Introduction of AOTAS modules and source code

- Module Overview

Module Name	Description
Download	Download data files To download dataset data files from source and save to **‘raw’** folder, using the original format, **‘csv’.**
ETL Module	Read ‘raw’ data, perform cleansing transformation and save to ‘processed’ location
Integration Module	Read ‘processed’ data, conduct data warehouse modeling, create dimension/fact tables and save to ‘integration’ location
Utility Module	Provide common utilities that will be used throughout the pipeline process

- Module details

Module	Scripts
Top Entry point	.AOTA_main.py. ● Entry point of the pipeline ● It will run the main file of download module, ETL module , and integration module to complete their duties.
Download Module	(see table below)

Description	To download dataset data files from source and save to **‘raw’** folder, using the original format, **‘csv’.**
Entry point file	**download_manager.py**
Major Files	**downloader.p**y– to provide tools/methods to download flight time dataset Two major methods

Method	Description

		download_small_dataset: To download following small datasets: airport, carrier, plane **download_dataset_flight**: To download flight time dataset of multiple years (specified in array ‘`flight_dfiles`’)
ETL Module	Description	Provide following functionalities: 1. Read data from **‘raw’** files, 2. perform cleansing/transformation 3. save to **‘processed**’ folder, using ‘**parquet’**
	EntryPoint File	**ETL_Manager.py**
	Major Files	**ETL_Airport.py**: to process airport data **ETL_Carrier.py:** to process carrier data **ETL_Plane.py**: to process plane data **ETL_Flight.py**:to process flight data
Integration Module	Description	Provide following functionalities : 1. Read data from **‘process’** files, 2. Perform DW modeling, 3. Create dimension table and fact table datasets 4. Save to **‘integration**’ folder, using ‘**parquet’**
	EntryPoint File	**intgr_Manager.py**
	Major Files	**1.Intgr_stg_table.py** (main method: **`cr_stg_flight`**) Read ‘processed’ flight data and create a staging table dataset, **stg_flight**, which works as a temp table for downstream processes. It will be saved to “integrated” folder, usin “parquet” **2.Intgr_dim_table.py** Read data from “processed’ folder and use ‘STG_FLIGHT’ data created in step 1, create following dimension table datasets and save to “integration”folder, using “parquet”. **dim_plane** **dim_origin_dest** **dim_date** **dim_carrier**

		3.Intgr_fact_table.py(main method: **cr_fact_flight**) 1) Read from **stg_flight** (created in **step 1**) and **dim_* datasets** (Created in **step 2**), 2) Create the flight fact table dataset **‘fact_flight’** 3) Save to “integration”folder, using “parquet”.
Utility module	Description	Provide different types of utilities to help the whole process.
	Major files	**ut_store.py:** work as a configuration file, which stores various parameters (file name,process name,etc). **ut_base.py**: provide some most basic utilities that will used by all modules **ut_log.py**: provide logging related functionalities that will be used by Download, ETL and interaction modules **Ut_spark.py:** Provided utilities that are related to Spark, such as, create a spark session, read from/write to files, etc **Ut_pipeline.py:** Provided utilities that are related to pipeline processing, such as log metrics, read data from ‘raw’, or save to ‘processed’, etc. They will be mainly used by ETL and Integration modules.