Homework 2 - Make Card Transaction Variables

Cleaning/Imputation Logic in the Notebook:

- We have a CSV format dataset that includes columns with missing values represented as 'NaN'. We will disregard these columns for our analysis.

- The date column's current data type is 'object', and we will transform it to the 'datetime' format.

- To adhere to the given requirements, we will filter the data to only include transactions with a transaction type of 'P' and remove any other transactions.

- Furthermore, there is an unusual transaction where the amount exceeds $3000000, which we will remove from the dataset as an outlier.

- Once we have cleaned the data, we will determine the number of rows that contain missing or 'NaN' values.

Clean and impute merchnum:

- First, it replaces any values of '0' in the 'Merchnum' column with NaN (not a number) using the replace() method increasing the null values from 3198 to 3251.

- Next, it creates a dictionary called 'merchdes_merchnum' to map 'Merch description' values to their corresponding 'Merchnum' values. This is done by iterating over the DataFrame rows where both 'Merch description' and 'Merchnum' are not null and adding each unique 'Merch description' to the dictionary along with its corresponding 'Merchnum' value.

- The code then fills in any remaining NaN values in the 'Merchnum' column by mapping each non-null 'Merch description' value to its corresponding 'Merchnum' value in the 'merchdes_merchnum' dictionary using the fillna() method.

- Any remaining NaN values in the 'Merchnum' column corresponding to adjustment transactions are assigned the value 'unknown' using the mask() method.

- Finally, any remaining NaN values in the 'Merchnum' column that do not have a corresponding 'Merch description' value are assigned a new unique 'Merchnum' value. This is done by creating a dictionary called 'merchnum_create' to map each unique 'Merch description' value to a new 'Merchnum' value, starting from the current maximum 'Merchnum' value in the DataFrame plus one. The fillna() method is then used again to fill in these new 'Merchnum' values.

Akansha Lalwani (A59019733)

- It checks the number of null values for each column in the DataFrame to confirm that the 'Merchnum' column has no remaining null values after cleaning and imputation.

Clean and impute State:

- The code checks the number of missing values in the Merch state column and finds that there are 1020 missing values.

- It identifies the unique non-null zip codes for the transactions with missing Merch state values.

- It creates a dictionary zip_state that maps each of the zip codes to the corresponding state. For example, the zip code '00926' is mapped to 'PR' (Puerto Rico).

- It creates dictionaries merchnum_state and merchdes_state that map each non-null Merchnum and Merch description to the corresponding Merch state.

- It fills in the missing Merch state values by mapping each row's Merch zip, Merchnum, or Merch description to the corresponding Merch state using the dictionaries created in steps 3 and 4.

- It assigns 'unknown' to the Merch state values for the adjustment transactions.

- It changes the non-US states to 'foreign' and leaves missing values as they are.

- Finally, it checks the number of missing values in the Merch state column again and finds that there are 346 missing values.

Clean and impute zip:

- The missing values in zip are first identified using the isnull() function. The missing values in the ‘Merch zip’ column are 4300.

- Then, it creates two dictionaries merchnum_zip and merchdes_zip to map each unique value of ‘Merchnum’ and ‘Merch description' to its corresponding ‘Merch zip’. This is done by iterating through the non-null values of the ‘Merchnum’ and ‘Merch description’ columns and adding their respective ‘Merch zip’ values to the dictionaries if they are not already present in the dictionaries.

- Next, the missing values in the ‘Merch zip’ column are filled in by mapping the values in the ‘Merchnum’ and ‘Merch description’ columns to their corresponding "Merch zip" values using the dictionaries created earlier. This is done using the map() function.

- After this step, there are still 2658 missing values in the ‘Merch zip’ column. To handle this, it assigns the value ‘unknown’ to the missing values for transactions labeled as ‘RETAIL CREDIT ADJUSTMENT’ and ‘RETAIL DEBIT ADJUSTMENT’. This is done using the mask() function.

- Finally, it displays the first 50 rows where ‘Merch zip’ is missing to verify the imputation visually.

New Cleaning/Imputation logic: (implemented in the notebook)

- I have initialized a dictionary called 'state_zip' to hold non-null values of the 'Merch state' column as keys and the corresponding non-null values of one of the zip codes associated with that state.

- Using this dictionary, I have replaced any null values in the 'Merch zip' column with the appropriate zip code from the dictionary based on the 'Merch state' column's value.

Summary Table:

Description	# Variables Created
Date of week target encoded: average fraud percentage of that day	1
Month target encoded: average fraud percentage of that month	1
State Risk target encoded: Top 15 Fraud Merchant State with highest fraud percentage	1
Benford’s Law Card number: count the # of first digits in card number	1
Benford’s Law Merchant number: count the # of first digits in merchant number	1
Benford’s Law Merchant Zip: count the # of first digits in merchant zip	1
Day Since: Number of days since application with that entity was seen	18
Velocity: Number of records with the same entity over the last {0,1,3,7,14,30,60} days	1134

Relative Velocity: Count of records and total amount with same entities seen in the past {0,1} day divided by the number of applications with those same entities seen in the last {3,7,14,30} days	288
Velocity Density: Count of records with same entities seen in the past {0,1} day divided by day since those same entities seen in the last {3,7,14,30} days	144
Cross Entity Uniqueness: Number of records with unique combinations of all the entities in the dataframe	306
Entity Amount Variability: Amount variability with the amount of same entity seen over the last {0,1,3,7,14,30,60} days	324
Counts by Entity: Number of unique entities for a particular fields over the last {0,1,3,7,14,30,60} days	1836
Relative Velocity (squared): Count of records and total amount with same entities seen in the past {0,1} day divided by square of the number of applications with those same entities seen in the last {3,7,14,30} days	144
Binning Amount: Creates 5 equal sized bins to divide amount column and assigns labels (1,5) to each bin	1