test_ingestion

October 20, 2022

Data Source: Flight Status Prediction | Kaggle

0.1 Task:

- Take any csv/text file of 2+ GB of your choice. — (You can do this assignment on Google colab)
- Read the file (Present approach of reading the file)
- Try different methods of file reading eg: Dask, Modin, Ray, pandas and present your findings in term of computational efficiency
- Perform basic validation on data columns : eg: remove special character , white spaces from the col name
- As you already know the schema hence create a YAML file and write the column name in YAML file. –define separator of read and write file, column name in YAML
- Validate number of columns and column name of ingested file with YAML.
- Write the file in pipe separated text file (|) in gz format.
- Create a summary of the file:

  Total number of rows,

  total number of columns

  file size

```
In [ ]: !pip install ray --user
```

```
In [ ]: !pip install modin
```

```
In [ ]: !pip install --upgrade pandas
```

```
In [1]: import os
        # import ray
        import time
        import numpy as np
        import pandas as pd
        # import modin.pandas as modin
        from dask import dataframe as dd
```

```
/home/ychen306/.local/lib/python3.7/site-packages/pandas/compat/_optional.py:138: UserWarning:
  warnings.warn(msg, UserWarning)
/home/ychen306/.local/lib/python3.7/site-packages/dask/dataframe/utils.py:15: FutureWarning: pa
  import pandas.util.testing as tm
```

```
In [2]: !ls
```

```
Combined_Flights_2019.csv  test_ingestion.ipynb
```

```
In [3]: %%writefile testutility.py
        import logging
        import os
        import subprocess
        import yaml
        import pandas as pd
        import datetime
        import gc
        import re


        ################
        # File Reading #
        ################

        def read_config_file(filepath):
            with open(filepath, 'r') as stream:
                try:
                    return yaml.safe_load(stream)
                except yaml.YAMLError as exc:
                    logging.error(exc)


        def replacer(string, char):
            pattern = char + '{2,}'
            string = re.sub(pattern, char, string)
            return string

        def col_header_val(df,table_config):
            '''
            replace whitespaces in the column
            and standardized column names
            '''
            # df.columns = df.columns.str.lower()
            df.columns = df.columns.str.replace('[^\w]','_',regex=True)
            df.columns = list(map(lambda x: x.strip('_'), list(df.columns)))
            df.columns = list(map(lambda x: replacer(x,'_'), list(df.columns)))
```

```
    # expected_col = list(map(lambda x: x.lower(),  table_config['columns']))
    expected_col = table_config['columns']
    expected_col.sort()
    # df.columns = list(map(lambda x: x.lower(), list(df.columns)))
    df = df.reindex(sorted(df.columns), axis=1)
    if len(df.columns) == len(expected_col) and list(expected_col)  == list(df.columns
        print("column name and column length validation passed")
        return 1
    else:
        print("column name and column length validation failed")
        mismatched_columns_file = list(set(df.columns).difference(expected_col))
        print("Following File columns are not in the YAML file",mismatched_columns_fil
        missing_YAML_file = list(set(expected_col).difference(df.columns))
        print("Following YAML columns are not in the file uploaded",missing_YAML_file)
        logging.info(f'df columns: {df.columns}')
        logging.info(f'expected columns: {expected_col}')
        return 0
```

```
Writing testutility.py
```

0.1.1 Write YAML file

```
In [4]: %%writefile file.yaml
        file_type: csv
        dataset_name: testfile
        file_name: Combined_Flights_2019
        table_name: edsurv
        inbound_delimiter: ","
        outbound_delimiter: "|"
        skip_leading_rows: 1
        columns:
            - FlightDate
            - Airline
            - Origin
            - Dest
            - Cancelled
            - Diverted
            - CRSDepTime
            - DepTime
            - DepDelayMinutes
            - DepDelay
            - ArrTime
            - ArrDelayMinutes
            - AirTime
            - CRSElapsedTime
            - ActualElapsedTime
            - Distance
            - Year
            - Quarter
            - Month
            - DayofMonth
            - DayOfWeek
            - Marketing_Airline_Network
            - Operated_or_Branded_Code_Share_Partners
            - DOT_ID_Marketing_Airline
            - IATA_Code_Marketing_Airline
            - Flight_Number_Marketing_Airline
            - Operating_Airline
            - DOT_ID_Operating_Airline
            - IATA_Code_Operating_Airline
            - Tail_Number
            - Flight_Number_Operating_Airline
            - OriginAirportID
            - OriginAirportSeqID
            - OriginCityMarketID
            - OriginCityName
            - OriginState
            - OriginStateFips
            - OriginStateName
            - OriginWac
            - DestAirportID
            - DestAirportSeqID
            - DestCityMarketID
            - DestCityName
            - DestState
            - DestStateFips
            - DestStateName
            - DestWac
            - DepDel15
            - DepartureDelayGroups
            - DepTimeBlk
            - TaxiOut
            - WheelsOff
            - WheelsOn
            - TaxiIn
            - CRSArrTime
            - ArrDelay
            - ArrDel15
            - ArrivalDelayGroups
            - ArrTimeBlk
            - DistanceGroup
            - DivAirportLandings
```

```
Writing file.yaml
```

```
In [2]: # Read config file
        import testutility as util
        config_data = util.read_config_file("file.yaml")
```

```
In [6]: config_data['inbound_delimiter']
```

```
Out[6]: ','
```

```
In [7]: #inspecting data of config file
        config_data
```

```
Out[7]: {'file_type': 'csv',
         'dataset_name': 'testfile',
         'file_name': 'Combined_Flights_2019',
         'table_name': 'edsurv',
         'inbound_delimiter': ',',
         'outbound_delimiter': '|',
         'skip_leading_rows': 1,
         'columns': ['FlightDate',
          'Airline',
          'Origin',
          'Dest',
          'Cancelled',
          'Diverted',
          'CRSDepTime',
          'DepTime',
          'DepDelayMinutes',
          'DepDelay',
          'ArrTime',
          'ArrDelayMinutes',
          'AirTime',
          'CRSElapsedTime',
          'ActualElapsedTime',
          'Distance',
          'Year',
          'Quarter',
          'Month',
          'DayofMonth',
          'DayOfWeek',
          'Marketing_Airline_Network',
          'Operated_or_Branded_Code_Share_Partners',
          'DOT_ID_Marketing_Airline',
          'IATA_Code_Marketing_Airline',
          'Flight_Number_Marketing_Airline',
          'Operating_Airline',
          'DOT_ID_Operating_Airline',
          'IATA_Code_Operating_Airline',
          'Tail_Number',
          'Flight_Number_Operating_Airline',
          'OriginAirportID',
          'OriginAirportSeqID',
          'OriginCityMarketID',
          'OriginCityName',
          'OriginState',
          'OriginStateFips',
          'OriginStateName',
          'OriginWac',
          'DestAirportID',
          'DestAirportSeqID',
          'DestCityMarketID',
          'DestCityName',
          'DestState',
          'DestStateFips',
          'DestStateName',
          'DestWac',
          'DepDel15',
          'DepartureDelayGroups',
          'DepTimeBlk',
          'TaxiOut',
          'WheelsOff',
          'WheelsOn',
          'TaxiIn',
          'CRSArrTime',
          'ArrDelay',
          'ArrDel15',
          'ArrivalDelayGroups',
          'ArrTimeBlk',
          'DistanceGroup',
          'DivAirportLandings']}
```

```
In [3]: # read the file using config file
        file_type = config_data['file_type']
        source_file = "./" + config_data['file_name'] + f'.{file_type}'
        print("",source_file)
```

```
 ./Combined_Flights_2019.csv
```

0.1.2 Pandas

```
In [9]: # Normal reading process of the file
        import pandas as pd
        filename="Combined_Flights_2019.csv"
        start = time.time()
        df_sample = pd.read_csv(filename,delimiter=',')
        end = time.time()
        print("Read 2.82GB file using pandas: ",(end-start),"sec")
        print(f"\n{df_sample.shape}")
        df_sample.head()
```

```
Read 2.82GB file using pandas:  65.52308750152588 sec

(8091684, 61)
```

```
Out[9]:    FlightDate    Airline Origin Dest  Cancelled  Diverted  CRSDepTime  \
        0  2019-04-01  Envoy Air    LIT  ORD      False     False        1212
        1  2019-04-02  Envoy Air    LIT  ORD      False     False        1212
        2  2019-04-03  Envoy Air    LIT  ORD      False     False        1212
        3  2019-04-04  Envoy Air    LIT  ORD      False     False        1212
        4  2019-04-05  Envoy Air    LIT  ORD      False     False        1212

           DepTime  DepDelayMinutes  DepDelay  ...  WheelsOff  WheelsOn  TaxiIn  \
        0   1209.0              0.0      -3.0  ...     1219.0    1342.0     8.0
        1   1200.0              0.0     -12.0  ...     1210.0    1339.0     9.0
        2   1203.0              0.0      -9.0  ...     1214.0    1336.0     6.0
        3   1435.0            143.0     143.0  ...     1452.0    1615.0     6.0
        4   1216.0              4.0       4.0  ...     1234.0    1357.0    13.0

           CRSArrTime  ArrDelay  ArrDel15  ArrivalDelayGroups  ArrTimeBlk  \
        0        1405     -15.0       0.0                -1.0   1400-1459
        1        1405     -17.0       0.0                -2.0   1400-1459
        2        1405     -23.0       0.0                -2.0   1400-1459
        3        1405     136.0       1.0                 9.0   1400-1459
        4        1405       5.0       0.0                 0.0   1400-1459

           DistanceGroup  DivAirportLandings
        0              3                   0
        1              3                   0
        2              3                   0
        3              3                   0
        4              3                   0

        [5 rows x 61 columns]
```

```
In [8]: start = time.time()
        df = pd.read_csv(source_file,delimiter=config_data['inbound_delimiter'])
        end = time.time()
        print("Read 2.82GB file using pandas: ",(end-start),"sec")
```

```
Read 2.82GB file using pandas:  76.70126724243164 sec
```

0.1.3 Dask

```
In [ ]: start = time.time()
        df = dd.read_csv(source_file) #,delimiter=config_data['inbound_delimiter'],header=1
        end = time.time()
        print("Read 2.82GB file using dask: ",(end-start),"sec")
```

0.1.4 Ray

```
In [ ]: ray.shutdown()
        ray.init()
        start = time.time()
        df = pd.read_csv(source_file,config_data['inbound_delimiter'])
        end = time.time()
        print("Read 2.82GB file using ray: ",(end-start),"sec")
```

```
2022-10-21 02:25:00,814        INFO worker.py:1518 -- Started a local Ray instance.
```

0.1.5 Validate

```
In [10]: #validate the header of the file
         util.col_header_val(df,config_data)
```

```
column name and column length validation passed
```

```
Out[10]: 1
```

```
In [11]: print("columns of files are:" ,df.columns)
         print("columns of YAML are:" ,config_data['columns'])
```

```
columns of files are: Index(['FlightDate', 'Airline', 'Origin', 'Dest', 'Cancelled', 'Diverted
       'CRSDepTime', 'DepTime', 'DepDelayMinutes', 'DepDelay', 'ArrTime',
       'ArrDelayMinutes', 'AirTime', 'CRSElapsedTime', 'ActualElapsedTime',
       'Distance', 'Year', 'Quarter', 'Month', 'DayofMonth', 'DayOfWeek',
       'Marketing_Airline_Network', 'Operated_or_Branded_Code_Share_Partners',
       'DOT_ID_Marketing_Airline', 'IATA_Code_Marketing_Airline',
       'Flight_Number_Marketing_Airline', 'Operating_Airline',
       'DOT_ID_Operating_Airline', 'IATA_Code_Operating_Airline',
       'Tail_Number', 'Flight_Number_Operating_Airline', 'OriginAirportID',
       'OriginAirportSeqID', 'OriginCityMarketID', 'OriginCityName',
       'OriginState', 'OriginStateFips', 'OriginStateName', 'OriginWac',
       'DestAirportID', 'DestAirportSeqID', 'DestCityMarketID', 'DestCityName',
       'DestState', 'DestStateFips', 'DestStateName', 'DestWac', 'DepDel15',
       'DepartureDelayGroups', 'DepTimeBlk', 'TaxiOut', 'WheelsOff',
       'WheelsOn', 'TaxiIn', 'CRSArrTime', 'ArrDelay', 'ArrDel15',
       'ArrivalDelayGroups', 'ArrTimeBlk', 'DistanceGroup',
       'DivAirportLandings'],
      dtype='object')
columns of YAML are: ['ActualElapsedTime', 'AirTime', 'Airline', 'ArrDel15', 'ArrDelay', 'ArrDe
```

```
In [12]: if util.col_header_val(df,config_data)==0:
             print("validation failed")
             # write code to reject the file
         else:
             print("col validation passed")
             # write the code to perform further action
             # in the pipleine
```

```
column name and column length validation passed
col validation passed
```

0.1.6 Summary of file

```
In [13]: print(df.shape)
```

```
(8091684, 61)
```

```
In [16]: file_size = os.path.getsize(source_file)/1024/1024/1024
         print("File Size is :", file_size, "GBs")
```

```
File Size is : 2.6277403542771935 GBs
```

```
In [17]: df.describe
```

```
Out[17]: <bound method NDFrame.describe of          FlightDate                   Airline Origi
         0        2019-04-01                 Envoy Air    LIT  ORD      False
         1        2019-04-02                 Envoy Air    LIT  ORD      False
         2        2019-04-03                 Envoy Air    LIT  ORD      False
         3        2019-04-04                 Envoy Air    LIT  ORD      False
         4        2019-04-05                 Envoy Air    LIT  ORD      False
         ...             ...                       ...    ...  ...        ...
         8091679  2019-01-23  ExpressJet Airlines Inc.    MEM  IAH      False
         8091680  2019-01-24  ExpressJet Airlines Inc.    MEM  IAH      False
         8091681  2019-01-25  ExpressJet Airlines Inc.    MEM  IAH      False
         8091682  2019-01-26  ExpressJet Airlines Inc.    MEM  IAH      False
         8091683  2019-01-28  ExpressJet Airlines Inc.    MEM  IAH      False

                  Diverted  CRSDepTime  DepTime  DepDelayMinutes  DepDelay  ...  \
         0           False        1212   1209.0              0.0      -3.0  ...
         1           False        1212   1200.0              0.0     -12.0  ...
         2           False        1212   1203.0              0.0      -9.0  ...
         3           False        1212   1435.0            143.0     143.0  ...
         4           False        1212   1216.0              4.0       4.0  ...
         ...           ...         ...      ...              ...       ...  ...
         8091679     False         640    634.0              0.0      -6.0  ...
         8091680     False         640    631.0              0.0      -9.0  ...
```

```
8091681     False         640    632.0              0.0      -8.0  ...
8091682     False         640    630.0              0.0     -10.0  ...
8091683     False         640    632.0              0.0      -8.0  ...

         WheelsOff  WheelsOn  TaxiIn  CRSArrTime  ArrDelay  ArrDel15  \
0           1219.0    1342.0     8.0        1405     -15.0       0.0  
1           1210.0    1339.0     9.0        1405     -17.0       0.0  
2           1214.0    1336.0     6.0        1405     -23.0       0.0  
3           1452.0    1615.0     6.0        1405     136.0       1.0  
4           1234.0    1357.0    13.0        1405       5.0       0.0  
...            ...       ...     ...         ...       ...       ...  
8091679      710.0     847.0     6.0         840      13.0       0.0  
8091680      657.0     820.0    10.0         840     -10.0       0.0  
8091681      654.0     822.0     6.0         840     -12.0       0.0  
8091682      656.0     825.0     6.0         840      -9.0       0.0  
8091683      652.0     813.0    12.0         840     -15.0       0.0  

         ArrivalDelayGroups  ArrTimeBlk  DistanceGroup  DivAirportLandings
0                      -1.0   1400-1459              3                   0
1                      -2.0   1400-1459              3                   0
2                      -2.0   1400-1459              3                   0
3                       9.0   1400-1459              3                   0
4                       0.0   1400-1459              3                   0
...                     ...         ...            ...                 ...
8091679                 0.0   0800-0859              2                   0
8091680                -1.0   0800-0859              2                   0
8091681                -1.0   0800-0859              2                   0
8091682                -1.0   0800-0859              2                   0
8091683                -1.0   0800-0859              2                   0

[8091684 rows x 61 columns]>
```