Predicting Financial Inclusion in India: A Comparative Analysis of Machine Learning Models

Anushka[1] **and Puneeth**[2]

[1]PhD Candidate, Department of Humanities and Social Sciences, IIT Roorkee; [2]BS-MS Dual Degree, Department of Humanities and Social Sciences, IIT Roorkee

This study leverages the World Bank Global Findex 2021 microdata to predict financial inclusion using machine learning models, from decision trees to neural networks so as to improve efficiency of welfare schemes provision. All ML models performed better than logistic regression, and particularly DT based methods were best suited for this classification problem.

Decision Trees | Ensemble | SVM | Neural Networks

The effectiveness of contemporary welfare in India is mainly centered on Direct Benefit Transfer schemes, which depend directly on the financial inclusion of its intended beneficiaries.The *JAM* trinity (*Jan Dhan–Aadhaar–Mobile*) assumes that individuals not only hold bank accounts but can also use them reliably to receive digital payments. Major programs such as *PM-KISAN (income support for farmers), PAHAL (LPG subsidy reform)*, and the *National Social Assistance Program* channel their benefits solely through these bank linked digital platforms. When specific groups such as low-income rural women or informal-sector workers lack access to such accounts, the resulting *exclusion errors* weaken the impact of welfare policies and risk deepening existing inequalities.

Predicting financial inclusion using observable socioeconomic and demographic indicators therefore becomes necessary and it allows policymakers to estimate, ex ante, whether a target population has the capacity to absorb digital welfare transfers. Thus, accurate prediction supports better policy delivery, minimizes leakages, and enhances the impact of welfare spending.

Introduction

Financial inclusion is at the center of India's development agenda, influencing not only the welfare of households, but also the success of public programs that increasingly rely on financial channels. In this study, we examine the determinants of financial inclusion in India by leveraging the 2021 World Bank Global Findex microdata. The central research question is: **Can we reliably predict whether an individual holds a financial account based on their demographics, socioeconomic position, financial behaviour, and access to technology?** The answer has direct implications for how the state designs and targets welfare schemes that depend on account based transfers.

To address the research question, we apply ML models that begin with data preparation and feature engineering, including cleaning missing values, standardising variables, and encoding key factors such as gender, income, education, employment, and access to digital technology. Using this processed dataset, machine learning models are applied: Decision Trees, Ensemble Methods (Random Forest, Extra Trees, AdaBoost, Gradient Boosting, and Bagging), Support Vector Machine, and a feedforward neural network (FNN/MLP). Each model's performance is then evaluated using accuracy, Precision, Recall, F1-score, and ROC-AUC. Comparing these results along with how interpretable each model is, we draw final conclusions about which methods predict financial inclusion most effectively.

Dataset

For the analysis we used the Indian subsample of the World Bank's Global Findex 2021 dataset, a widely used global survey that provides information on how adults access and use formal and digital financial services. Using the raw microdata (3,000 observations and 119 variables) [*], each respondent is weighted through the survey weight variable `wgt` to reflect the broader population. Several potential measures of financial inclusion are available in the dataset – **account**, `account_fin`, `account_mob`, `anydigpayment`, and `merchantpay_dig`. Among these, we select `account` as the target variable, as it best captures whether an individual possesses any formal financial account (bank, cooperative, microfinance institution, post office) or a mobile money account. This measure captures other variables (such as digital payments) and is also consistent with metrics used by the World Bank, the Reserve Bank of India, and NITI Aayog. The binary nature of the target variable aligns with the classification models used in this project.

The features are classified into four broad categories: demographic characteristics (e.g., `female`, `age`, `educ`), socioeconomic indicators (e.g., `inc_q`, `emp_in`, `urbanicity_f2f`), technology and access variables (e.g., `mobileowner`, `internetaccess`), and financial behaviours over the past year (e.g., `saved`, `borrowed`, `receive_wages`, `pay_utilities`, `receive_transfers`, `remittances`, `receive_pension`, `receive_agriculture`). These variables are drawn from a combination of direct survey responses and constructed indicators derived from multiple questionnaire items.[†]

Following data preprocessing, we narrow the feature set to 17 key variables, along with the survey weight `wgt`, for use in the subsequent modelling stage, as shown in Figure 1.

[*] Raw data can be accessed here: https://github.com/PuneethRaavi/ML_Project/blob/main/Data/RawData.csv or https://microdata.worldbank.org/index.php/catalog/4653/data-dictionary/F1?file_name=micro_ind.dta

[†] Variable definitions and questionnaire are documented here: https://microdata.worldbank.org/index.php/catalog/4653/related-materials

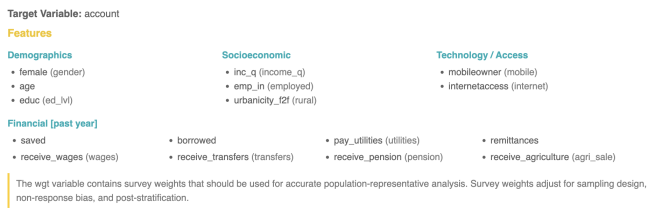

Fig. 1. Overview of the feature space selected for the predictive models.

1. Data Preprocessing

For the analysis, we start with the Indian subsample of the Global Findex 2021 dataset, which includes 3,000 observations and 119 variables. Each respondent comes with a survey weight (`wgt`) that we keep for population level inference.

The preprocessing steps are straightforward (refer to `DataPreProcessing.ipynb`): We first drop variables that serve no purpose in a single country study, such as `economy` and `economycode`. We then remove the large set of questionnaire-level items (those beginning with `fin...`) as these are used only to build the constructed financial indicators and are the source of all missing values in the raw data. Since the constructed indicators are already present and complete, keeping the questionnaire items is unnecessary. Next, we remove a few constructed variables `account_fin`, `account_mob`, `anydigpayment`, and `merchantpay_dig` because they overlap with the dependent variable, `account`. As explained earlier, `account` is the broadest and most widely used measure of financial inclusion, so keeping the redundant variants does not add value. After removing irrelevant fields, no missing values or duplicates remain. The weight variables are rounded to two decimals, and a few variable names are simplified for readability (e.g., renaming `urbanicity_f2f` to "rural").

Fig. 2. Missing value pattern in the raw dataset.

Figure 2 shows the missing value pattern, which confirms that all missing values come from the dropped questionnaire variables. This final feature set, shown in Figure 1, forms the basis for upcoming sections.

2. Feature Engineering

Feature engineering is necessary to ensure compatibility with both linear and non-linear learning algorithms. After preprocessing, the dataset contained 18 variables: 16 categorical predictors, one continuous predictor (`age`), and the survey weight `wgt`. (refer to `FeatureEngineering.ipynb`):

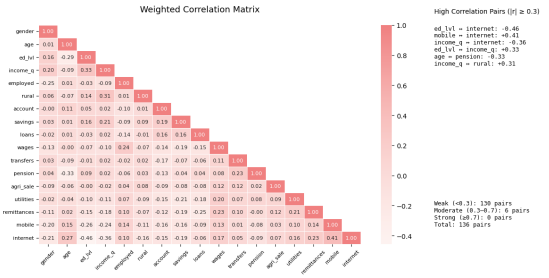

Fig. 3. Correlation matrix for all the variables.

We first examined the predictors using a correlation heatmap, shown in Figure 3. Correlations across features were generally weak—only six out of 136 feature pairs showed mild association. Since all correlations were far below multicollinearity thresholds, dimensionality-reduction methods such as PCA was unnecessary. Mild correlation is easily absorbed by regularized linear models (e.g., SVMs, penalized logistic regression) and is mostly irrelevant for tree-based methods. Next, we classified the predictors into categorical, ordinal, and continuous types. The categorical block included variables such as `gender`, `employed`, `mobile`, `internet`, and several other financial behaviour indicators. Two variables—`ed_lvl` and `income_q` were treated as ordinal. Based on this classification subsequent encoding, merging, and scaling are done.

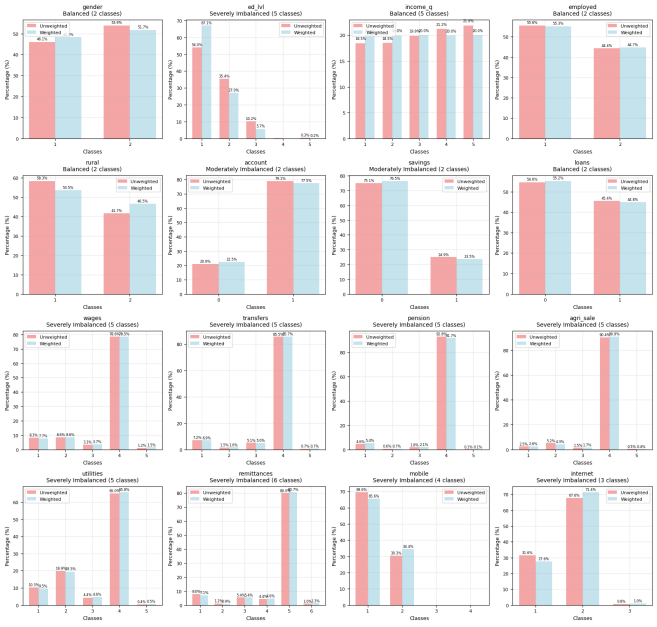

Fig. 4. Correlation matrix for all the variables.

Class distributions are then inspected for the dependent variable and all predictors. The target variable `account` showed only slight imbalance, which is addressed during model training using cost-sensitive learning. Several categorical predictors contained very low frequency categories ("don't know" or "refused" responses under 1%). These are removed to avoid rare category instability. Additionally, rare but meaningful categories were merged: higher education levels are combined into a unified "tertiary or above" group, and categories 2 and 4 in the `remittances` variable are merged due to conceptual similarity and low frequency. For certain financial behaviour variables (e.g., `wages, transfers, utilities, pensions, agricultural sales`), the original three-level structure (1, 2, 3) was consolidated into a binary indicator (sent/received vs. not), improving class balance without sacrificing interpretability. Updated category distributions are presented in Figure 4.

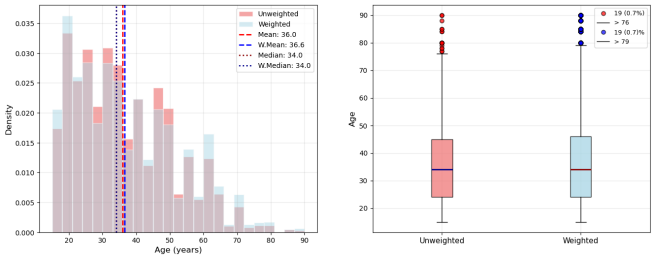

Fig. 5. Age distribution on the left and box plots on the right.

The continuous variable `age` required additional cleaning. A small number of observations above age 76 (0.3% of the sample) are removed, given life expectancy of India is below 72 years and the presence. The age distribution showed mild right-skewness, to resolve this we applied square-root transformation to construct the variable `age_sqrt`. This transformation was selected because the skew was moderate and the square root preserves more information than log based transformations. Tree based algorithms use the raw `age` variable, while SVMs, logistic regression, and neural networks use the transformed version. The distribution of age before trimming and its outliers are shown in Figure 5 and can be compared to its distribution after transformation as showm in Figure **??**.

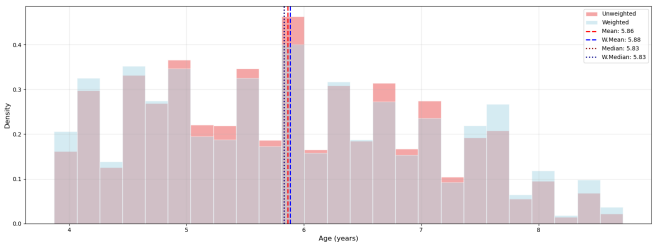

Fig. 6. Age distribution on the left and box plots on the right.

Categorical variables were encoded according to their structure. Binary variables are re-encoded into a standard 0/1 format, while multiclass nominal variables are one-hot encoded with the first category dropped to avoid collinearity. Ordinal predictors (`educ`, `inc_q`) natural ordering are preserved. For scaling, MinMax normalization is applied to the continuous and ordinal predictors (`age_sqrt`, `educ`, `inc_q`) when training models is sensitive to feature scale—SVMs, logistic regression, and neural networks. Tree based models (Decision Trees, Random Forests, Extra Trees, AdaBoost, Bagging, and Gradient Boosting) are trained on the unscaled feature matrix.

Methodology

After preprocessing, the dataset is split into training and testing sets, 25% data is used for testing.[‡] To capture patterns in the data, we use the following models: Logistic Regression, Decision Trees, Ensemble methods (Random Forest, Gradient Boosting, AdaBoost, and Bagging), Support Vector Machine with kernels, and a (MLP) neural network.

Models are tuned using `GridSearchCV` with cross-validation, which allows to find best hyperparameters such as tree depth, number of estimators, learning rates, kernel parameters, and others. Because the target variable shows mild imbalance, cost-sensitive learning is used by assigning higher penalties to errors involving the minority class. This helps balance the relative impact of false positives and false negatives.

1. Logistic Regression

To establish a benchmark, we utilized Logistic Regression. When the model says 'No account', it is right about 82% of the times where its precision is comparatively low around 35%. It is unable to find people who actually have no account. F1 score is also moderate. Accuracy is around 65%. When the model predicts that a person has an account, precision rises to 93% whereas recall falls to 60% and f1 scores improve . This is happening because data is imbalanced.

Table 1. Logistic Regression

Class	Precision	Recall	F1-Score	Support
No account	0.35	0.82	0.49	149
Have account	0.93	0.60	0.73	570
Accuracy	—	—	0.65	719
Macro Avg	0.64	0.71	0.61	719
Weighted Avg	0.81	0.65	0.68	719

2. Decision Tree

Decision Trees provide an intuitive non-linear modelling approach that partitions the feature space into simple decision rules. Their interpretability makes them robust for understanding the structure of financial inclusion in the data. Hyperparameters are tuned using an extensive grid search with `RepeatedStratifiedKFold` cross-validation (5 folds, 3 repetitions). The search grid spans 250 combinations of `max_depth` $\in \{2, 4, 10, 20, \text{None}\}$, `min_samples_split` $\in \{2, 4, 10, 20, 50\}$, `min_samples_leaf` $\in \{2, 4, 10, 20, 50\}$, and {`gini`, `entropy`} criteria, resulting in 3,750 fitted models.

For both accuracy and F1, the best-performing configuration is a relatively shallow tree (`criterion=gini`, `max_depth=4`, `min_samples_leaf=20`, `min_samples_split=2`). In contrast, ROC–AUC favours a deeper but more regularised structure (`criterion=entropy`, `max_depth=10`, `min_samples_leaf=20`,

[‡] Random state is kept 24 throughout the modelling analysis for reproducibility.

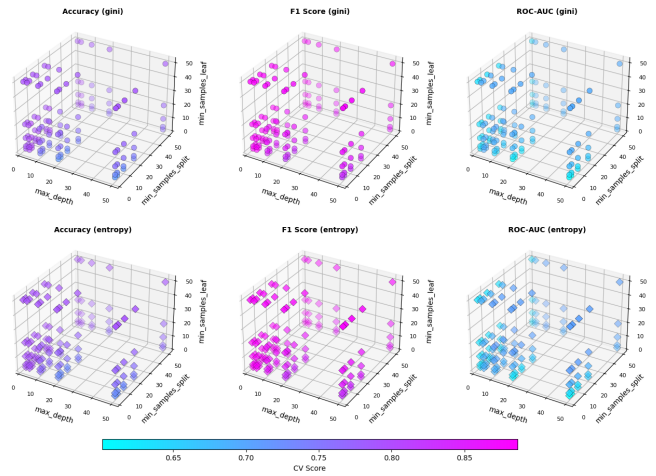

Fig. 7. 3D plot with different hyperparameters and scoring criteria.

`min_samples_split=50`). To understand these patterns, we visualise the three key hyperparameters (`max_depth`, `min_samples_leaf`, `min_samples_split`) in a 3D plot, shown in Figure 7). [§] The plot reveals three regularities: (1) performance declines steadily as `min_samples_split` increases; (2) `max_depth` and `min_samples_leaf` act as substitutes, deeper trees require larger leaf sizes to avoid overfitting and (3) Gini and Entropy criteria give nearly identical scores.

Based on these results and prioritising interpretability, we use the following hyperparameters for the final model: `criterion=gini, max_depth=10, min_samples_leaf=50, min_samples_split=2, min_impurity_decrease=0.001`. This gives a balance between F1 and ROC-AUC performance while protecting against overfitting. Decision Tree is given in Figure 8

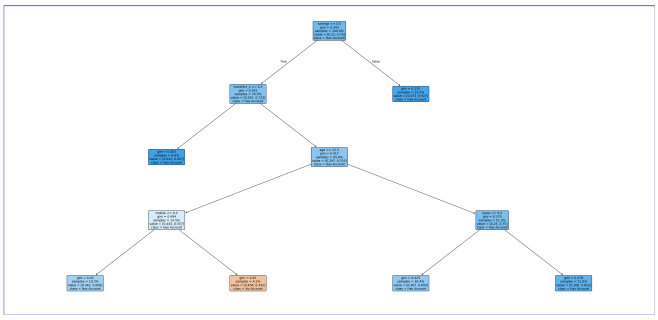

Fig. 8. Age distribution on the left and box plots on the right.

The resulting confusion matrix shows strong performance for the "included" class, but the model generates a high number of false positives (Type I errors):

$$\text{FP} = 131, \quad \text{FN} = 17, \quad \text{Type I} = 0.873, \quad \text{Type II} = 0.030.$$

To explore whether threshold adjustment can reduce false positives or false negatives, we evaluate three cost scenarios: (1) balanced costs ($c_{\text{FP}} = c_{\text{FN}} = 1$), (2) high false-negative cost ($c_{\text{FN}} = 2.5$), (3) high false-positive cost ($c_{\text{FP}} = 2.5$). The optimal thresholds and performance metrics are shown in Figure 9, and the threshold performance curves are plotted in Figure 10.

[§] An interactive version is included in the notebook `DecisionTree.ipynb`.

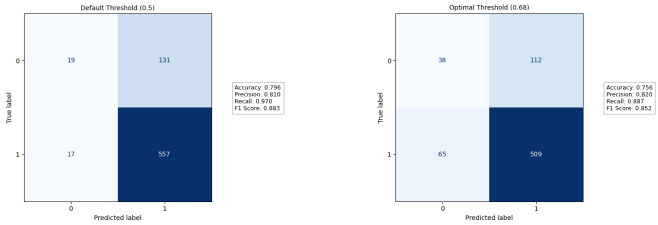

Fig. 9. Confusion Matrix comparision with different threshold from cost optmisation

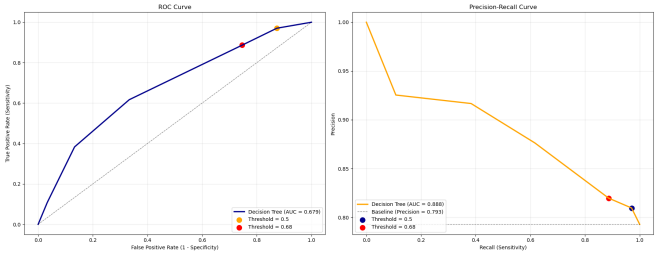

Fig. 10. ROC-AUC curves at different thresholds.

Two findings stand out. First, increasing the cost of FN shifts the model toward the positive class, eliminating FN entirely without reducing accuracy, this is desirable feature in contexts where missing an excluded individual is costly. Second, raising FP cost does not reduce false positives. The ROC curve in Figure 10 shows limited ability to trade off FP for precision without sharply lowering recall. This reflects the structure of the data, where the non-account class is extremely sparse and concentrated among specific subgroups.

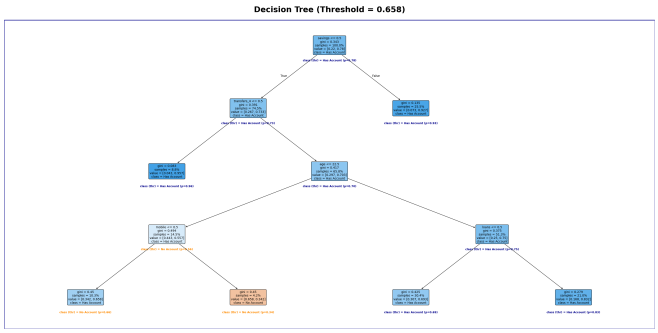

Fig. 11. Desicion Tree predictions for cost optimised threshold.

The structure of the fitted tree in Figure 8 gives a clear view of how the model distinguishes financially included and excluded individuals. Under the balanced threshold setting, the tree identifies only a single terminal node corresponding to the "no account" class. This reflects the high prevalence of financial inclusion in the dataset: almost all decision paths terminate in the "included" class. The one exclusion node is intuitive, it captures young individuals(below 23 years of age) who do not engage in any financial activity such as saving, borrowing, or receiving transfers, despite having access to a mobile phone. This group represents the small but distinct segment of financially excluded youth in the sample. When the hyperparameters

are relaxed to allow deeper trees (e.g., `max_depth=10`) while maintaining regularisation through larger leaf sizes additional "no account" nodes emerged, largely driven by low income quintile, lack of internet access, and absence of borrowing history.

Figure 11 shows how predictions shift when we adjust the decision threshold under different cost scenarios. Under the high false-negative cost ($c_{\mathrm{FN}} = 2.5$), the model becomes conservative and predicts the positive class more, eliminating false negatives entirely. This comes at the cost of higher false positives, but it is useful in contexts where missing is more costly than classifying an included one.

Finally, calibration curves (Figure [P8]) show that the tree is well calibrated at low predicted probabilities but tends to overestimate inclusion at higher probabilities, reflecting the high baseline prevalence of account ownership in the sample.

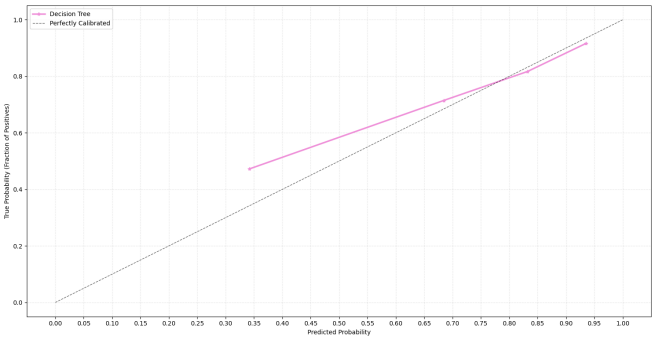

Fig. 12. Age distribution on the left and box plots on the right.

While the Decision Tree give clear interpretability, its ability to reduce false positives is limited. Threshold adjustments improve false negatives but do not reduce false positives, suggesting that deeper nonlinear models may be better suited for capturing the boundary of financial exclusion. In the upcoming sections we will just see the results from fitting the models and compare it in the final section.For Detailed interpretation, check out the DecisionTree.ipynb file

3. Ensemble Methods

The ensemble models build directly on the structure learned by the baseline Decision Tree, but combine multiple weak learners to improve stability and predictive power. As shown in Table 2, all ensemble methods deliver performance very similar to the tuned Decision Tree, but with modest gains in recall and F1-scores. Random Forest, Extra Trees, and Gradient Boosting perform particularly well, each achieving F1-scores around 0.885. These models are able to capture more complex patterns in the data than a single tree, while still inheriting its interpretability at the aggregate level. Notably, AdaBoost attains perfect recall, correctly identifying all financially included individuals, although this comes at the cost of lower precision due to a higher number of false positives. Bagging, which simply averages multiple trees trained on bootstrap samples, performs slightly worse than the more sophisticated ensemble variants but remains competitive.

The ROC curves for these models, presented in Figure 13, reinforce these observations. All ensemble learners show broadly

Table 2. Comparative Performance Metrics of Ensemble Models

Model	Accuracy	Precision	Recall	F1-Score
Bagging	0.7762	0.8209	0.9181	0.8668
Random Forest	0.7970	0.8011	0.9895	0.8854
Extra Trees	0.7970	0.8037	0.9843	0.8849
AdaBoost	0.7928	0.7928	1.0000	0.8844
Gradient Boosting	0.8011	0.8190	0.9617	0.8846

similar discrimination performance, with only marginal differences in their ability to separate included and excluded individuals.

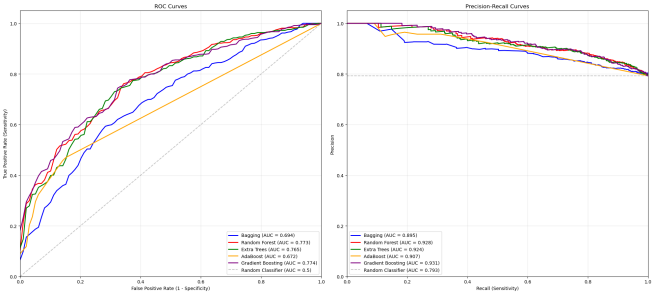

Fig. 13. Age distribution on the left and box plots on the right.

4. Support Vector Machine

Now after ensemble learning models, we have moved to Support Vector machines. Here the first step would be kernel selection. We have taken three types of kernels : linear,rbf and poly kernels. First we are training baseline SVMs.

Table 3. Performance Comparison of Different SVM Kernels

Kernel	Test Accuracy	ROC-AUC	CV Mean
Linear	0.7928	0.7453	0.7928
Poly	0.7983	0.7197	0.8016
RBF	0.7955	0.7183	0.7951

Here accuracy for all models is almost similar.Now we use grid search to find best parameters for the SVM classifiers. Now the model is trying to treat both classes equally. C=1, means there is moderate regularisation. The class weights are balanced and decision boundary is linear. The test accuracy is low i.e around 58%. But ROC score is better We need to compare precision,Recall and F1 score.

When the model says 'No account', it is right about 86% of the times where its precision is very low. It is unable to find people who actually have no account. F1 score is also low. When the model predicts that a person has an account, Precision ,recall and f1 scores improve . This is happening because data is imbalanced. The model learns class 1 better than class 0.

The total number of support vectors used to define the decision boundary is $1,427$, indicating that the data is not easily separable. Class 0 (No account) comprises 299 support vectors, whereas Class 1 requires a significantly higher number, reflecting its status as the majority class. regarding feature

Table 4. SVM Metrics

Class	Precision	Recall	F1-Score	Support
No account	0.31	0.86	0.46	149
Have account	0.93	0.51	0.66	570
Accuracy	—	—	0.58	719
Macro Avg	0.62	0.68	0.56	719
Weighted Avg	0.80	0.58	0.62	719

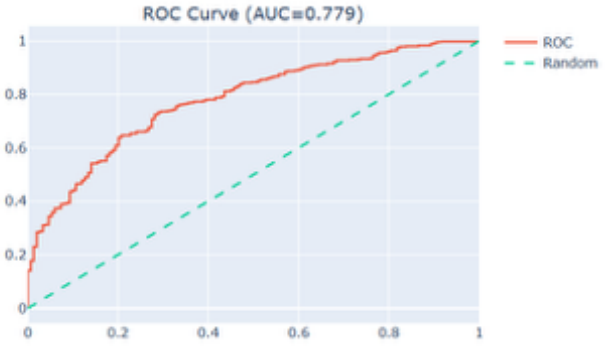

Fig. 14. ROC curve for SVM model.

importance, the SVM model identifies `utilities_4` (did not make utility payments) as the most important variable. This is followed by `transfers_4` (did not receive transfers) and `wages_4` (did not receive wages) as the second and third most significant predictors, respectively.

5. Neural Networks

Out of all neural networks we have chosen Feed Forward Neural network. Since it is best suited for our Tabular and structural data. Our data do not have temporal dependencies. Now we run a simple feed forward neural network. Input layers are the number of features in the dataset. For nonlinearity in hidden layers we use ReLu activation function. The output layer uses sigmoid function which uses binary classification since we are predicting whether a person has an account or not. Now we train the Neural Network using scaled data

Table 5. FNN Metrics

Class	Precision	Recall	F1-Score	Support
No account	0.33	0.74	0.46	149
Have account	0.90	0.61	0.73	570
Accuracy	—	—	0.64	719
Macro Avg	0.61	0.67	0.59	719
Weighted Avg	0.78	0.64	0.67	719

When the model says 'No account', it is right about 74% of the time whereas its precision is again very low(33%). It is unable to find people which actually have no account. F1 score is also low(0.46). When the model predicts that a person has an account, Precision ,recall and f1 scores improve ; 0.90,0.61 and 0.73 respectively. This is happening because data is imbalanced. And there is the problem of overfitting.

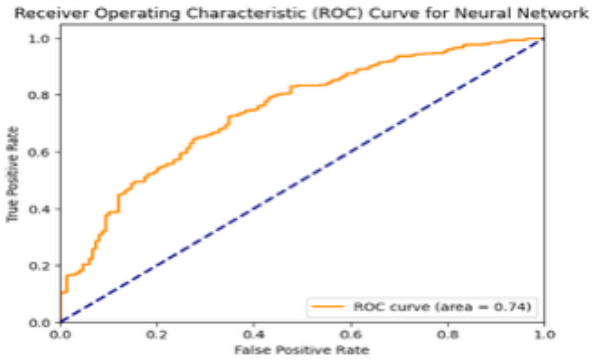

Fig. 15. ROC curve for FNN, Neural Network model.

Discussion

Across all specifications, the machine-learning models outperform the baseline Logistic Regression classifier as seen in Table 6, confirming that nonlinear socioeconomic and financial interactions are better captured by tree based and ensemble methods. Among them, the Decision Tree and ensemble models particularly Gradient Boosting gave the strongest accuracy. However, all models struggled to reliably identify the minority "no-account" class, which constitutes only a small fraction of the sample. Cost-sensitive analysis showed this limitation, increasing the penalty for false negatives causes the Decision Tree to predict nearly all individuals as financially included, while raising the penalty for false positives fails to reduce FP errors without harming accuracy.

Table 6. Model Comparision

Model	Accuracy	F1	Precision	Recall
Logistic Regression	0.65	0.68	0.35	0.82
Decision Tree	0.79	0.88	0.81	0.97
Bagging	0.77	0.86	**0.82**	0.92
Random Forest	0.79	**0.88**	0.80	**0.99**
SVM	0.58	0.62	0.31	0.86
Neural Network (FNN)	0.64	0.67	0.33	0.74
Gradient Boosting	**0.80**	0.88	0.82	0.96

Under balanced-cost thresholds, the Decision Tree performs well and remains highly interpretable. It identified an intuitive split, younger respondents (below 22 years) with mobile access but no savings, borrowing, or payment transfers did not have an account, rest all had an account. This also shows that financial inclusion is high in India and welfare schemes should not ideally be hindered by financial inclusion.

ACKNOWLEDGMENTS. We would like to express our sincere gratitude to Prof. Hari Venkatesh for his instruction during this course and for providing valuable guidance throughout the project.

We also genuinely enjoyed learning throughout this course, it made the project meaningful and helped us appreciate how these modelling ideas work in real policy contexts.