LOAN DEFAULT DATA PIPELINE

DATA

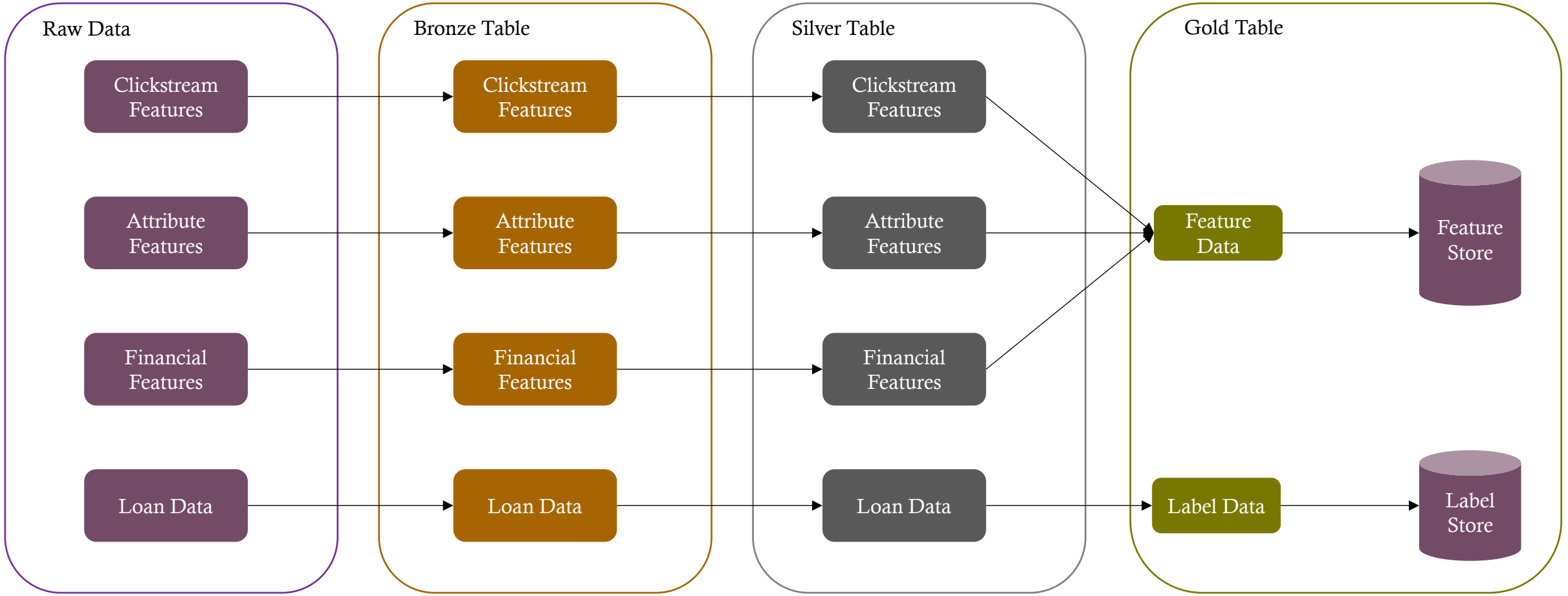
Inputs

- Clickstream Data
  - App user data
- Attribute Data
  - Job and Age Data
- Financial Data
  - Number and Types of Loans
  - Salary and Investment Data
  - Credit Score and Payment Behaviours
- Loan Data
  - Payments schedule and Amounts overdue
  - Instalments Number and amount

Outputs

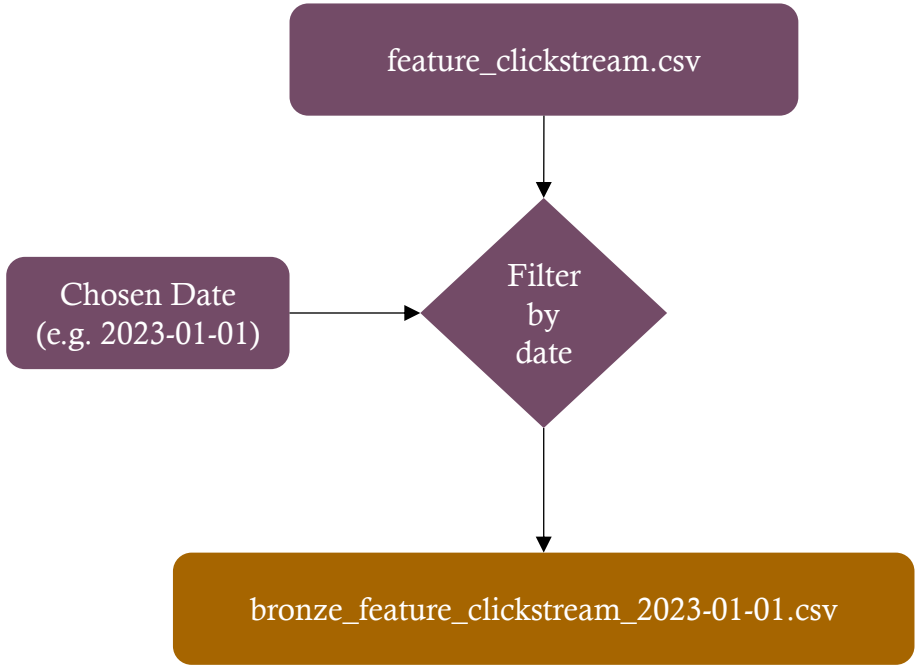
- Feature Store Containing all necessary features
- Label Store

DATA PIPELINE



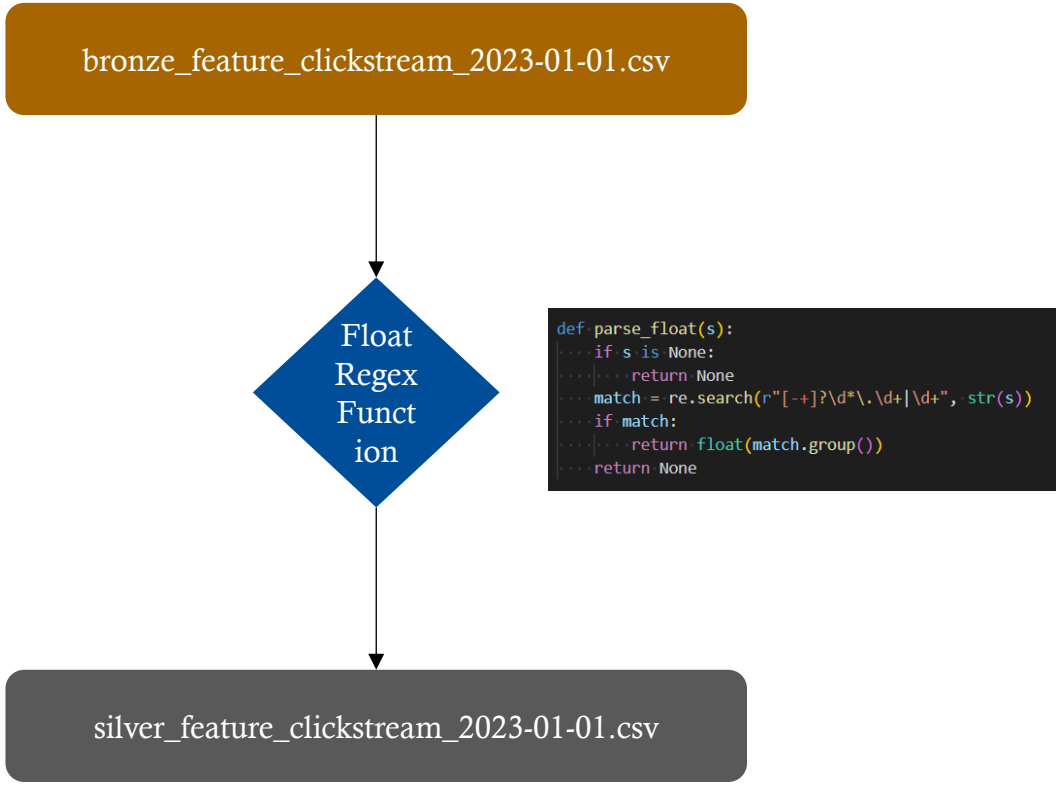
RAW DATA TO BRONZE TRANSFORMATION(ALL)

For bronze table, all data is filtered by snapshot date, where the first day of each month is chosen and the raw data is split by date



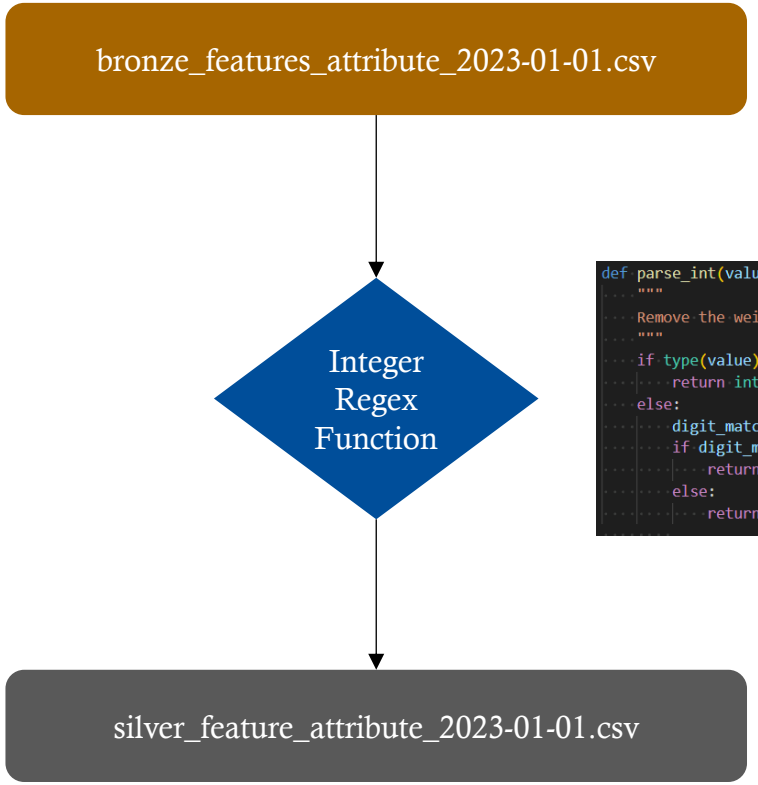
BRONZE TO SILVER TRANSFORMATION (CLICKSTREAM)

- For clickstream data, all values are seemingly floats
- To ensure that all values in the data are floats, a regex function was run on all the feature columns to ensure the output was a float



BRONZE TO SILVER TRANSFORMATION (ATTRIBUTES)

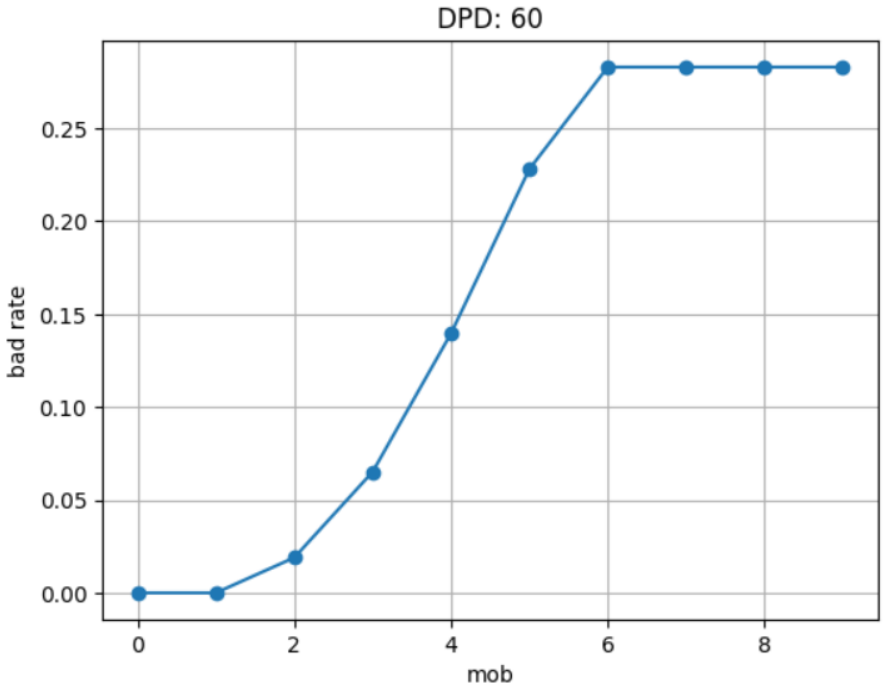
- For attribute data, some people do not have jobs listed, so value was changed to “Unknown”
- The age data is also unclean, so we needed to parse the integer data using regex as well




```python
def parse_int(value : str):
    """
    Remove the weird values from number of loan
    """
    if type(value) == int:
        return int(value)
    else:
        digit_match = re.search(r'\d+', value)
        if digit_match:
            return int(digit_match.group())
        else:
            return None
```

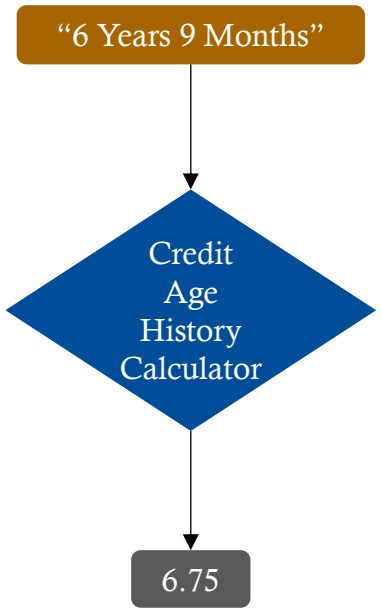
BRONZE TO SILVER TRANSFORMATION (LOAN DATA)

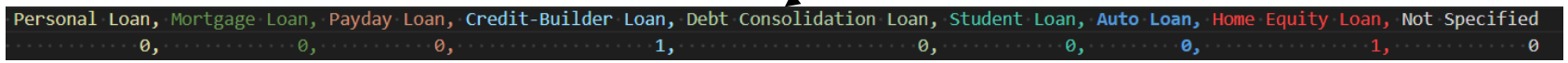
- Based on Singapore system as example, I chose a date past due (DPD) of 60 based on their standards, and 7 mob was used
  - https://www.mas.gov.sg/regulation/explainers/ongoing-credit-checks-and-requirements/borrowers-who-are-60-days-past-due
- Upon inspection of the dataset, there does not seem to be any anomalous data, so cleaning was not done



BRONZE TO SILVER TRANSFORMATION (FINANCIALS)

- Financials had very dirty integer and float values, which were fixed with the previous int and float parsers
- Numerical variables that were not represented numbers we parsed into floats
- Recounted the number of loans based on “Type_of_loan”
- Split type of loan into columns which counted the number of loans an individual had





SILVER TO GOLD TRANSFORMATION

Label Table

- Used the 7th installment and checked whether any installment was passed 60 days, was deemed as a default
- Made a new label called 60dpd_7mob to represent this

Feature Table

- Joined all data using Customer_ID as the key
- Join order was Clickstream -> attributes -> financials in order from cleanest to dirtiest, using inner join
- We set the date of the snapshot of data as the latest date between the 3 features

THINGS TO NOTE

- Data poisoning may happen since the date recorded of the features maybe be later than the loan defaulting parameter, so remember to compare the snapshot date of features and snapshot date of the label, making sure that the label date is later than the feature date