**Describe the dataset of your team, What does the dataset consist of and what applications could it be used for in a ML/DL context?**

1. **Describe the dataset of your team**

**1.1 Dataset name:** Credit Card Classification - clean data

**1.2 Source:**

<https://www.kaggle.com/samuelcortinhas/credit-card-classification-clean-data>

**1.3 Context**

Credit cards are a popular risk management strategy in the financial sector. To estimate the likelihood of future defaults and credit card borrowing, it uses the personal information and data provided by credit card applicants. The bank has the authority to choose whether or not to give the applicant a credit card. The degree of risk can be accurately measured by credit scores.

Credit score cards are typically based on past data, once experiencing significant economic swings, previous models might no longer be as accurate. A prominent method for credit scoring is the logistic model. Because Logistic can determine the coefficients of each feature and is appropriate for binary classification applications. The score card will round the logistic regression coefficient and multiply it by a specific number (such as 100) to make it easier to understand and use.

Currently, as a result of the advancement of machine learning algorithms. Credit card scoring now incorporates more predictive techniques like Boosting, Random Forest, and Support Vector Machines. These techniques, however, frequently lack good transparency. It could be challenging to give customers and regulators a justification for approval or rejection.

**1.4 Content of the dataset**It contains data of various clients personal information, and their background information.

**1.5 General Information about the data**

Number of rows: 9921

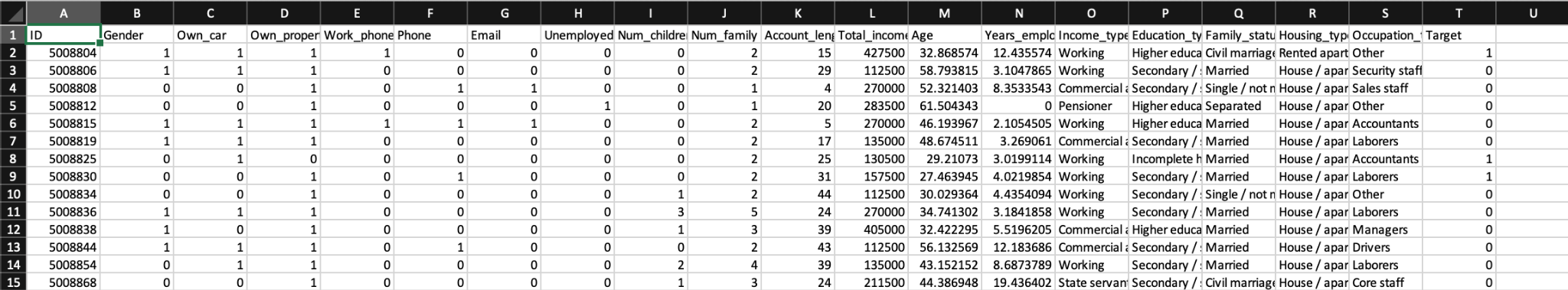
Number of columns: 20

| **Column** | **Description** | **Value** |
| --- | --- | --- |
| ID | Unique identifier of the client | Unique number  assigned to person |
| Gender | Gender of the person | 1 = Male 0=Female |
| Own\_Car | Does the person own car | 1=Yes 0 =No |
| Own\_Property | Does the person own property | 1=Yes 0 =No |
| Work\_phone | Does the person have work phone | 1=Yes 0 =No |
| Phone | Does the person have personal phone | 1=Yes 0 =No |
| Email | Does the person have email ID | 1=Yes 0 =No |
| Unemployed | Is the person unemployed | 1=Yes 0 =No |
| Num\_children | How many children does the person have | Integer value |
| Num\_family | How many family members does the person has | Integer value |
| Account\_length | Number of months credit card has been owned | Integer value in years |
| Total\_income | What is the total income of the person | Integer Value |
| age | Age of the person | Decimal value of age which includes months and days |
| Years\_employed | Number of years employed | Integer value in years |
| income\_type | Type of income source | Categorical values like(pensioner, commercial associate etc.) |
| Education | Highest level of education | Categorical values( higher secondary, secondary etc.) |
| familiy\_status | Status of family | Categorical values( single, married etc.) |
| housing\_type | Type of housing | Categorical values(House/apartment, rented apartment etc.) |
| occupation\_type | Type of occupation | Categorical values( Labourer, Driver, Core staff etc.) |
| Target | Is the person at high credit risk or low credit risk | 1 = High risk 0= low risk |
| Date | Date of the entry | Date in mm/dd/yyyy format  (this column has been added for calculation purpose) |

**1.7 Size of the Dataset**

The size of the dataset is 1.46 MB.

**1.8 Screenshot of the Dataset**

****

**1.9 Application**

A machine learning model to predict if an applicant is trustable client or not can be build after measuring the data. Vintage analysis(illustrate the behavior after an account was opened), widely-used method for managing credit risk can also be used to construct the label (Based on same origination period, calculate charge-off ratio of a loan portfolio).

**2. Base measures data collection procedure**

It is important to have an explicit data collection procedure in order to ensure consistency and reliability of the collected data. Our team planned a small overview of the data collection procedures in step 4 and we followed the collection guide to collect the data and calculate the corresponding base measures based on it. Overall, the process was:

* Data is collected from a credible source and was stored locally to perform measurement operations on it.
* Since in our case the data is static in nature, we divide the dataset into three parts assuming each part as a separate timeframe

This overview of the guideline was used to collect data for all the required base measures.

**2.1 DA01 Nds**

The **number of datasets** was collected using the Kaggle data source provided to the team [1]. The data was in excel sheet format and there was only one dataset available for use.

**2.2 DA02 Nds\_cr**

To find the **number of credible datasets**, it was assumed that the dataset provided was credible

.

**2.3 DA03 Nrec\_comp**

The dataset was parsed using a python function where all the data columns of each record was checked if they follow certain standards and format and they don’t adversely impact the quality of data.

The list of datasets are fed into getNumberOfCompliantRecords() function where all the records are validated if they are compliant using isCompliant() function and returns the count of the records.

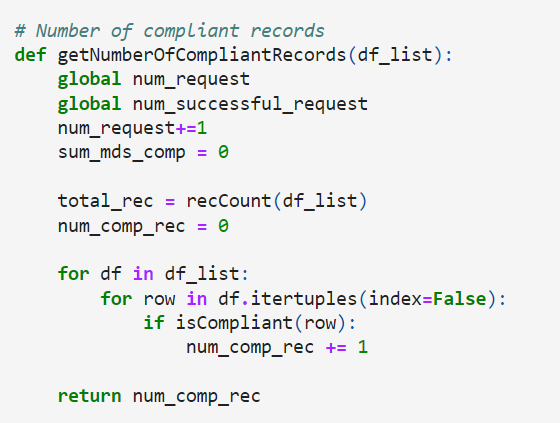


Figure: getNumberOfCompliantRecords() function

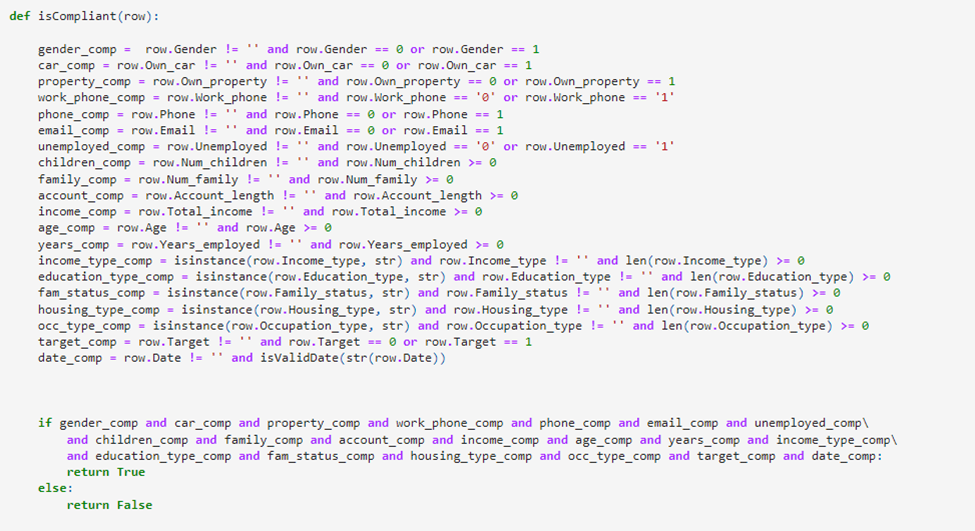
****

Figure: IsCompliant() function

**2.4. DA04 Rec\_trace**

To find the **number of traceable records** an approach similar to DA03 was used where each record in the dataset was parsed through a function which verified if the record provide an audit trail of access to the data and any change made to it in the specific context of use.

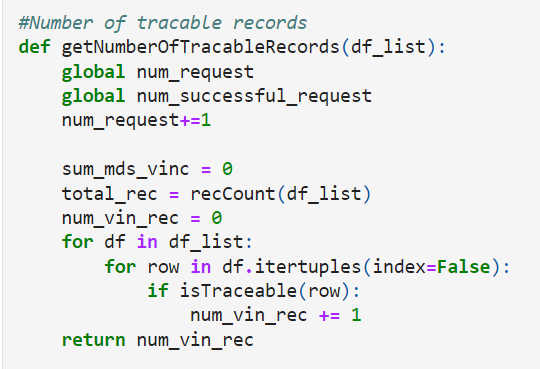


Figure: getNumberOfTracableRecords() function

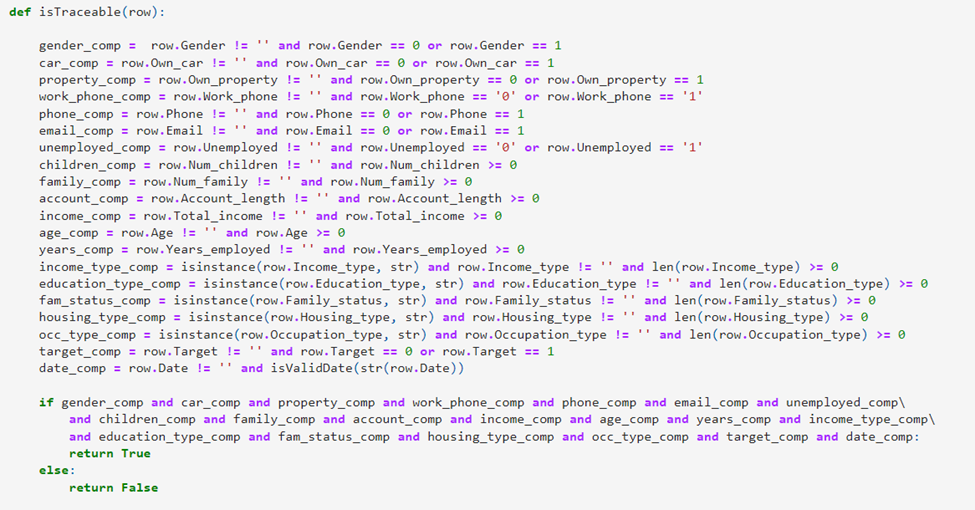
****

Figure: isTraceable()

**2.5 DA05 Ldst**

Below procedure was taken to collect the **length of record** for the available datasets

1. Dataset is loaded using the excel file into jupyter notebook.
2. A utility function written in python is used to count the total number of occurrences of data elements in the dataset

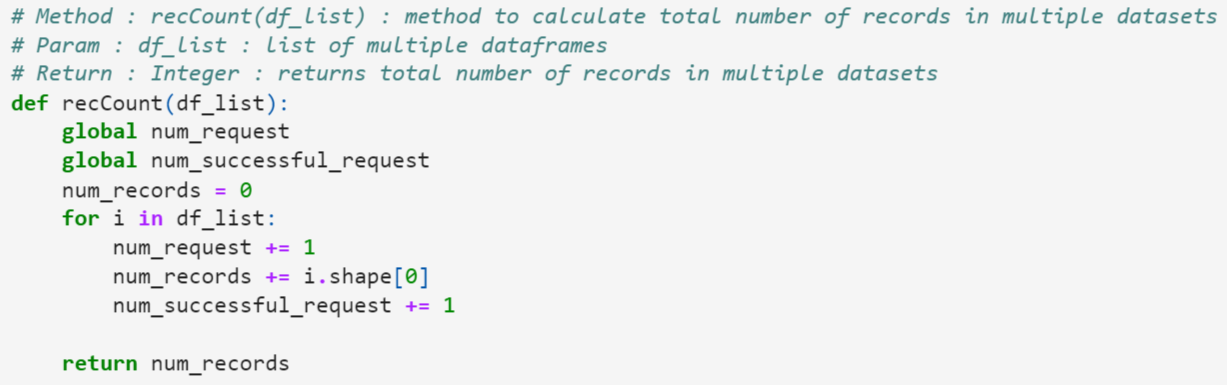
****

Figure: recCount()

**2.6 DA06 Lbd**

After obtaining the data, the **total number of records in dataset** was calculated using recCount() utility function written in python

**2.7 DA07 Rec\_acc\_age**

Following procedures was followed to find the **total number of records with ages that fall within the acceptable range**:

1. Dataset is loaded using the excel file into jupyter notebook.
2. The transaction dates were converted to data format using python
3. The lower and upper quartiles of the box and whisker model were calculated based on transaction dates
4. The count of all the records within the acceptable range were counted which gave us rec\_acc\_age

getNumberOfCurrentRecords() takes the dataset as an argument and filters out the record based on the acceptable values of the date.

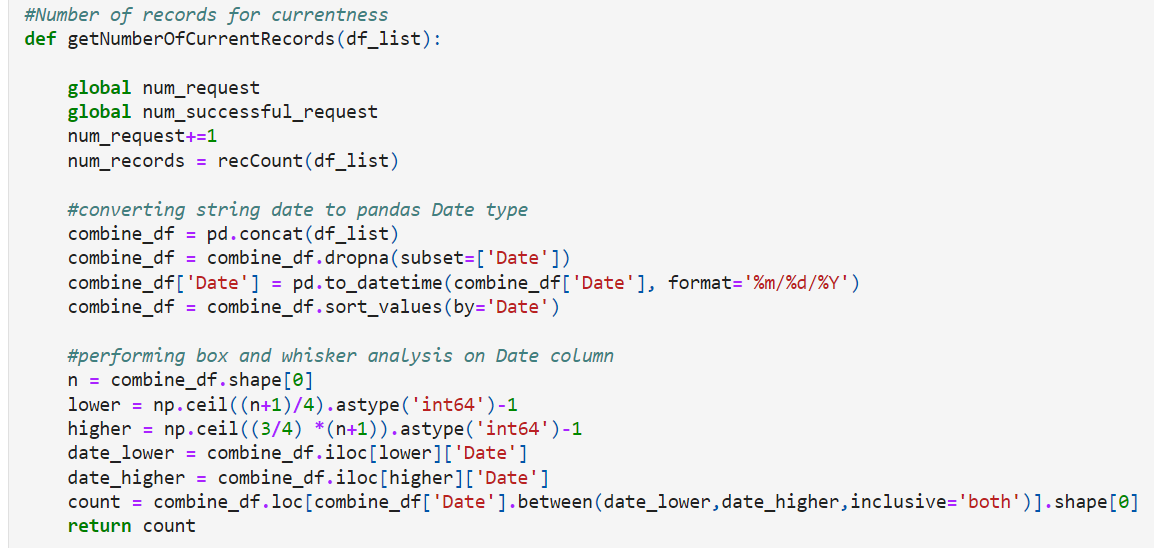


Figure: getNumberCurrentRecords()

**2.8 DA08 N\_succ\_req**

According to our measurement process guide, generally API request or queries are logged for future/audit references therefore count the number of requests for which the correct responses have been returned from the database or dataset.

But in our situation, It is assumed that dataframe is equivalent to database and each call to dataframe is considered as request. If the call does not return any error then num\_successful\_requests is incremented.

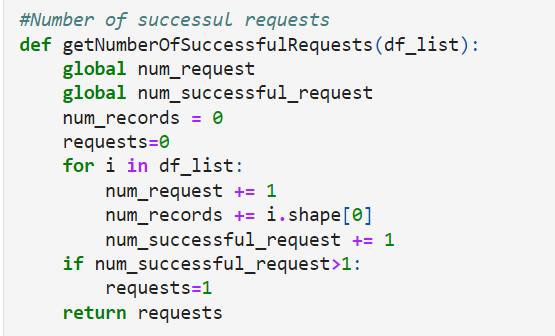


Figure: getNumberOfSuccessfulRequests() - Each call to a dataset is considered a successful request if the call doesn’t return any error

**2.9 DA09 N\_req**

It was assumed that all the calls made to datasets is considered **a request made by an authorized users and/or applications**.

**2.10 DA10 Rec\_no\_null**

To find the **frequency of records in MDS with no null values**, the dataset was parsed using python and isnull() function was used to check if the record is empty.

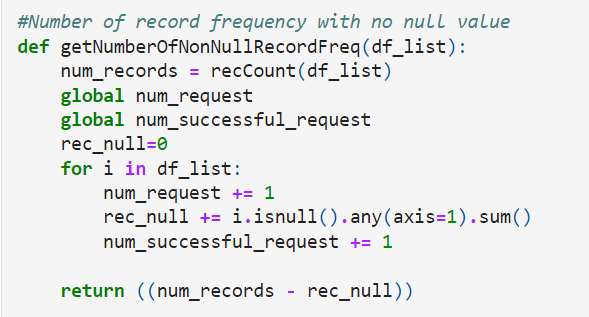


Figure: getNumberOfNonNullRecordsFreq()

**2.11 DA11 P(j)**

The **total number of duplicate items** and their specific count in each dataset was calculated using inbuilt using the inuilt functions present with python code. The data was stored locally and the the dataset was parsed using python code and converted into pandas dataframe. The inbuilt duplicated() was used to calculate P(j).

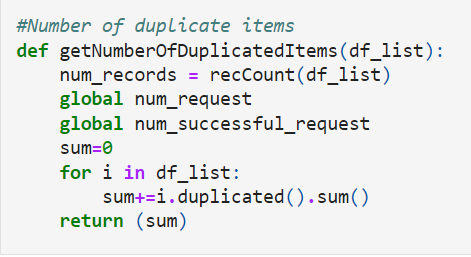
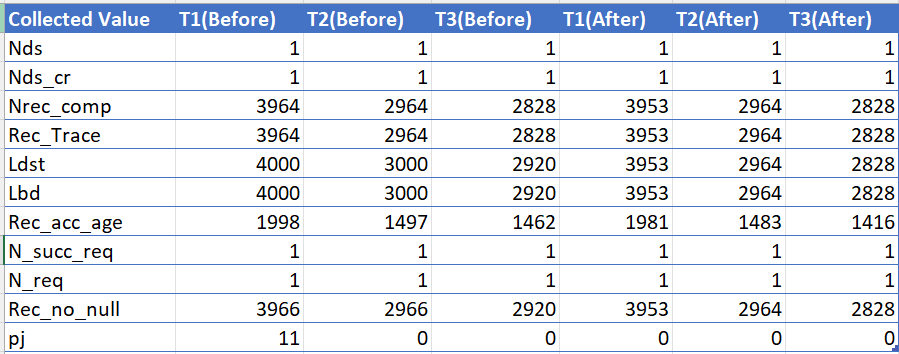


Figure: getNumberOfDuplicatedItems()

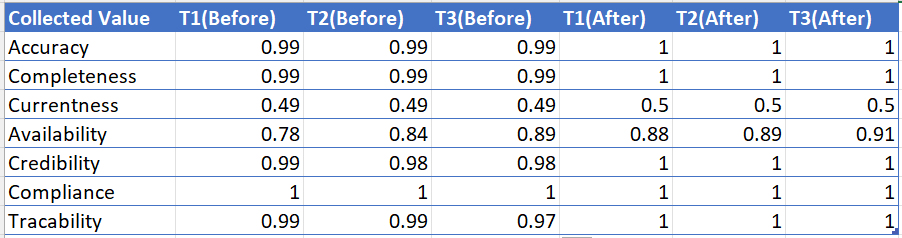
**3. Attach the collected data values (excel file, etc.)**

Below are the values calculated for three Vs. Before means the values for the raw data or dataset without doing processing. We have also calculated the measure values for after the processing of the data**.**

****

**4. For each of the 3 V’s indicators:**

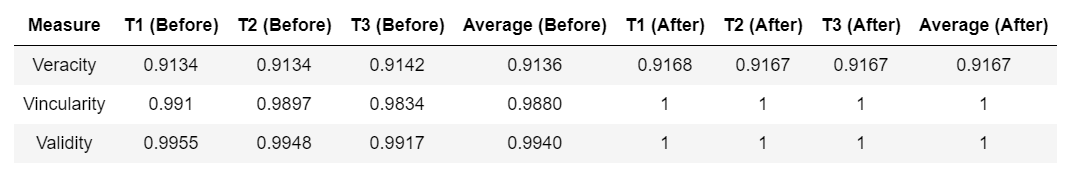
**4.1 Attach the values of the corresponding derived measure(s), where applicable (excel file, etc.)**

****

**4.2 Values of the V’s (Validity, Vincularity and Veracity) at each time frame / data pipeline phase**

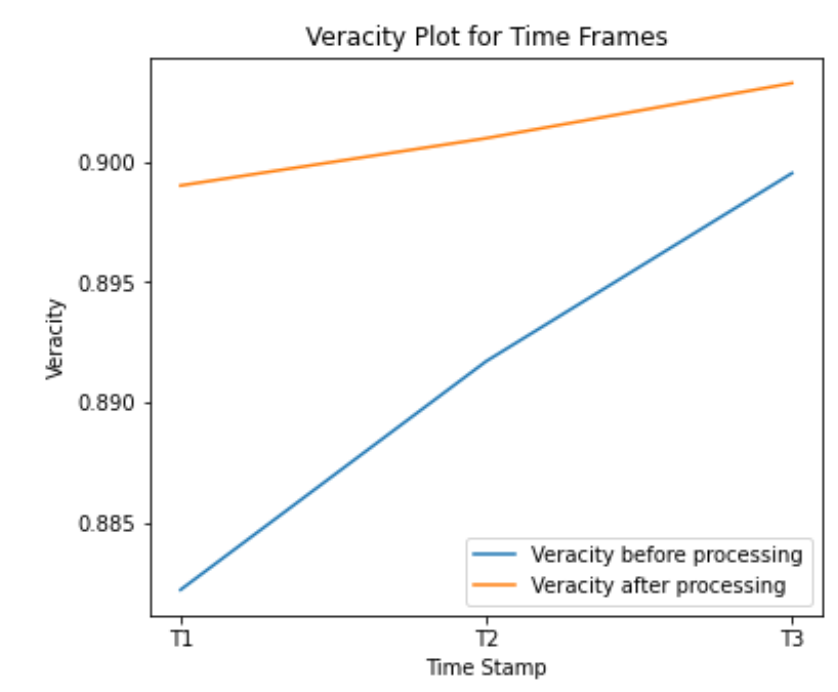
**4.3 Average value of each of the V’s (Validity, Vincularity and Veracity) at the end of the process**

**4.4 Final Value of each of the V’s (Validity, Vincularity and Veracity) at the end of the process**

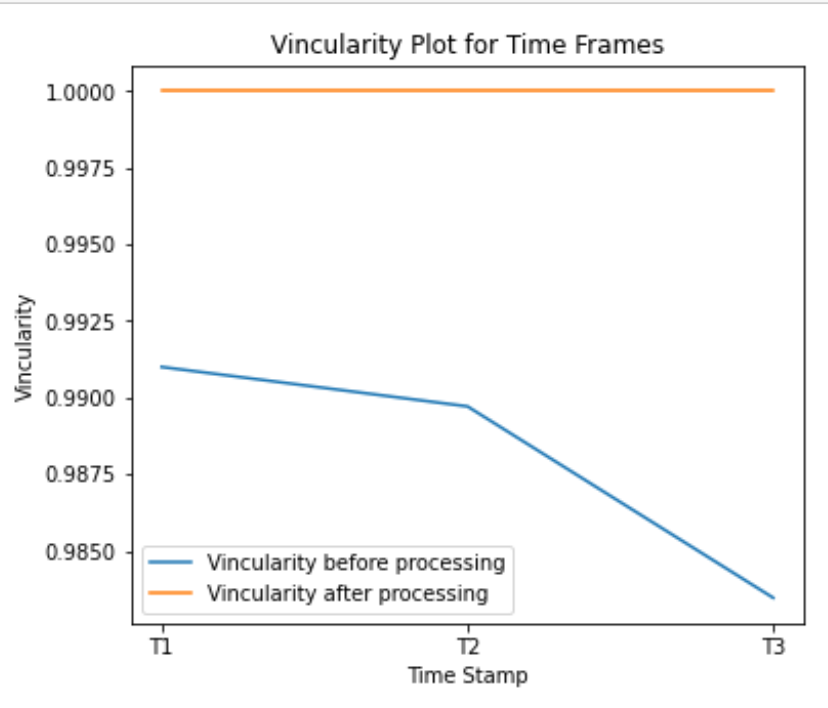
****

**4.5 Draw the graphs of the indicators Mval, Mvinc, Mver generated from the values of the derived measures**

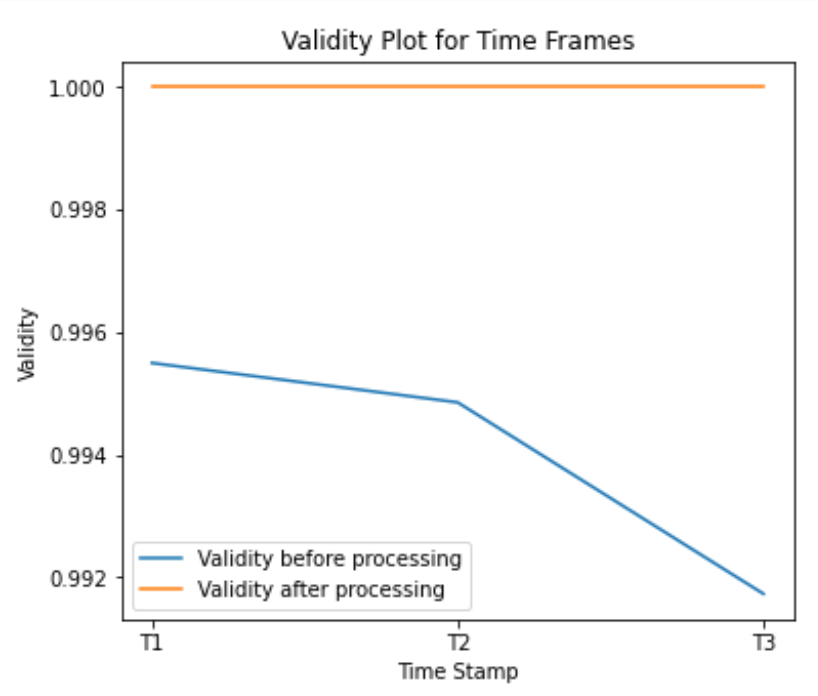
1. **Veracity**

****

1. **Vincularity**

****

1. **Validity**

****

**5. Write your conclusions: Would your data be usable for a machine learning algorithm? Justify your answer.**

**Veracity:** Veracity refers to the degree that data is accurate, trusted and precise. It is not only the accuracy of the data itself but the trustworthiness of the data source, type, and processing of it. In our case when we performed the measurement before processing of the data. Average accuracy value we got over the three different time frames is 91.37%. After processing of data, we performed various operations in order to remove the noise from the data such as removing duplicates, null values or even invalid data values for any of the records. In this way, we tried to make sure our data is more cleaned and could perform well for machine learning algorithms. We again performed measurements on the processed data, in this case we were able to get the veracity value of 91.67%.

**Vincularity:** Vincularity refers to the degree of connectivity and linkage of data. We calculated the vincularity before processing the data and the average value we got was 98.80% which is good for many of the machine learning algorithms. After processing the data, we got the average value 100%. When we have the vincularity of 100%, all the records from the datasource are traceable.

**Validity:** Validity of Big Data is defined in terms of its accuracy and correctness for the purpose of usage. Accuracy and correctness are important factors for any machine learning algorithms. If the data is not accurate and correct, machine learning models will give false results which can lead to business losses for the customers. We performed the validity calculation on row data, and we got the value for validity 99%. After processing the data, we achieved a validity value of 100%. So after processing data, our dataset is more correct and accurate.

After analyzing the above data we can conclude that the quality of big data before and after processing is within acceptable standards and it can be used to train machine learning models.

**6 Link to Project code**

<https://github.com/Kshitij13579/SOEN-6611-Project-Software-Measurement>

**References:**

1. <https://www.kaggle.com/datasets/samuelcortinhas/credit-card-classification-clean-data>