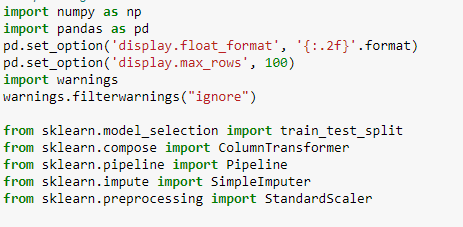
**ML Loan Default Prediction:**

# Created by Kaushik Kar

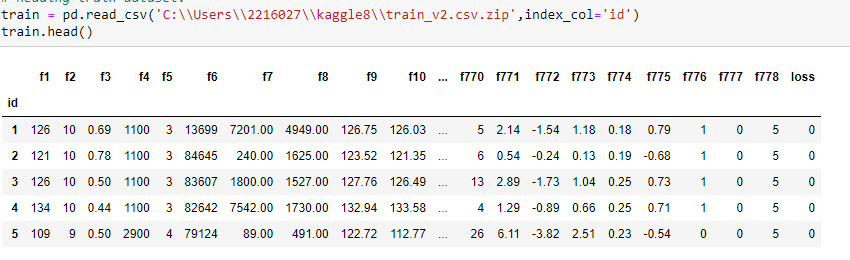
# Employment id- 2216027

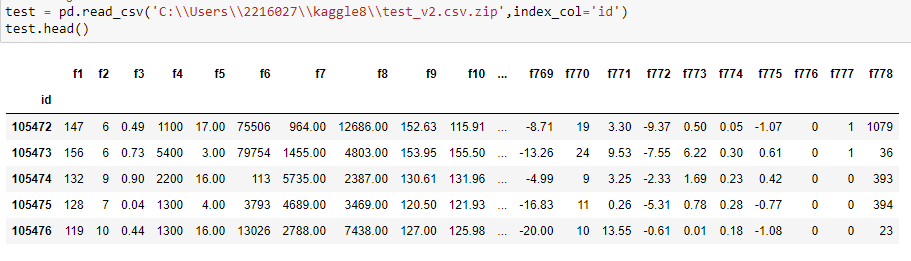
# Importing Libraries:

First, importing the important external Python packages using the pip package manager.

****

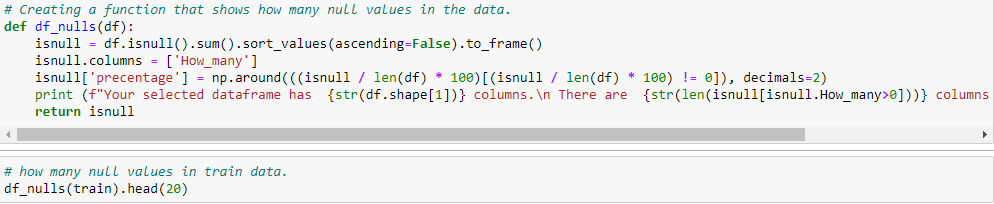
1. NumPy is used for mathematical operations like addition, subtraction, multiplication, division, etc. on arrays and matrices.
2. Pandas provides data structures for efficiently storing and manipulating large datasets, and tools for reading and writing data to and from various file formats, including CSV, Excel, and SQL databases.
3. Imports the warnings module uses its filterwarnings() function to ignore all warnings that may be raised in the code.
4. train\_test\_split is used to split the data into training and testing sets.
5. ColumnTransformer, Pipeline, SimpleImputer, and StandardScaler are used for data preprocessing.
6. **Upload the data and display:**





With the help of pandas library, we can read and upload the data in csv form. we can display the first five rows of the data

**Null Values:** isnull() is the process by which we can check whether there is any null value present in the data or not i.e. isnull() is part of the data cleaning stage. Here in the data, we found that Your selected dataframe has 770 columns.There are 525 columns that have missing values. Your selected data frame has 769 columns. There are 530 columns that have missing values.

****

****

Getting the name of the numeric columns

****

**Then choosing only the numeric columns from the test and train data and then drop the rows with null values from the loss column and then apply train test and split .**

The numerical transformer, defined by the variable num\_trans, is a pipeline that performs two steps on the numerical columns of the dataset: imputation and scaling. The SimpleImputer step replaces missing values in the numerical columns with the mean value of the corresponding column, while the StandardScaler step scales the numerical features to have zero mean and unit variance.

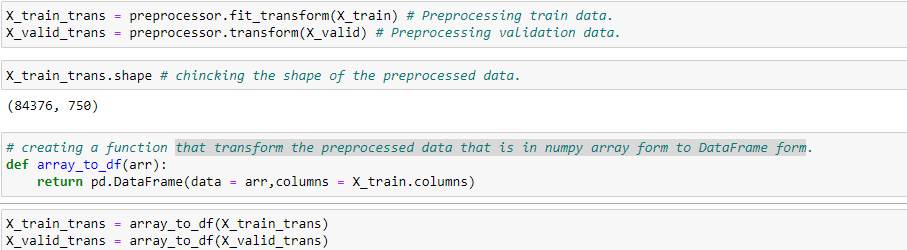
The num\_columns variable is a list of column names that are numerical features in the dataset, and the code removes the 'loss' column from this list, as it is not a numerical feature.

The preprocessor combines the numerical transformer with the column transformer by specifying the transformers argument as a list of tuples, where each tuple contains a transformer and a list of column names to apply the transformer to. In this case, there is only one tuple, with the numerical transformer and the list of numerical column names.

The remainder argument is set to "drop", which means that any columns not explicitly listed in the transformers argument will be dropped from the dataset.

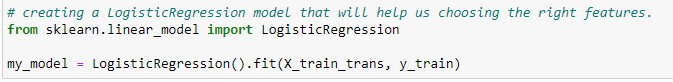
Overall, this preprocessor can be used as a preprocessing step for a machine learning model that requires numerical features to be imputed and scaled**.**

**Then preprocessing the training and test data and creating a function** **that transform the preprocessed data that is in numpy array form to DataFrame form.**

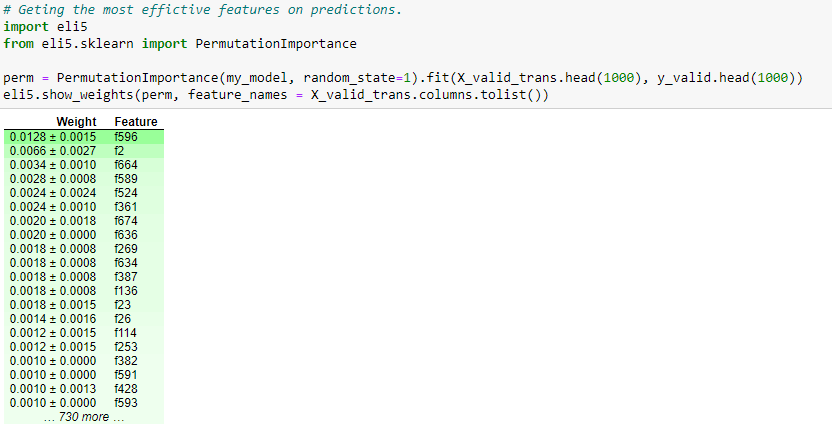
****

**Model:**

Logistic Regression is a type of classification algorithm that is used to predict binary outcomes (i.e., outcomes that have only two possible values, such as 0 or 1). It works by fitting a logistic function to a set of input variables, which maps the input variables to a probability between 0 and 1. This probability represents the likelihood that a particular data point belongs to the positive class (1) or the negative class (0).

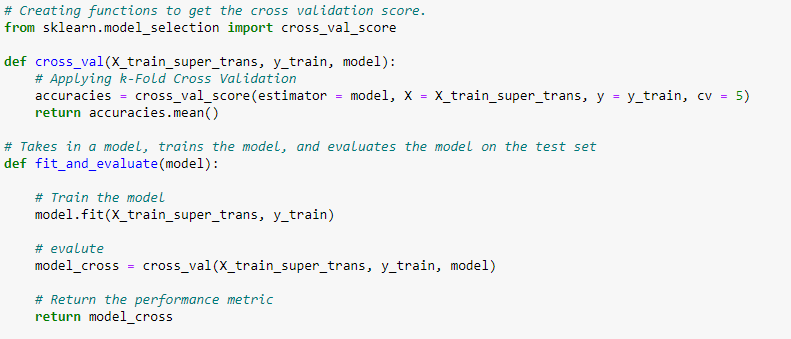


The Permutation Importance function from the eli5 library to calculate the feature importance of a machine learning model (my\_model) using permutation importance.



this code can be used to identify the most important features in a machine learning model, which can help with feature selection and understanding the underlying factors that drive the model's predictions. However, it is important to note that feature importance scores should be interpreted with caution and in the context of the specific dataset and model being used.

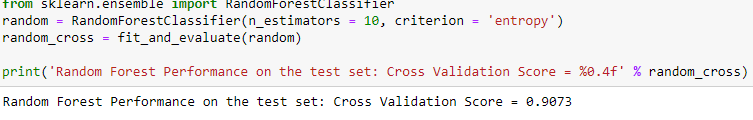
**Cross validation score:**



These functions can be used to streamline the process of training and evaluating a machine learning model using cross-validation, making it easier to compare the performance of different models and select the best one. However, it's important to keep in mind that the performance of a model on the training dataset may not generalize well to unseen data, so it's important to also evaluate the model on a separate test dataset before deploying it in a real-world setting.

**Random Forest Classification:**

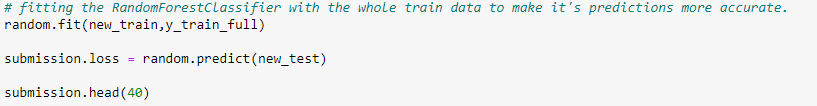
Random Forest Classification is a popular machine learning algorithm used for classification tasks. It is an ensemble learning method that combines multiple decision trees to make predictions. Each decision tree in the random forest is trained on a different subset of the training data and a different subset of the features.

This code trains a Random Forest Classification model using the RandomForestClassifier class from scikit-learn, with 10 decision trees and the entropy criterion for splitting. The fit\_and\_evaluate function is then used to train and evaluate the model on the preprocessed training data (X\_train\_super\_trans and y\_train) using cross-validation. The resulting cross-validation score is stored in the random\_cross variable.

Finally, the code prints the cross-validation score of the model on the training dataset using the print function.

Preprocessing and filtering the whole train and test data after checking its Cross Validation Score.

Fitting the RandomForestClassifier with the whole train data to make it's predictions more accurate.



This code fits the RandomForestClassifier model (random) to the entire preprocessed training data (new\_train and y\_train\_full). The goal is to obtain more accurate predictions on the test dataset.After fitting the model, the code uses the predict method to make predictions on the preprocessed test data (new\_test) and assigns the predicted loss values to the submission.loss column.

Finally, the code displays the first 40 rows of the submission DataFrame, which now includes the predicted loss values for the test dataset.