**Link of the Dataset->**

The dataset used is a subset of the LendingClub dataset obtained from Kaggle: <https://www.kaggle.com/wordsforthewise/lending-club>

LendingClub is a US peer-to-peer lending company, headquartered in San Francisco, California. It was the first peer-to-peer lender to register its offerings as securities with the Securities and Exchange Commission (SEC), and to offer loan trading on a secondary market. LendingClub is the world's largest peer-to-peer lending platform.

**Aim of the Project:**

Given historical data on loans given out with information on whether or not the borrower defaulted (charge-off), can we build a model that can predict wether or nor a borrower will pay back their loan? This way in the future when we get a new potential customer, we can assess whether or not they are likely to pay back the loan.

The "loan\_status" column contains the label.

Introduction: The Problem statement of this project is that LendingClub is a US peer-to-peer lending company, headquartered in San Francisco, California, and it lends loan to it's customers but the problem is that it's customers dosen't repay back the loan, so the main motive is to build a model using Artificial Neural Network(which is a part of Deep Learning), so that we can predict whether or not the borrower will repay back their loan, which will help the company to assess whether or not the customer will repay back their loan on the basis of their data.

Data Analysis libraries are used for data analysis, Exploratory Data Analysis is done to assess properly, about the data given. A lot of Feature Engineering work is done and dummy variables are also created on the dataset, so as to fill the missing values, and deal with the missing columns.

**# The outcome of feature engineering:**

1. Because of feature engineering, a lot of correlation was found between some columns of the dataset.
2. Since the dataset was highly imbalanced, the technique of over sampling is also applied to balance the label which we are trying to predict, as without balancing the accuracy was around 89%, but using the SMOTE using the imblearn library, the accuracy of the model increased to 91%
3. There are other resampling techniques, such as, under sampling could also be done but then a large amount of data might have got lost, and so over-sampling was chosen as the final technique.
4. When training the model, Adam optimizer could have been used, but then there was a huge increment in the loss and the accuracy came out to be around 52%, so then Stochastic Gradient Descent technique was applied as the optimizer.
5. SMOTE was applied before train\_test\_split, it could also have been applied before that since it doesn’t create duplicate data.

**Batch Gradient Descent:** The weights are adjusted after all the rows have run into the network.

**SGD:** The weights are adjusted after each of the row gets fed into the model. Helps to provide the global minima.

**# Data - pre-processing, Exploratory Data analysis and Feature Engineering:**

1. Exploratory Data Analysis is done to get an overview about the classes, and further resampling techniques such as SMOTE is applied to balance out the dependent variable.
2. Correlation of the dataset is found out to get an idea about the columns which are highly corelated to each other.
3. Checked for any null values/missing values, and what percentage of contribution these null values have.
4. Calculating the summary statistics described by the boxplot.
5. The dependent variables are encoded to numerical values, since the model only predicts 0,1 values.
6. Further EDA is done to get an overview of the columns which need to be dropped.
7. The mortgage account column had a lot of missing values which had a major contribution in the dataset, hence this column can’t be dropped, so these were filled with the mean of the values in this column.
8. Some columns in which there was a lot less missing data were dropped.
9. The dataset is cleaned, so that it does not contain any null values.
10. Further, the categorical columns have been converted into dummy variables, so the model can process it (one-hot encoder could have been used).
11. The dataset is further split into training, and test set and scaling is done through normalization (therefore, MinMaxScaler is used)
12. Since the accuracy is already going to be high, the main evaluation metrics are going to be precision and recall.
13. In reality, there wouldn’t be any issue\_d when the model will be deployed

**# CONCLUSION:**

The recall, precision, and f1-score were affected and were improved a lot, since the classes were balanced out before feeding into the model.

The main evaluation metrics in this problem statement were f1-score, recall, and precision. Accuracy couldn’t be a good evaluation metric to measure the performance of the model, since the dataset was imbalanced.

The f1-score, precision, and recall was the main criteria to decide the model’s performance, and it was low when the dataset was not balanced, but after balancing the dataset, the performance of the model improved.

**# Testing the model on New Observations:**

Finally, the model was tested on random generated data.