**Explanation Document**

**Software Requirements:**

* Python 3
* Anaconda (Jupyter Notebook)

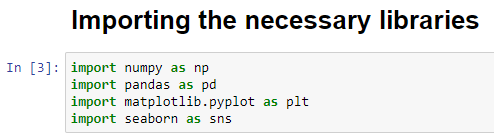
**Processing Steps**

**Step #1: Data Collection**

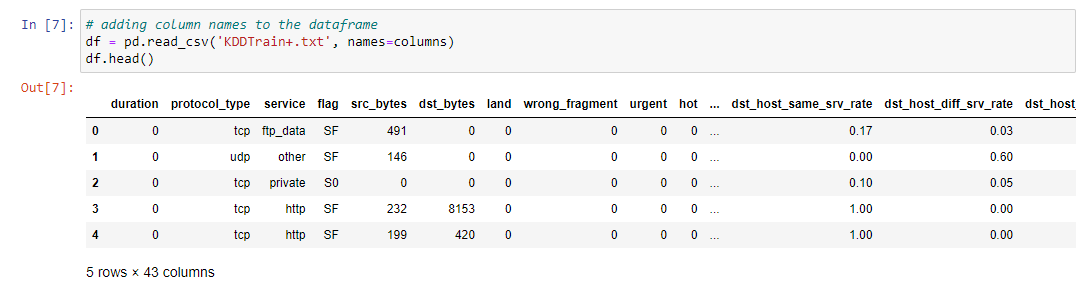
* Machines initially learn from the data that we give to them.
* This is the utmost importance to collect reliable data so that your machine learning model can find the correct patterns.
* The quality of the data that you feed to the machine will determine how accurate our model is.
* Data Collection can be done from various sources like kaggle, UCI etc.,
* NSL-KDD dataset: <https://www.kaggle.com/datasets/hassan06/nslkdd>

**Step #2: Data Preprocessing:**

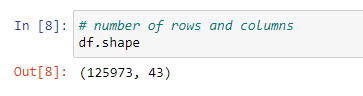
* Data preprocessing is a process of preparing the raw data and making it suitable for a machine learning model.
* It is the first and crucial step while creating a machine learning model.
* First, Importing the necessary libraries



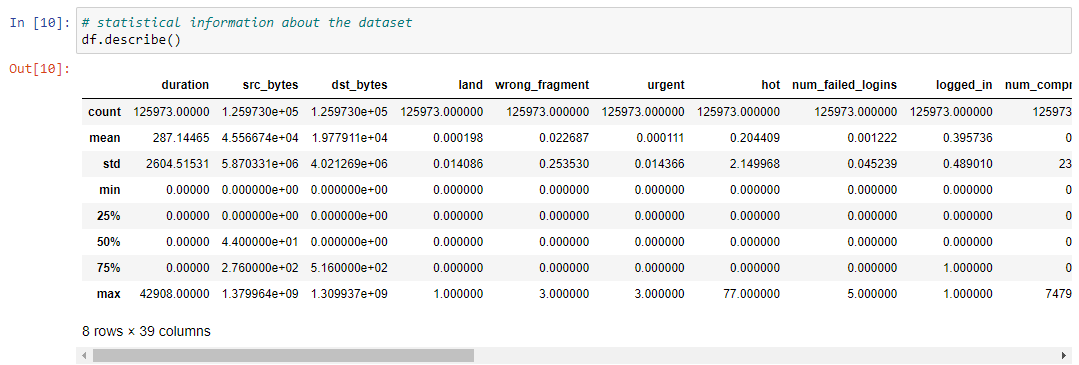
* Next, then Load the dataset as a DataFrame using the pandas library.



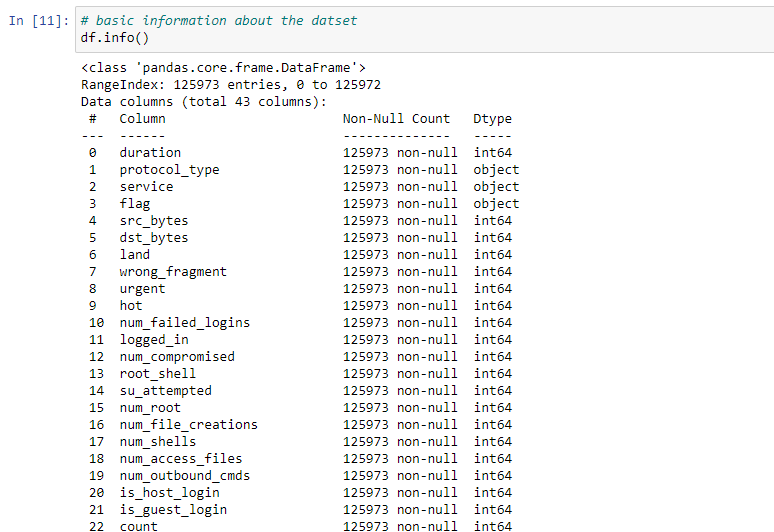
* We will leverage several Pandas features and properties to understand some basic information.
* ****.head()**** - This allow us to view an arbitrary number of rows (by default 5) from the beginning or end of the dataset.
* **.shape() -** This returns us a pair of numbers that represent the dimensionality(rows and columns) of our dataset.



* ****.describe() -** This** provides statistical information about the dataset.

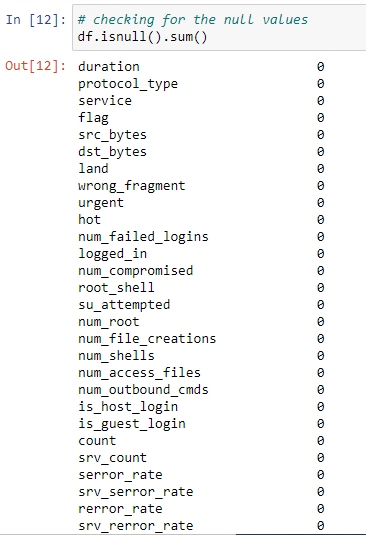


* ****.info()-** This gives** us information about the data type, non-null values and memory usage.



**Finding Missing Data**

The next step of data preprocessing is to handle missing data in the datasets. If our dataset contains some missing data, then it may create a huge problem for our machine learning model.

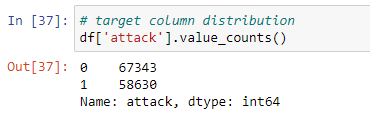


We don’t have null values in our dataset.

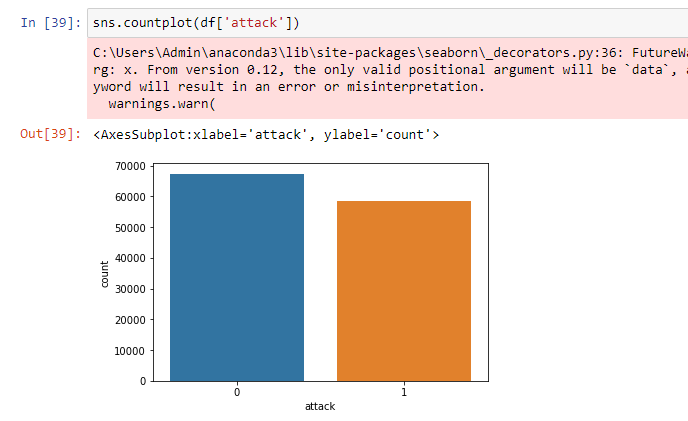
**Categorical Variables**

****.value\_counts() -** This** is one of the most important functions to understand how many values of a given variable there are in our dataset.

Let’s do it with the target variable,

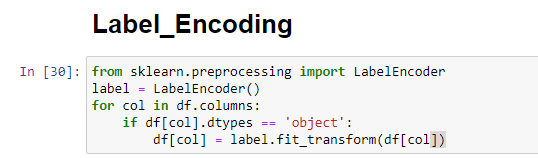


Visualizing the above one, with the help of countplot(),



**Label Encoding**

* In machine learning, we usually deal with datasets that contain multiple labels in one or more than one columns.
* These labels can be in the form of words or numbers.
* **Label Encoding** refers to converting the labels into a numeric form so as to convert them into the machine-readable form.

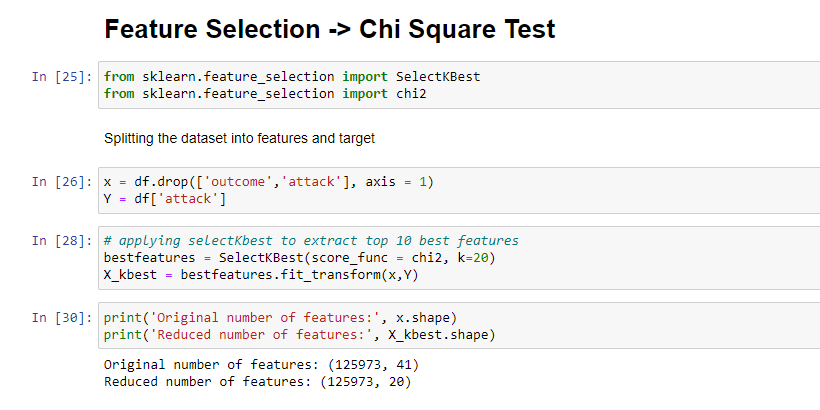


**Feature Selection**

* ****Feature selection**** is the process of reducing the number of input variables when developing a predictive model, to both reduce the computational cost of modeling and, in some cases, to improve the performance of the model.

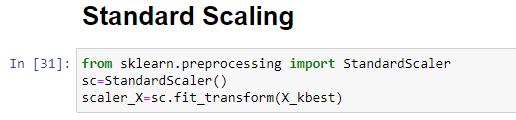
**Feature Selection using Chi-Square Test**

* Chi-square test is used for categorical features in a dataset.
* We calculate Chi-square between each feature and the target and select the desired number of features with best Chi-square scores.



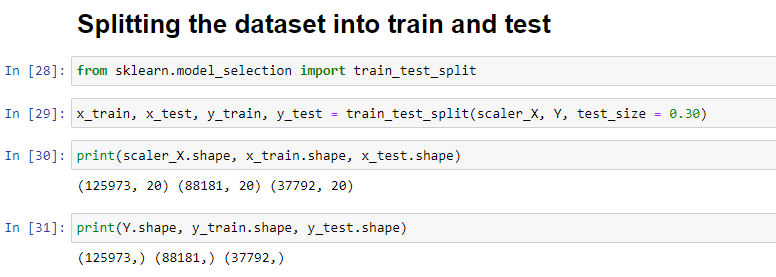
**Standard Scaling**

**Standard Scaler** follows **Standard Normal Distribution (SND)**. Therefore, it makes mean = 0 and scales the data to unit variance.



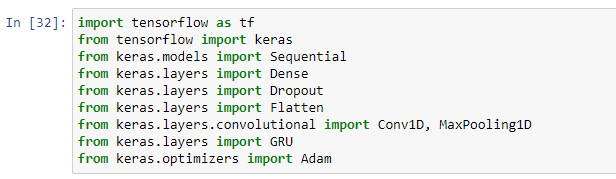
**Splitting the data into Train and Test**

Train and Test datasets are the two key concepts of machine learning, where the training dataset is used to fit the model, and the test dataset is used to evaluate the model.

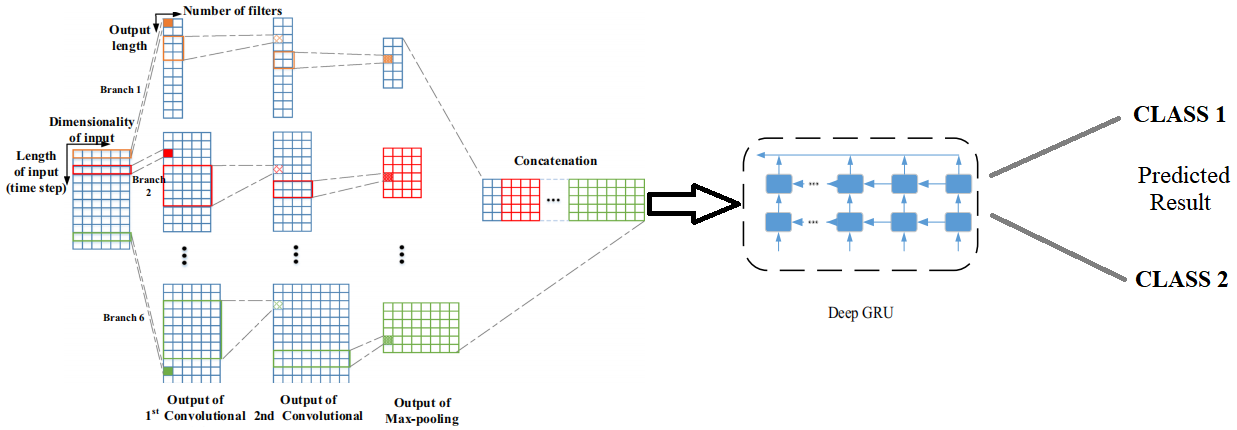


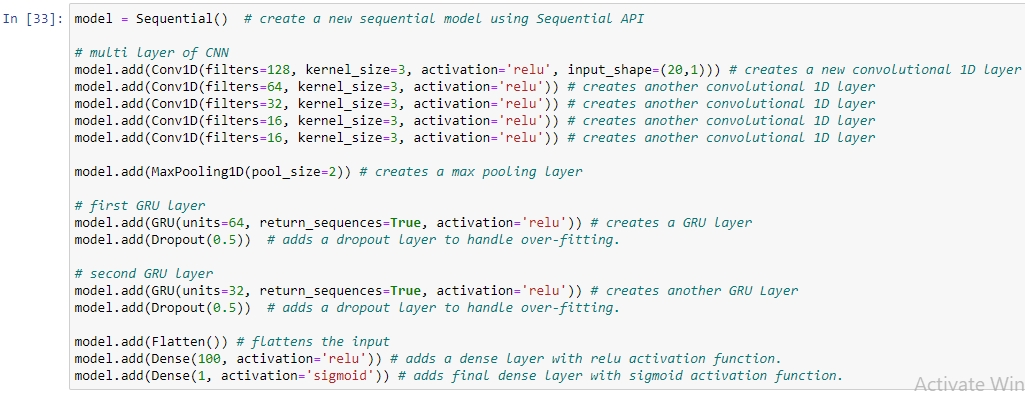
* In the first line of the above code, we have imported the train\_test\_split function from the ****sklearn**** library.
* In the second line, we have used four variables, which are
* **x\_train**: It is used to represent features for the training data
* **x\_test**: It is used to represent features for testing data
* **y\_train**: It is used to represent dependent variables for training data
* **y\_test**: It is used to represent independent variable for testing data
* In the **train\_test\_split()** function, we have passed four parameters. Which first two are for arrays of data, and test\_size is for specifying the size of the test set.
* The last parameter, **random\_state** is used to set a seed for a random generator so that you always get the same result.

**Importing the required libraries for model building**



**Model building using Multi-scale Convolutional Gated Recurrent Unit network (MCGRU)**





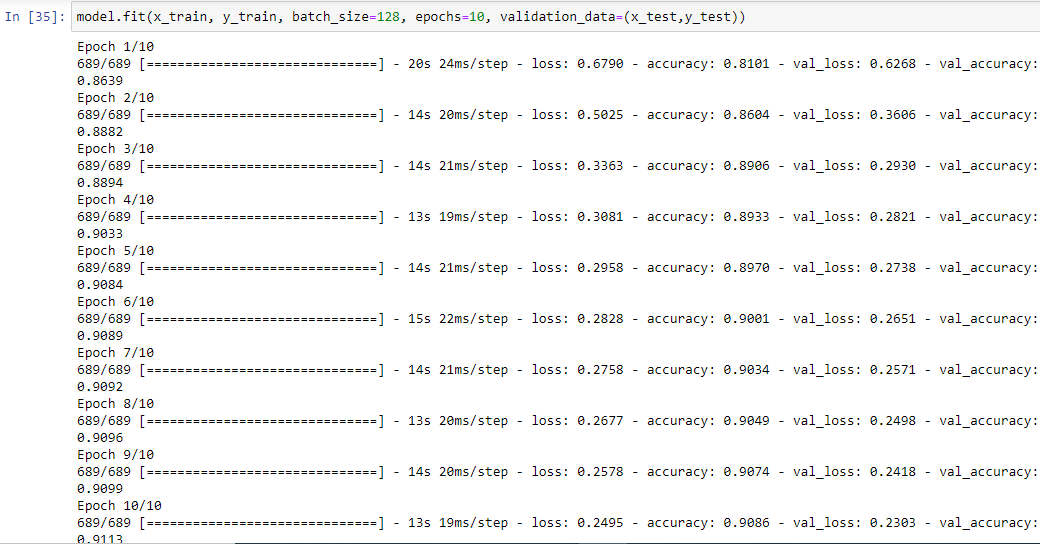
**Model Compilation**

Compiling the model using selected loss function, optimizer and metrics.



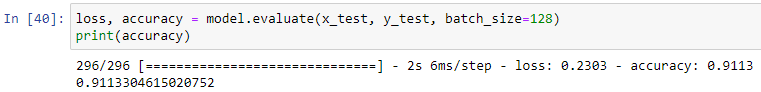
**Training the model**

Training our model using the fit() method



## Model Evaluation

Evaluation is a process during development of the model to check whether the model is best fit for the given problem and corresponding data.



**Note:** In next update, optimization will be added.