

Maratha Vidya Prasarak Samaj’s

**K.R.T Arts, B.H. Commerce & A.M. Science College, Nashik-422002**

Affiliated to Savitribai Phule Pune University, Pune

A Project Report on

**“Bank Loan Default Prediction using Machine Learning Models”**

By

**Aaradhya Balasaheb Jadhav**

**Sumit Bhaskar Jagdale**

**Vaibhav Babasaheb Kotkar**

Under the guidance of

**Mr. D. B. Uphade**

**Assistant Professor,**

**Department of Statistics,**

**KTHM College, Nashik**

**Certificate**

This is to certify that the project entitled **“Bank Loan Default Prediction Using Machine Learning Models ”** is being submitted by Jadhav Aaradhya Balasaheb, Jagdale Sumit Bhaskar, Kotkar Vaibhav Babasaheb as partial fulfilment for the award of the degree of the Master of Science (Statistics).

This is a record of considerable work carried out by them under my supervision and guidance.

Place: Nashik

Date:

**Project Guide** **Examiner H.O.D.**

Prof. D. B. Uphade Dr. G. S. Phad

Assistant professor, Department of Statistics,

**Acknowledgement**

We have satisfaction on completion of our project entitled “Bank Loan Default Prediction Using Machine Learning Models” at Department of Statistics in “M.V.P. Samaj’s K.R.T. Arts, B.H. Commerce, A.M. Science College, Nashik” during the academic year of 2022-2023.

We owe Professor D. B. Uphade our gratitude for his direction and persistent supervision, as well as for providing the project with the essential knowledge and for helping us finish the project. We were able to finish the assignment allocated to us on time because to his consistent direction and desire to impart his wide knowledge.

We would like to show our gratitude to Dr. G. S. Phad, Head of the Statistics Department, for allowing us the chance to work on this project and for providing us with all the support we needed.

Additionally, we appreciate the invaluable advice and excellent cooperation from all the faculty in the Department of Statistics.

**Content**

**Abstract**

The prediction of customer loan default has always been a critical and persistent issue in the banking industry. Traditional manual methods of loan prediction require significant time and manpower resources, often leading to suboptimal results. To overcome these challenges and enhance prediction accuracy while saving time, this project employs various machine learning models.

The first key point addressed in this project is the fact that customer loan prediction is a lifelong issue and requires continuous monitoring and effective mitigation strategies. The second key point focuses on the limitations of manual loan prediction methods, such as labour-intensive, time-consuming, and prone to human errors. Finally, the project emphasizes the advantages of utilizing different machine learning models for loan default prediction. Overall, this project highlights the importance of leveraging machine learning models to address the lifelong issue of customer loan prediction, by automating the process, reducing manual efforts, and utilizing diverse models.

**Introduction**

This research aims to create a sophisticated machine learning-based system for predicting bank loan default. The goal is to find the best model(s) for loan default prediction by applying a variety of algorithms, such as logistic regression, decision trees, random forests, and neural networks. The system will automate the loan review process and produce predictions that can help banks identify clients who are likely to default on their loans. Additionally, the proposed system will enable a continual review of loan applicants during their relationship with the bank. Overall, the use of machine learning algorithms to anticipate bank loan defaults has great promise for revolutionising the sector. This project aims to provide banks with an effective and accurate prediction system that enhances decision-making, optimises resource allocation, and enables proactive risk management.

The initiative hopes to improve loan default prediction procedures, making them more precise, effective, and automated a development that will eventually be advantageous to banks and customers alike.

**Objectives**

1) to Enhance Prediction Accuracy: By utilising the capabilities of multiple machine learning models

2) To Automate loan prediction process to save time and resources.

3) To Identifying high-risk loan applicants helps banks allocate resources more efficiently.

4) To Enable Proactive Risk Management: Banks can reduce financial risks by using proactive risk management strategies to anticipate loan defaults.

5) To Facilitate Informed Decision Making: The initiative seeks to help banks make informed decisions about loan approvals and risk assessment, promoting a better financial ecosystem and ensuring the stability of lending institutions.

**Motivation**

The concept of predicting bank loan defaults using machine learning models was inspired by the urgent need for precise, effective, and automated solutions to handle the problems posed by loan defaults in the banking sector. The significance and applicability of this project are influenced by the following factors: Financial Stability: Banks and lending institutions' ability to maintain their financial stability is seriously threatened by loan defaults; Timely Decision-Making: Banks and customers depend on quick loan approval procedures; Resource Optimisation: Manual loan appraisal demands a significant time and resource commitment; Enhanced Prediction Accuracy: Conventional loan default prediction techniques might not have the accuracy required to pinpoint clients who are at high risk of default.

The project of Bank Loan Default Prediction using Machine Learning Models aims to increase forecast accuracy, reduce financial risks, and revolutionise the loan default prediction process by leveraging the power of modern technologies and leveraging the capabilities of machine learning models.

The goal is to create a more effective and customer-focused lending environment by creating a system that can predict loan defaults with accuracy and efficiency, assisting in the stability and sustainability of the banking sector. Machine learning algorithms have the capacity to examine intricate patterns and glean insightful information from big datasets, allowing banks to decide more intelligently about loan approvals, risk assessment, and customising loan terms. Technology advancements have created new opportunities for the banking sector's difficult issues, and this project intends to revolutionise the loan default prediction process by leveraging the power of modern technologies and leveraging the capabilities of machine learning models.

**Methodology**

**1)SVM (Support Vector Machines):**

SVM is a binary classification algorithm that finds the best hyperplane in a high-dimensional feature space to separate data points of different classes. It aims to maximize the margin between the hyperplane and the nearest data points of each class. The decision function for SVM can be represented as:

f(x) = sign(w^T \* x + b)

where 'w' represents the weight vector, 'x' represents the input feature vector, and 'b' represents the bias term.



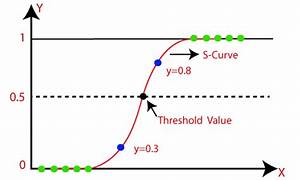
Source : Java T Point

**2)Logistic Regression:**

* Logistic regression is a popular binary classification algorithm. It models the relationship between the input features and the probability of belonging to a certain class. The logistic regression equation is:

p(y=1|x) = 1 / (1 + exp(-(w^T \* x + b)))

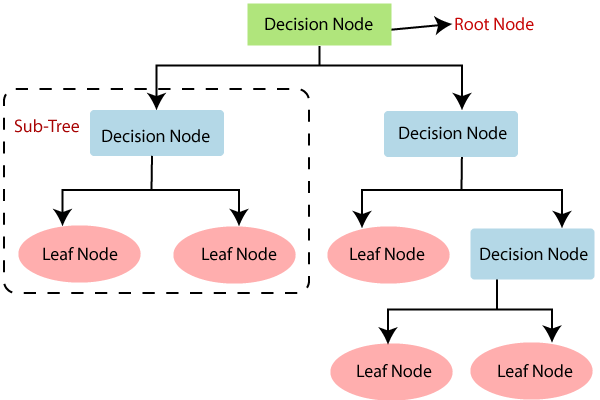
* where 'w' represents the weight vector, 'x' represents the input feature vector, 'b' represents the bias term, and 'exp()' is the exponential function.



Source: Google

**3)Decision Tree:**

* Decision Tree is a Supervised learning technique that can be used for both classification and Regression problems, but mostly it is preferred for solving Classification problems. It is a tree-structured classifier, where internal nodes represent the features of a dataset, branches represent the decision rules and each leaf node represents the outcome.
* In a Decision tree, there are two nodes, which are the Decision Node and Leaf Node. Decision nodes are used to make any decision and have multiple branches, whereas Leaf nodes are the output of those decisions and do not contain any further branches.
* The decisions or the test are performed on the basis of features of the given dataset.
* It is a graphical representation for getting all the possible solutions to a problem/decision based on given conditions.
* It is called a decision tree because, similar to a tree, it starts with the root node, which expands on further branches and constructs a tree-like structure.
* In order to build a tree, we use the CART algorithm, which stands for Classification and Regression Tree algorithm.
* A decision tree simply asks a question, and based on the answer (Yes/No), it further split the tree into subtrees.
* Diagram explains the general structure of a decision tree:



Source: Java T Point

**Why use Decision Trees?**

* There are various algorithms in Machine learning, so choosing the best algorithm for the given dataset and problem is the main point to remember while creating a machine learning model. Below are the two reasons for using the Decision tree:
* Decision Trees usually mimic human thinking ability while making a decision, so it is easy to understand. The logic behind the decision tree can be easily understood because it shows a tree-like structure.

**4) Random Forest:**

Random Forest is an ensemble learning algorithm that combines multiple decision trees. Each tree in the forest is trained on a random subset of the training data and a random subset of the input features.

The final prediction of the random forest is made by averaging the predictions of all individual trees. Random Forest does not have a single mathematical equation but is based on the principles of decision trees and averaging.



Source: Java T Point

Assumptions for Random Forest

Since the random forest combines multiple trees to predict the class of the dataset, it is possible that some decision trees may predict the correct output, while others may not. But together, all the trees predict the correct output. Therefore, below are two assumptions for a better Random forest classifier:

* There should be some actual values in the feature variable of the dataset so that the classifier can predict accurate results rather than a guessed result.
* The predictions from each tree must have very low correlations.

Why use Random Forest?

Below are some points that explain why we should use the Random Forest algorithm:

* It takes less training time as compared to other algorithms.
* It predicts output with high accuracy, even for the large dataset it runs efficiently.
* It can also maintain accuracy when a large proportion of data is missing.

**5)KNN (K-Nearest Neighbours):**

* K-Nearest Neighbour is one of the simplest Machine Learning algorithms based on Supervised Learning technique.
* K-NN algorithm assumes the similarity between the new case/data and available cases and put the new case into the category that is most similar to the available categories.
* K-NN algorithm stores all the available data and classifies a new data point based on the similarity. This means when new data appears then it can be easily classified into a well suite category by using K- NN algorithm.
* K-NN algorithm can be used for Regression as well as for Classification but mostly it is used for the Classification problems.
* K-NN is a **non-parametric algorithm**, which means it does not make any assumption on underlying data.
* It is also called a **lazy learner algorithm** because it does not learn from the training set immediately instead it stores the dataset and at the time of classification, it performs an action on the dataset.
* KNN algorithm at the training phase just stores the dataset and when it gets new data, then it classifies that data into a category that is much similar to the new data.
* **Example:** Suppose, we have an image of a creature that looks similar to cat and dog, but we want to know either it is a cat or dog. So for this identification, we can use the KNN algorithm, as it works on a similarity measure. Our KNN model will find the similar features of the new data set to the cats and dogs images and based on the most similar features it will put it in either cat or dog category.



Source : Java T Point

How does K-NN work?

The K-NN working can be explained on the basis of the below algorithm:

* **Step-1:** Select the number K of the neighbours
* **Step-2:** Calculate the Euclidean distance of **K number of neighbours**
* **Step-3:** Take the K nearest neighbours as per the calculated Euclidean distance.
* **Step-4:** Among these k neighbours, count the number of the data points in each category.
* **Step-5:** Assign the new data points to that category for which the number of the neighbour is maximum.
* **Step-6:** Our model is ready.
* Firstly, we will choose the number of neighbours, so we will choose the k=5.
* Next, we will calculate the **Euclidean distance** between the data points. The Euclidean distance is the distance between two points, which we have already studied in geometry. It can be calculated as:



Source: Java T Point

* By calculating the Euclidean distance we got the nearest neighbours, as three nearest neighbours in category A and two nearest neighbours in category B. Consider the below image:



Source: Java T Point

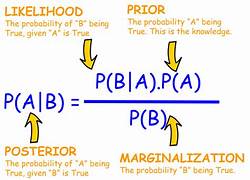
* As we can see the 3 nearest neighbours are from category A, hence this new data point must belong to category A.

**6)Naive Bayes:**

Naive Bayes is a probabilistic classification algorithm based on Bayes' theorem. It assumes independence between the features, given the class label. The equation for Naive Bayes can be written as:

p(y|x\_1, x\_2, ..., x\_n) = p(y) \* p(x\_1|y) \* p(x\_2|y) \* ... \* p(x\_n|y)

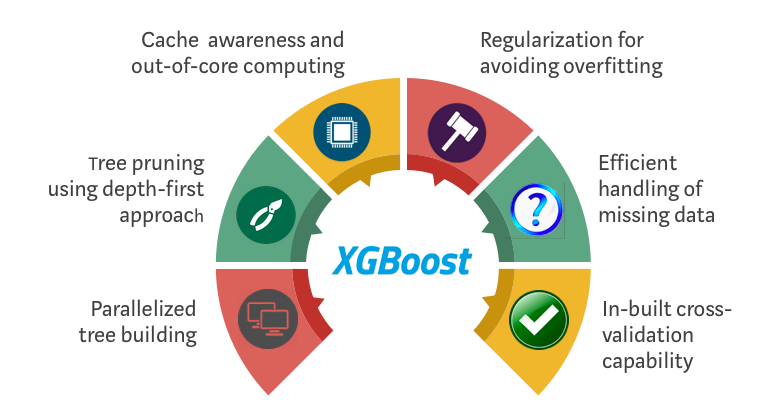
where 'p(y)' represents the prior probability of class 'y', and 'p(x\_i|y)' represents the likelihood of feature 'x\_i' given class 'y'.



Source: Java T Point

**7)XG-Boosting (Extreme Gradient Boosting):**

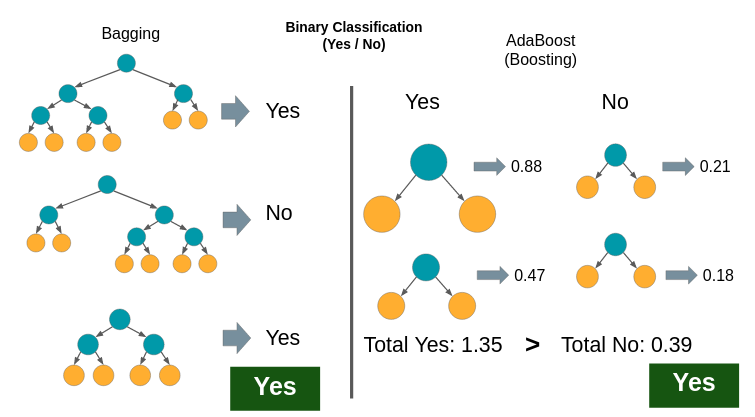
XGBoost is a powerful boosting algorithm that combines weak learners, typically decision trees, to create a strong predictive model. The final prediction is a weighted sum of the predictions from individual weak learners. The mathematical equations in XGBoost involve optimizing an objective function that incorporates the training loss and regularization terms.



Source: google

**8)Ada-Boost (Adaptive Boosting):**

AdaBoost is another boosting algorithm that iteratively trains weak learners, such as decision trees, and adjusts the weights of the training samples to focus on previously misclassified samples. The final prediction is a weighted combination of the predictions from all weak learners. The mathematical equations in AdaBoost involve updating sample weights and calculating the contribution of each weak learner.



Source: Google

**Precision:-**

Precision measures the proportion of correctly predicted positive instances (true positives) out of all instances predicted as positive (true positives + false positives). It focuses on the accuracy of positive predictions and is calculated using the following formula:

Precision = True Positives / (True Positives + False Positives)

Precision indicates how well the model performs when it predicts a positive outcome. A high precision value means that the model has a low false positive rate, which is desirable when avoiding false alarms is important.

**Recall :-**

Recall, also known as sensitivity or true positive rate, measures the proportion of correctly predicted positive instances (true positives) out of all actual positive instances (true positives + false negatives). It focuses on capturing as many positive instances as possible and is calculated using the following formula:

Recall = True Positives / (True Positives + False Negatives)

Recall indicates how well the model captures positive instances. A high recall value means that the model has a low false negative rate, which is important when it's crucial to avoid missing positive instances.

**F1 Score:**-

The F1 score is the harmonic mean of precision and recall, providing a balanced measure that takes into account both metrics. It can be seen as a single metric that represents the overall performance of a model. The F1 score is calculated using the following formula:

**F1 Score = 2 \* (Precision \* Recall) / (Precision + Recall)**

The F1 score combines precision and recall into a single value, and it ranges between 0 and 1, with 1 being the best possible score. It is useful when you want to consider both precision and recall simultaneously.

The F2 score is a weighted harmonic mean of precision and recall, where recall is given more weight. The weight is controlled by the beta parameter, and the F2 score is given by the formula:

**F2 = (1 + beta^2) \* (precision \* recall) / (beta^2 \* precision + recall)**

**Data Description**

Kaggle is a popular online platform for data science and machine learning competitions, datasets, and community collaboration. It was founded in 2010 and acquired by Google in 2017. Kaggle offers a diverse range of resources and tools that facilitate data analysis, modelling, and sharing within a vibrant community of data scientists, machine learning enthusiasts, and researchers.

**Data Description**

Dataset Name: Bank Loan Default Prediction.

Link:<https://www.kaggle.com/datasets/hemanthsai7/loandefault?select=train.csv>

Size [ Row – 67464 , cols – 35 ].

Splitting [70% , 30% respectively].

No of Variables: 35 [ numerical = 24 , categorical= 9 ].

Dependent variables is “Loan Status”.

Description of Variables:

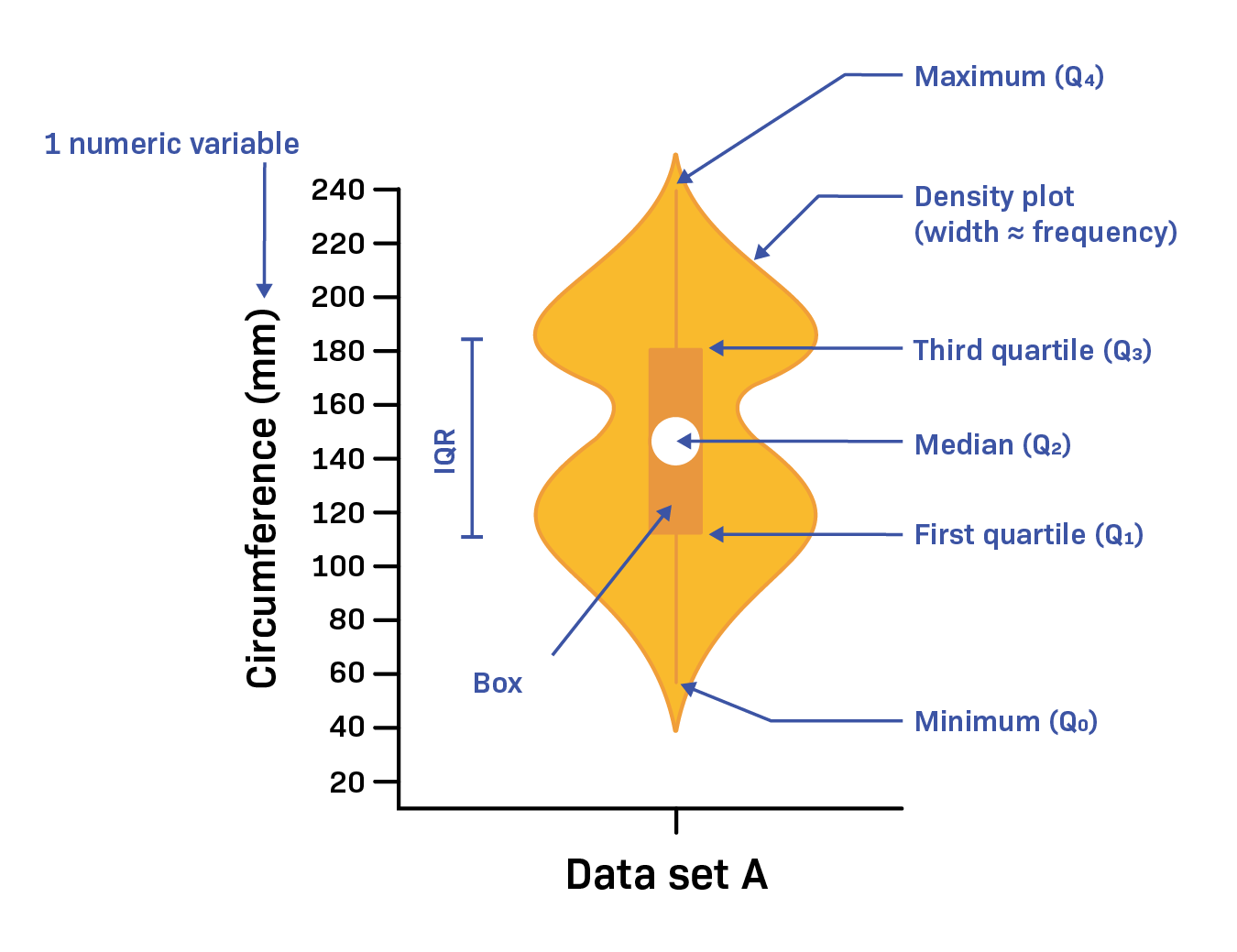
1. Loan Amount: loan amount applied.
2. Funded Amount: loan amount funded.
3. Funded Amount Investor: loan amount approved by the investors.
4. Term: term of loan (in months).
5. Batch Enrolled: batch numbers to representatives.
6. Interest Rate: interest rate (%) on loan.
7. Grade: grade by the bank.
8. Sub Grade: sub-grade by the bank.
9. Debit to Income: ratio of representative's total monthly debt repayment divided by self-reported monthly income excluding mortgage.
10. Delinquency - two years: number of 30+ days delinquency in past 2 – years.
11. Inquires - six months: total number of inquiries in last 6 months.
12. Open Account: number of open credit line in representative's - credit line.
13. Public Record: number of derogatory public records.
14. Revolving Balance: total credit revolving balance.
15. Revolving Utilities: amount of credit a representative is using - relative to revolving-balance.
16. Total Accounts: total number of credit lines available in - representatives credit line.
17. Initial List Status: unique listing status of the loan - - W(Waiting), F(Forwarded).
18. Total Received Interest: total interest received till date.
19. Total Received Late Fee: total late fee received till date.
20. Recoveries: post charge off gross recovery.
21. Collection Recovery Fee: post charge off collection fee.
22. Collection 12 months Medical: total collections in last 12 months - excluding medical collections.
23. Application Type: indicates when the representative is an individual or joint.
24. Last week Pay: indicates how long (in weeks) a representative has paid EMI after batch enrolled.
25. Accounts Delinquent: number of accounts on which the representative is delinquent.
26. Total Collection Amount: total collection amount ever owed.
27. Total Current Balance: total current balance from all accounts.
28. Total Revolving Credit Limit: total revolving credit limit.
29. Loan Status: 1 = Defaulter, 0 = Non-Defaulters

**GRAPHICAL PRESENTATION**

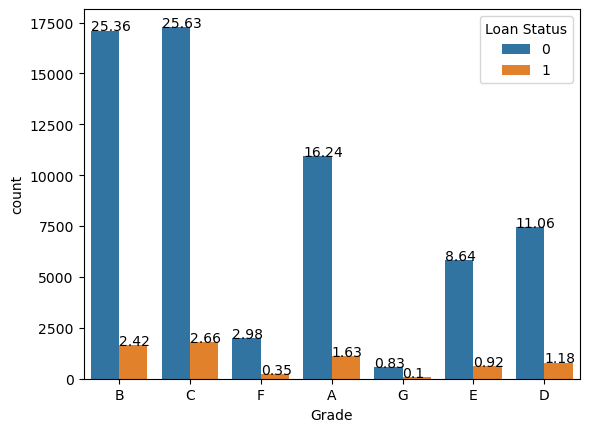
**#Violin Plot:-**

A violin plot is a hybrid of a box plot and a kernel density plot, which shows peaks in the data. It is used to visualize the distribution of numerical data. Unlike a box plot that can only show summary statistics, violin plots depict summary statistics and the density of each variable.

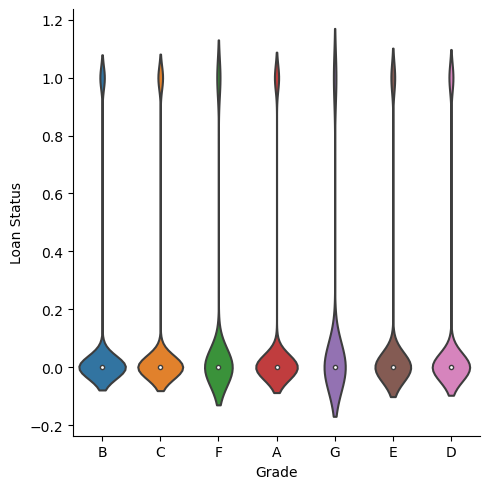
Violin plots are similar to box plots, except that they also show the probability density of the data at different values, usually smoothed by a kernel density estimator. Typically, a violin plot will include all the data that is in a box plot: a marker for the median of the data; a box or marker indicating the interquartile range; and possibly all sample points, if the number of samples is not too high.



**In this graphical representation we plot the graph of different categorical variables from our dataset**.

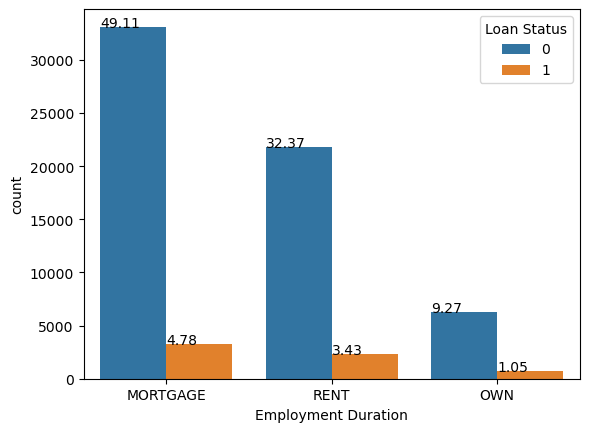
1)

This graph represents the percentwise count plot of the variable grade with respect to loan status. This grade characterized loan into different risk categories. In this dataset there are 7 different categories as A, B, C, D, E, F, G. In above graph Blue bar denotes the Non-defaulter applicants and orange bar denotes the defaulter applicants.

 So, from this graph we interpreted that Grade C has highest number of Non-defaulter and defaulter applicants .

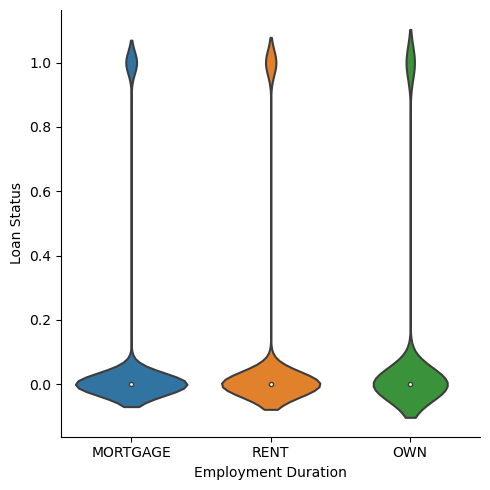
This is the violin plot of the Variable grades; it shows the contribution or density of 0(Non-defaulter) and 1 (Defaulter)of our depend variable ‘Loan Status’.

2)



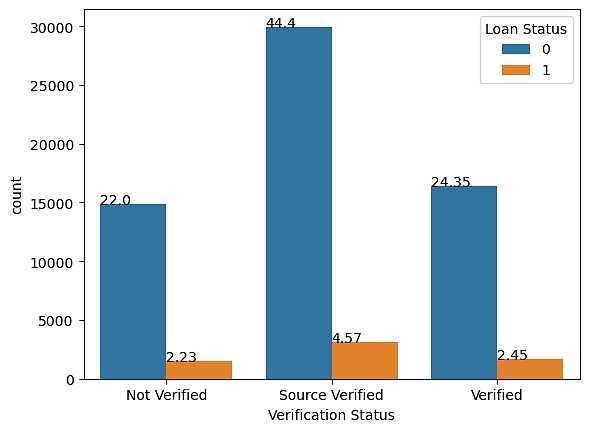
**Home ownership**

Similarly this is the count plot of variable ‘Home Ownership’. In this count plot of there are 3 different types as ‘MORTAGE’, ‘RENT’ and ‘OWN’. From this graph we can interpret that ‘MORTAGE’ Home ownership has the highest number of Non-defaulter and defaulter applicants.



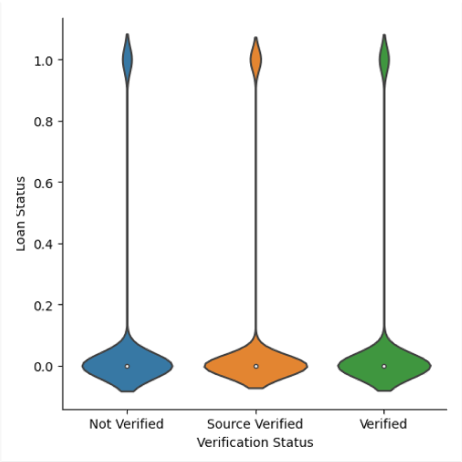
**Home ownership**

This is the violin plot of the Variable ’Home Ownership’ ,its shows the contribution or density of 0(Non-defaulter ) and 1 (Defaulter) of our depend variable ‘Loan status’.

3)

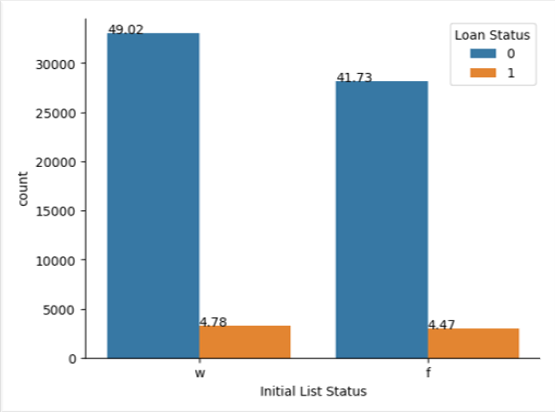
This graph shows the percentwise count plot of the variable ‘Verification Status’ with respect to ‘Loan status’. There are 3 types of the ‘Verification Status’ as ‘Not Verified’ , ‘ Source Verified’ and ‘Verified’. ( In this ‘Verified’ indicates that the lender has successfully verified the applicant information such as income, address etc. where ‘Not-Verified’ shows the lender has not conduct verification or unable to verify applicant information due to various reason and ‘Source verified’ shows lender has verified specific information from third party.)

From above graph we can interpret that ‘Source Verified’ shows highest number of of Non-defaulter and defaulter applicants.

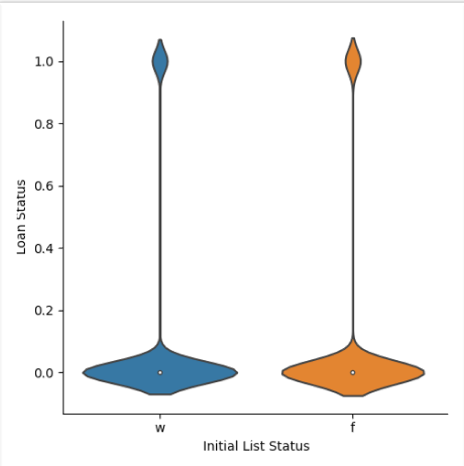


This is the violin plot of the Variable ’Verification Status’ vs ‘Loan status’ , its shows the density of 0(Non-defaulter ) and 1 (Defaulter) of our depend variable ‘Loan status’.

4)

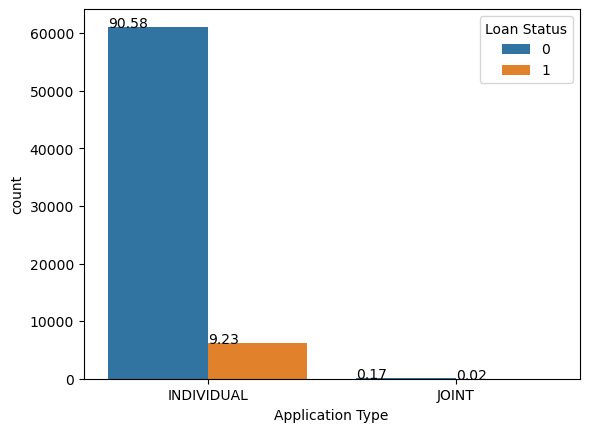


Similarly this graph is the count plot of categorical variable ‘Initial List Status’ with respect to ‘Loan status’. In this plot ‘w’ denotes waited, mean applicant loan account is still under the review or pending resolution. Whereas ‘f’ indicates Forwarded ,means applicant loan account has been passed or forwarded to another team or department within the lending institute. From this graph we can interpreted that the ‘w’(waited) ‘Initial list status’ has highest number of Non-defaulter and defaulter applicants .

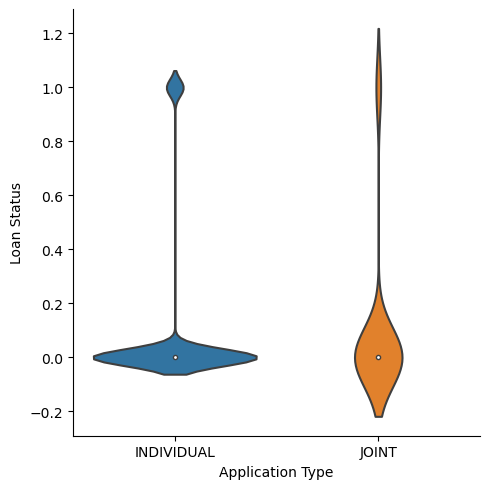


This graph shows the violin plot of variable ‘Initial List Status’ Vs ‘Loan Status’. its shows the density of 0(Non-defaulter ) and 1 (Defaulter) of our depend variable ‘Loan status’.

5)



The above graph is the count plot of the variable ‘Application Type’ vs “Loan status’. From this graph we interpret that the Non-defaulter and Defaulter applications are maximum for Individual Application type.(Or number of applicants are larger than the joint Application type)



The above graph is the violin plot of ‘INDIVIDUAL Application type’ Vs ‘JOINT Application type’ with respect to ‘Loan status’. This graph shows the density of 0(Non-defaulter ) and 1 (Defaulter) of our depend variable ‘Loan status’.

**ANALYSIS**

For this project we use “Bank Loan Default prediction” Dataset. By analysing this dataset we found that this dataset is imbalance , So we need to balance this dataset and for this purpose we use oversampling method.

Chart, bar chart

Description automatically generated

In this Oversampling method minority class is increased or created and compare it with the majority class.

In this method samples are increased in synthetically and artificially manner.So by this method we balanced our dataset.