Using ML to Predict Social Impact Bond Outcomes

Serena Yin, Nadia Bozeman, Cyra Alesha, Sophie Hollowell

Table of Contents

01

Problem & Data

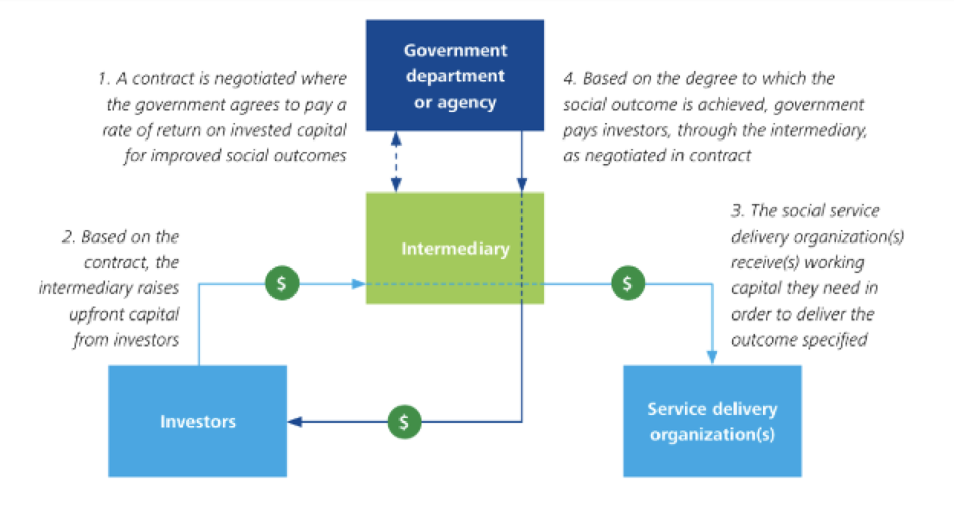
Introduction

Our project seeks to develop a trading strategy that impact investors can utilize to maximize return and profit when investing in Social Impact Bonds.

Background

Social Impact Bonds

- Financing mechanism used to fund social programs
- Private investors are not paid back until the social program is "successful"
- Complex to structure and evaluate
- Relatively few have reached maturity globally



Problem Statement

How can we develop a trading strategy to help advise impact investors when investing in Social Impact Bond to maximize return?

What is the likelihood of the “success” of an SIB?

If an SIB is successful, what is the expected internal rate of return?

Data

Data Overview

- Indigo Impact Bond Dataset
- Curated by the Government Outcomes Lab at the University of Oxford
- Includes records of individual SIB projects globally

Data Features

- Project Characteristics
    - Name, development stage, # of beneficiaries, SDG goals, etc.
- Outcomes and Results
    - Outcome metrics, targets, validation methods, etc.
- Investments and Repayments
    - Investment instruments, repayment structures, etc.
- Timeline and Status
    - Project duration, start and end dates, etc.

Data

Data Quality

- Very few social impact bonds within dataset have reached maturity
- Dataset was sparse and didn't allow for robust analysis due to lack of data points

ChatGPT Supplement

- Generated 4 linked datasets:
    - Projects
    - Outcome Metrics
    - Investments
    - Outcome Payments
- Preserved original project IDs and names for consistency
- Used controlled randomization & domain-informed logic to simulate realistic relationships

Preventing Data Leakage

1

Pre-Investment Data

Simulate real-world conditions using only pre-investment data

2

Outcome-based Features

Removed fields that directly reflect project performance

3

Pre-Contractual Features

Curated inputs limited to pre-contractual or projected values

02

Methodology

SIB Success Prediction Methodology

1. Join projects, outcome metrics, and investments datasets
2. Drop features that would cause data leakage, including actual IRR, other actual outcome finances, and actual outcome metrics
3. Process categorical variables via one-hot encoding
4. Split data using a stratified train-test split as one method to overcome class imbalance
5. Developed three classification models for success prediction: logistic regression, random forest, and XGBoost

SIB Success Models

Model	Rationale	Details
Logistic Regression With Lasso	Used as a baseline to compare performance of other models	Uses the success feature in the test set as the prediction
Random Forest	Captures complex nonlinear patterns and interactions between features, reducing noise	Used 100 estimators and default parameters
XGBoost	Uses gradient boosting to optimize predictions and tends to outperform on tabular data	Trained with n_estimators=100, max_depth=4, and learning_rate=0.1

SIB Success Feature Selection

Feature Selection

Ensemble / Consensus Feature Selection

1. Manually filtered out leaky or post-investment features
2. Selected only features available prior to investment decisions
3. Trained logistic regression with Lasso (L1 penalty) to shrink unimportant features
4. Trained Random Forest and XGBoost, extracted top features
5. Final set = safe, statistically important features

Model Inputs

- Maximum Potential Loss of a Project
- Maximum Potential Return of a Project
- Maximum Potential Outcome Payment
- Potential Return to Loss
- Targeted Number of Service Users
- Return on Investment Estimate

Model Output

- Success (binary target feature)

IRR Prediction Methodology

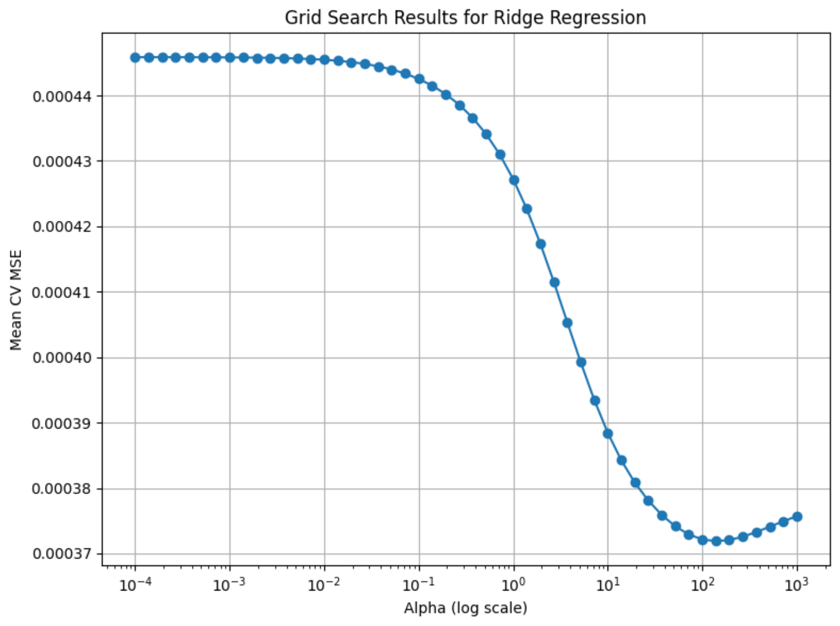
1. Merge projects, outcome metrics, and investments datasets, and filter for only successful projects
2. Drop features that would cause data leakage, including actual outcome payment amounts and actual outcome metrics
3. Process categorical variables via one-hot encoding
4. Split data and develop a naïve average predictor as a baseline model
5. Develop regression models to compare against the baseline

IRR Models

Model	Rationale	Details
Naïve Average Predictor	Used as a baseline to compare performance of other models	Uses the average IRR in the test set as the prediction
Linear Regression	Easily interpretable and implementable	Input features were standardized using StandardScaler
Lasso Regression	Applied to perform feature selection	Optimized alpha = 0.001
Ridge Regression	Address potential overfitting issues	Optimized alpha = 10
Decision Tree	Simple, interpretable nonlinear model	Tree depth = 1 | Min. Sample Leaves = 39
Bagged Trees	Bagging applied to decision tree to address potential overfitting	Number of Trees = 50
Random Forest	Handle noisy features and improve generalization performance	Num of Trees = 50 | Max Tree Depth = 5 | Min. Sample Leaves = 58
XGBoost	Gradient boosting method to capture complex nonlinear patterns and interactions	Max Depth = 3, N Estimators = 50

IRR Model Parameters

Model	Parameters Tuned
Lasso Regression	• Optimal stepsize • GridSearch CV with 5-fold cross validation
Ridge Regression	• Optimal stepsize • GridSearch CV with 5-fold cross validation
Decision Tree	• Tree Depth & Min. Samples in Each Leaf • GridSearch CV with 5-fold cross validation
Bagged Trees	• Number of Trees • GridSearch CV with 5-fold cross validation
Random Forest	• Number of Trees & Min. Samles in Each Leaf • GridSearch CV with 5-fold cross validation
XGBoost	• Learning rate, max depth, n estimators • GridSearch CV with 5-fold cross validation



IRR Model Feature Selection

Feature Selection

Ensemble / Consensus Feature Selection

1. Trained all models with all features
2. Extracted feature importance for all models
3. Combined rankings of feature importance by taking union of top 10 from each model
4. Used the union of top features as final features selected

Model Inputs

- Anticipated Project Duration
- Outcome Metric Definition and Target
- Primary SDG goal and targets
- Secondary SDG goal and targets
- Targeted population and # of beneficiaries
- Jurisdiction/ geography

Model Output

- Latest Internal Rate of Return

03

Results

SIB Success Models

Model	Accuracy	Precision (Class 1)	Recall (Class 1)	F-1 Score (Class 1)	ROC AUC Score
Logistic Regression	0.87	0.87	1.00	0.93	0.7477
Random Forest	0.89	0.89	1.00	0.94	0.5682
XGBoost	0.73	0.88	0.8	0.84	0.5159

- XGBoost underperforms relative to the other models
  - May be overfitting, poorly tuned, or not suited to this dataset
- Random Forest shows strongest overall performance
  - Highest accuracy, perfect recall, and strong precision

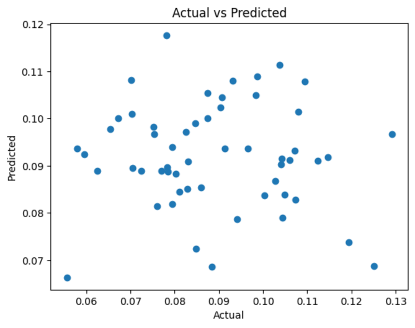
Baseline Comparisons

Model	Accuracy	No. of projects invested in	Estimated Return	Notes
Naïve Classifier	87.30%	63	$4M	High return and accuracy reflects the class imbalance in the dataset.
Monte Carlo	87.21%	18 (random)	~ $1.2M	Important to learn from project features rather than investing randomly.
Logistic Regression	89.83%	59	~$3.57M	Correctly identified 53 projects as successful.
Random Forest	89.13%	46	~$2.7M	Correctly identified 41 projects as successful.
XGBoost	88.89%	36	~$2.23M	Conservative nature likely limited returns.

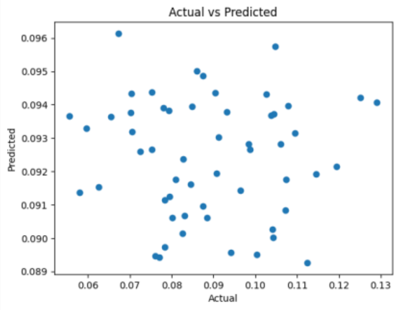
Baseline Comparisons Reflections

- ML models consistently **outperform naive and random baselines** in both accuracy and return
- **Logistic Regression** provided the best return-to-risk balance
- **Naïve accuracy is misleading** due to class imbalance — true learning matters
- **XGBoost was most selective**, but possibly too conservative
- Results reinforce the value of **data-driven project evaluation** for impact investing

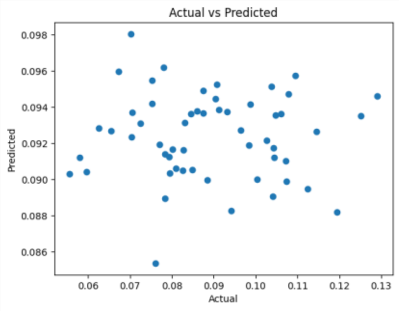
IRR Model Predictions



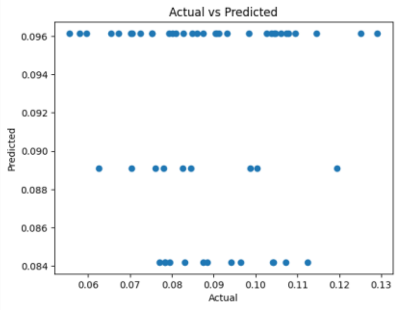
Linear Regression



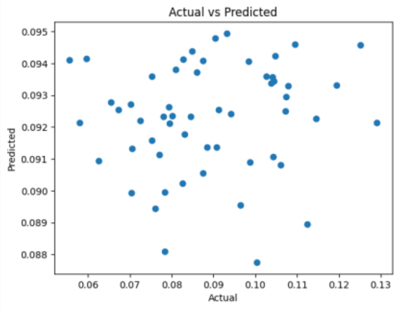
Lasso Regression



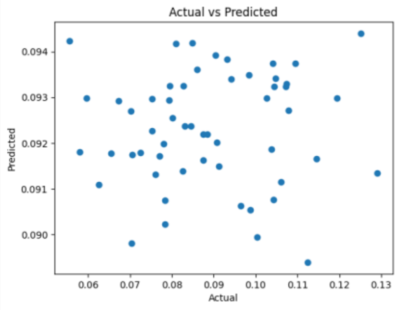
Ridge Regression



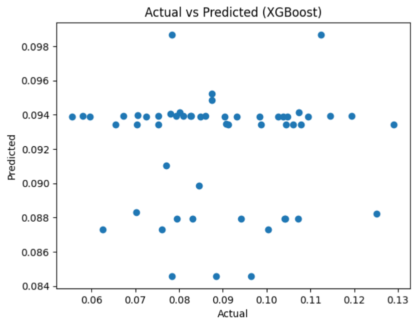
Decision Tree



Bagged Tree



Random Forest



XGBoost

IRR Model Evaluation Metrics

Model	Train MSE	Train RMSE	Test MSE	Test RMSE	Train R-squared	Test R-squared
Bagging	0.000358	0.018918	0.000299	0.017305	0.043143	-0.026616
Random Forest	0.000366	0.019134	0.000302	0.017389	0.021125	-0.036647
Naïve Average	N/A	N/A	0.000303	0.017403	N/A	-0.038309
Lasso	0.000365	0.019092	0.000308	0.017553	0.025400	-0.056338
Ridge	0.000340	0.018436	0.000309	0.017591	0.091290	-0.060918
XGBoost	0.00324	0.017993	0.000314	0.01778	0.134424	-0.075037
Decision Tree	0.000350	0.018713	0.000336	0.018317	0.063782	-0.150214
Linear	0.000283	0.016810	0.000443	0.021039	0.244540	-0.517514

IRR Model Evaluation

Test MSE

Only Bagging and Random Forest Models performed better than the Naïve Average Predictor

Low R-squared

R-squared values are close to 0, indicating low predictive power of selected features

Overfitting

Linear regression had signs of overfitting, while other models had lower errors on the test set than the training set

Overall Investment Strategy

1. Use logistic regression with lasso regularization to predict success of the bond.
2. Utilize Bagging or Random Forest models to predict IRR. Otherwise, investors should assume the historical average for the IRR of SIB.
3. Based on investors' risk appetite, investors can choose to invest in the Social Impact Bond based on whether the bond would achieve their return goals.

04

Conclusion

Reflections

The Challenges

Data Quality

Class Imbalance

Small Dataset Size

Weak Predictive Signal

Our Responses

- In-depth pre-processing
- Implementation of a variety of techniques
- Tested many model types
- Robust pipeline of pre-investment features

Key Takeaways

1

Simpler Models Can Outperform on Imperfect, Real-World Data

2

Framing ML Through an Investor Lens Enhances Relevance

3

Robustness to Imbalance and Noise Can Matter More than Accuracy Alone

Potential Next Steps

Quantify Prediction Confidence

Integrate uncertainty estimates within current models to better represent the confidence that an investor should have in SIB predictions to guide their investing strategy

Enrich Dataset

Expand the current dataset or feature coverage by adding external macroeconomic, policy context, or geographic features that may influence SIB outcomes

Bridge to Real-World Use

Develop investing integration through creating tools to make models applicable to decision-making

Thank You!

Appendix

Further Exploratory Models

Model	Rationale	Details
OLS Linear Regression	Used as a baseline due to its simplicity and interpretability	Used calculated irrAchievedRatio as the target variable
Lasso Regression	Reducing noise of redundant information	Optimized alpha = 0.001
Ridge Regression	Reduce model variance and mitigate multicollinearity	Optimized alpha = 0.1
Feedforward MLP	Learn feature interactions and capture hidden patterns not handled by linear models	Two hidden layers and 30 epochs
MLP With Dropout Layer	Address overfitting which might occur due to small dataset	Dropout layer with a rate of 0.2

Further Exploratory Models

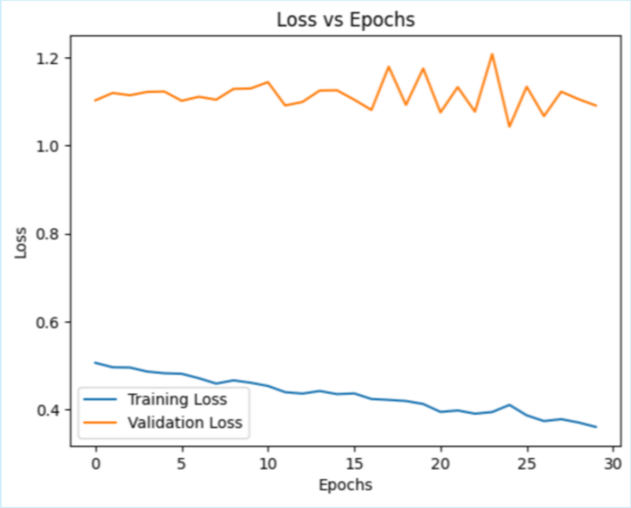
Model	Train MSE	Train RMSE	Test MSE	Test RMSE	Train R-squared	Test R-squared
Linear	0.028031	0.167425	0.028699	0.169407	0.136786	0.054612
Lasso	0.028411	0.168556	0.027655	0.166298	0.125084	0.088992
Ridge	0.028399	0.168518	0.027667	0.166334	0.125467	0.088599

Linear Model Evaluation

- MSE and RMSE similar across models suggests minimal overfitting
- R^2 is very low (~0.09) suggests that linear models explain little of the variation in IRR
- Features lack predictive signaling for IRR suggests that a linear model may not be suitable

Feedforward MLP

Model	Test MSE	Test MAE	R-squared
Feedforward MLP	0.8815	0.7549	0.0571
MLP with Dropout Layer	0.8997	0.7690	0.0375



- Neither MLP Model outperformed the linear model
- Adding a dropout layer further reduced performance
- Dropout is designed to prevent overfitting – but in this case, **underfitting** was the bigger issue
- Dataset size (only **314 samples**) likely limits the model's ability to extract deeper patterns

Exploratory Model Reflections

Investment Strategy Evaluation

- Strategy: invest in the **top 8 projects by predicted IRR Ratio**
- This **improved average return per project** from ~$64K to ~$100k

Limitations and Considerations

- Top-N performance boost is likely due to **granularity of IRR Ratio predictions**, not model precision
- The match between top-ranked predictions and high-return projects may be **coincidental**
- **Narrow top-N selection** can create an illusion of strong performance even with weak models