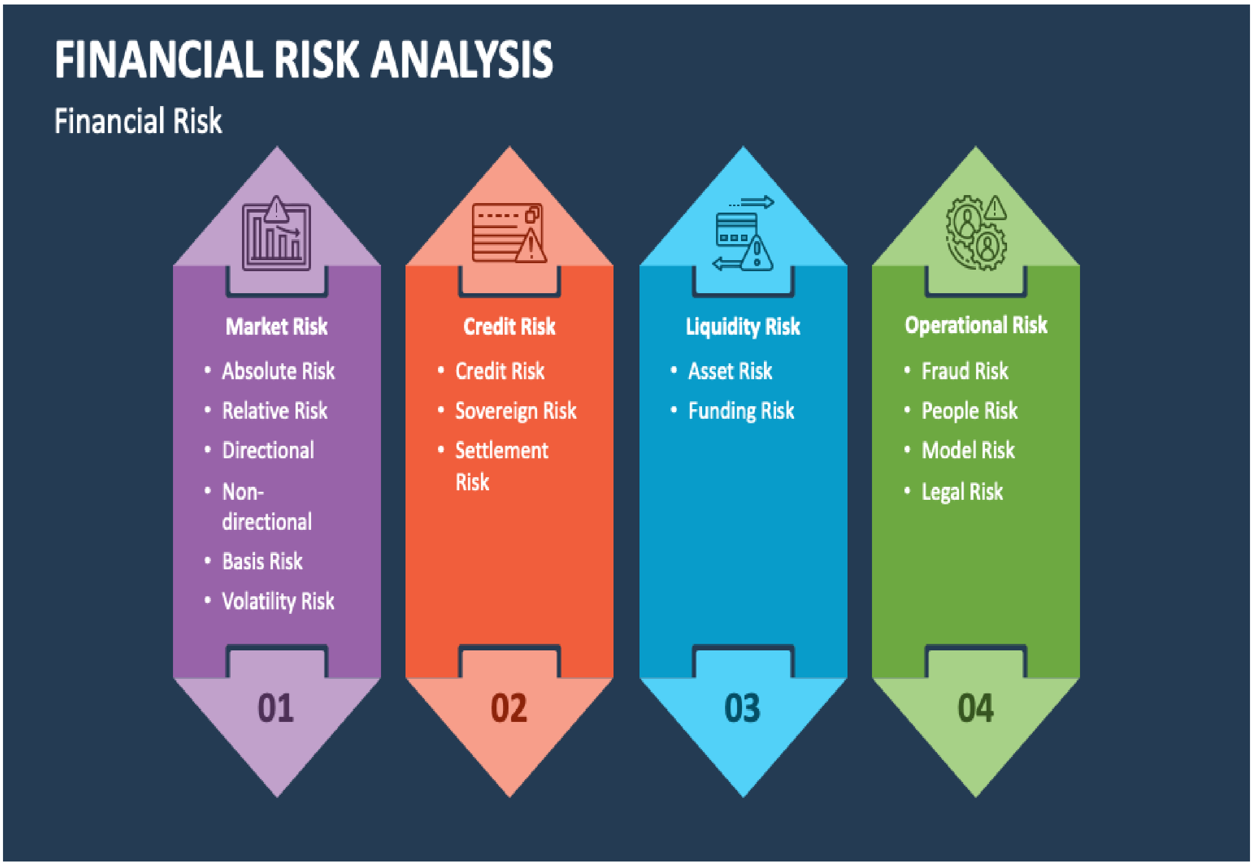
SISTER NIVEDITA UNIVERISTY

Name: Susanta Dhurua
Reg: 1811108010005
Topic: Credit Risk Analysis



FINANCIAL RISK ANALYSIS

Financial Risk

Market Risk

- Absolute Risk
- Relative Risk
- Directional
- Non-directional
- Basis Risk
- Volatility Risk

01

Credit Risk

- Credit Risk
- Sovereign Risk
- Settlement Risk

02

Liquidity Risk

- Asset Risk
- Funding Risk

03

Operational Risk

- Fraud Risk
- People Risk
- Model Risk
- Legal Risk

04

Financial Analysis Report

Abstract:

Credit risk analysis is a vital aspect of financial institutions' operations, aiming to assess the likelihood of borrowers defaulting on their loan obligations. This project focuses on performing credit risk analysis using logistic regression in Python. Logistic regression is a widely used statistical technique for binary classification problems, making it well-suited for predicting credit default events.
The logistic regression model is then trained on the prepared dataset to predict the probability of default. The model is evaluated using appropriate performance metrics, including accuracy, precision, recall, and area under the receiver operating characteristic curve (AUC-ROC). Furthermore, feature importance analysis is conducted to identify the most influential factors affecting credit risk.
The project demonstrates the practical application of logistic regression in credit risk analysis, providing insights into the factors contributing to credit default events. The outcomes of this analysis can assist financial institutions in making informed decisions, optimizing credit risk management strategies, and improving loan portfolio performance.
Overall, this project done in Python. Logistic regression in credit risk analysis, offering valuable insights into the assessment and management of credit risk in the financial industry.

Introduction:

The word "Finance" refers to Money. Money refer to three things in the real world, the three things is Cash, Debt and Equity. We all are familiar with the word but we have not clear image when we use the word Debt and Equity. To fully understand the concepts of Debt and Equity we could took a situation to understand.

Debt situation (conversations between Bank Manager and the Customer Ram)

BM: Hello sir, Good morning
Ram: Good morning
BM: How could I help you?
Ram: I want a Housing Loan.
BM; No problem sir could I have your monthly employer slip
Ram: Yeah Of course
BM: Yes sir you are eligible for the loan, before the proceedings of loan i should inform you that the loan will give in the interest rate of 8.05%-8.55% are you comfortable to proceed further.
Ram: Yeah I am aware of that we can proceed further.
BM: ok sir

From the above conversation between bank manager and the customer ram is
Ram want a housing loan to build a dream house for his future generation and he asked to the bank to pay the loan to him. Bank agreed to give him the loan at 8.05% having some collateral from ram. Bank gives a specific time period at that particular time ram have to repay his loan with 8.05% interest rate.

Debt is refer to an amount of money borrowed by one party from another, with the agreement to repay the borrowed amount along with any applicable interest within a specific period of time.

Equity situation (conversations between Ram (founder of company) and his friend Shyam)

Ram founded company namely ABC he invested all his income to start the business and the business run good and he made profit. By the time he is the only decision makers in his company because 100% equity is hold by him. And like the time grows he have an idea but to bring that idea into the market he needs investments and suddenly he met his friends in the auction party.

Ram: Hey Shayam, how are you?
Shayam: hello ram I am good what about you?
Ram: I am fine.
Shayam: Hey I heard about you new product lunch, it's very awesome.
Ram: Thanks bro
Shayam: I am very happy for you
Ram: Thanks, by the way which job are you doing these days?
Shayam: I am working as an investment banker.
Ram: That's great to hear, I have one plan have you some time to listen?
Shayam: Yak tell me
Ram: So this is the plan " "and I need investors who will invest in the plan have you know anyone?
Shayam: The plan sounds very nice, if you don't mind then I will invest in your plans.
Ram: Yak that's fine, so for 10% equity you should invest $50000.
Shayam: Ok that's fine.

From the above conversation between two friends we have a clear idea of what' the situation is. Situation is very clear as the time grows Ram business also grown up and to support the innovative ideas the founders decided to sell a 10% equity stake to an investors to raise a additional capital support.
As he gives 10% equity, now the equity ownership is distributed as

- Founder: 90% equity ownership (remaining 90% after selling 10% equity)
- Investor: 10% equity ownership (purchased 10% from the founder)

Equity refers to the ownership interest or stake that an individual or entity holds in a company or property. It represents the residual value of an asset after deducting liabilities. In simple words, it is the portion of the asset's value that belongs to the owner.

While we are talking about the finance then the Bank will come to the picture. The life cycle of bank is very simple it takes money as deposit and give as loan to the borrower.

Have a situation is that two people deposit their money and third person asked for the loan and bank agreed to give the loan having some collateral, by the end of time if the borrower didn't repay the loan then bank will sell the collateral and fill the treasures. In bank there is two types one is depositor and second is borrower.

While the bank provide loan to an individual then there should be a "question of risk?" rised what if the borrowed might not able to pay the loan, this type of risk is called credit risk.
And our main motto of our project is Credit risk analysis.

There are two other types of risk is there:
Liquidity risk: is there enough liquid cash available?
Operational risk: What is someone rob the banks?

So, our main focus is to do the credit risk analysis for which we have collected the data, which will be helpful throughout the analysis and finally we will conclude that in which condition loan are repaid and not.

Dataset Description:

The dataset contains 10000 rows and 4 columns. The columns namely default, student, balance and income.
Description of the dataset is

- Default: Whether the person is able to repay the loan or not. It has a binary class (Yes or No). If the default is No then it signifies that the person is not defaulter (he/she will repay their loan) and the vice versa.
- Student: It contains whether the person is student or not.
- Balance: it is the outstanding balance, outstanding balance refers to the amount of money that remains unpaid on loan, credit card or any form of credit. In simple words the column balance shows the unpaid balance that has to be paid a lender.
- Income: It shows the total income of the persons.

(Note: All the balance, income are in dollars)

The data related to the credit risk, so we are doing the credit risk analysis, it has the data related to different borrower and we are try to understand that this borrower would be a defaulter or not.

In simple words, all the customer would be pay back their loan or not.

Exploratory Data Analysis:

Under exploratory data analysis we have done the following things:

- Data cleaning and Preprocessing
- Data Visualization
- Summary Statistics
- Feature Engineering

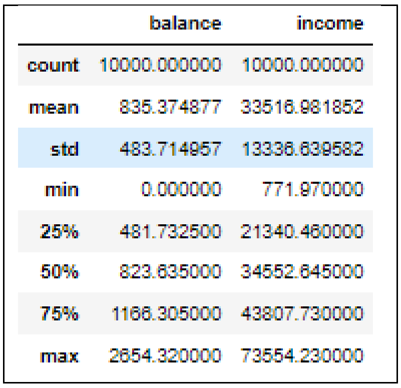
Giving the full focus on the EDA terms

- Data cleaning and Preprocessing: In the data cleaning and preprocessing we have identify and handling the missing data, outliers or inconsistencies in the dataset.

To make our data readable in python we have import some of the library that are required.
The required libraries are: Pandas, Numpy, Matplotlib.pyplot and Seaborn

We have check the data shape of our dataset, and the shape of our dataset is **10000**, 4
This means **10000** rows and 4 columns.

In our dataset there are 2 categorical columns and 2 numerical columns are there. So to get a rough idea of the two numerical columns we have applied describe function that will describe the outer information.



	balance	income
count	10000.000000	10000.000000
mean	835.374877	33516.981852
std	483.714957	13336.639582
min	0.000000	771.970000
25%	481.732500	21340.460000
50%	823.635000	34552.645000
75%	1166.305000	43807.730000
max	2654.320000	73554.230000

From the data description we have the knowledge of the mean, std, Min, the quartile function (25%, 50% and 75%) and the max.

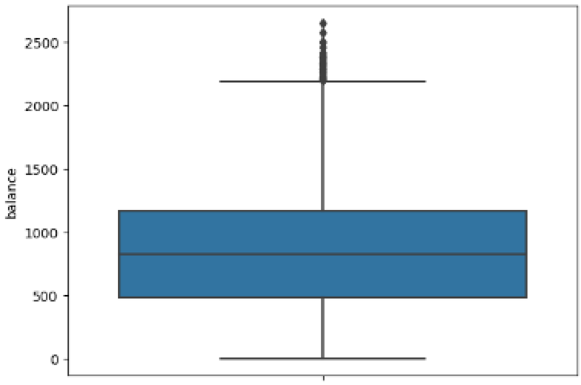
These information will help in detecting the outliers.

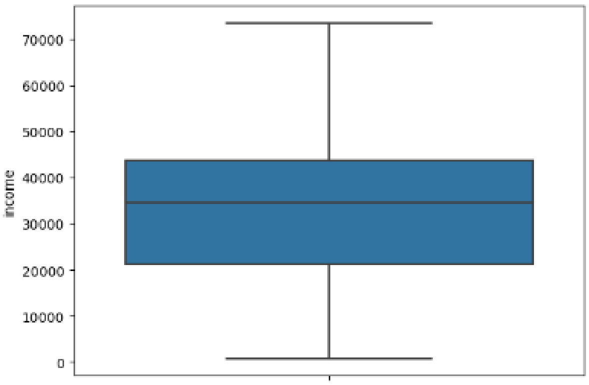
To identify the outliers (means the data points), we have perform the boxplot. Before we go to box plot output we should know about the boxplot.

Boxplot: Boxplot is also known as box and whisker plot, is a graphical representation of the distribution of a dataset. It display key statistical measures and provides a visual summary of the data's central tendency, spread, and presence of outliers. The components of a boxplot are Median (Q2), Interquartile range (IQR), whiskers and outliers.

As we know that boxplot are typically used to visualize the distribution of continuous numerical variables.

According to our data, there are 4 variables and from the 4 variables 2 variables are categorical and 2 are continuous numerical variables namely balance and income.





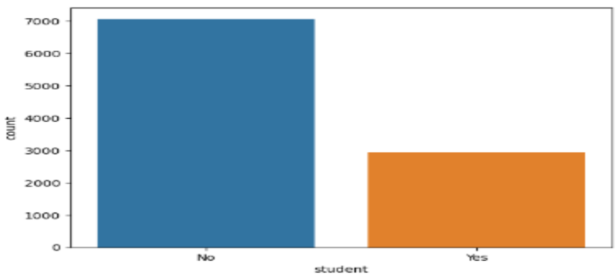
Interpretation:

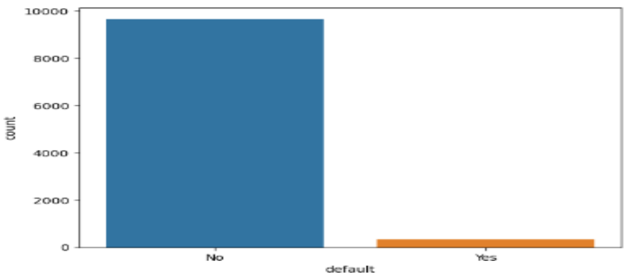
From the visualization of balance boxplot we can signifies that the minimum balance is zero and maximum balance is above 2500.
Some of the outliers are present in our balance data which is clearly shown the points which are above the 100 percentile.

From the visualization of income boxplot we could signifies that the minimum income is zero and maximum income is above 7000.
From the visualization we could clearly see that there is no outliers present in the graph.

According to the dataset there are 2 categorical variables namely the student and default attributes we perform count plot to count the observations.

Count plot is a type of graphical representation used to display the count or frequency of observations in different categories of a categorical variable. It is particularly useful for understanding the distribution and relative frequency of each category within a dataset.





Interpretation:

From the visualization of count plot of student attributes, we could summarize the yes and no. The count of yes is 2944 and no is 7056.

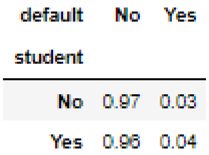
Similarly from the visualization of count plot of default attributes, we could summarize the yes and no. The count of yes is 333 and no is 9667.

We could get the percentage of the respective two attributes, we have two attributes namely student and default.

For the attributes student: Yes 0.2944 (29% are yes) and No 0.7056 (70% are no)
For the attributes default Yes 0.0333 (3.33% are yes) and No 0.9667 (96% are no)

To visualize it very clearly we could perform the crosstabs. Before we perform the crosstabs we should know about crosstabs.

A crosstab is a table or matrix that displays the relationship between two or more categorical variables. It provides a summary of the distribution of data by showing the frequency or count of observations that fall into each combination of categories.



default	No	Yes
student		
No	0.97	0.03
Yes	0.96	0.04

Output: If the persons is not a student's then 97% chance is there that he/she is not a defaulter.
If the person is a student's then 4% chance is there that he/she is a defaulter.

Confirming that there is a presence of outlier in a balance attributes. We move forward to the next element of data cleaning and preprocessing.

Checking the presence of missing values or not.

Very clearly, in our dataset there are no single missing values present.

Successfully performing all the elements of data cleaning and preprocessing, we have the most important question arrived in middle of our analysis is that "how could we treat the outliers?"

There are several methods from which we can treat the outliers namely imputation, transformation, trimming and model based techniques. Before moving forward we should about the above mentioned terms.

Imputation: in most of the cases, generally we replace outliers with more reasonable values. Imputation techniques such as mean, median or mode substitution can be used to replace outliers with central tendencies of the data.

Transformation: it generally used the mathematical transformations to the data to reduce the impact of outliers. By apply positively skewed data, logarithmic or square root transformation can help normalize the distribution and mitigate the effect of outliers.

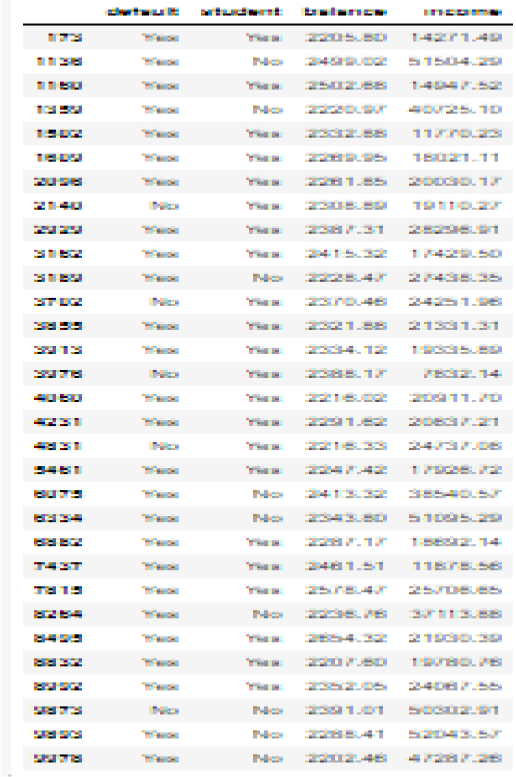
Trimming: it basically removing the outliers altogether from the datasets.

Model based techniques: it is used when the impact of outliers needs to be minimized. This techniques contain robust statistical methods or outlier detection algorithms.

In our case, one solution arrived that we can use trimming. But whether the trimming is applicable in this case. The case where the count of yes in target variables is 333 and from the 333 we have 31 outliers and from this we could summarize that approx. 10% of valuable information is hold by outliers, so it's not a good idea to drop the outliers.

There is method which can bring the outliers in the 100 percentile by using interquartile range.
Interquartile range mathematical formula is Q3-Q1
Where the Q1 is 25% and Q3 is 75%
The formula for finding the lower and upper limit is
LL: Q1-1.5*(IQR)
UL: Q3+1.5*(IQR)
We have to work with the upper limit so, the values came for the UL is 2193.16375

After we got the value of UL we set the values of outliers present in the balance attributes greater than the upper limit.



	default	student	balance	income
173	Yes	Yes	2205.80	14271.49
1136	Yes	No	2499.02	51504.29
1160	Yes	Yes	2502.68	14947.52
1359	Yes	No	2220.97	40725.10
1502	Yes	Yes	2332.88	11770.23
1609	Yes	Yes	2269.95	18021.11
2098	Yes	Yes	2261.85	20030.17
2140	No	Yes	2308.89	19110.27
2929	Yes	Yes	2387.31	28296.91
3162	Yes	Yes	2415.32	17429.50
3189	Yes	No	2228.47	27438.35
3702	No	Yes	2370.46	24251.96
3859	Yes	Yes	2321.88	21331.31
3913	Yes	Yes	2334.12	19335.89
3976	No	Yes	2388.17	7832.14
4060	Yes	Yes	2216.02	20911.70
4231	Yes	Yes	2291.62	20637.21
4831	No	Yes	2216.33	24737.06
5461	Yes	Yes	2247.42	17926.72
6075	Yes	No	2413.32	38540.57
6334	Yes	No	2343.80	51095.29
6862	Yes	Yes	2287.17	18692.14
7437	Yes	Yes	2461.51	11678.56
7815	Yes	Yes	2578.47	25706.65
8264	Yes	No	2238.76	37113.88
8425	Yes	Yes	2654.32	21930.39
8832	Yes	Yes	2207.60	19780.76
8952	Yes	Yes	2352.05	24087.55
9873	No	No	2391.01	50302.91
9893	Yes	No	2285.41	52043.57
9976	Yes	No	2202.46	47287.26

From the beside image we could visualize the values of outliers are change to the values more than the UL.

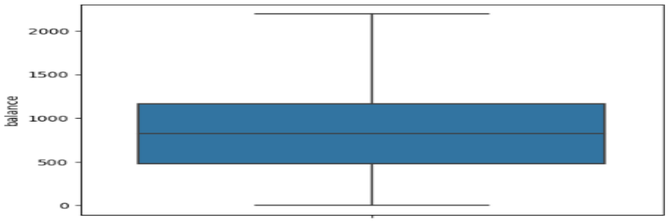
And from the curious we want to check how many yes and no is there from 31 outliers.

From the outliers there is 26 Yes and 5 No is there which means there is 83.871% of Yes is present and 16.129% of No is there

Finally we treat the outliers by apply a condition that if the balance data value is greater than UL then make it same as UL and keep the other data values same.

- Data Visualizations: In the data visualizations we create a visual representations of the data to gain a better understanding of its distributions, patterns and trends.

Successfully fixed the outliers we could visualize it with the box plot.



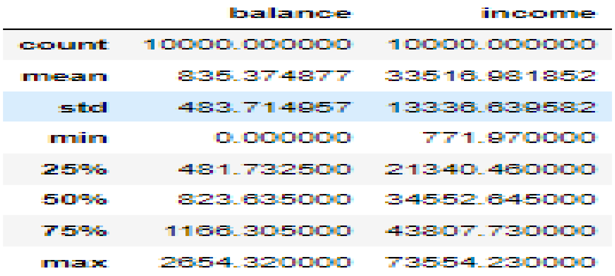
Interpretation: from the boxplot we are failed to find the outliers, now we could confirmed that in the balance attributes there is no outliers present.

- Summary Statistics: Computing descriptive statistics such as mean, median, mode, variance, or correlation coefficients to summarize the data's central tendency and variability.

The summary statistics of our dataset is:

Mean: 835.374877
Median: 823.635
Std: 483.714957

Mean: 33516.98
Median: 34552.64
Std: 13336.63



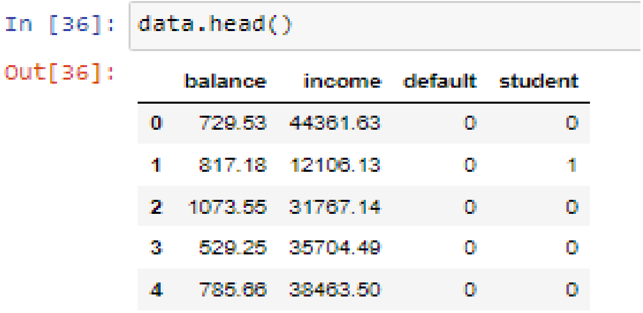
	balance	income
count	10000.000000	10000.000000
mean	835.374877	33516.981852
std	483.714957	13336.639582
min	0.000000	771.970000
25%	481.732500	21340.460000
50%	823.635000	34552.645000
75%	1166.305000	43807.730000
max	2654.320000	73554.230000

- Feature Engineering: creating new features or transforming existing features to improve the dataset's representation or performance in subsequent analysis. Identifying relevant attributes that might be useful for prediction or modelling tasks.

According to our dataset we have four columns and 4 columns are equally important to take our analysis forward. But one important question is arrived that "two of the columns is in categorical" here we could possibly do one transformation. The transformation technique called binary encoding or binary mapping.
Before we could move forward with the transformation we should know little bit about the Binary mapping.

When we have the situation of categorical variable yes or no and we have to transform it into the numerical form as 0 and 1. This transformation is called the Binary mapping. We are converting the categorical variables into the numerical form, because we have to perform various operations and analyses that require numeric input. This conversion is commonly used when dealing with binary or Boolean variables, where the presence of a certain condition or attributes is denoted by 1, and the absence is denoted by 0.

The same techniques concept is applied in our dataset, where the "yes" is represented by 1 and "no" is represented by 0. Here the sample of our transformation.



```
In [36]: data.head()
Out[36]:
```

	balance	income	default	student
0	729.53	44361.63	0	0
1	817.18	12106.13	0	1
2	1073.55	31767.14	0	0
3	529.25	35704.49	0	0
4	785.66	38463.50	0	0

From the table we have saw the binary mapping of the categorical variables.

Here Yes is represented by 1
No is represented by 0

Model Building:

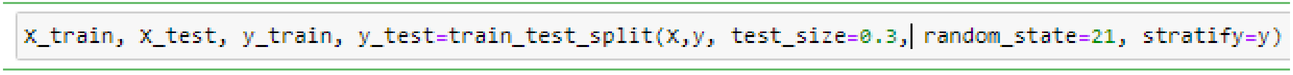
Before we build our model we have split our dataset into the training set and testing set. The purpose of the division is to evaluate the performance and generalization ability of a model on unseen data. Before we move forward to our splitting we should look a glance of training set and testing set.

Training set: the training set is a subset of the original dataset that is used to train a machine learning model. It contains a known set of input data (features) and their corresponding target values (labels or outputs).

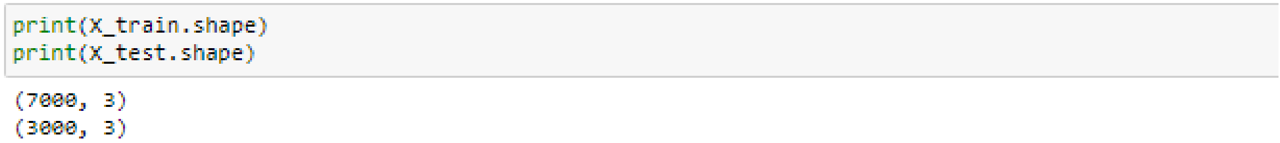
Testing set: The testing set is also known as the validation set or holdout set, is another subset of the original dataset that is used to evaluate the performance of the trained model. It contains data that the model has not seen during the training process.

From the required library sklearn.model_selection we import train_test_split.
We have create x and y where x contains all the attributes except the target variables (default), and the y contains only the target variables (default).

The code to split the dataset is



```
x_train, x_test, y_train, y_test=train_test_split(x,y, test_size=0.3, random_state=21, stratify=y)
```



```
print(x_train.shape)
print(x_test.shape)
```

```
(7000, 3)
(3000, 3)
```

Working of the above code is:
Here x represents the independent variables or features of the dataset, and y represents the dependent variable or target variable that we want to predict or classify.

The following parameters are in the train_test_split are

test_size=0.3: specifies the proportion of the dataset that should be allocated to the testing set. In this case 0.3 means 30% of the data will be assigned to the testing set, while the remaining 70% will be used for training.

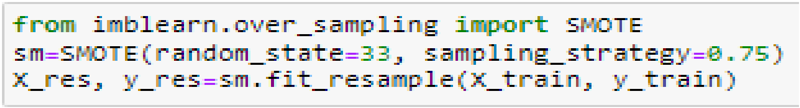
random_state=21: It ensures that the data split will be the same every time the code is executed with the same random_state value.

stratify=y: stratified sampling ensures that the class distribution in the original dataset is maintained in both the training and testing sets.

The splitting between the two set is 70% data go to training set and remaining 30% data go to testing set.

And using shape we are able to know the rows and columns it contains.

As we know that we our data is imbalanced so we used one machine learning techniques to make our data balanced.
From imblearn.over_sampling we import SMOTE and after that we write the code.



```
from imblearn.over_sampling import SMOTE
sm=SMOTE(random_state=33, sampling_strategy=0.75)
x_res, y_res=sm.fit_resample(x_train, y_train)
```

Working of the above code is:
We import the SMOTE from the following library. The following parameters are in the SMOTE are random_state=33 the parameters sets a seed to ensure reproducibility, and the sampling_strategy=0.75, a ratio of 0.75 means that the number of samples in the minority class will be increased to 75% of the number of samples in the majority class.

X_res, y_res=sm.fit_sample(x_train, y_train): the fit_samples method oversamples the minority class using SMOTE techniques, creating synthetic samples to balance the class distribution.

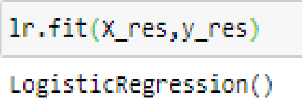
Now from the sklearn.linear_model we will import Logistic Regression.

Before we move forward we should took a glance towards the Logistic Regression

Logistic Regression: Logistic regression is a statistical modelling techniques used for binary classification problems. It widely employed to predict the probability of an event to classify data into two distinct classes based on input variables or features. The logistic regression model uses a logistic or sigmoid function to map their linear combination of the input features to the predicted probability.

Logistic regression is commonly used in credit risk analysis. Credit risk analysis involves assessing the probability of default or the likelihood that a borrower will fail to repay a loan or credit obligation.

We have successfully fit the logistic regression to the training data. The code is

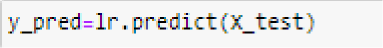


```
lr.fit(X_res,y_res)

LogisticRegression()
```

When the code get executed, its fits the logistic regression model to the training data (x_res, y_res). This process involves adjusting the model's internal parameters based on the provided training data to find the best-fit line or decision boundary that separates the two classes.

After successfully fit the logistic regression on the training data we have create the predicted variable The code



```
y_pred=lr.predict(X_test)
```

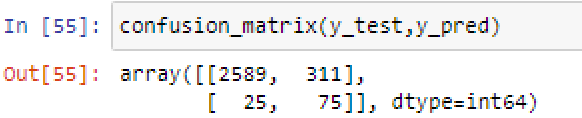
When the above code is get executed the logistic regression model uses the learned patterns and relationship from the training process to predict the class labels or outcomes for the test data.

From sklearn.metrics we have import confusion matrix and classification report.
Before moving forward we should look a glance towards confusion matrix and classification report.

Confusion matrix: In simple terms, a confusion matrix is a table that summarizes the performance of a classification model.

- True Positive (TP): The model correctly predicted instances as belonging to the positive class.
- True Negative (TN): The model correctly predicted instances as belonging to the negative class.
- False Positive (FN): The model incorrectly predicted instances as belonging to the positive class when they actually belong to the negative class.
- False Negative (FN): The model incorrectly predicted instances as belonging to the negative class when they actually belong to the positive class.

Classification report: in simple terms a classification report is a summary of the performance of a classification model. It provides detailed information about how well the model is predicting different classes or categories.
A classification report contains Precision, Recall, F1-score, support.



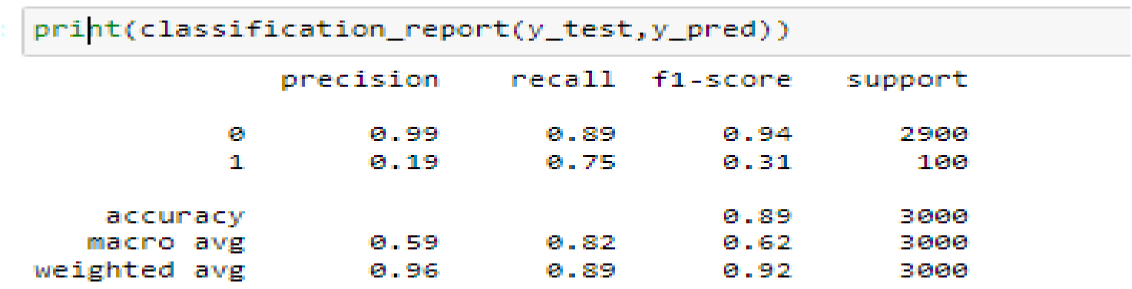
```
In [55]: confusion_matrix(y_test,y_pred)

Out[55]: array([[2589,  311],
                [  25,   75]], dtype=int64)
```

Interpretation:
True Positives (TP): there are 75 true positive classes.
True Negative (TN): there are 2589 true negative classes.
False Positives (FP): there are 311 falsely positive classes.
False Negative (FN): there are 25 false negative classes.

In summary we could say that
. There are 75 observations correctly classified as positive.
. There are 2589 observations correctly classified as negative.
. There are 311 observations falsely classified as positive.
. There are 25 observations falsely classified as negative.

And now move forward to the classification report. The output of the classification report is



```
print(classification_report(y_test,y_pred))
```

```
              precision    recall  f1-score   support

           0       0.99      0.89      0.94      2900
           1       0.19      0.75      0.31       100

    accuracy                           0.89      3000
   macro avg       0.59      0.82      0.62      3000
weighted avg       0.96      0.89      0.92      3000
```

Interpretation:

- Precision: in this case, for class 0 the precision is 0.99 which means that 99% of the instances predicted at class 0 were actually class 0. For class 1 the precision is 0.19 indicating that only 19% of the instances predicted as class 1 were actually class 1.
- Recall: for class 0, the recall is 0.89 meaning that the model correctly identified 89% of the actual instances belonging to class 0. For class 1, the recall is 0.75 indicating that the model identified 75% of the actual instances belonging to class 1.
- F1-score: for class 0, the F1-score is 0.94 and for class 1, it is 0.31
- Support: support represents the number of occurrences of each class in the actual dataset. For class 0 the support is 2900 and for class 1 the support is 100.
- Accuracy: In this case the accuracy is 0.89, which means that the model is predicted 89% of the instances correctly.
- Macro avg: macro average calculates the average of precision, recall and F1-scores across all classes, without considering class imbalance. The macro average precision is 0.59, the recall is 0.82, and the F1-score is 0.62.
- Weighted avg: weighted average calculates the average of precision, recall and F1-score, weighted by the support of each class. This takes into account the class imbalance. The weighted average precision is 0.96, the recall is 0.89 and the F1-score is 0.92

Conclusions:

The model exhibits high precision and recall for class 0 (low credit risk) , suggesting it is effective at identifying instances with low risk. However, it performs relatively poorly for class 1 (high credit risk), with low precision and moderate recall. The overall accuracy of the model is good, but there is room for improvement, particularly in predicting high credit risk instances. Further analysis and refinement of the model may be required to enhance its performance on high-risk cases.