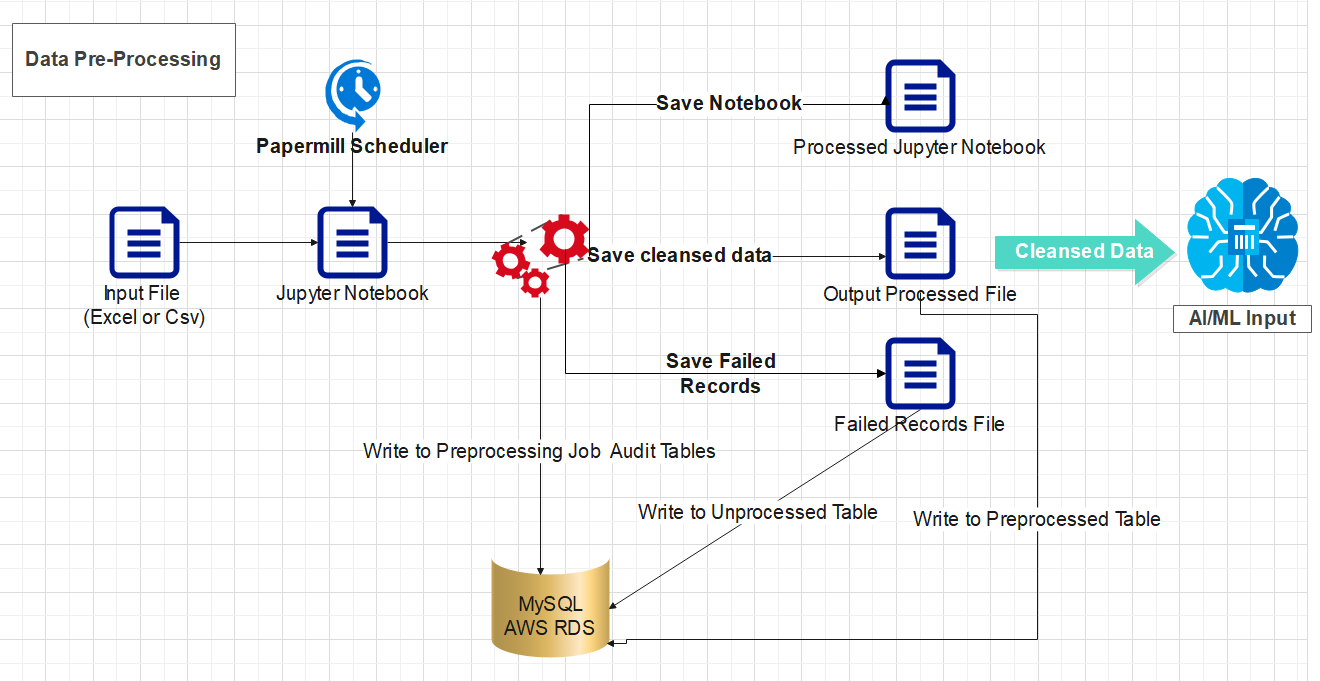
**Data Pre-processing Architecture:**



**Technologies Used**:

* Python 3.9
* Papermill
* Pandas
* MySQL Connector

**Data Pre-processing Strategies and Implementations:**

In order to convert our data from the source into a clean data set, it was checked for missing values, noisy data. As part of dealing with missing values, we removed all the records having null values and we did so using the following methods:

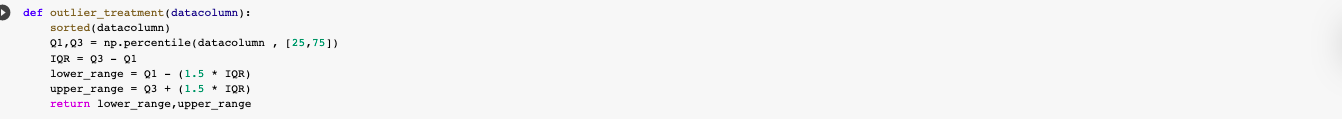
1. **Handling Missing Data by Redirecting records**

* To handle missing data, each column is treated separately. The rule of thumb followed is: If a column has more than 30% data missing, then this treatment cannot be done and that particular column must be rejected.
* We utilized the MODE value to impute the missing values for categorical variables as evident in the image below.

Graphical user interface, text, application

Description automatically generated

1. **Outlier Treatment**:

In addition to missing data, it is important to also treat outliers that interfere with data processing. We use the following function to solve these four specific issues:

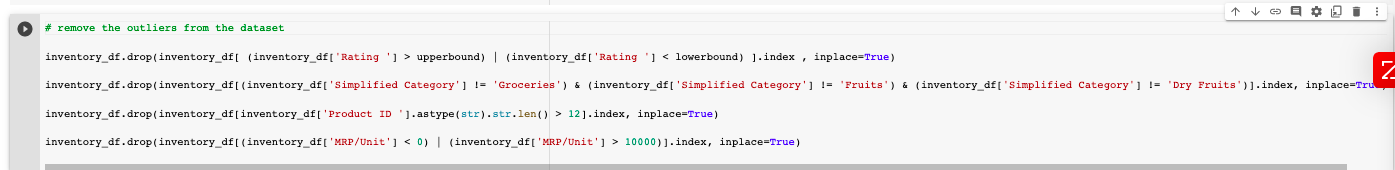
* **Restricting to Selected Category Products**:
  + To start off, we clean values that fall outside our pre-selected simplified categories: ‘Groceries’, ‘Fruits’, ‘Dry Fruits’ as below:
* **Handling Invalid Product ids**:
  + Graphical user interface, text, application, email

    Description automatically generatedSince unique ids allow for easy identification, product ids require a standardised number. To do this, we employed the following code:
* **Handling Invalid Prices**:
  + Graphical user interface, application

    Description automatically generatedSince we are looking at relatively economical products that are commonly sold, it is safe to place a pricing limit to 10,000 and remove any products that are priced above that in the following way:
* **Handling Rating Outliers**:
  + Graphical user interface, application

    Description automatically generatedA major factor that pushes our recommendation model is product ratings. Our dataset only includes those values whose ratings fall between 0 and 5.

Taking all of these into considerations, these models below conclude our data cleaning process:



**Data reprocessing Scheduling**:

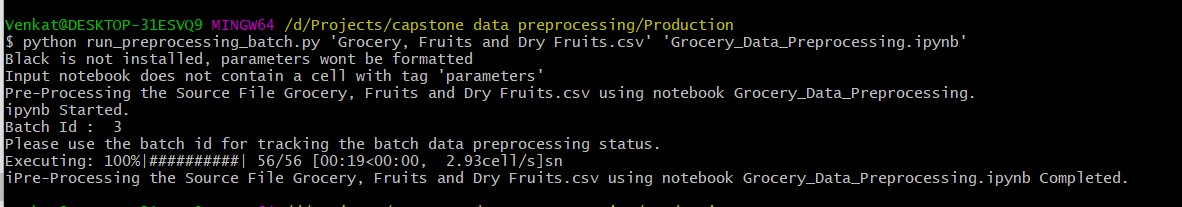
Papermill is a tool that allows us to parameterize and execute notebooks. It transforms your Jupyter notebook on a data workflow tool, by executing each cell sequentially, without having to open JupyterLab (or Notebook).

Each version of notebook is maintained in the local folders/ AWS S3 for source code version history.

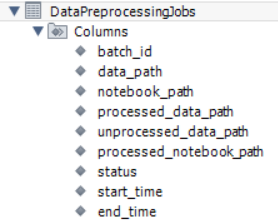
The notebooks are parameterized with following parameters and run as a data pre-processing job.

1. Notebook path (Provided by Client)
2. Source csv data file path (Provided by Client)
3. Output Notebook path
4. Output Pre-processed file path
5. Output Unprocessed file path

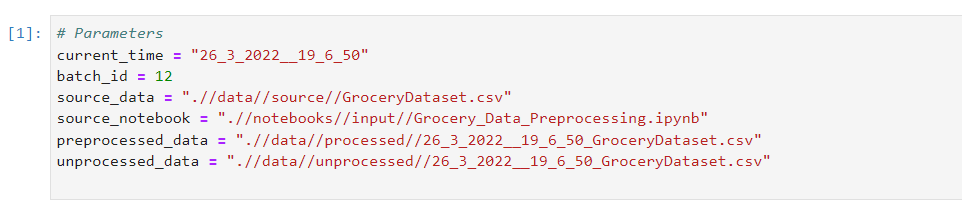
**Pre-processing Execution**:



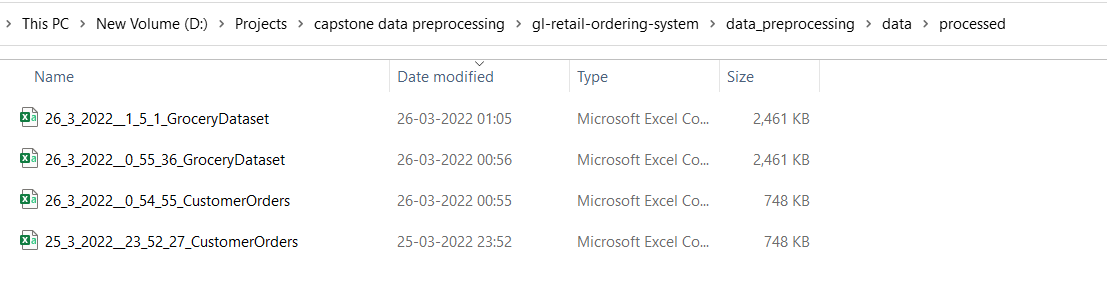
The job audit is maintained in the backend MySQL/ AWS RDS Database.



\*The Output Notebook has a cell with the exact parameter values used to run the pre-processing which could be used while troubleshooting or post-mortem analysis e.g.



**Data Pre-processing Data Versioning:**



**Data Pre-processing Notebook Versioning:**

