, ‘likeCount’, ‘publishAt’, ‘source’]

* Reason to select three above tables is we can combine the tables together for future data analytics use.

**Steps Summary**

We load three tables named as **new\_topics\_data\_df** into relational database MYSQL. The schema name is crypto\_db. The three tables were loaded into crpto\_db as three tables named as below:

1. bitcoin\_reddit : save **new\_topics\_data\_df** into csv file; import csv files into database crypto\_db named as table Bitcoin\_reddit
2. bitcoin\_twitter: use SQLAlchemy library in Python to load **new\_topics\_data\_df** into database crypto\_db named as table Bitcoin\_twitter.
3. bitcoin\_youtube: use SQLAlchemy library in Python to load **new\_topics\_data\_df** into database crypto\_db named as table Bitcoin\_youtube.