LOAN DATA PREPROCESSING USING NUMPY IN PYTHON

Introduction:

A Lending Company has provided us with the loan-data.csv file, which contains the information of Loan applicants. We need to use this information to generate a report on the creditworthiness of each applicant.

Observation:

When opening the loan-data.csv file, we have observed many missing values in the data. And also we can see so much text data which cant be used for further analysis. So we need to preprocess this data which can be analysed further.

The description for each Column names are as follows:

Id	: A unique assigned ID for the loan listing
Issue_d	: The month which the loan was funded
loan_amnt	: The listed amount of the loan applied for by the borrower. If at some point in time, the credit department reduces the loan amount, then it will be reflected in this value.
loan_status	: Current status of the loan
funded_amnt	: The total amount committed to that loan at that point in time.
term	: The number of payments on the loan. Values are in months of 36 or 60.
int_rate	: Interest Rate on the loan
installment	: The monthly payment owed by the borrower if the loan originates.
grade	: LC assigned loan grade
sub_grade	: LC assigned loan subgrade
verification_status	: Indicates if the borrowers income was verified by LC, not verified, or if the income source was verified
url	: URL for the LC page with listing data.
addr_state	: The state provided by the borrower in the loan application
total_pymt	: Payments received to date for total amount funded

IMPORTING DATA:

Importing & Checking for Incomplete Data :

- Data is imported into the kernel.
- When checked for the total number of missing values, we found a total of 88005 missing values in the dataset.

Missing values :

- Checked for missing values and can see there are 88005 missing values in the dataset.
- So assuming a temporary_fill = maximum values in dataset + 1
- And also creating an array named temporary_stats which has 3 arrays namely minimum values, mean values and maximum values of data set along each column.

Re-importing the Dataset :

- Reimporting the dataset into two variables in which string data columns are stored in *loan_data_strings* and numeric data columns are stored in *loan_data_numeric* , while importing numeric data, the missing values are assigned with temporary_fill which can be filtered and changed easily in future.
- Similarly, headers are loaded into *header_strings* and *header_numeric.*

MANIPULATING STRING COLUMNS :

Issue Date :

- The data in issue_d is in the format as MMM-YY.
- After inspecting using np.unique() we found this data belongs to 2015. So, Suffix “-15” is stripped from the column.
- Then month of issue_d “Jan, Feb, Mar,.....” is replaced with “1,2,3,4…….” and missing values in this column are assigned with “0”.
- Header for this column is changed from “string_d” to “string_date”

Loan Status:

- In this loan_status field, the status is given in the for of text in 8 types namely ['Charged Off', 'Current', 'Default', 'Fully Paid', 'In Grace Period', 'Issued', 'Late (16-30 days)', 'Late (31-120 days)']
- From this [missing values, Charged Off, Default, Late (31-120 days)] were replaced with 0 considering these are negative impacts on a loan application.
- And all other items were replaced with 1 for ease analysis.

Term:

- In the term field, there are two variants which are 36 months and 60 months.
- Initially, stripped “ months” characters from the field.
- Then assigned ‘60’ for missing values considering the worst case scenario.

Grade and Subgrade:

- When inspected using np.unique(), the grade is categorised from “A - G” and subgrade is categorised into “1-5” with Grade as Prefix for that particular application. And also there are missing values present in both Grade and Subgrade.
- For missing values in Subgrade where grade is already available, the subgrade is assigned by using grade with a prefix of ‘5’ considering the worst case scenario.
- Removed the grade field from the database as we already have subgrade which is enough.
- Still there are 9 missing values. These missing values are assigned with a new grade ‘H1’.
- Converting these subgrades into numbers from 1 to 36 using a dictionary with keys and values as lists.

Verification Status:

- When inspected, verification status has 3 categories ['Not Verified', 'Source Verified', 'Verified'] and missing values.
- Assigned ‘0’ for Not Verified and missing values.
- Assigned ‘1’ for Source Verified and Verified.

URL:

- When inspected, URL has ‘https://www.lendingclub.com/browse/loanDetail.action?loan_id=’
- as a prefix for all records. So stripping this prefix from all records.
- When comparing the resulting field with the “id” field in loan_data_numeric using np.array_equal, we found both are the same. So this field has been removed from the table.

State Address:

The State Address was replaced with [1,2,3,4,0] based on below details

1.states_west = ['WA', 'OR','CA','NV','ID','MT', 'WY','UT','CO', 'AZ','NM','HI','AK']

2.states_south =['TX','OK','AR','LA','MS','AL','TN','KY','FL','GA','SC','NC','VA','WV','MD','DE','DC']
3.states_midwest = ['ND','SD','NE','KS','MN','IA','MO','WI','IL','IN','MI','OH']
4.states_east = ['PA','NY','NJ','CT','MA','VT','NH','ME','RI']
0. Missing values

Converting to Numbers:

- We have replaced all string items into numbers in loan_data_strings. But still these numbers are in string format itself.
- In this step, we will change the datatype of loan-data_strings from ‘str’ to ‘int’
- Here, we created a checkpoint named Checkpoint-Strings and loaded header_strings and loan-data_strings to this Checkpoint.

MANIPULATING NUMERIC COLUMNS:

- We have initially assigned temporary_fill for all missing values in the Numeric dataset which is loan_data_numeric. Now this temporary_fill is to be replaced with the most appropriate value.
- We already have temporary_stats which have minimum, mean and maximum values from each field before assigning the temporary_fill for missing values. This temporary_stats can be used for assigning most appropriate value for missing values.

ID:

- Checked for temporary_fill in ID field. There were no instances of temporary_fill in the ID field. So, it's clear.

Funded Amount:

- Check for temporary_fill in this field and replaced the temporary_fill with minimum value of the field taken from temporary_stats.

Loaned Amount, Interest rate, total Payment, Instalment:

- Check for temporary_fill in the field and replace the temporary_fill with the mean of the respective field from temporary_stats.
- Interest rate is in percentage and it has been changed to decimal.

- All these values from header_numeric and loan_data_numeric are saved into Checkpoint_numeric as header and data.

SAVING THE PREPROCESSED DATA:

Created a complete data set named ‘loan_data” step wise

1. Loan_data is assigned with checkpoint_numeric[‘data’] and checkpoint_strings[‘data’] and stacking horizontally.
2. Assigned checkpoint_numeric[‘header’] and chekpoint_strings[‘data’] into header_full
3. Assigned header_full and loan_data by vertically stacking into loan_data
4. Finally saving the resultant dataset as “loan-data-preprocessed.csv”