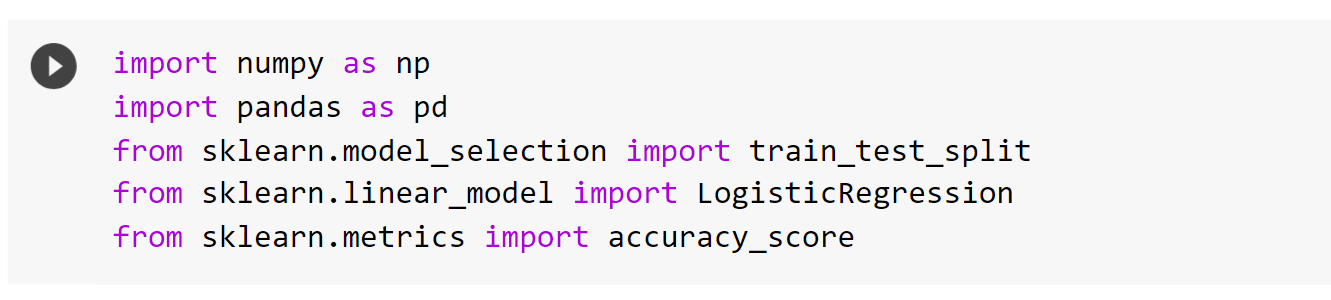
**Credit Card Fraud Detection**

**Phase 3: Development Part1**

1. **Importing the required libraries:**

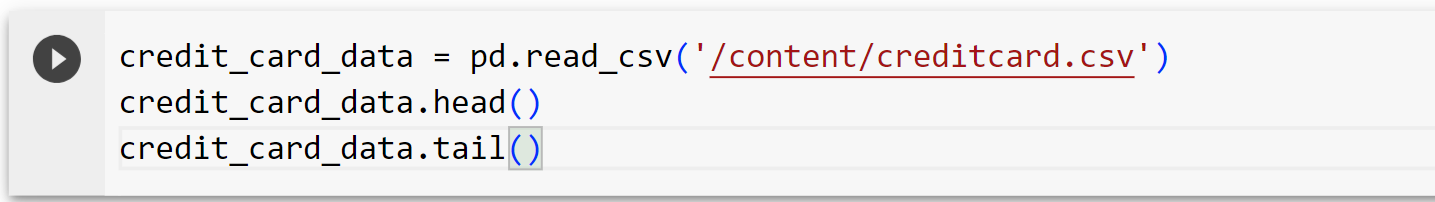
Importing the necessary Python libraries, including NumPy, Pandas, scikit-learn functions for model training, and performance evaluation which is used to perform credit card fraud detection for the given dataset.



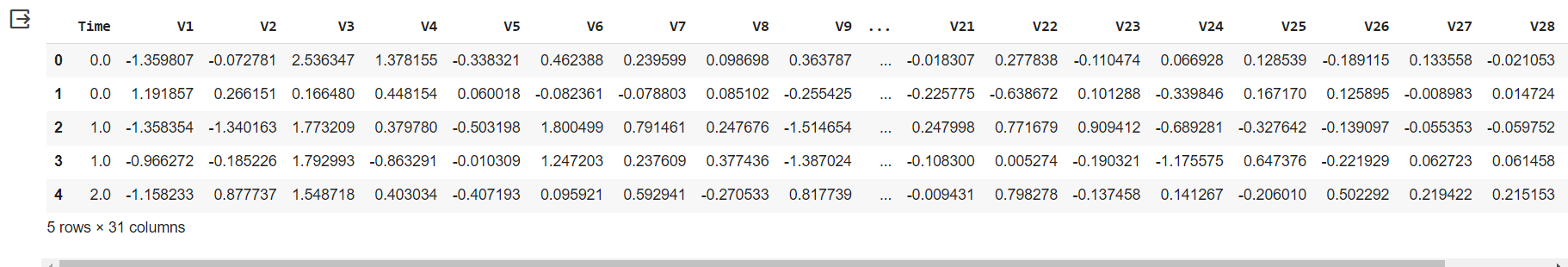
1. **Loading the Dataset:**

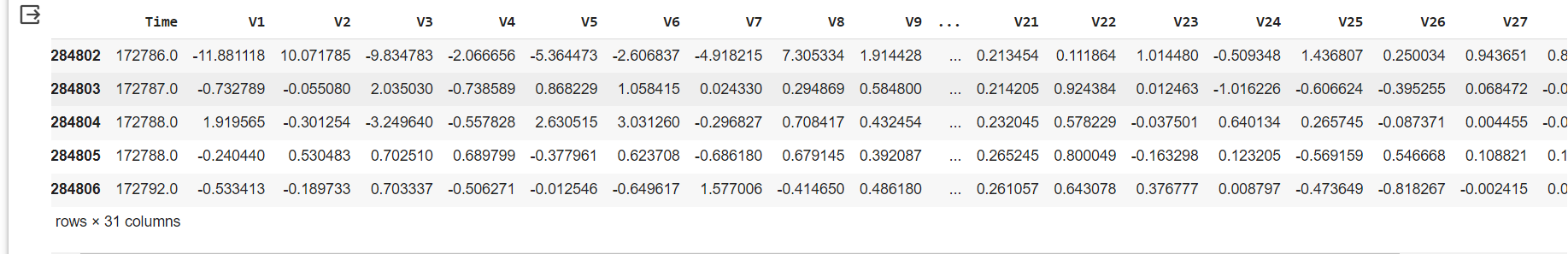
In this step we are going to import the dataset that is going to be used across this project.

* Load the credit card transaction data from a CSV file named 'creditcard.csv' located at '/content/creditcard.csv'.
* Use the ‘**pd.read\_csv’** function from Pandas to read the data.
* credit\_card\_data.head() is used to display the first few rows of the data set .By default, it displays the first 5 rows of the Data Frame.
* credit\_card\_data.tail() is used to display the last few rows of the dataset. Similar to head(), it displays the last 5 rows by default



**Output**

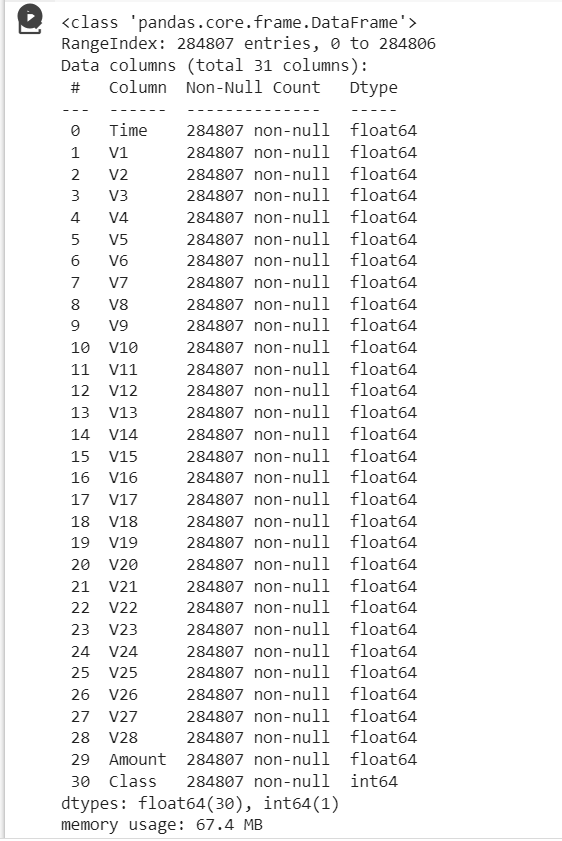




1. **Dataset Information:**

The credit\_card\_data.info() is used to obtain essential information about the dataset stored in the credit\_card\_data. It provides a concise summary of the dataset's characteristics.

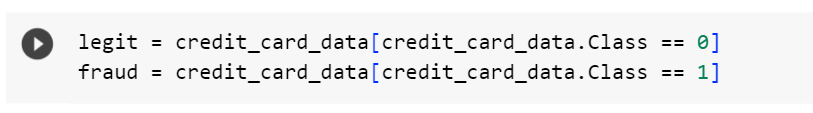




The output of credit \_card\_ data. Info () will give you a quick overview of the dataset's structure, including the data types of the columns, which can be crucial for understanding your data, checking for missing values, and ensuring that data types are appropriate for analysis. This information is valuable during the data preprocessing and data exploration phases.

1. **Separating the Data for Analysis:**

This step is done to separate the original credit card transaction dataset, credit\_card\_data, into two separate DataFrames based on the value of the 'Class' column. This column typically represents whether a transaction is legitimate (Class 0) or fraudulent (Class 1).



1. **Balancing the dataset:**

It is used to balance the dataset by selecting a subset of legitimate transactions to match the number of fraudulent transactions (that is 492 fraudulent cases). Balancing the dataset is essential in scenarios where one class is significantly underrepresented, such as in credit card fraud detection, to ensure that the model doesn't become biased towards the majority class.



1. **Checking for Missing Values:**

This step helps identify and handle missing values in the dataset. The code credit\_card\_data.isnull().sum() is used to count and display the number of missing (null) values in each column of the credit\_card\_data.

