AI/ML Loan Summarization System

Final Report

Parlay Team 1

Team Members

- JunSoo Jung (junsoo.jung@colorado.edu)
- Dhairya Veera (dhairya.veera@colorado.edu)
- Yashwanth Reddy Thallapalli (yath8924@colorado.edu)
- Sai Preetham Vinnakota (savi3618@colorado.edu)
- Venkata Harshith Nikhil Samudrala (venkataharshithnikhil.samudrala@colorado.edu)

Company and Mentors

Company: Parlay Finanace

Mentors: James (james@parlay.finance), Aneesh Khole (aneesh.khole@colorado.edu)

Contents

1 Abstract

This report outlines the progress made on the AI/ML Loan Summarization System, a project aimed at automating the loan eligibility assessment process. The system leverages machine learning models to evaluate applicant profiles and recommend suitable loan types. The report covers the project's introduction, data analysis, model performance, and preliminary conclusions. The team has made significant strides in data preparation, model selection, and evaluation, with XGBoost and CatBoost emerging as the top-performing models. Future work will focus on hyperparameter tuning and further refining the system for deployment.

2 Introduction

The AI/ML Loan Summarization System is developed to transform the traditional approach to assessing loan eligibility by introducing automation and intelligent decision-making. Historically, loan approvals have relied on manual processes in which lenders evaluate various parameters such as credit scores, income levels, and employment history. These assessments, while functional, are labor-intensive, inconsistent, and prone to human error, especially when scaled across large applicant volumes. The goal of this project is to address these limitations by leveraging machine learning to automate and standardize the eligibility evaluation process. By doing so, the system aims to deliver faster, more accurate, and data-driven decisions that not only enhance operational efficiency but also improve the overall borrower experience.

The primary stakeholders of this system include Parlay, a financial services company, and its clients who depend on seamless and efficient loan processing. Traditionally, loan officers determined applicant eligibility by manually scoring each application based on predefined rules. For instance, an applicant with a credit score above a certain threshold might be granted access to a broader range of loan products. While this rule-based scoring system provided some level of objectivity, it lacked the flexibility, scalability, and predictive accuracy that modern machine learning models offer. As the financial industry moves toward digital transformation, there is a growing need for systems that can evolve with changing market

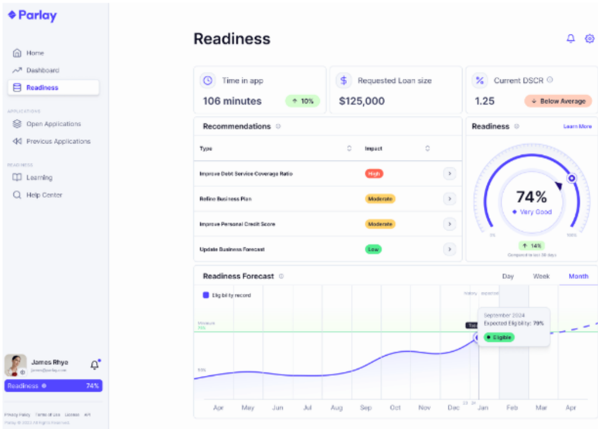

Figure 1: Analysis of Parlay Service View from Borrower

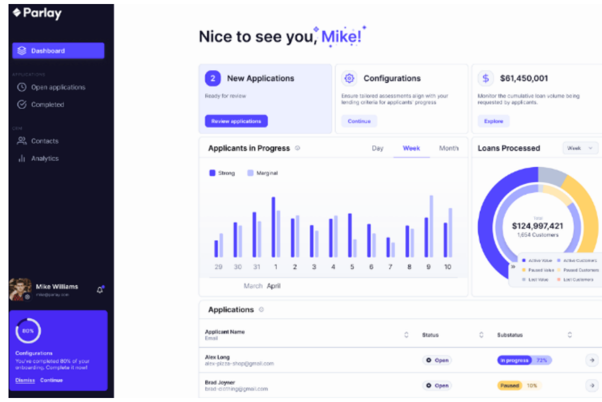

Figure 2: Analysis of Parlay Service View from Lender

demands and regulatory frameworks.

This project brings significant value by reducing manual workload, cutting operational costs, and accelerating the loan approval timeline. Automating eligibility assessments allows financial institutions to reallocate human resources to higher-value functions such as customer engagement and personalized service offerings. Moreover, machine learning models can dynamically adjust to new patterns in data, ensuring the decision-making process remains both compliant and effective over time. The end result is a system that enhances both business performance and customer trust.

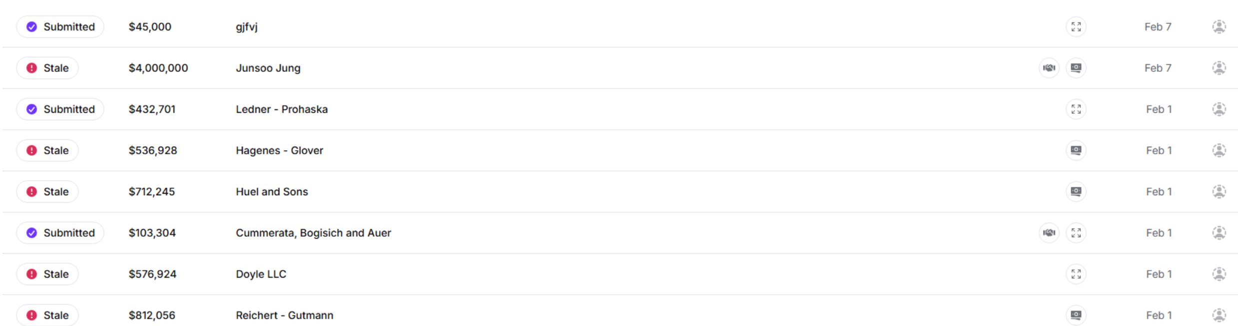

Figure 3: Example of Loan Desicion Making

To train and evaluate the models, the team used synthetic data carefully generated to reflect realistic applicant profiles, incorporating features such as credit scores, income levels, employment history, and other financial indicators. A scoring mechanism was then developed to compute an eligibility score, which serves as the basis for recommending suitable loan products. This approach mimics traditional decision-making logic while introducing statistical rigor and automation.

In the following sections of this report, we will explore the technical implementation in detail, covering the steps taken for data preparation, model selection, and performance evaluation. We will also highlight the technologies used throughout the project and discuss the key challenges encountered during development. Ultimately, this project aims to deliver a scalable, reliable, and intelligent solution for modernizing the loan approval pipeline in real-world financial settings.

3 Analysis

3.1 About the Data

The dataset used in this project is synthetic, designed to simulate real-world loan applicant profiles. It includes a wide range of parameters such as personal credit scores, income levels, employment history, and other financial metrics. The data was provided in a CSV format, with each row representing an applicant and each column representing a specific feature. The dataset also includes an "eligibility" column, which indicates whether the applicant is eligible for specific loan types (e.g., 7A, 504, Express loans). The team was provided with a script to generate synthetic data, which included predefined eligibility rules. However, to improve the model's performance, the team added an "accrued score" column. This score was calculated using a custom function (calculate_score) that evaluated specific parameters deemed important for loan eligibility. The accrued score was used to filter and clean the data, ensuring that only relevant and high-quality data was fed into the models.

Link to GitHub

Applicant ID	Business Structure	Country	Location	NAICS	Business Ownership (1)	Business Ownership (2)	Business Ownership (3)	Business Ownership (4)	Business Ownership (5)	...	Acquisition or Improvement	Business Acquisition or Buyout	Refinancing Existing Debt	Emergency Funds	Franchise Financing	Contract Financing	Licensing or Permits	Line of Credit Establishment	Eligibility	Eligibility Score
1	Sole Proprietorship	US	New Orleans, LA	931370	100	0	0	0	0	...	False	False	False	True	False	False	False	True	[Express]	90.027262
2	LLC	US	Sioux Falls, SD	711130	50	50	0	0	0	...	False	True	True	True	False	False	False	True	[]	25.307292
3	S Corporation (S-Corp)	US	Phoenix, AZ	931430	100	0	0	0	0	...	True	False	False	True	True	False	True	True	[]	98.104346
4	Sole Proprietorship	US	Montgomery, AL	931470	100	0	0	0	0	...	False	False	True	True	True	True	True	True	[Express]	88.342808
5	Sole Proprietorship	US	St. Petersburg, FL	339950	78	22	0	0	0	...	False	True	False	True	True	False	True	True	[]	42.154253

Figure 4: Snippet of Synthetic Data Generated

3.2 Data Cleaning and Preparation

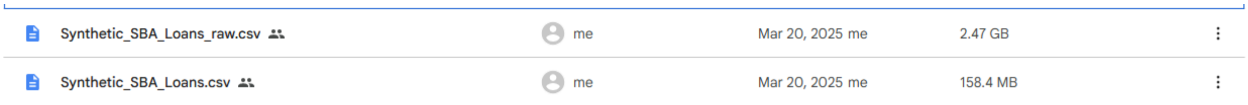

Figure 5: Raw Data Size(2.47gb) and Cleaned Data Size(158.4mb)

- **Declining the Lower 25% of Scores:** Removed bottom scorers. Resulted in overfitting.
  - The team removed the bottom 25% of applicants based on their accrued scores. This approach was intended to eliminate applicants who were clearly ineligible.
  - **Outcome**: This strategy led to overfitting. The model only learned from eligible applicants, resulting in poor generalization. When presented with ineligible applicants, the model incorrectly classified them as eligible, leading to a high rate of false positives.
- **Randomly Declining 25%:** More balanced data improved generalization.
  - Instead of removing the lowest-scoring applicants, the team randomly removed 25
  - **Outcome**: This approach yielded better results. The model was able to generalize better, as it learned from a more balanced dataset. Precision, recall, and F1 scores improved significantly compared to the first approach.

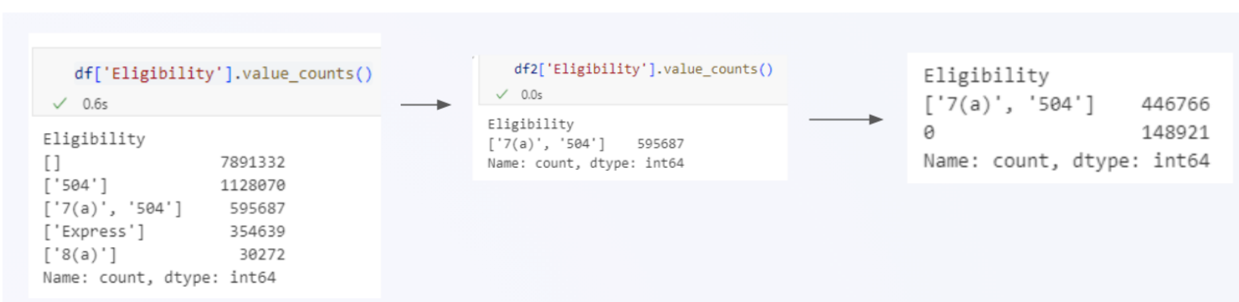

Figure 6: Showing How Data Changed

- **Removing Specific Columns:** Reduced noise and improved focus.

– The team identified and removed columns that were not contributing significantly to the model's performance. This was done by analyzing feature importance scores and removing low-impact features.

– **Outcome**: Removing irrelevant columns improved model performance by reducing noise and focusing on the most important features. This also helped in reducing overfitting.

3.3 Models and Methods

The team experimented with several machine learning models to classify loan eligibility. The models used include:

```
Voting Classifier Details:

Estimator: rf
RandomForestClassifier(random_state=42)

Estimator: gb
GradientBoostingClassifier(random_state=42)

Estimator: xgb
XGBClassifier(base_score=None, booster=None, callbacks=None,
              colsample_bylevel=None, colsample_bynode=None,
              colsample_bytree=None, device=None, early_stopping_rounds=None,
              enable_categorical=False, eval_metric='logloss',
              feature_types=None, gamma=None, grow_policy=None,
              importance_type=None, interaction_constraints=None,
              learning_rate=None, max_bin=None, max_cat_threshold=None,
              max_cat_to_onehot=None, max_delta_step=None, max_depth=None,
              max_leaves=None, min_child_weight=None, missing=nan,
              monotone_constraints=None, multi_strategy=None, n_estimators=None,
              n_jobs=None, num_parallel_tree=None, random_state=42, ...)

Estimator: lgbm
LGBMClassifier(random_state=42)

Estimator: cat
<catboost.core.CatBoostClassifier object at 0x798cd4d72dd0>
```

Figure 7: Voting Classifier Model for Express Loan Eligibility

- LightGBM
  – A high-performance gradient boosting framework that uses tree-based learning algorithms optimized for speed and efficiency, especially with large datasets.

- XGBoost
  - An optimized gradient boosting framework that builds decision trees sequentially to minimize a specified loss function, incorporating regularization to prevent overfitting.
- Logistic Regression
  - A linear classification algorithm used to model the probability of a binary outcome based on one or more input features by applying the logistic sigmoid function. It estimates the relationship between the features and the target variable using maximum likelihood estimation, making it interpretable and efficient for linearly separable data.
- Random Forest
  - Random Forest is an ensemble learning method that builds multiple decision trees during training and outputs the majority vote like classification or average prediction like regression from all trees.
- CatBoost
  - A gradient boosting algorithm developed by Yandex that is particularly effective at handling categorical features without extensive preprocessing. It uses ordered boosting and symmetric trees to reduce overfitting and improve generalization, making it efficient and accurate for both classification and regression tasks.
- Neural Network
  - A machine learning model inspired by the human brain, consisting of layers of interconnected nodes that learn complex patterns from data through weighted transformations and nonlinear activation functions.
- Meta Learner

- Ahigher-level model that learns how to best combine predictions from multiple base models to improve overall performance. It leverages the strengths of diverse models by training on their outputs, typically using a simple algorithm like logistic regression or a decision tree to make final predictions.

- Voting Classifier
  - The Voting Classifier combines multiple models like Random Forest, XGBoost, LightGBM, and CatBoost to improve prediction accuracy. It uses soft voting, meaning it averages predicted probabilities. This ensemble approach leverages the strengths of each model for more robust results.

These models were chosen based on their performance in similar classification tasks. The team prioritized recall as the primary metric because minimizing false negatives (i.e., eligible applicants being classified as ineligible) is critical in this context. However, precision and F1 scores were also considered to ensure a balanced evaluation.

3.4 Tools and Technologies

The project utilized the following tools and technologies:

- Python for data processing and model training.
- Pandas and NumPy for data manipulation.
- SMOTE for data pre-processing technique used to address class imbalance in classification tasks
- Scikit-learn for implementing baseline models like Logistic Regression and Random Forest.
- Pytorch for deep learning and advanced models like Neural Network.
- And Several Other Models listed above for implementing advanced boosting models.
- Matplotlib and Seaborn for data visualization and performance analysis.
- Joblib for importing models for Future Use.

4 Results

4.1 Model Performance

- Neural Network and Meta Learning
  - These models emerged as the top performers.
  - Mostly, Neural Network is the best performing method. But the Meta Learner is the one making sure it is stable.
  - Meta Learning has Neural Nets, XGBoost and LightGBM calculated by Logistic Regression in this test
- XGBoost and CatBoost:
  - These models perviously emerged as the top performers.
  - XGBoost, CatBoost achieved Higher recall and accuracy than most of any other models.
  - Both models demonstrated consistent performance across different thresholds, with XGBoost showing slightly better stability.
- Logistic Regression and Random Forest:
  - These models showed lower performance compared to XGBoost and CatBoost.
  - Precision dropped to zero after a certain threshold, indicating that the models became too strict and classified all applicants as ineligible.
  - This behavior was particularly problematic for Logistic Regression, which struggled to maintain a balance between precision and recall.
- Gradient Boosting Machine (GBM):
  - GBM performed moderately well but was outperformed by XGBoost and CatBoost.
  - The model showed a dip in precision at certain thresholds, indicating instability.

- Voting Classifier:
  - The VotingClassifier showed improved performance compared to individual base models, particularly in terms of recall and balanced accuracy.
  - By combining diverse learners, it reduced overfitting and handled class imbalance more effectively.
  - The ensemble consistently outperformed standalone models like XGBoost or CatBoost, especially after applying SMOTE, making it a reliable choice for predicting SBA Express loan eligibility.

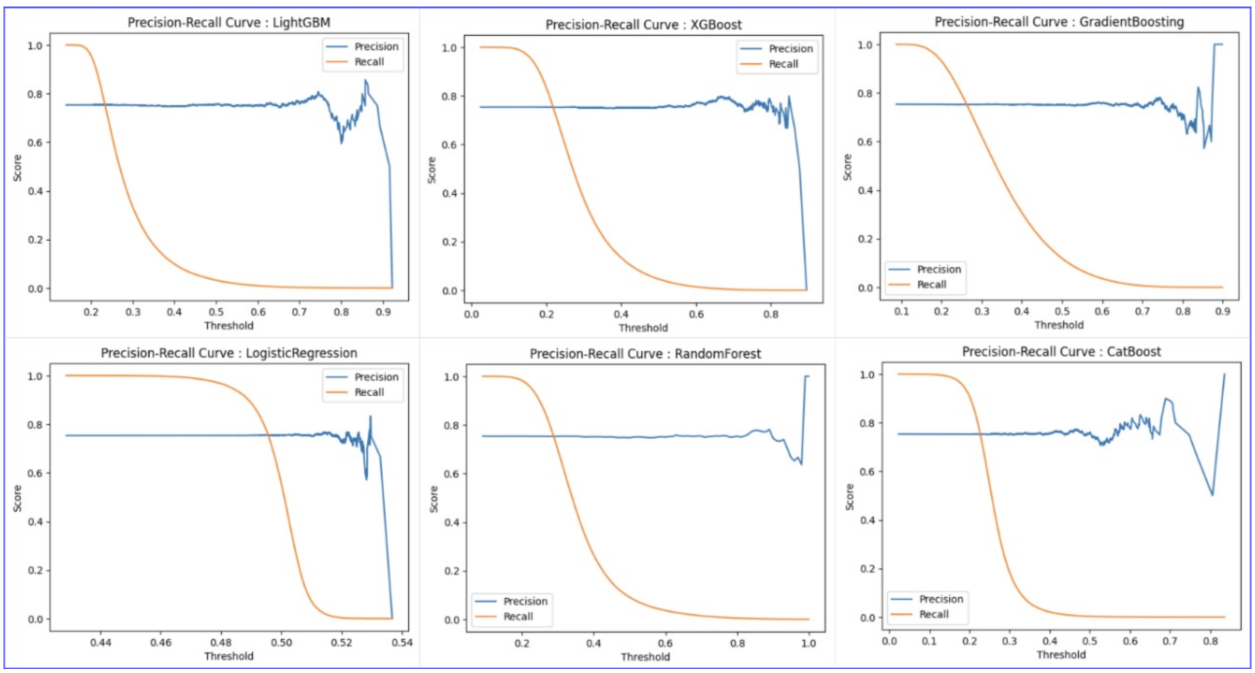

Figure 8: Precision-Recall Curves for the Models Trained

4.2 Handling Class Imbalance

When analysing the data at first, the team encountered a significant class imbalance in the dataset, particularly for the 504 loan type. Out of 100,000 samples, only 17,000 were eligible. To address this, the team tested two strategies:

- **Oversampling:** Recall dropped to 0.16.

  - The team increased the number of eligible samples by duplicating them.
  - **Outcome:** This approach led to poor performance, with recall dropping to 0.16. The model struggled to generalize, as the oversampled data introduced bias.

- **Undersampling:** Recall improved to 0.9.
  - The team reduced the number of ineligible samples to balance the dataset.
  - **Outcome:** This approach significantly improved performance, with recall increasing to 0.9. The model was able to generalize better, as it learned from a more balanced dataset.

- **SMOTE:**
  - SMOTE used in this project to address the significant class imbalance, where SBA loan approvals represented only a small fraction of the dataset. Rather than simply duplicating existing minority class samples—which can lead to overfitting—SMOTE generates new synthetic examples by interpolating between existing minority instances and their nearest neighbors.
  - **Outcome:** For example, if two eligible loan applicants had DSCR values of 1.2 and 1.4, SMOTE might create a new sample with a DSCR of 1.3. This approach enriched the representation of the minority class in the training data. As a result, it improved the model's ability to identify positive cases, increasing recall while maintaining balanced.

```
Out[3]:
Eligibility
[]                   78994
['504']              11142
['7(a)', '504']       6054
['Express']           3509
['8(a)']               301
Name: count, dtype: int64
```

Figure 9: Class Imbalance in the Data Used

4.3 Feature Importance

The team analyzed the importance of different features in determining loan eligibility. The top five features for the Express loan type were identified and used to refine the models. Removing these features and retraining the models led to the following observations:

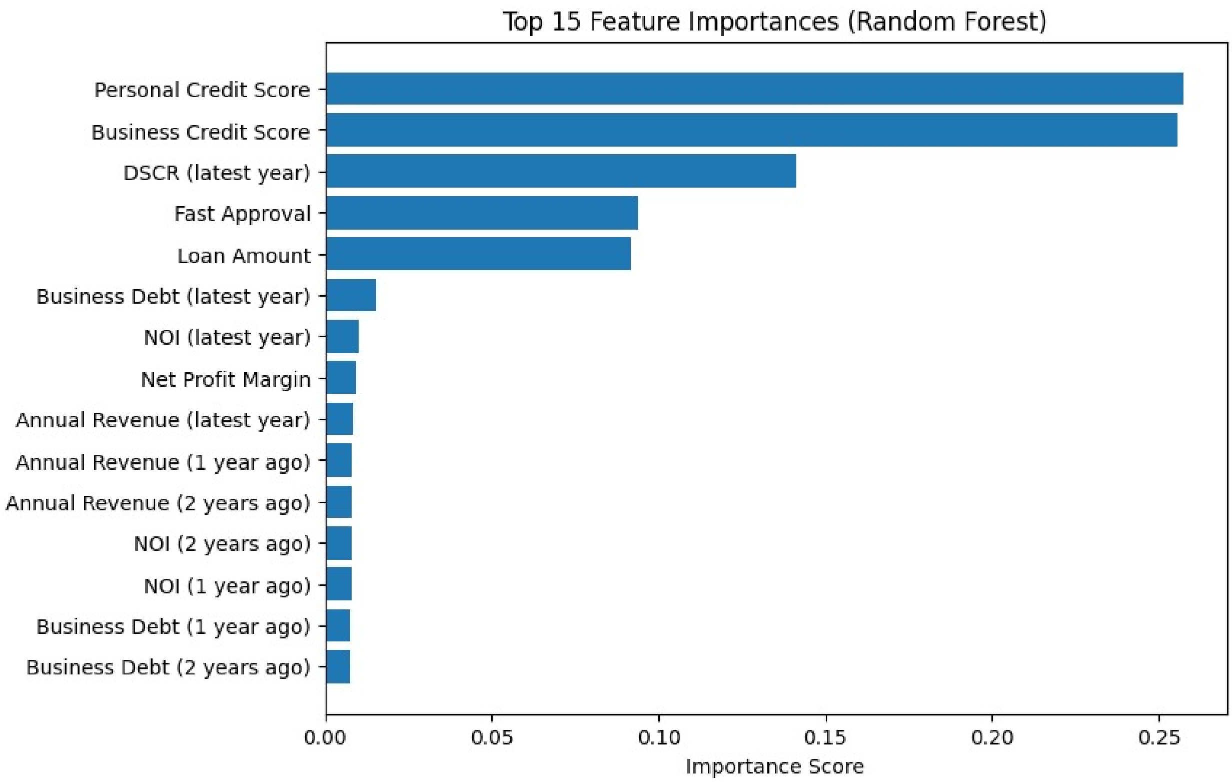

Figure 10: Feature Importance

- **XGBoost:** maintained consistent performance, with no significant drop in recall or F1 score.
- **CatBoost:** showed a slight decline in recall (from 1.0 to 0.93) but still performed well.
- Other models, such as Logistic Regression and Random Forest, showed significant drops in performance, indicating their reliance on specific features.

5 Key Findings

- **XGBoost** is the most stable and reliable model for this task, demonstrating consistent performance across different thresholds and datasets.
- **CatBoost** is a close second, with high recall and F1 scores but slightly less stability compared to XGBoost.
- **Undersampling** is more effective than oversampling for handling class imbalance in this dataset.
- The top five features identified for the Express loan type play a critical role in determining eligibility. Removing these features impacts model performance, highlighting their importance.

LightGBM Results:

	precision	recall	f1-score	support
0	0.24	0.03	0.06	29382
1	0.75	0.97	0.85	89756
accuracy			0.74	119138
macro avg	0.50	0.50	0.45	119138
weighted avg	0.63	0.74	0.65	119138

ROC-AUC: 0.5003

XGBoost Results:

	precision	recall	f1-score	support
0	0.25	0.05	0.08	29382
1	0.75	0.95	0.84	89756
accuracy			0.73	119138
macro avg	0.50	0.50	0.46	119138
weighted avg	0.63	0.73	0.65	119138

ROC-AUC: 0.5016

GradientBoosting Results:

	precision	recall	f1-score	support
0	0.25	0.12	0.16	29382
1	0.75	0.88	0.81	89756
accuracy			0.69	119138
macro avg	0.50	0.50	0.49	119138
weighted avg	0.63	0.69	0.65	119138

ROC-AUC: 0.5020

LogisticRegression Results:

	precision	recall	f1-score	support
0	0.24	0.55	0.34	29382
1	0.75	0.44	0.56	89756
accuracy			0.47	119138
macro avg	0.50	0.50	0.45	119138
weighted avg	0.63	0.47	0.50	119138

ROC-AUC: 0.4963

RandomForest Results:

	precision	recall	f1-score	support
0	0.25	0.09	0.14	29382
1	0.75	0.91	0.82	89756
accuracy			0.71	119138
macro avg	0.50	0.50	0.48	119138
weighted avg	0.63	0.71	0.65	119138

ROC-AUC: 0.5018

CatBoost Results:

	precision	recall	f1-score	support
0	0.24	0.00	0.01	29382
1	0.75	1.00	0.86	89756
accuracy			0.75	119138
macro avg	0.50	0.50	0.43	119138
weighted avg	0.63	0.75	0.65	119138

ROC-AUC: 0.5006

Figure 11: Model Classification Reports

6 Challenges and Limitations

- **Overfitting :** The initial approach of removing the lowest-scoring applicants led to overfitting, as the model only learned from eligible applicants.
- **Severe class imbalance :** The dataset had a significant class imbalance, particularly for the 504 loan type. This required careful handling to ensure the model could generalize well.
- **Hyper Tuning :** Used default parameters for the models, which may not be optimal.

Future work will focus on hyperparameter tuning to further improve performance.

- **No Opportunity of Real Data :** Due to the lack of availability of real data, the models trained and fine-tuned as a part of this project's performances and results cannot be translated to real data. The results and performance is restricted to the quality of the synthetic data at hand.

7 Conclusions

The AI/ML Loan Summarization System has made significant progress in automating the loan eligibility assessment process. By leveraging synthetic data and advanced machine learning models, the team has developed a system that can accurately evaluate applicant profiles and recommend suitable loan types. The project has demonstrated the potential of machine learning to streamline complex financial processes, reduce operational costs, and improve customer satisfaction.

	CatBoost_Prediction	GradientBoosting_Prediction	LightGBM_Prediction	LogisticRegression_Prediction	NeuralNet_Prediction	RandomForest_Prediction	XGBoost_Prediction	Meta_Learner_Prediction
0	1328	438	630	6066.0	6066.0	1915	NaN	630
1	4738	5628	5436	NaN	NaN	4151	6066.0	5436

Figure 12: Result of each model's prediction

The results indicate that LightGBM, XGBoost and CatBoost are the most effective models for this task, with XGBoost showing superior stability across different thresholds. However, there is still room for improvement, particularly in the area of hyperparameter tuning. Future work will focus on optimizing the models further and testing the system on real-world data. In conclusion, the AI/ML Loan Summarization System represents a significant step forward in the automation of financial services. By reducing the reliance on manual processes, the system has the potential to transform the way loans are approved, making the process faster, more accurate, and more efficient. The team is confident that with further refinement, the system will be ready for deployment, delivering tangible benefits to both lenders and borrowers.

A major challenge encountered during the project was the significant class imbalance present in the dataset, particularly for certain loan types like the 504 loan. The team

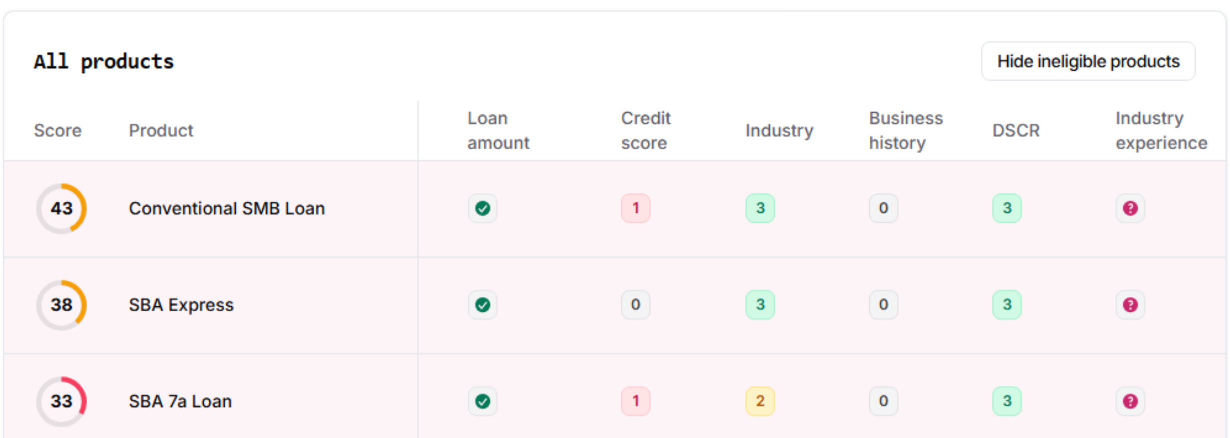

Figure 13: Loan Acceptance Model currently used at Parlay

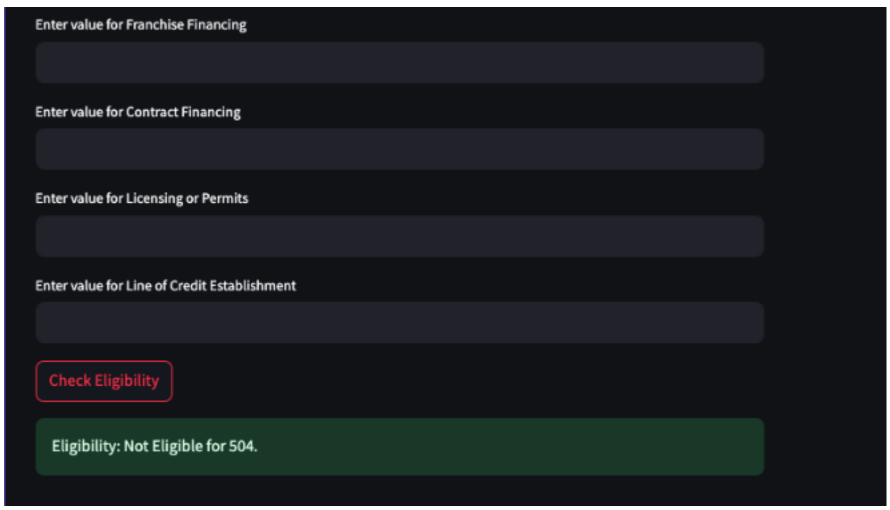

Figure 14: Testing of Model Performance with Applicant Data

experimented with both oversampling and undersampling strategies, ultimately finding that undersampling led to better generalization and higher recall. Additionally, the importance of careful feature selection was highlighted, as removing key features led to noticeable drops in model performance for some algorithms. These insights underscore the need for rigorous data preprocessing and thoughtful model design in financial applications.

Despite these achievements, the project faced limitations, most notably the reliance on synthetic data due to the lack of access to real-world company datasets. This constraint limited the ability to fully validate the models in practical settings. Furthermore, the models were primarily tested with default hyperparameters, suggesting that further improvements could be realized through systematic hyperparameter tuning and validation on real loan application data. Addressing these limitations will be essential for future work aimed at deploying the system in production environments.

In conclusion, the AI/ML Loan Summarization System represents a significant step forward in the automation of financial services. By reducing the reliance on manual processes, the system has the potential to transform the way loans are approved, making the process faster, more accurate, and more efficient. The team is confident that with further refinement, the system will be ready for deployment, delivering tangible benefits to both lenders and borrowers. Future efforts will focus on model optimization, integration with real-world data sources, and continuous monitoring to ensure compliance, fairness, and adaptability in a rapidly evolving financial landscape.