PROJECT ON LOAN PREDICTION

Title: Predicting Loan Default Using Machine Learning: An Imbalanced Dataset Approach.

Objective: Evaluate and compare the performance of various classification algorithms, including logistic regression, decision trees, and ensemble methods, to identify the most effective model for predicting loan defaults.

About Data: The dataset sourced from Kaggle comprises various features related to loan applicants, including age, income, loan amount, credit score, employment details, and loan terms. Additional attributes such as education level, marital status, and loan purpose are also included. The dataset exhibits class imbalance, with a disproportionate distribution between defaulted and non-defaulted loans. This imbalance necessitates techniques such as oversampling to ensure balanced representation for robust model training and accurate prediction of loan defaults.

Project:

- Importing libraries which are required and reading a dataset.

1) Importing libraries like pandas , numpy, matplotlib, seaborn and other important libraries.

2) Read the file and check size, shape of dataset.

By using info function in pandas we fot to know that theire are no null values in dataset.

- ➢ Balancing The imbalanced Data.

3) The output column contains imbalanced data where 0 class have rows 225694 and 1 class have rows 29653. Before sampling remove loanId column. Split the dataset into x and y.

4) By using random over sampler we did a oversampling and concat that x_sampled and y_sampled data in data.

5) After describing a data we got to know about mean, std, max, min etc of each columns.

- ➢ Data Preprocessing

6) On Eduacation column we applied Ordinal encoding because we find the order in that column by using sklearn's Ordinal Encoding. Change the datatype from object to int.

7) On Employement type we did one hot encoding by using pandas function pd.get_dummies. To avoid multicollinearity drop the first column and change the data type to int. Simultaneously we are removing the original column.

8) Similarly we do One hot encoding on column like MaritalStatus, LoanPurpose because there were no order in the classes.

9) In one hot encoding we create a variable in which we stores the column values which gets encoded and then by concat function we add them into a original dataset.

10) After that the columns like HasCoSigner, HasMortgage, HasDependents which contains a class labels yes and no. To convert them into a number we did a label encoding and changes the datatype of that column.

- Exploratory Data Analysis

11) To check for outliers we generally use boxplot. That's why we can use boxplot for Income, Loan amount, Credit score, Months employed. By using subplots we can visualize all four box plots in one figure. There were no outliers.

12) Using heatmap we can see the correlation between the columns.

13) Check wheather the data is normally distributed or not. To check that we firstly import scipy.stats. We do various types of transformation to make column to normally distributed and one of the transformation is box_cox transformation and we import that from scipy.

14) Figsizevis generally used to fit two or more figures or charts in one frame. By using a histplot first we check for Age column wheathe it is normally distributed or not and simultaneously we plot a Q-Q plot to visualize properly. We observed that the column was uniformally distributed.

15) We applied boxcox transformation to make it normally distributed. But there were no affects. Then after that applied sqrt, reciprocal etc transformation but data remains as it is.

16) After that checked for the Income column the data was uniformally distributed. So I applied boxcox, log, sqrt, reciprocal transformation and simultaneously checked the Q-Q plot.

17) We did same process on rest of the columns and visualize various transformation.

- Splitting a Data

18) After EDA now we can apply an algorithms on a dataset.First of all we split data into x and y in which x will contain independent columns and y will contain dependent column.

19) Split the data into training and testing where train size will be 0.77. To put a data on to a same scale we can use standardization to scaled down a data.

- Algorithms Selection
  - Random Forest

20) First we will apply Random forest. Training score is coming out to be 1 and the accuracy was 0.99. We also visualize confusion matrix and a classification report.

	precision	recall	f1-score	support
0	0.98	1.00	0.99	51267
1	1.00	0.98	0.99	52553
accuracy			0.99	103820
macro avg	0.99	0.99	0.99	103820
weighted avg	0.99	0.99	0.99	103820

- Logistic Regression

21) After that we perform logistic regression which give me accuraccy of 0.68 the we did hyperparameter tunning but the accuracy remain same. Classification report is like:

	precision	recall	f1-score	support
0	0.67	0.69	0.68	51024
1	0.69	0.68	0.69	52796
accuracy			0.68	103820
macro avg	0.68	0.68	0.68	103820
weighted avg	0.68	0.68	0.68	103820

- KNN

22)KNN gives me accuraccy of 0.82.

	precision	recall	f1-score	support
0	0.69	0.95	0.80	37793
1	0.97	0.76	0.85	66027
accuracy			0.83	103820
macro avg	0.83	0.86	0.83	103820
weighted avg	0.87	0.83	0.83	103820

- AdaBoost

23) AdaBoost Classifier gives us accuracy of 0.68.

	precision	recall	f1-score	support
0	0.68	0.69	0.68	51565
1	0.69	0.68	0.68	52255
accuracy			0.68	103820
macro avg	0.68	0.68	0.68	103820
weighted avg	0.68	0.68	0.68	103820

24) After that I used PCA where We select n_compone nts as 4. And the scaled the data. On PCA data we used random forest algorithm whose accuracy comes out to be 0.98.

	precision	recall	f1-score	support
0	0.96	1.00	0.98	65456
1	1.00	0.96	0.98	69961
accuracy			0.98	135417
macro avg	0.98	0.98	0.98	135417
weighted avg	0.98	0.98	0.98	135417

25) From all the algorithms the best accuracy was 0.99 which is given by Random forest.

Conclusion:Through rigorous preprocessing, exploratory analysis, and modeling, the developed machine learning model demonstrated promising performance in predicting loan defaults. The use of oversampling techniques to address class imbalance, along with appropriate feature encoding and scaling, contributed to the model's effectiveness. Future work may involve fine-tuning model hyperparameters and exploring advanced ensemble techniques to further enhance predictive accuracy and robustness.