Home Loan Approval Prediction

Sparsh Gupta[1], Alok Desai[2], and Adit Rastogi[3]

[*]BITS Pilani Hyderabad Campus
[*]CSF415 Data Mining Project
[1]Sparsh Gupta: f20221309@hyderabad.bits-pilani.ac.in
[2]Alok Desai: f20221358@hyderabad.bits-pilani.ac.in
[3]Adit Rastogi: f20221330@hyderabad.bits-pilani.ac.in

Abstract

This paper delves into the concept of automated loan approval using data mining techniques. This problem is significantly challenging due to the high stakes involved in accurate predictions, which directly impact the profitability of individuals and financial institutions. In this paper, we pre-process the data, train different machine learning models and evaluate their performance using different evaluative metrics. The results indicate a high degree of accuracy in the model's predictions, suggesting a potential in real-world applications.

Keywords: Loan Approval, Machine Learning, Risk Assessment, Predictive Modelling, Classification Metrics.

1. Introduction

1.1 Objective

In this project, we aim to address the challenges financial institutions face in their loan approval process. Our objective here is to apply classification techniques like XGBoost, Random Forest and Decision Tree algorithms to predict whether a person should be granted a loan or not based on various applicant attributes.

1.2 Importance

It is both interesting and important primarily due to three reasons,

1. Relevance to the financial industry: The finance industry heavily relies on the system of loan approvals. Improving the efficiency and accuracy of these processes can lead to reduced risks, increased profits, and enhanced customer satisfaction.

2. Making these practices fair: Borrowers demand fair and transparent processes of loan approval to ensure equal access to credit opportunities.

3. Traditional methods of loan approval often rely on manual assessment processes, which can be time-consuming, error-prone, and subject to biases.

1.3 Challenges

This problem also comes with some challenges due to:

1. High Dimensionality: Loan application datasets often contain a large number of features. For these datasets, there is a need to handle both numerical and categorical attributes. High-dimensional datasets also have noise and redundant features, which might not be contributing significantly to the required outcome.

2. Missing Data/ Data Quality issues: Loan application datasets often suffer from missing or inconsistent data, which can hinder the model's performance. It is important to deal with such data by applying the appropriate preprocessing techniques.

1.4 Why hasn't it been solved before?

1. Complex Decision Making: Loan approval decisions are often complex and are determined by a combination of many factors, including both qualitative and quantitative ones, thus making it difficult to incorporate all the relevant information into the model effectively.

2. Dynamic Nature of Financial Markets: There are constant changes in economic conditions, regulatory policies, and consumer behaviors, hence making it difficult for the model to adapt to them, which leads to the model being obsolete due to its reduced predictive accuracy over time.

1.5 Key components of our approach

1. Dealing with high dimensionality by thoroughly analyzing each of the attributes and their contribution to the decision-making process. We applied appropriate preprocessing techniques which helped us reduce the dimensionality considerably without losing out on much information.

2. Handling inconsistent values, outliers, and noise by applying exploratory data analysis and appropriate preprocessing techniques

3. Applying classification techniques, XGBoost, Random Forest, and Decision Tree monitoring their predictive accuracy, by using metrics like confusion matrix and f1 score, and concluding with the most suitable technique for our model, based on our observations.

2. Related Works

1. ([1]) This study employs machine learning to analyze data from 1000 loan applicants in Dhaka, Bangladesh, aiming to predict loan approval and repayment ability. Techniques like Decision Tree, Gradient Boosting, and Random Forest are used to improve the accuracy of loan approval predictions, potentially streamlining the process for banks and borrowers alike. At the conclusion of the study, the Random Forest method has the highest accuracy. The research effectively explores machine learning algorithms for loan prediction, emphasizing data preprocessing and model evaluation. However, limited transparency in data collection and quality control processes may introduce bias, requiring closer examination for reliable results.

2. ([2]) This project improves loan eligibility prediction using machine learning with data from multiple banks. It compares various models and finds decision tree with AdaBoost which is its strength because it is most accurate for predicting customer loan eligibility, surpassing traditional methods. AdaBoost uses a progressively learning boosting technique. Hence high-quality data is needed in examples of AdaBoost vs Random Forest. It is also very sensitive to outliers and noise in data requiring the elimination of these factors before using the data.

3. ([3]) This study employs machine learning algorithms to predict loan defaulters by analyzing past customer data, aiming to improve banks' risk assessment procedures. Random Forest outperforms other algorithms in accuracy, highlighting its potential for enhancing loan approval decisions and minimizing financial losses. One of the main advantages of random forest is that it reduces the risk of overfitting and the required training time. Additionally, it offers a high level of accuracy. On the other hand training time is more than other models due to its complexity

3. Methodology

3.1 Problem

The problem being tackled in this project is the challenge faced by financial institutions in effectively evaluating loan applications. Specifically, the focus is on developing predictive models using data mining techniques to determine whether a loan application should be approved or rejected based on various applicant attributes and other relevant factors.

3.2 Requirements

We mainly need an appropriate dataset to work on. The dataset should have included all the relevant attributes that contribute to the loan approval process. The dataset should have less inconsistent values, which will help in the smooth functioning of the model.

We were able to get our dataset from Kaggle.com. This dataset matched most of our requirements to build our model.

3.3 Dataset and its properties

Initially, our dataset had 614 rows and 13 attributes. The attributes are as follows:

- Loan ID
- Gender
- Married
- Dependents
- Education
- Self Employed
- Applicant Income
- Co-applicant Income
- Loan Amount
- Loan Amount Term
- Credit History
- Property Area
- Loan Status

This had a mix of both numerical and categorical attributes. This dataset didn't have any duplicate values but had several missing values.

3.4 Techniques Used

3.4.1 Random Forest Algorithm

It is based on the concept of ensemble learning, which is a process of combining multiple classifiers to solve a complex problem and improve the performance of the model. ([4]). The algorithm comprises of the following steps:

1. Select random K data points from the training set.
2. Build the decision trees associated with the selected data points (Subsets).
3. Choose the number N for decision trees that you want to build.
4. Repeat Steps 1 and 2.
5. For new data points, find the predictions of each decision tree, and assign the new data points to the category that wins the majority votes.

3.4.2 XGBoost Algorithm

XGBoost algorithm belongs to ensemble learning category, specifically to the gradient boosting category. XGBoost stands for eXtreme Gradient Boosting, and it works by building predictive models by combining predictions of many individual models like Decision Trees, in an iterative fashion. ([5]). The algorithm comprises of the following steps:

1. Create a single leaf tree.
2. (a) For the first tree, compute the average of target variable as prediction and calculate the residuals using the desired loss function.

   (b) For subsequent trees the residuals come from prediction made by previous tree.
3. Calculate Similarity scores using formula:

$$\text{Similarity Score} = \left(\frac{\text{Gradient}^2}{\text{Hessian} + \lambda} \right) \tag{1}$$

   [1] Use the similarity score to select appropriate node. Higher the similarity score, higher is the homogeneity.
4. Using similarity score, calculate information gain.

$$\text{Information Gain} = \text{Left Similarity} + \text{Right Similarity} \tag{2}$$

   Information gain gives the difference between old similarity and new similarity and thus tells how much homogeneity is achieved by splitting the node at a given point.
5. Create the tree of desired length using the above method. Pruning and regularization can be done by playing with the regularization hyperparameter.
6. Predict the residual values using the Decision Tree you constructed.

$$\text{New Residuals} = \text{Old Residuals} + \rho\Sigma\text{Predicted Residuals} \tag{3}$$

7. Go to 1, and repeat the process for all trees.

3.4.3 Decision Tree Algorithm

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. Decision

[1] Hessian = number of residuals. Gradient^2 = squared sum of residuals. λ is a regularization hyperparameter.

Trees were chosen for their simplicity, interpretability, and effectiveness in capturing complex data relationships, improving loan prediction accuracy.

The various steps in the Decision Trees algorithm include:

1. Splitting: At each node, the algorithm selects the feature that best splits the data into groups based on a criterion like Gini impurity or entropy.

2. Recursive partitioning: This process continues recursively, splitting the data into smaller subsets until a stopping criterion is met, such as reaching a maximum depth or having a minimum number of samples in each leaf node.

3. Tree pruning: After the tree is fully grown, pruning techniques may be applied to reduce overfitting by removing nodes that do not significantly improve predictive accuracy on a validation dataset.

4. Prediction: Once the tree is constructed, new instances are classified by traversing the tree from the root node to a leaf node, where a class label is assigned based on majority voting or probability estimates.

4. Experiments

4.1 Dataset

4.1.1 What pre-processing methods do you need to apply?

First, we visualized the data using exploratory data analysis, by plotting the graphs of different attributes to get an idea of the distribution between them. Further, we used stack graphs to get the percentages of loans approved based on different attributes.

We also applied the following preprocessing methods:

1. Missing value treatment : Now we replace the categorical attributes with their mode and numerical attributes with their median.

2. Feature Creation(Total Income): From EDA, we could see many missing values in co-applicant income, also the total income was not affecting the loan status significantly. Hence creating a feature, total income, by adding applicant income and co-applicant income was an essential part of our data preprocessing.

3. Handling Outliers: We handled the outliers on total income and Loan Amount using a log transformation, which earlier had a skewed distribution.

4.1.2 What would the final processed dataset look like?

Our final dataset has 614 rows and 9 attributes. The final processed dataset also has a combination of both numerical and categorical attributes. As a result of appropriate preprocessing, all the missing values are treated already.

	Married	Dependents	Education	LoanAmount	Loan_Amount_Term	Credit_History	Property_Area	Loan_Status	TotalIncome
0	No	0	Graduate	4.859812	360.0	1.0	Urban	1	8.674197
1	Yes	1	Graduate	4.859812	360.0	1.0	Rural	0	8.714732
2	Yes	0	Graduate	4.204693	360.0	1.0	Urban	1	8.006701
3	Yes	0	Not Graduate	4.795791	360.0	1.0	Urban	1	8.505525
4	No	0	Graduate	4.955827	360.0	1.0	Urban	1	8.699681

Figure 1: Final Pre-processed Dataset

4.2 Evaluation Method / Metrics

We've mainly used four evaluation metrics, namely accuracy, precision, f1-score and confusion matrix.

1. Accuracy: It is the ratio of the number of correct predictions to total number of predictions. This is the most intuitive performance measure as it simply measures the proportion of correct predictions.

2. Precision: It is the ratio of the number of true positives (loans correctly identified as approved) to the sum of true positives and false positives (loans incorrectly identified as approved). It's a good measure to determine when the costs of false positive is high.

3. F1-Score: It is the harmonic mean of precision and sensitivity, it tries to balance the two. Since precision and recall both provide valuable information about a model's performance, it's often convenient to combine the two into a single metric called the F1 score.

4. Confusion matrix: It is a matrix that describes the performance of the model. It includes values for true positives, true negatives, false positives, and false negatives. It gives a more detailed view of what's going on than just the proportion of correct classifications (accuracy). It's especially useful for imbalanced classes.

4.3 Experimental setup

4.3.1 Methods Used

Exploratory Data Analysis

Performing EDA will help us in performing appropriate data preprocessing, through visualisation of attributes and their relations with each other in the form of graphs.

Random Forest Algorithm

1. To apply this algorithm, we first split our dataset by using the train_test_split from sklearn.model_selection library. Further, we we import the RandomForestClassifier from sklearn.ensemble library.

2. For our random forest classifier, we are considering three hyperparameters ([6]), which are

   - **maxdepth:** for limiting the total depth to which we want our random forest to grow,
   - **nestimators:** tells the number of decision trees, and
   - **randomstate:** which helps ensure reproducibility of the results.

3. Inorder to find the best suited values for these parameters we use GridSearchCV from the sklearn.model_selection library.

4. We import confusion matrix, accuracy score and f1_score from sklearn.metrics library to judge our model performance.

XGBoost Algorithm

1. We first split our dataset into X_train, X_test, y_train and y_test by using train_test_split with test_size=0.2 and random_state=142.

2. Using GridSearchCV , we determine our hyperparameters ([7]).

   - **max_depth:** This parameter sets the maximum depth of a tree and is used to control over-fitting. Typically, a value of 3-10 is chosen depending on the number of features and samples. We start with a value of 3.
   - **colsample_bytree:** This is the subsample ratio of columns when constructing each tree. Subsampling occurs once for every tree constructed. Generally, a value of 0.2-1 is used. We start with a value of 0.3.
   - **colsample_bylevel:** This is the subsample ratio of columns for each level. Subsampling occurs once for every new depth level reached in a tree.Generally, a value of 0.3-1 is used. We start with a value of 0.1.
   - **subsample:** This is the subsample ratio of the training instances. Subsampling occurs once in every boosting iteration. Generally, a value of 0.3-1 is used. We start with 0.1.

3. We then run GridSearchCV over a certain range for each hyperparameter, and find the best parameters by judging performance using roc_auc.

Decision Tree Algorithm

1. We first split our dataset into X_train, X_test, y_train and y_test by using train_test_split with test_size=0.2 and random_state=142. Further, we we import the DecisionTreeClassifier from sklearn.tree library.
2. For our decision tree classifier, we are considering three hyperparameters ([8]), which are
   - **maxdepth:** for limiting the total depth of the decision tree
3. Inorder to find the best suited values for these parameters we use GridSearchCV from the sklearn.model_selection library.
4. We import confusion matrix, accuracy score and f1_score from sklearn.metrics library to judge our model performance.

4.3.2 What are the characteristics and properties of the Data Mining techniques used in the proposed method?

Random Forest Algorithm

Random Forest is a classifier that contains several decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset. The greater number of trees in the forest leads to higher accuracy and prevents the problem of overfitting. It has the following characteristics:

- It is based on the concept of ensemble learning.
- It is a supervised learning technique.
- It is capable of handling datasets with high dimensionality, and with high accuracy.
- It takes less training time as compared to other algorithms.
- It can also maintain accuracy even with some missing data.

XGBoost Algorithm

The XGBoost algorithm is classified as an ensemble learning technique ([9]), specifically belonging to the gradient boosting category. Its name is an acronym for "eXtreme Gradient Boosting," and its functionality involves creating high-precision predictive models by combining the predictions of multiple individual models, including Decision Trees, in an iterative manner. The algorithm is characterized by the following attributes:

- It is based on ensemble learning, specifically gradient boosting.
- It is capable of parallel processing, making it faster than many other gradient boosting techniques.
- It incorporates a regularization term in the loss function to prevent overfitting.
- It has an in-built routine to handle missing values.
- It prunes trees backwards, removing splits beyond which there is no positive gain.

Decision Tree Algorithm
A decision tree algorithm is a machine learning algorithm that uses a decision tree to make predictions. It follows a tree-like model of decisions and their possible consequences. The algorithm works by recursively splitting the data into subsets based on the most significant feature at each node of the tree.

- It is a nonparametric method.
- It is a supervised learning technique.
- It takes less time as compared to other algorithms
- Decision Trees offer interpretability, allowing easy understanding of decision rules.
- Decision Trees require minimal data preprocessing compared to other algorithms.
- They are robust to outliers and missing values in the data.

5. Results and discussion

5.1 Outcomes of proposed methods

5.1.1 EDA

Some notable observations upon performing exploratory data analysis were as follows

- The proportion of male and female applicants was nearly the same for both approved and unapproved loans. Since it's also unethical to deny loans based on gender, we dropped the gender attribute.
- Graduates have higher proportions of approved loan
- Nothing could be inferred from the Self Employed graph, thus we dropped it.

- Applicants having a Credit History of 1 have a far higher approval percentage than applicants having a Credit History of 0.
- The proportion of loans getting approved in semi-urban areas is higher as compared to that in rural or urban areas.
- Loan Status has a good correlation with only Credit History and has a weak correlation with other numerical attributes.

5.1.2 Random Forest Algorithm

1. We were able to successfully split our data into train and test using a proposed methods.
2. **Hyperparameter values:** Upon using the *GridSearchCV*, we were able to get the best suited values for our chosen hyperparameters which were:
   - max_depth=7
   - n_estimators=21
   - random_state=90
3. Our model was successfully trained using the hyperparameter values mentioned above.

5.1.3 XGBoost Algorithm

1. We were able to successfully split our data into train and test using a proposed methods.
2. **Hyperparameter values:** Using *GridSearchCV*, we were able to get the best hyperparameters:
   - max_depth=19
   - colsample_bylevel=0.9
   - colsample_bytree=0.3
   - subsample=0.1
3. We successfully trained our model with the above hyperparameter values.

5.1.4 Decision Tree Algorithm

1. We were able to successfully split our data into train and test using a proposed methods.
2. **Hyperparameter values:** Using *GridSearchCV*, we were able to get the best hyperparameters:

- max_depth=2

3. We successfully trained our model with the above hyperparameter values.

5.2 Outcomes of the evaluation metrics

5.2.1 Random Forest Algorithm

Upon successfully training our model, we were able to get the values of our evaluation matrix using the sklearn.metrics library. The results were:

- Accuracy = 89.43%
- F1-Score = 93.26%
- Precision = 89.1%

	Predicted: Yes	**Predicted: No**
Actual: Yes	90 (TP)	2 (FN)
Actual: No	11 (FP)	20 (TN)

Table 1: Resultant Confusion Matrix

5.2.2 XGBoost Algorithm

After training our model, we obtained the evaluation metrics values using the sklearn.metrics library. The results were:

- Accuracy = 80.54%
- F1-Score = 87.41%
- Precision = 79.11%

	Predicted: Yes	**Predicted: No**
Actual: Yes	125 (TP)	3 (FN)
Actual: No	33 (FP)	24 (TN)

Table 2: Resultant Confusion Matrix

5.2.3 Decision Tree Algorithm

After training our model, we obtained the evaluation metrics values using the sklearn.metrics library. The results were:

- Accuracy = 81.3%
- F1-Score = 88%
- Precision = 80.76%

	Predicted: Yes	**Predicted: No**
Actual: Yes	84 (TP)	3(FN)
Actual: No	20 (FP)	16 (TN)

Table 3: Resultant Confusion Matrix

5.3 Comparative analysis of the proposed method with current state of art and existing techniques

5.3.1 Comparing with Wells Fargo XAI Algorithm

Wells Fargo is known to develop and utilise a specific type of machine learning for loan prediction called Explainable AI (XAI). This differs from some of the general algorithms mentioned earlier. While traditional machine learning models can be highly accurate, their inner workings can be like a black box - difficult for humans to understand. XAI, as the name suggests, aims to create models that are accurate and interpretable ([10]). This allows Wells Fargo to:

1. Demonstrate fairness: By understanding the factors influencing the model's decisions, Wells Fargo can ensure no bias against certain demographics in loan approvals.

2. Explain decisions to customers: The bank can provide more precise explanations to borrowers about why their loan application was approved or rejected.

According to Wells Fargo's research, their XAI model, called LIFE, offers high accuracy and interpretability, making it more valuable than traditional machine learning algorithms for loan prediction.

5.3.2 Comparing with Support Vector Machines (SVMs)

The primary concept behind SVMs is to identify a hyperplane in an N-dimensional space ([11]), where N represents the number of features that divide the data points into distinct classes. The most optimal hyperplane that separates the classes is the one with the maximum margin, which is essentially the farthest distance between data points of both classes. SVMs can be powerful for certain datasets but are extremely slow to train on large datasets and require careful tuning of the kernel and regularization parameters ([12]). XGBoost, Decision Trees and Random Forest can handle large datasets and are less sensitive to parameter tuning, making them a more practical choice for loan prediction tasks.

5.3.3 Comparing with Logistic Regression

Logistic Regression is a fast and efficient algorithm that assumes a linear relationship between features and the target variable, making it a highly reliable choice for predictive modelling ([13]). However, XGBoost, Random Forest and Decision Trees, being tree-based models, can capture complex non-linear relationships, often resulting in superior loan prediction performance.

6. Conclusion

6.1 Summary

In this project, we created and assessed a model for chances of loan acceptance. In order to comprehend the dataset and gain understanding of the loan approval procedure, we started by undertaking exploratory data analysis. In order for address missing values, we replaced them with the suitable values. We also performed scaling and log transformation to deal with noise and outliers. Then, we trained several classification models on our dataset including the Random Forest Classifiers, the XGBoost Classifier and the Decision Tree Classifier. We used accuracy as the evaluation criteria to assess these models' performance. Based on our findings, we discovered that the Random Forest Classifier outperformed the other models and had the greatest accuracy among all of our other models. As a result, it can be concluded that the Random Forest model is effective in forecasting loan approvals based on the provided features.

6.2 Future scope of work

Although our model performed decent results, there is still a scope of some improvement in the future ([14]):

1. Ensemble Approaches: Combining the predictions of various models may help in improving the performance of the overall model.

2. Expanding our dataset: We can expand our dataset by adding more relevant entries to it and also bby adding other important attributes, and train our model on the same.

3. Deployment and Monitoring: We can deploy our model and constantly monitor it to constantly keep a check of model accuracy and constantly training it to make it sustainable in the dynamic financial conditions.

References

[1] A. T. Rahman, M. R. H. Purno & S. A. Mim, 'Prediction of the approval of bank loans using various machine learning algorithms,' in *2023 IEEE World Conference on Applied Intelligence and Computing (AIC)*, Sonbhadra, India, 2023, pp. 272–277. DOI: 10.1109/AIC57670.2023.10263880.

[2] C. N. Kumar, D. Keerthana, M. Kavitha & M. Kalyani, 'Customer loan eligibility prediction using machine learning algorithms in banking sector,' in *2022 7th International Conference on Communication and Electronics Systems (ICCES)*, Coimbatore, India, 2022, pp. 1007–1012. DOI: 10.1109/ICCES54183.2022.9835725.

[3] P. Tumuluru, L. R. Burra, M. Loukya, S. Bhavana, H. M. H. CSaiBaba & N. Sunanda, 'Comparative analysis of customer loan approval prediction using machine learning algorithms,' in *2022 Second International Conference on Artificial Intelligence and Smart Energy (ICAIS)*, Coimbatore, India, 2022, pp. 349–353. DOI: 10.1109/ICAIS53314.2022.9742800.

[4] JavaTpoint, *Machine learning random forest algorithm*, https://www.javatpoint.com/machine-learning-random-forest-algorithm, 2022. [Online]. Available: https://www.javatpoint.com/machine-learning-random-forest-algorithm (visited on 2024).

[5] Monis Khan, *Explain the step by step implementation of xgboost algorithm.* https://www.csias.in/explain-the-step-by-step-implementation-of-xgboost-algorithm/, 2021. [Online]. Available: https://www.csias.in/explain-the-step-by-step-implementation-of-xgboost-algorithm/.

[6] scikit-learn developers, *Sklearn.ensemble.randomforestclassifier*, https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html, 2022. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html (visited on 2024).

[7] Aarshay Jain, *Xgboost — how does this work*, https://www.analyticsvidhya.com/blog/2016/03/complete-guide-parameter-tuning-xgboost-with-codes-python/?, 2024. [Online]. Available: https://www.analyticsvidhya.com/blog/2016/03/complete-guide-parameter-tuning-xgboost-with-codes-python/?.

[8] TutorialsPoint, *What are the characteristics of decision tree induction?* https://www.tutorialspoint.com/what-are-the-characteristics-of-decision-tree-induction, 2022. [Online]. Available: https://www.tutorialspoint.com/what-are-the-characteristics-of-decision-tree-induction (visited on 2024).

[9] Prathamesh Sonawane, *Xgboost — how does this work*, https://medium.com/@prathameshsonawane/xgboost-how-does-this-work-e1cae7c5b6cb, 2023. [Online]. Available: https://medium.com/@prathameshsonawane/xgboost-how-does-this-work-e1cae7c5b6cb.

[10] S. Martin, *Wells fargo examines explainable ai for modeling lending risk*, https://blogs.nvidia.com/blog/wells-fargo-examines-explainable-ai-for-modeling-lending-risk/, Apr. 2021. [Online]. Available: https://blogs.nvidia.com/blog/wells-fargo-examines-explainable-ai-for-modeling-lending-risk/ (visited on 2024).

[11] N. A. Bhagwat, *Designing single guide rnas for crispr/cas9*, 2019. DOI: 10.31979/etd.6zqq-pdc4. [Online]. Available: https://doi.org/10.31979/etd.6zqq-pdc4.

[12] JavaTpoint, *Machine learning support vector machine algorithm*, https://www.javatpoint.com/machine-learning-support-vector-machine-algorithm, 2022. [Online]. Available: https://www.javatpoint.com/machine-learning-support-vector-machine-algorithm (visited on 2024).

[13] JavaTpoint, *Logistic regression in machine learning*, https://www.javatpoint.com/logistic-regression-in-machine-learning, 2022. [Online]. Available: https://www.javatpoint.com/logistic-regression-in-machine-learning (visited on 2022).

[14] V. Viswanatha & A. Ramachandra, 'Prediction of loan approval in banks using machine learning approach,' *International Journal of Engineering and Management Research*, 13, pp. 7–19, 2023. DOI: 10.31033/ijemr.13.4.2.