**Module Description**

**config**

config basically reading json file which contains configuration/properties details like kafka,database,elastic search etc.

Property file

Properties.json

Package: Gson

Here **gson** package using to read json file.

**logger**

logger function logged the log/message to elastic search that visible to Kibana.

Package: scalaj-http

Scalaj-http use for post to elastic search

**utils**

**DFOperations:** It contains basic operation related to dataframe like create customize schema,release dataframe and read a fine and convert to dataframe.

**FileOperations:** FileOperations basically use for common purpose like check is file empty,file rename, is file exist or not,get current time stamp.

Package:MD5 to get the checksum of columns and file contains.

**sourcestoragesystem**

**ConsumerNProducer:** This function is used to consume and produce using kafka and spark streaming. To consume log from any messaging queue system like Kafka, RabbitMQ.

**brokerUri**: "kafka-0-broker.dev-xxx-kafka02.cdt.dcos:9625,kafka-1-broker.dev-xxx-kafka02.cdt.dcos:9625, kafka-2-broker.dev-xxx-kafka02.cdt.dcos:9625,kafka-3-broker.dev-xxx-kafka02.cdt.dcos:9625, kafka-4-broker.dev-xx-kafka02.cdt.dcos:9625"

**topic**: "devl-xxx-catest-evt"

**ConsumeThroughApi:** Through post call to execute the application and load data into data lake it supports both file and log, it is implemented through finchrest api.

curl -H **"**Content-Type: application/json**"** -X POST http://localhost:8000/msg-consumer-api -d '{

“filename”:”yellow\_tripdata\_2018-02.csv”,

“folder”: “/input/path/location/”

“unique id” :”12w34fg4567asds3fgh0paszxq1234”

}'

**PullFromDB:** This basically read data from database.

Packages:Finch, KafkaConsumer, KafkaProducer,jdbc

**cryptographyalgo**

This module is use to decrypt and encrypt the file using pgp.

Package: BCPGPEncryptor and BCPGPDecryptor

**deduplication**

Function is use to identity unique and duplicate record.

History table:

CREATE TABLE historytabyellotaxi (md5\_32bitCol String, filename STRING,load\_date TIMESTAMP)

Primary key(md5\_32bitCol)

COMMENT 'DCP audit details'

ROW FORMAT DELIMITED

FIELDS TERMINATED BY '\t'

LINES TERMINATED BY '\n'

STORED AS PARQUET;

Duplicate table:

CREATE TABLE duplicatetabyellotaxi (md5\_32bitCol String, filename STRING,load\_date TIMESTAMP)

COMMENT 'DCP audit details'

ROW FORMAT DELIMITED

FIELDS TERMINATED BY '\t'

LINES TERMINATED BY '\n'

STORED AS PARQUET;

**transformationprocess**

Find the peak time of pickup/drops on a daily basis. Consider peak time

to be interval of 1 hour.

2. Find the average total amount per payment type.

3. Find anomalies in the average trip distance per Vendor Id.

4. If the trip distance for a vendor Id increases above the mean value mark

the trip as anomalous. Assume period of 3 days for the trips.

5. Find the average trip distance per RateCodeId.

6. Rank the Vendor Id/per day based on the fare amount.

7. Categorise the trips based on its total amount to high, medium and low.

8. Create three consecutive ranges, label them high, medium and low in

decreasing order. Set the range limits by yourself.

For example:- If the three ranges are 30+, 16-29, 15-1, and the total

amount per is 17, then mark it as medium.