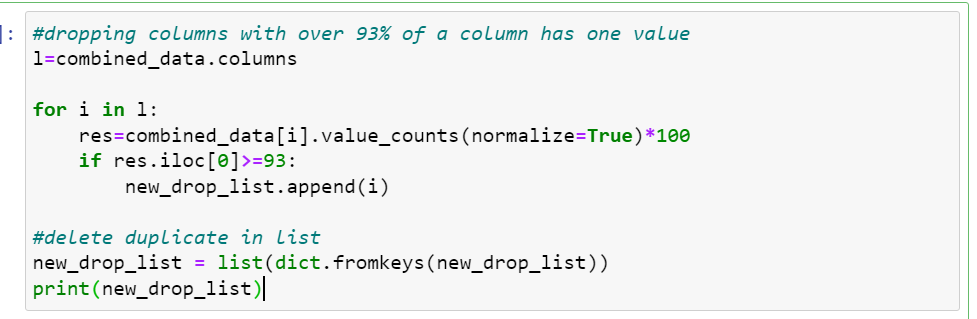
1. **Describe**
2. **Combine, Clean and Prepare**
3. **Clean and Prepare**
4. **Dropping Columns**

* Delete Columns That Contain Over 93% On a Single Value:

****

****

We have identified

[

'FLAG\_DOCUMENT\_2', 'FLAG\_DOCUMENT\_10',

'FLAG\_DOCUMENT\_12', 'FLAG\_DOCUMENT\_13',

'FLAG\_DOCUMENT\_14', 'FLAG\_DOCUMENT\_15',

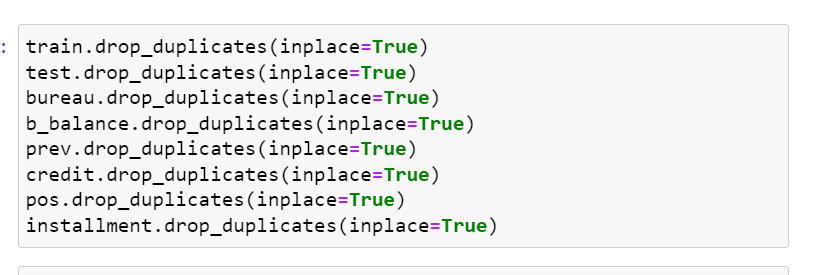
'FLAG\_DOCUMENT\_16', 'FLAG\_DOCUMENT\_17',

'FLAG\_DOCUMENT\_19', 'FLAG\_DOCUMENT\_20',

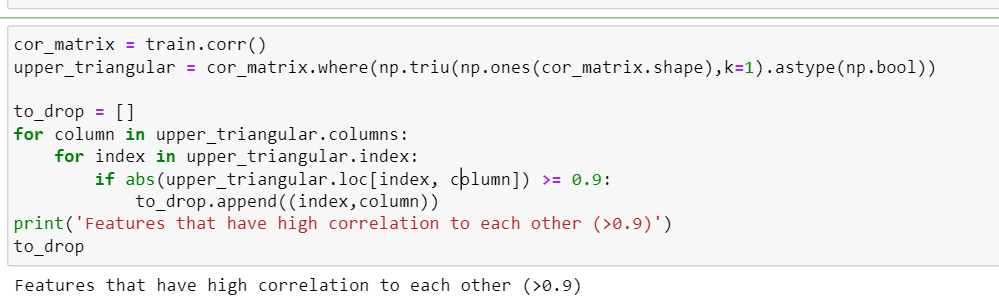
'FLAG\_DOCUMENT\_21', 'FLAG\_MOBIL', 'FLAG\_CONT\_MOBILE', 'REG\_REGION\_NOT\_LIVE\_REGION', 'REG\_REGION\_NOT\_WORK\_REGION', 'LIVE\_REGION\_NOT\_WORK\_REGION', 'HOUSETYPE\_MODE', 'EMERGENCYSTATE\_MODE', 'FLAG\_DOCUMENT\_4', 'FLAG\_DOCUMENT\_5', 'FLAG\_DOCUMENT\_7', 'FLAG\_DOCUMENT\_9', 'FLAG\_DOCUMENT\_11', 'FLAG\_DOCUMENT\_18', 'AMT\_REQ\_CREDIT\_BUREAU\_HOUR', 'AMT\_REQ\_CREDIT\_BUREAU\_DAY', 'AMT\_REQ\_CREDIT\_BUREAU\_WEEK', 'CREDIT\_DAY\_OVERDUE', 'CNT\_CREDIT\_PROLONG', 'AMT\_CREDIT\_SUM\_OVERDUE', 'NFLAG\_LAST\_APPL\_IN\_DAY'

] to be fitting the issue.

* Delete Rows that Contain Duplicate Data



* Hand picking from column pairs with over 0.9 correlation



Application\_train and test: [

'FLAG\_DOCUMENT\_2', 'FLAG\_DOCUMENT\_10', 'FLAG\_DOCUMENT\_12', 'FLAG\_DOCUMENT\_13', 'FLAG\_DOCUMENT\_14', 'FLAG\_DOCUMENT\_15', 'FLAG\_DOCUMENT\_16', 'FLAG\_DOCUMENT\_17', 'FLAG\_DOCUMENT\_19', 'FLAG\_DOCUMENT\_20', 'FLAG\_DOCUMENT\_21', 'AMT\_GOODS\_PRICE', 'FLAG\_EMP\_PHONE', 'REGION\_RATING\_CLIENT\_W\_CITY', 'LIVINGAPARTMENTS\_AVG', 'LIVINGAREA\_AVG', 'APARTMENTS\_MODE', 'BASEMENTAREA\_MODE', 'YEARS\_BEGINEXPLUATATION\_MODE', 'YEARS\_BUILD\_MODE', 'COMMONAREA\_MODE', 'ELEVATORS\_MODE', 'ENTRANCES\_MODE', 'FLOORSMAX\_MODE', 'FLOORSMIN\_MODE', 'LANDAREA\_MODE', 'LIVINGAPARTMENTS\_MODE', 'LIVINGAREA\_MODE', 'NONLIVINGAPARTMENTS\_MODE', 'NONLIVINGAREA\_MODE', 'APARTMENTS\_MEDI', 'BASEMENTAREA\_MEDI', 'YEARS\_BEGINEXPLUATATION\_MEDI', 'YEARS\_BUILD\_MEDI', 'COMMONAREA\_MEDI', 'ELEVATORS\_MEDI', 'ENTRANCES\_MEDI', 'FLOORSMAX\_MEDI', 'FLOORSMIN\_MEDI', 'LANDAREA\_MEDI', 'LIVINGAPARTMENTS\_MEDI', 'LIVINGAREA\_MEDI', 'NONLIVINGAPARTMENTS\_MEDI', 'NONLIVINGAREA\_MEDI', 'TOTALAREA\_MODE', 'OBS\_60\_CNT\_SOCIAL\_CIRCLE'

]

Bureau: [

'AMT\_DRAWINGS\_ATM\_CURRENT', 'AMT\_INST\_MIN\_REGULARITY', 'AMT\_PAYMENT\_TOTAL\_CURRENT', 'AMT\_RECEIVABLE\_PRINCIPAL', 'AMT\_RECIVABLE', 'AMT\_TOTAL\_RECEIVABLE', 'CNT\_DRAWINGS\_POS\_CURRENT'

]

Bureau\_balance: []

Credit\_card\_balance:[

'AMT\_DRAWINGS\_ATM\_CURRENT', 'AMT\_INST\_MIN\_REGULARITY', 'AMT\_PAYMENT\_TOTAL\_CURRENT', 'AMT\_RECEIVABLE\_PRINCIPAL', 'AMT\_RECIVABLE', 'AMT\_TOTAL\_RECEIVABLE', 'CNT\_DRAWINGS\_POS\_CURRENT'

]

Installment\_payment: [

'DAYS\_ENTRY\_PAYMENT', 'AMT\_INSTALMENT'

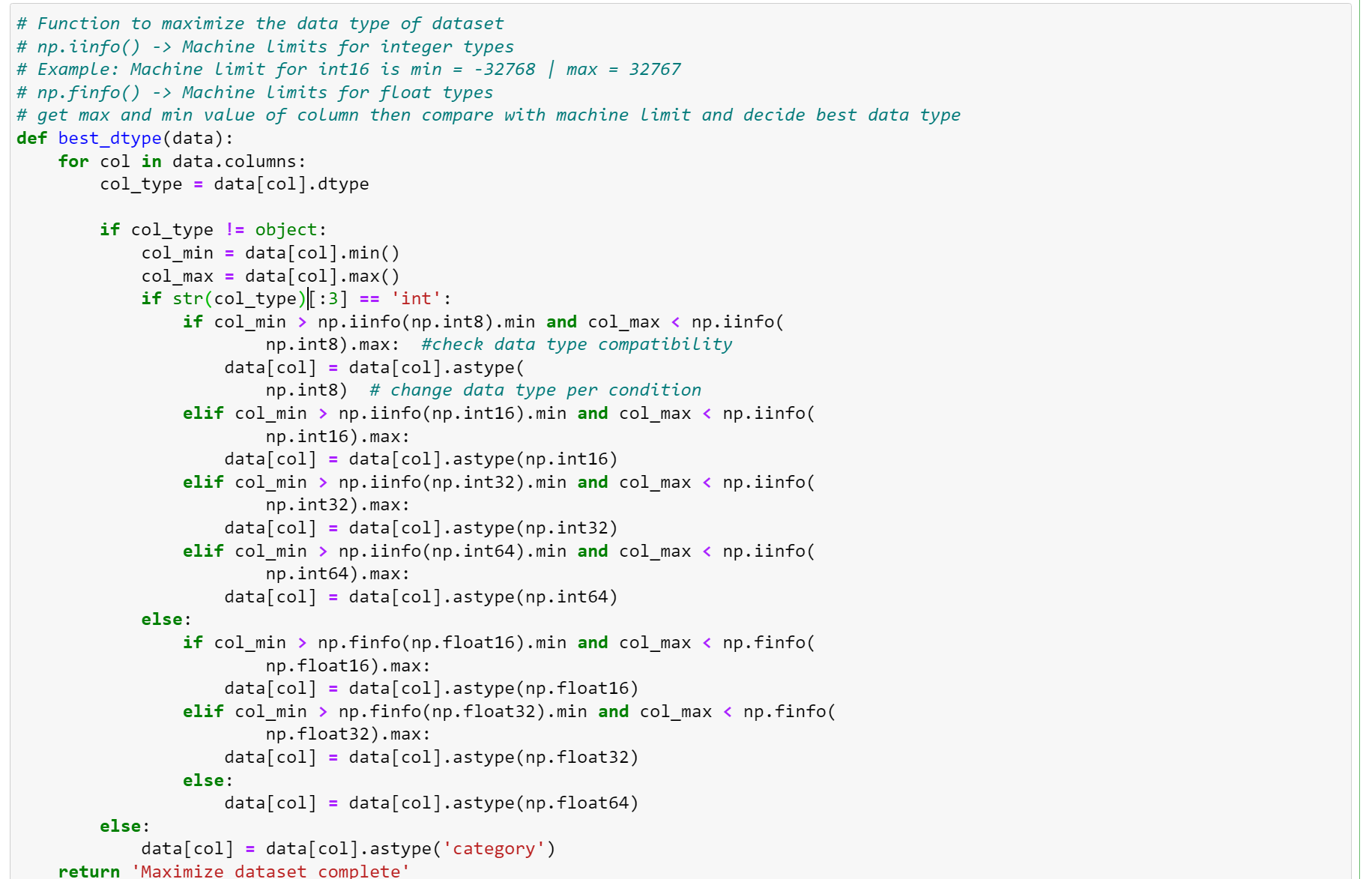
]

POS\_CASH\_balance: [

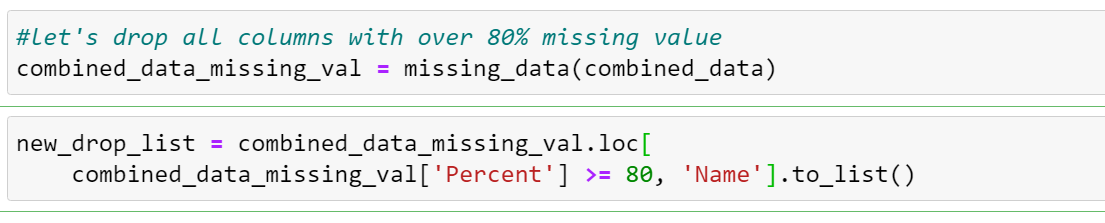
'CNT\_INSTALMENT\_FUTURE'

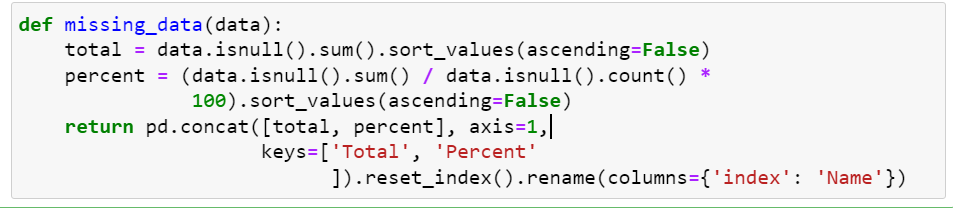
]

* Optimize the data types for each dataset:



* Drop columns with over 80% missing value:





New\_drop\_list = [

'RATE\_INTEREST\_PRIMARY', 'RATE\_INTEREST\_PRIVILEGED', 'COMMONAREA\_MODE',

'COMMONAREA\_MEDI', 'NONLIVINGAPARTMENTS\_MODE', 'NONLIVINGAPARTMENTS\_MEDI',

'LIVINGAPARTMENTS\_MEDI', 'LIVINGAPARTMENTS\_AVG', 'LIVINGAPARTMENTS\_MODE',

'FLOORSMIN\_MEDI', 'FLOORSMIN\_MODE', 'YEARS\_BUILD\_MODE', 'YEARS\_BUILD\_MEDI',

'LANDAREA\_MEDI', 'LANDAREA\_MODE', 'BASEMENTAREA\_MEDI', 'BASEMENTAREA\_MODE',

'NONLIVINGAREA\_MEDI', 'NONLIVINGAREA\_MODE', 'ELEVATORS\_MODE', 'ELEVATORS\_MEDI',

'APARTMENTS\_MEDI', 'APARTMENTS\_MODE', 'ENTRANCES\_MEDI', 'ENTRANCES\_MODE',

'LIVINGAREA\_MEDI', 'LIVINGAREA\_AVG', 'LIVINGAREA\_MODE', 'FLOORSMAX\_MEDI',

'FLOORSMAX\_MODE', 'YEARS\_BEGINEXPLUATATION\_MEDI',

'YEARS\_BEGINEXPLUATATION\_MODE', 'TOTALAREA\_MODE', 'OBS\_60\_CNT\_SOCIAL\_CIRCLE',

'FLAG\_DOCUMENT\_13', 'FLAG\_DOCUMENT\_15', 'FLAG\_DOCUMENT\_16',

'FLAG\_DOCUMENT\_17', 'FLAG\_DOCUMENT\_19', 'FLAG\_DOCUMENT\_20',

'FLAG\_DOCUMENT\_12', 'FLAG\_DOCUMENT\_10', 'FLAG\_DOCUMENT\_2',

'AMT\_GOODS\_PRICE\_x', 'FLAG\_EMP\_PHONE', 'REGION\_RATING\_CLIENT\_W\_CITY',

'FLAG\_DOCUMENT\_14', 'FLAG\_DOCUMENT\_21', 'AMT\_PAYMENT\_CURRENT',

'AMT\_DRAWINGS\_OTHER\_CURRENT', 'AMT\_DRAWINGS\_POS\_CURRENT',

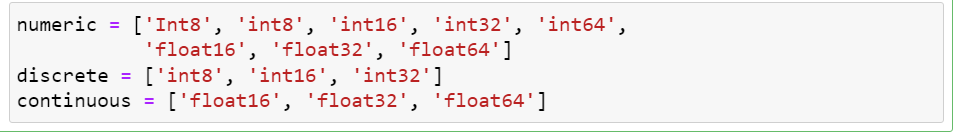
'CNT\_DRAWINGS\_ATM\_CURRENT', 'CNT\_DRAWINGS\_OTHER\_CURRENT'

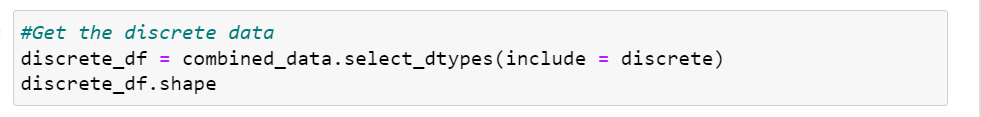
]

1. Handle missing value:

* Numeric columns:

Discrete data:

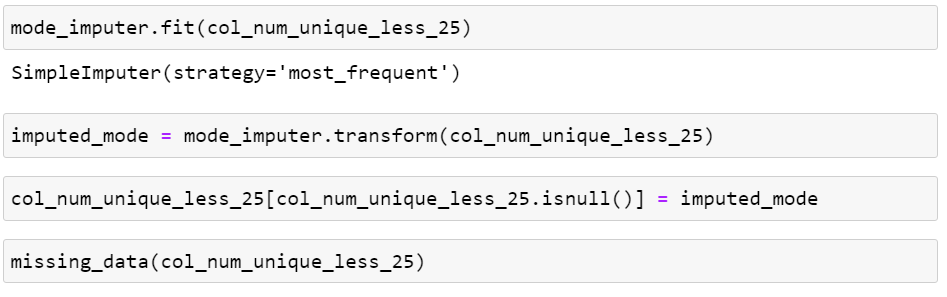


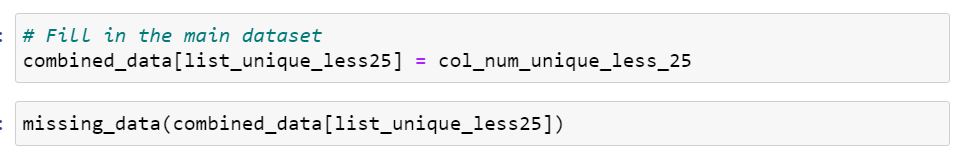


**=> Discrete data variables don't have missing data**

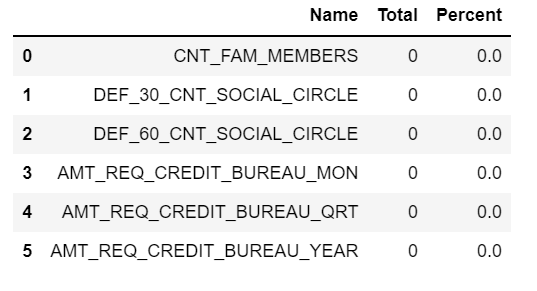
Continuous data:

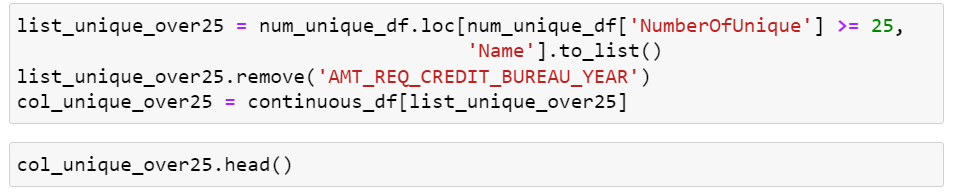
* **After consideration of the number of unique in continuous dataframe we will fill missing value by the method as follow. Variables with over 25 number of unique except will be filled with its mode. The rest will be filled with mean**
* **We will use imputer estimator to fill the missing values. Replace missing values using mean, median, mode ... (descriptive statistic) values for the corresponding columns**

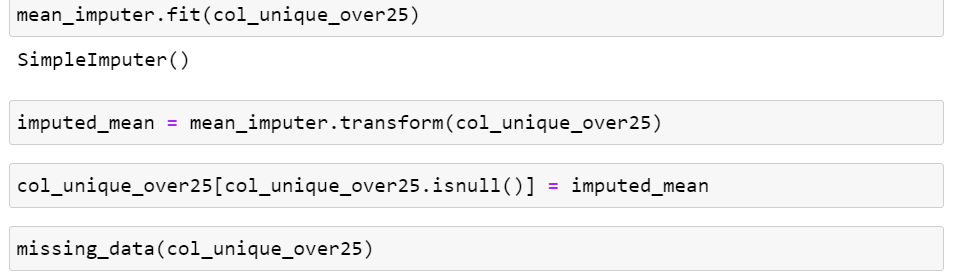
****

****

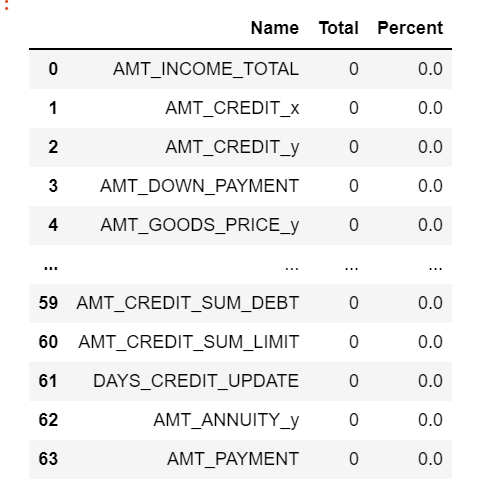
Output:

****

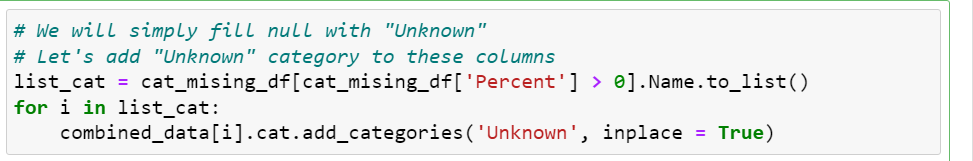
****

****

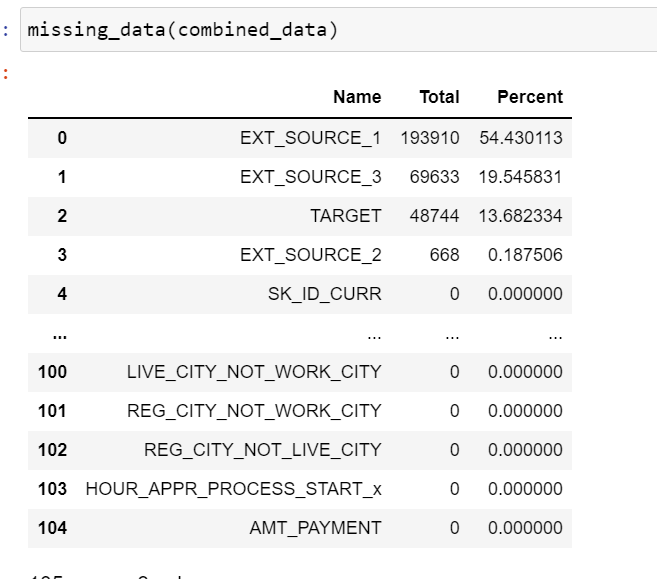
Output:



* Categorical columns:
* We will simply fill null with "Unknown".
* Let's add "Unknown" category to these columns



Output:



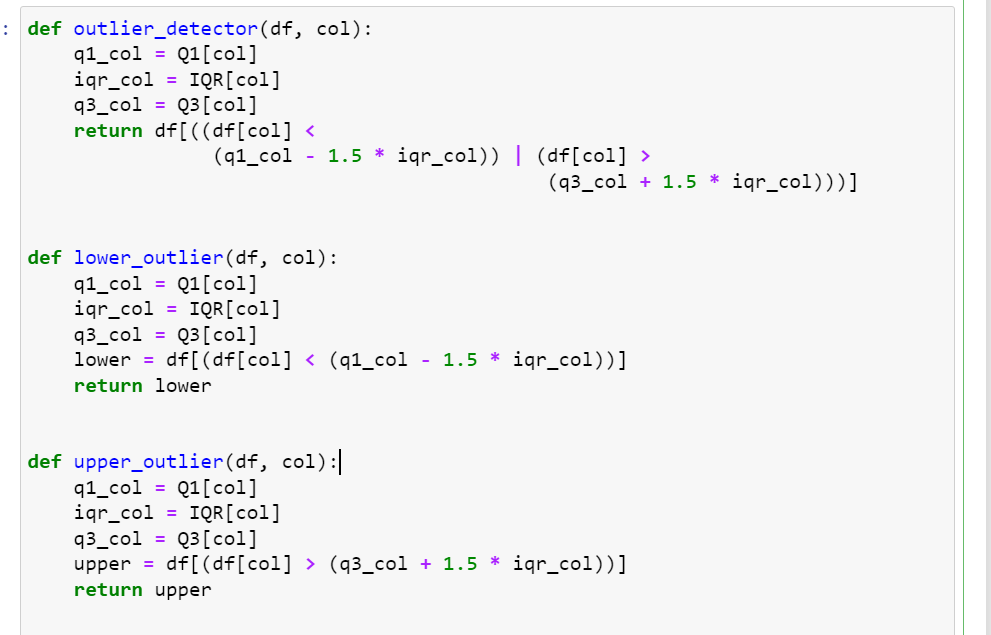
EXT\_SOURCE\_1, EXT\_SOURCE\_2, EXT\_SOURCE\_3, are very important features so we will not do anything to it at this moment.

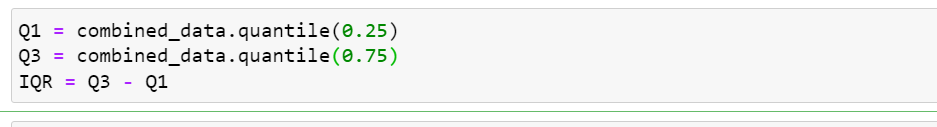
TARGET has missing value because application\_test don’t have that column

1. Handling Outliers:

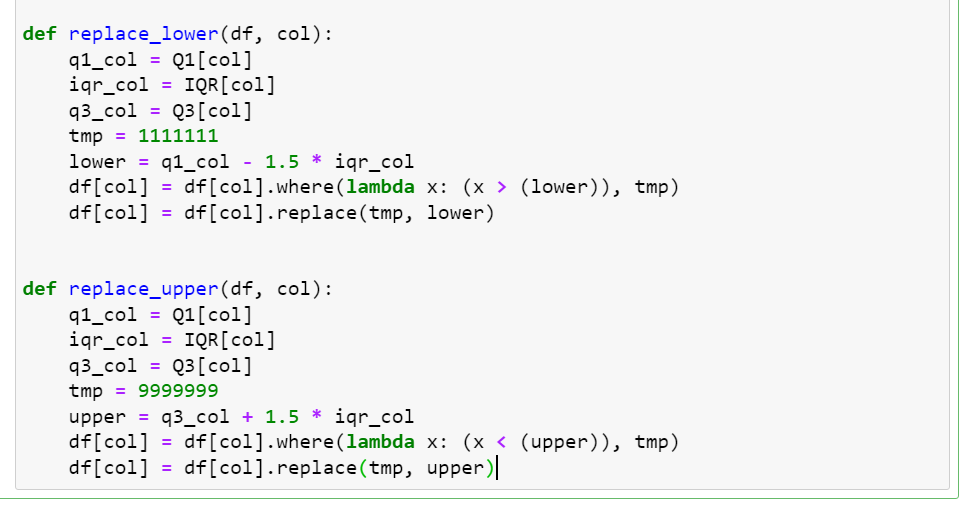
**Approach:**

* Sort the dataset in ascending order
* calculate the 1st and 3rd quartiles(Q1, Q3)
* compute IQR=Q3-Q1
* compute lower bound = (Q1–1.5\*IQR), upper bound = (Q3+1.5\*IQR)
* loop through the values of the dataset and check for those who fall below the lower bound and above the upper bound and mark them as outliers



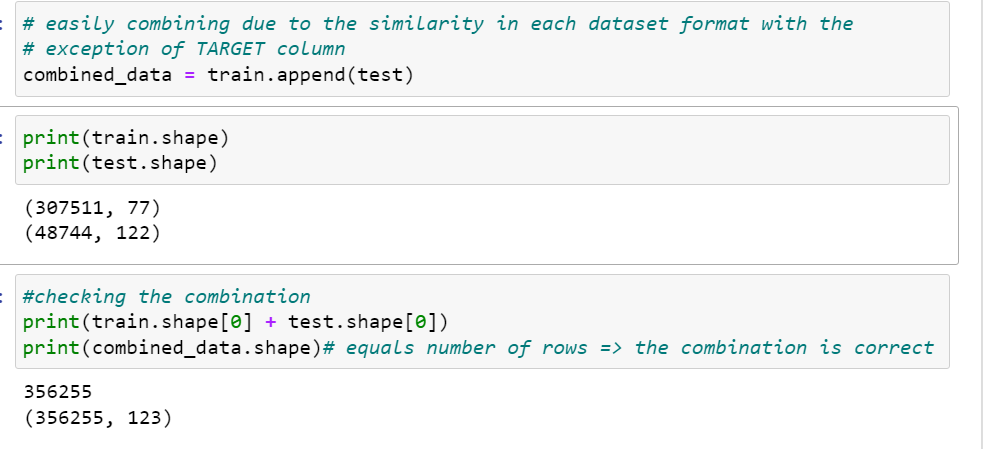


In this case, we replaced the out-of-bounds data with the value of Q3 + (1.5 \* IQR) and those before the lower limit position in the chart with the value of Q1 – (1.5 \* IQR)



1. Combine data:
2. Handling Train/test:

* Easily combining due to the similarity in each dataset format with the exception of TARGET column
* We will call the dataset combined\_data



1. Handling bureau and bureau\_balance:

* Identify the issue:

1. bureau dataset gives us the application of previous loans that client got from other institution reported to Credit Bureau

2. bureau\_balance dataset illustrates the monthly balance of credits in bureau dataset (Behaviral data)

3. bureau matches SK\_ID\_CURR columns with train/test

4. train/test has one row per customer

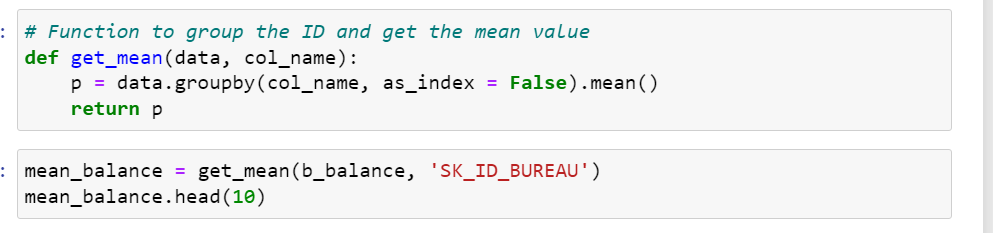
5. bureau has multiple row per customer because most customer apply for different loans

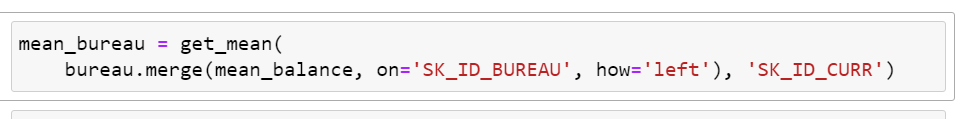
6. bureau\_balance contains rows for each month of credit reported to bureau\_balance

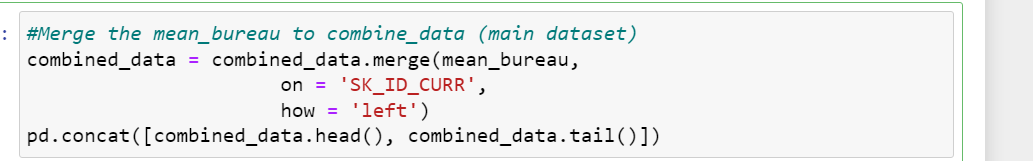
7. bureau\_balance matches SK\_ID\_BUREAU with bureau

* Approach the issue:

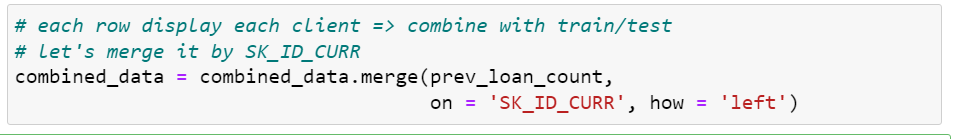
1. Get mean values of columns grouped by SK\_ID\_BUREAU to minimize bureau\_balance



2. Merge it with bureau

3. Do the same with bureau but grouped by SK\_ID\_CURR

4. Merge it with train/test



**We have added 14 new usable columns from bureau and bureau\_balance dataste after the train/test dataset**

1. Handling

previous\_application/credit\_card\_balance/installments\_payments/POS\_CASH\_balance:

* Identify the issue:

1. previous\_application shows data of client's previous loans from Home Credit (one row per previous application)

2. POS\_CASH\_balance shows the monthly balance of client's previous loans in Home Credit (Behavioral data) (one row for each month)

3. installments\_payments shows repayment history for previous credits with Home Credit (Behavioral data) (one row for each payment)

4. credit\_card\_balance monthly balance of client's previous credit card loans with Home Credit (Behavioral data) (one row for each month)

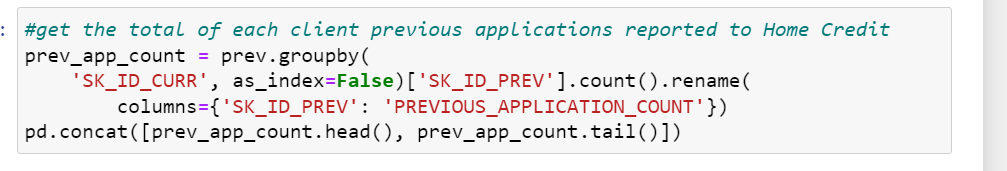
5. All dataset matches the SK\_ID\_CURR with combined\_data

6. previous\_application have SK\_ID\_PREV as its own column matching with the other 3 datasets.

* Approach the issue:

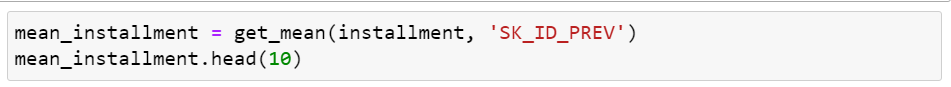
The approach will be similar to the approach of bureau dataset

1. Get mean values of columns grouped by SK\_ID\_PREV to minimize the three subsidiary datasets (credit, installment, pos)

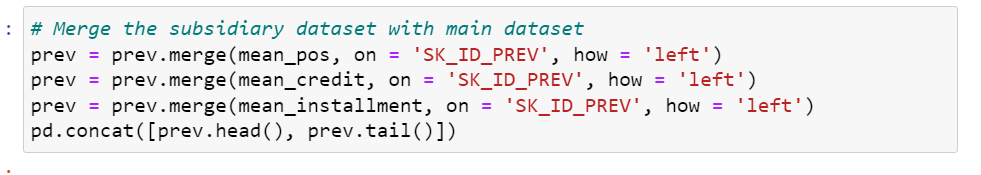








1. Merge the 3 dataset with the main dataset (prev)



1. From the merged prev dataset we now minimize it by getting mean values



1. Merge the minized prev dataset to the combined dataset



**We have added 52 new usable columns from previous\_application, credit\_card\_balance, POS\_CASH\_balance, installments\_payments dataset for the final combinated dataset of Home Credit Default Risk dataset**