Zan’s Capstone Project Report

Introduction

This report serves as a complimentary breakdown on the project study and implementation of building a loan default prediction model. The model is based on a subset of data from Lending Club, a peer to peer lending platform that serves over 4.8 million customers. The dataset features 100,000 instances of loan applications with roughly 143 columns of data. This means the subset is feature rich but also quite gargantuan to process without a proper framework and process to obtain focused insights.

This report has been designed to provide a high level overview to stakeholders of all levels about the approaches taken and the outcomes achieved through this project. For a deeper and more technical understanding of the workflow, data science practices and evaluation metrics used I strongly recommend to view the accompanying Jupyter Notebook file which has all outputs, insights , graphs and models embedded within the notebook itself.

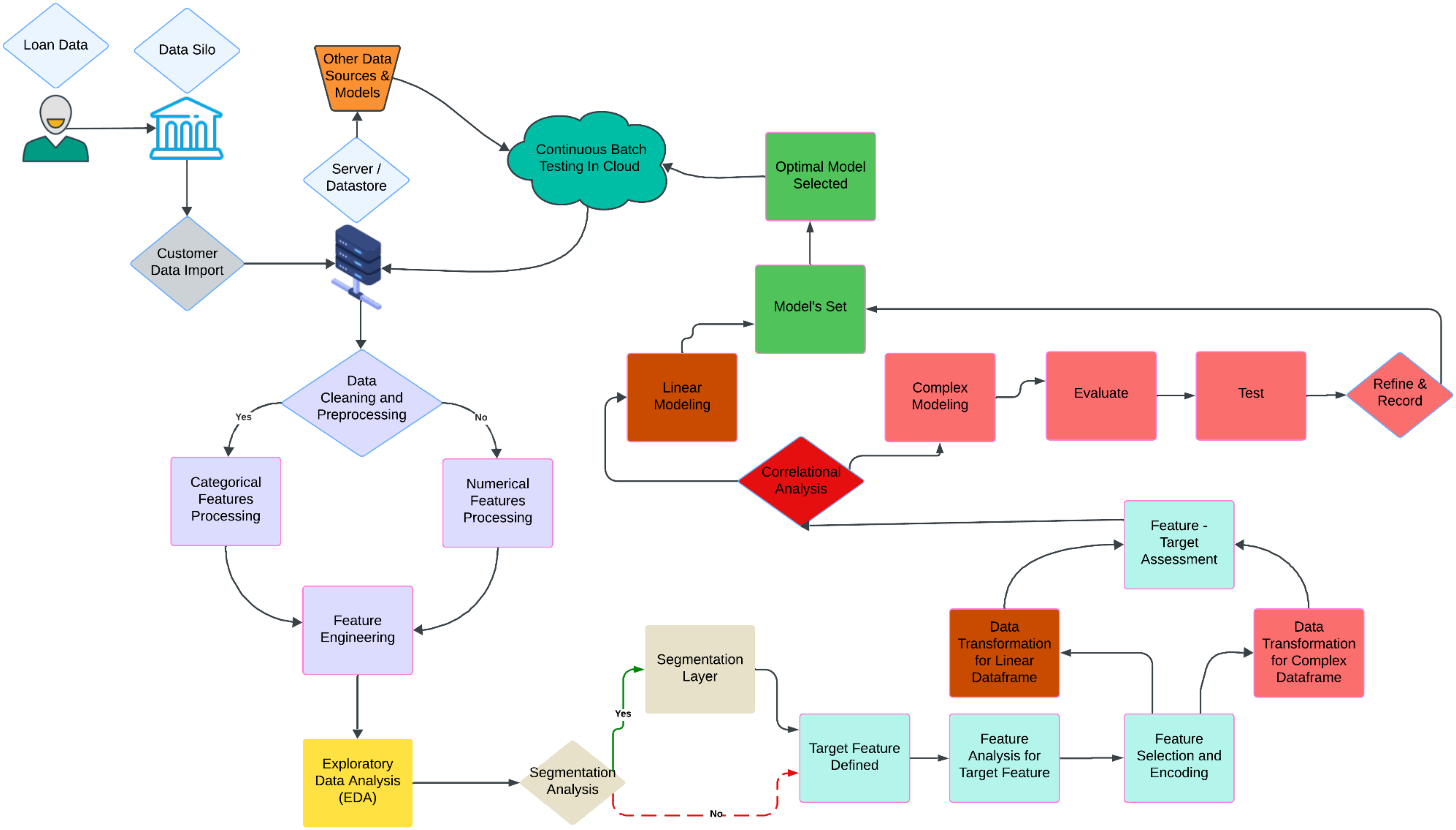
This report will walk you through how the data has been processed, the manner in which key insights were used to drive decision making and the findings and outcomes of this project. By reading this report you will hopefully have a better understanding of how the business can benefit from this project as well as any future projects based on the research and analysis done.

Deliverables

- Jupyter Notebook File with preprocessing steps, exploratory data analysis (EDA) insights, modelling frameworks and visualisations to understand the data better.
- GIT Repository that includes all project code with a basic README guide.
- Chosen modelling approach to be used at the discretion of senior stakeholders and applied to the rest of the business
- Basic architecture diagram detailing current processes and how they can be incorporated further into the business function of Lending_Club

Contents

Overview Of Project Framework & Business Implementation


Loan Data
Data Silo
Other Data Sources & Models
Server / Datastore
Continuous Batch Testing In Cloud
Optimal Model Selected
Customer Data Import
Model's Set
Data Cleaning and Preprocessing
Yes
No
Linear Modeling
Complex Modeling
Evaluate
Test
Refine & Record
Correlational Analysis
Categorical Features Processing
Numerical Features Processing
Feature - Target Assessment
Feature Engineering
Data Transformation for Linear Dataframe
Data Transformation for Complex Dataframe
Segmentation Layer
Yes
Target Feature Defined
Feature Analysis for Target Feature
Feature Selection and Encoding
Exploratory Data Analysis (EDA)
Segmentation Analysis
No

Methodology

The brief and simplified methodology used in the code was as follows:

1. Data Import and Setup

- Import the dataset and perform initial setup.

1. Data Cleaning and Preprocessing

- Clean the dataframe.
- Categorical Feature Cleaning and Preprocessing.
- Float Feature Cleaning and Removal.
- Create and set the target feature.

1. Feature Engineering

- Create new features based on the existing data.

2. Exploratory Data Analysis (EDA)

- Analyse the target feature distribution.
- Perform segmentation analysis:
    - **If Accepted**: Transform the target feature and loop back into EDA.
    - **If Rejected**: Continue with key feature relationship analysis.
- Examine key feature relationships.

2. Feature Selection and Encoding

- Select specific features based on analysis.
- Generate encoding recommendations and apply them.
- Split the target feature list into two dataframes: **Linear** and **Complex**.

2. Separate Data Transformation for Linear and Complex Dataframes

- Transform features separately for each dataframe.
- If transformations look unsuitable or show weak relationships to the target feature, drop those features.
- Verify transformations with inference testing, ensuring they align with the target feature.

2. Correlational Assessment and Feature Refinement

- Assess correlation and collinearity for features with significant differences.
- Refine and optimise the final feature lists based on correlation and relevance.

3. Modelling

- **Linear Model**: Train, test, and evaluate to identify the best-performing linear model.
    - Select the best linear model for the final comparison.
- **Complex Model**: Iterate through model refinements until the best model is found.
    - Choose the best complex model for the final comparison.

3. Final Model Selection

- Compare the best linear and complex models.
- Select and verify the overall best-performing model.

Section 1 Data Import, Cleaning & Preprocessing Summary

The overall approach to the dataset was to holistically and efficiently evaluate features as a whole, based on their nature as numeric features or categorical features. The LCD Data Dictionary proved useful here and so it was appended to the feature names in a separate list to make analysis within the notebook easier.

Feature Pre-Processing & Encoding
For categorical features I performed a broad analysis of the features in terms of how complete they were within the dataset to ascertain which features would be most useful in predictive relevance for loan defaults. Recommendations could be quickly made based on some basic statistics and insights on these categorical features and if I expected they would be weak for the modelling frameworks.

It's important to note, that for the purposes of this project a funnel down approach was used. This allowed me to provide insights and some pre-processing recommendations for features, which outside of this project could still be leveraged in future projects. Typically most data projects operate in this manner in order to start off with the scope of the data at hand as a whole and gradually narrow down the focus to make insights and outcomes more specific and more manageable with a high impact.

Once key categorical features were retained, encoding recommendations and implementations were performed to make these features more friendly to work with. As an example, loan grades and sub grades can provide really useful information about a loan application, their values such as grade A or A1 cannot be natively used in models. Encoding features into categories where for example Grade A = 0, Grade B = 1 and so on and so forth retains their categorical 'ranked' nature while also being suitable for modelling.

Some categorical features were also date/time based that I processed accordingly. Although this project didn't leverage them as much as it could have, the pre-processing steps are there for any future exploration of modelling approaches with a temporal nature.

Removal Of Features That Could Contribute to Data Leakage
Numeric features were also broadly assessed in the same manner. The first step was to identify any features that demonstrated a high number of missing values that may be linked to their nature of loan status. Loan status refers to the Lending Club's classification of loan applications, with the primary labels being Fully Paid, Charged Off, or Current. My approach here was to determine whether certain numeric features had more missing values or 0's in Current applications vs applications that already had an outcome. This was done to ensure our models had minimal data leakage. Data leakage can lead to poorly trained models that overfit data and don't perform well on unseen data. An example of a feature that might skew results and give false confidence in our model is the recoveries feature, which counts how many recoveries a loan application has had. Since this occurs after a default event, it would not be suitable to provide predictive relevance and would need to be dropped to avoid false confidence in model performance. Other features similar to recoveries were assessed and recommended implementations were performed to limit any data leakage in our modelling approach.

Final Steps
Other approaches in this step included handling missing values and encoding considerations to ensure the data was more complet , understandable and didn't lead to errors. In an ideal world we wouldn't have messy data or missing values. For the purposes of insight evaluation and modelling they had to be treated with care and attention based on personal understanding of the features.

New features were created using existing features to tie data together and potentially serve as better predictors of high risk loan applicants. Features such as leverage ratios , high interest loan ratios and others might be able to better pronounce the risk factor of any given applicant. The details and calculations behind these features are in the notebook.
I created the target feature (loan defaults)= 1 as a binary target variable to label loans that had defaulted based on loan status. The majority of defaulted loans were marked Charged Off but some had been labelled default. This gave me an idea which we will discuss in Section 2 but I think it's something that needs to be addressed.

Section 2 Exploratory Data Analysis

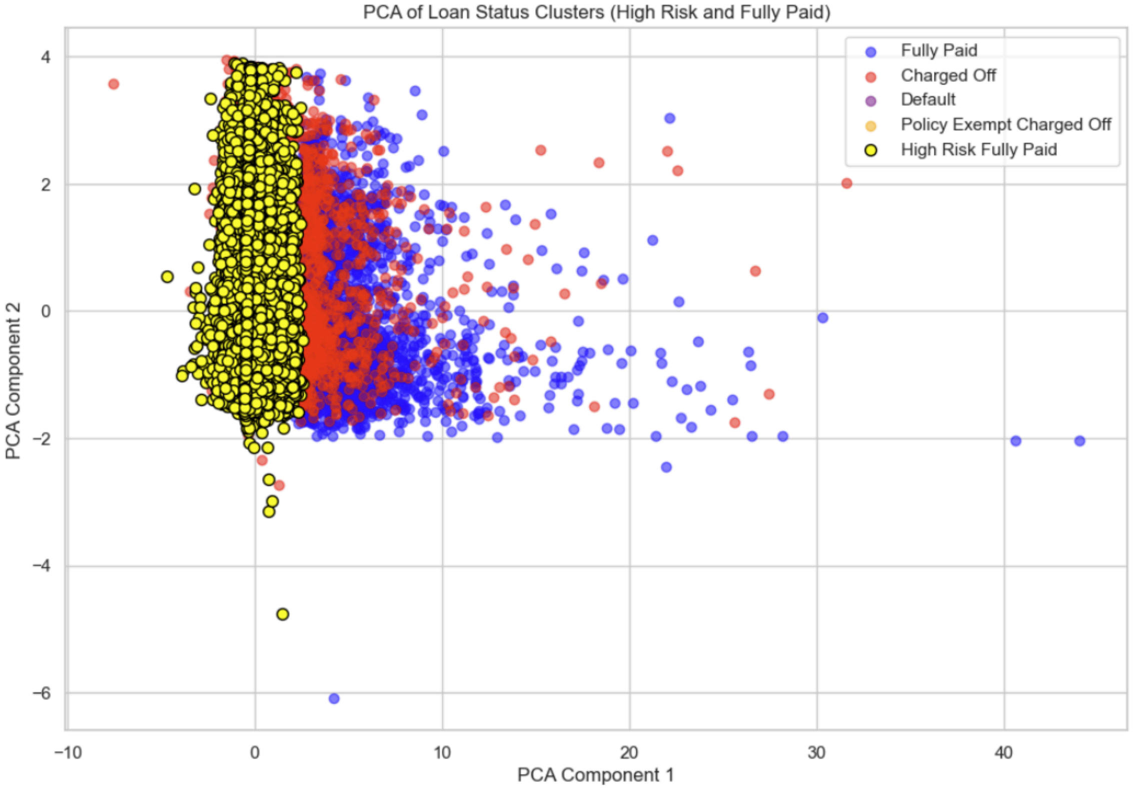
There was a huge benefit in having a holistic approach, even though the final modelling approach didn't leverage everything touched upon in Sections 1 and 2.

Key Finding: As mentioned previously, there were loan status values for 'Default' but only a small number (less than 50) as opposed to the 10,000+ Charged Off loans and the 60,000+ Fully Paid values. I realised that at some point, some of the Fully Paid loans at some point may have defaulted at least once, which opens up a number of questions and opportunities for the business.

Business Findings & Recommendations:

1. The data might benefit from a feature that is inherently added prior to any import or preprocessing, where Fully Paid loans are ranked in terms of how smooth they were i.e did they ever experience default once , twice etc before eventually turning around to reach the Fully Paid status. The modelling approaches seen here could benefit from having this feature as we can better assess riskier loans that could default once, but also assess loans that are risky and could eventually become Fully Paid.
2. Fully Paid loans that have defaulted at least once may inherently demonstrate a higher risk profile, understanding this risk profile would be useful not only for modelling but also to address class imbalance. As mentioned before, the sample dataset has too many Fully Paid loans as opposed to defaulted loans making some modelling approaches sensitive to class imbalance. In this instance they may identify more non defaults than defaults as there are so many of them compared to actual defaults to put it simply.

In order to understand this further, I decided to set up a segmentation layer that would use specific features indicating if Fully Paid data may have defaulted and share any resemblance with defaulted loans. Some of these features included delinquencies, months since collections to name a few. This was done through various means such as K Means Clustering and Principal Component Analysis. As you can see from the results below, there are indeed 'high risk' Fully Paid loans that based on certain features outside of our modelling approach reflect if Fully Paid loans could have defaulted. The reason these were not included in the model is many of them could lead to data leakage. The purpose of this approach was to determine if we could transition some Fully Paid loans into the defaulted category (as our model aim is to identify loans likely to default despite the end outcome being positive).



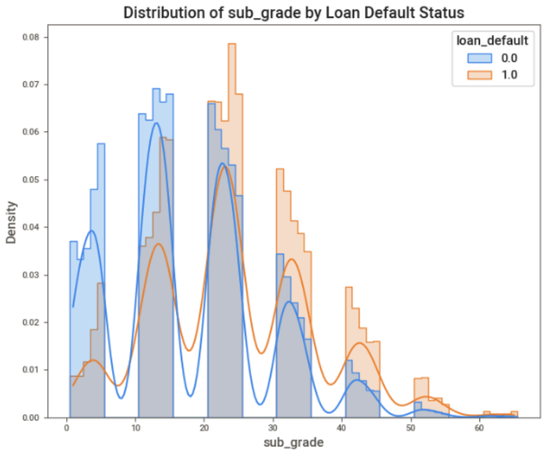
The graph above shows that indeed there are 'high risk' fully paid loans that may have defaulted. There is however extreme overlap with defaulted loans and a high concentration of Fully Paid Loans to the left of the plot, which diminish as you move to the right similar to Charged Off loans. This suggests there isn't enough probable cause or justification to change Fully Paid values to Charged Off ourselves.

I strongly recommend that the business functions and team's responsible for the management of recorded data add a feature that can identify or label Fully Paid loans that have defaulted before with clarity. I could have used an array of other features that suggest default activity but without certainty and clarity this did not seem suitable. The primary objective was to take actual recorded default and non default data and examine the potential use cases to predict defaults. Statistical testing did suggest that this method showed meaningful clusters and segmentation between high risk fully paid loans vs other fully paid loans. However when setting thresholds to then select how many fully paid loans to change to defaults, the change in imbalance was either too low or too extreme, suggesting sensitivity in the approach and so it was scrapped.

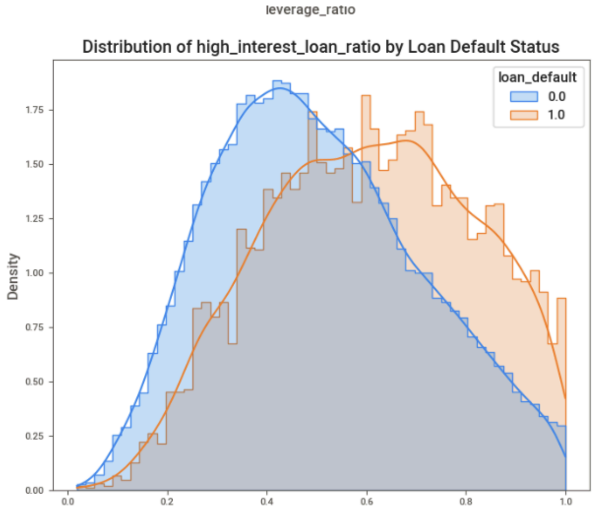
The rest of EDA got back to assessing features for predictive relevance through efficient methods such as automated visualisations on overall feature importance to the target variable via SweetViz (an automated tool to generate neat HTML reports). Distribution plots of the data and assessing feature quality was subsequently done after. A limited subset of features was selected, with justifications and rationale covered in more detail within the notebook. These features were then double checked to ensure encoding had been done properly before visualising their distribution and outliers as a whole, as well 2 separate group comparisons of each feature based on defaulted and non defaulted data. This approach excluded Current data as I wanted to explore the features of the target variable in a reliable manner to make meaningful observations about the relationships. Current loan status data would have added noise. Current data was only included when assessing the feature distributions as a whole.

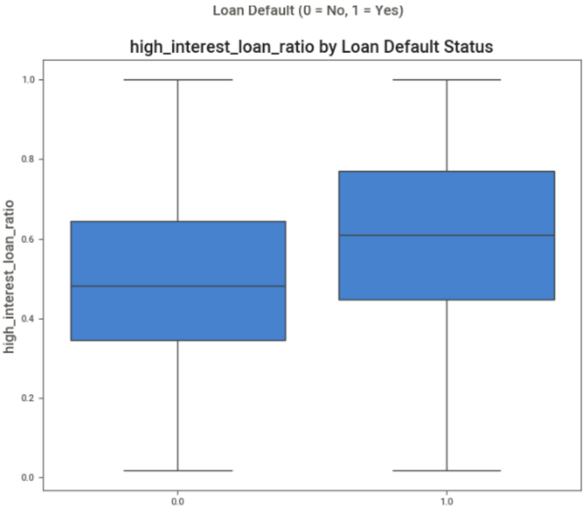
Following on from this step, 2 dataframes were created to have access to 2 variations of the features I had selected and the underlying data. A data frame is essentially a specific copy of the underlying CSV file, stored in memory by Python when running Jupyter Notebook.

Section 2.5 started with creating the data frame '**df_linear'**. It was in this section I thoroughly explored the relationships between each feature and the differences shown within them based on defaulted and non defaulted data. The top graph on the right shows how as sub_grade decreases, there is an increase in the density of defaulted loans, whereas the higher grades have more non defaulted loans (sub grades are encoded ordinally so the lower the value the better the grade).



The image below shows that the higher the high interest loan feature is , the more likely you will see defaulted loans as they have a larger density here vs non defaulted loans. The box plots for high interest loan ratio also show that on average , the general average for defaulted loans exhibit higher high interest loan rations.





Many other features displayed this sort of behaviour based on the target features to varying degrees, some more so and some barely negligible. The EDA for section 2.5 was based on df_linear, this dataframe took the selected sub features and applied transformation, imputation and outlier handling more suited to linear and logistic regression models. The other dataframe, df_complex, looked similar in terms of plots but it leveraged a more robust transformation method called Yeo-Johnson. Whilst this method performed better, for numeric features it transforms them into a range of positive and negative values and negative values can disproportionately impact linear and logistic regression models.

Recommendation: In the future it would be interesting to examine other modelling techniques and examine the differences when supplying them with the 2 different data frames to assess impacts on model performance. Through this approach the business might find the best way to optimise other modelling techniques.

With the features transformed in their respective datasets, correlation analysis was used to assess feature correlations with one another as well as loan defaults. It's important to not have too many features that are overly correlated with one another or even one that is extremely correlated with another feature as this can skew model performance and understanding. Both numeric features and categorical features were assessed in terms of their relationship to loan defaults and each other. Recommendations and further insights can be found in the notebook.

Based on interpreting the correlational analysis a further subset of features was selected to ensure a good array of predictors was used without overly relying on too many features highly correlated with one another.

Model Selection

Overview of Logistic Regression and CatBoost for Loan Default Prediction

In the process of building an effective loan default prediction model, I explored two different types of models: **Logistic Regression** and **CatBoost**, each justified by unique strengths that made them suitable for capturing patterns in the data. Here, I'll outline why these models were chosen, the benefits they bring to predicting loan defaults, and the considerations that influenced their selection.

Logistic Regression

Logistic Regression is one of the most widely used algorithms for binary classification problems, especially in financial sectors for risk assessment and binary decision-making (e.g., predicting loan default vs. non-default). Here are the main reasons it was considered appropriate for this project:

1. **Interpretability**:
    - Logistic Regression provides easily interpretable coefficients, making it transparent and straightforward to explain to business stakeholders. This is crucial for finance-based decisions, where understanding the influence of individual features (e.g., debt-to-income ratio or employment length) on the likelihood of default is important.
    - The interpretability makes it a particularly useful benchmark model, providing insights into feature importance and helping to set expectations for a more complex model.
2. **Baseline Comparison**:
    - Due to its simplicity, Logistic Regression serves as an ideal baseline for evaluating the performance of more advanced models like CatBoost. By comparing against a straightforward model, we were able to quantify the added value of complex models.
3. **Ability to Handle Imbalanced Data with Regularization**:
    - Given that loan default is often a minority class, we could apply regularisation techniques (e.g., L2 regularisation) and class weighting to Logistic Regression, which helps in reducing the influence of rare outliers and controlling for class imbalance. In cases where imbalanced classes are present, regularisation also stabilises the model, helping to avoid overfitting while capturing key signals from minority class data.**SMOTE for the best version of my logistic regression model with regularisation was used.**

4. **Suitability for Linear Relationships**:
   - Logistic Regression is effective for datasets where linear relationships exist between features and the target variable. In our exploratory data analysis (EDA), some features (such as debt-to-income ratio) demonstrated roughly linear correlations with loan default, making Logistic Regression an ideal candidate to capture these types of relationships.

Despite its advantages, Logistic Regression may struggle to capture complex, non-linear relationships in the data, and this is where CatBoost provided a significant advantage.

CatBoost

CatBoost (Categorical Boosting) is a powerful gradient boosting algorithm designed specifically to handle categorical features efficiently, which made it particularly attractive for our loan default prediction task. Here's why it was selected and why it proved to be valuable:

1. **Handling of Categorical Variables**:
   - The dataset contained numerous categorical variables (e.g., grade, sub_grade, home_ownership) that have high predictive power in loan default prediction. Unlike traditional gradient boosting models, CatBoost is designed to process categorical variables natively without needing extensive preprocessing like one-hot encoding, allowing it to preserve valuable information that could be lost in transformation.
2. **Performance in Complex, Non-linear Data**:
   - Loan default prediction involves complex, non-linear relationships between financial metrics and the likelihood of default. CatBoost's gradient boosting structure is particularly adept at capturing these patterns, providing a high degree of accuracy by aggregating numerous weak learners (decision trees) to form a strong predictive model.
   - By iteratively adjusting weights and re-evaluating based on previous errors, CatBoost was able to identify nuanced patterns and interactions between features that Logistic Regression could not capture effectively.
3. **Reducing Overfitting through Parameter Control**:
   - With hyperparameters like depth, iterations, and learning_rate, CatBoost provides significant flexibility to balance model complexity and generalizability. For instance, using early stopping rounds and adjusting class weights allowed us to control overfitting, which can be a risk when working with boosted models on imbalanced data.
   - Additionally, through parameter tuning, we managed to achieve high recall, which is essential for correctly identifying potential defaulters in loan portfolios.
4. **Handling Imbalanced Data with Class Weighting**:
   - Similar to Logistic Regression, CatBoost supports class weighting, allowing us to assign more importance to the minority class (defaults) and reducing the likelihood of default cases being misclassified. This is particularly valuable in high-stakes contexts like loan default prediction, where the cost of false negatives (missed defaults) is much higher than the cost of false positives.
5. **Improved Interpretability through SHAP Values**:
   - Although boosting models like CatBoost are often seen as black boxes, tools like SHAP (SHapley Additive exPlanations) help interpret feature importance in CatBoost, providing a clear view of which variables most influence predictions. This interpretability, combined with high predictive power, made CatBoost an ideal choice for this application.

Due to its superior performance in identifying a higher number of true defaults, Catboost was the chosen model I recommend for the business. I will delve into some of the advantages and limitations of this model approach below

Advantages and Disadvantages of CatBoost in the Context of Challenger Model 3

In deploying **Challenger Model 3** with CatBoost for loan default prediction, we encountered both notable advantages and several challenges. Here, I'll summarise the advantages CatBoost brought to the model as well as the key limitations observed, particularly around its computational requirements, iterative tuning needs, and reliance on specific features.

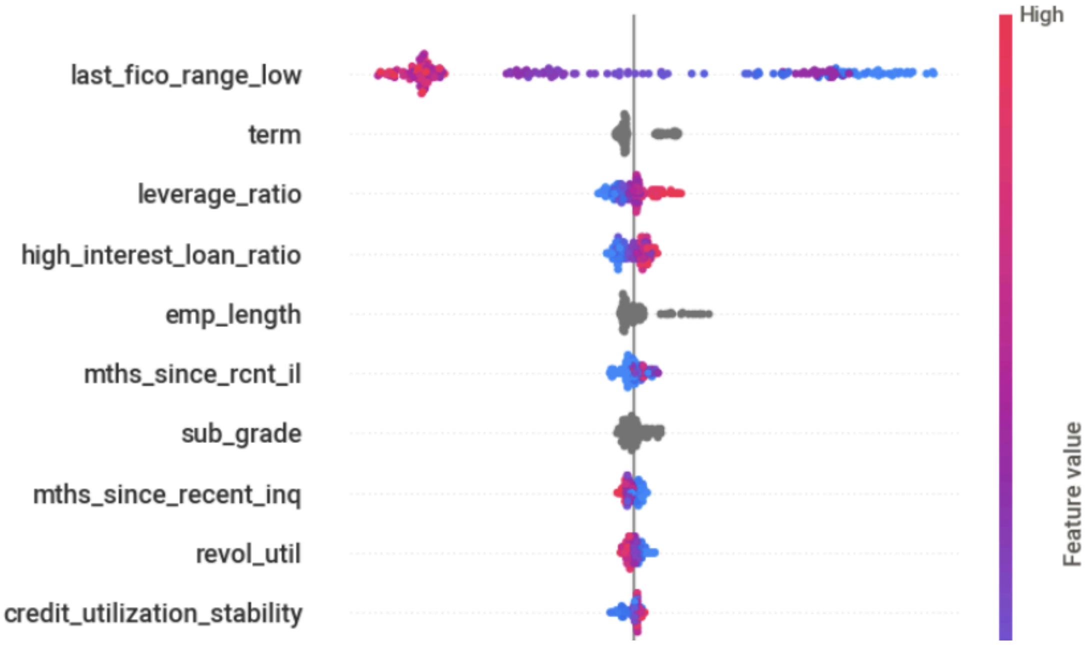
Advantages of CatBoost

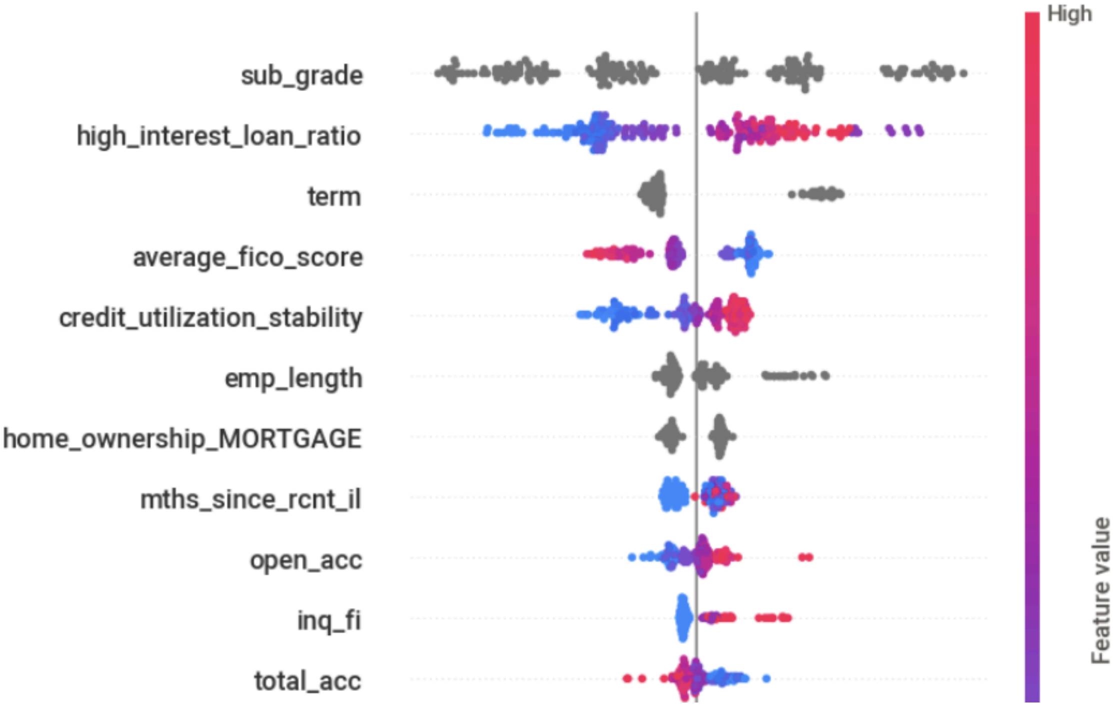
1. **Superior Performance on Imbalanced Data**:
   - CatBoost's support for **class weighting** allowed us to address the imbalance between defaulted and non-defaulted loans. By adjusting class weights, we could give more focus to default cases, enhancing recall and allowing for better identification of high-risk loans. This helped minimise the number of false negatives, which is particularly important for mitigating loan losses.
2. **Inherent Handling of Categorical Variables**:
   - CatBoost's ability to process categorical features without requiring extensive preprocessing was initially a major draw for this model. The categorical features (e.g., grade, sub_grade, home_ownership) could be processed natively, preserving their predictive power without information loss from transformations like one-hot encoding.
   - Despite this advantage, the final model did not rely on categorical features as much as expected, with numerical features such as last_fico_range_low and average_fico_score dominating the predictive influence.
3. **High Accuracy with Complex Relationships**:
   - As a gradient boosting model, CatBoost performed well with the non-linear relationships present in loan default data, especially as it stacked weaker models (decision trees) into a stronger overall predictor. This allowed for high precision in predictions and excellent recall scores, as it effectively identified complex patterns among financial metrics and behavioural data.
4. **Improved Interpretability with SHAP Values**:
   - Although CatBoost can be complex, using **SHAP values** allowed us to interpret the model by understanding which features were most influential in predicting defaults. This provided transparency, crucial in finance, where stakeholders need to understand the factors influencing predictions.

Disadvantages of CatBoost

1. **High Processing Time and Computational Cost**:
   - The **training time** for CatBoost was substantially longer compared to simpler models like Logistic Regression. This is because CatBoost processes numerous decision trees, making it computationally intensive. In our iterative refinement of Challenger Model 3, each parameter adjustment led to prolonged training times, which can be costly, especially in environments with large datasets or limited computational resources.
   - This extended processing time could pose challenges for model retraining and scalability if updates are needed frequently, especially in a production environment where quick turnaround is necessary.
   - Even though we compared 3 challenger models in the notebook based on CatBoost, while leveraging a fairly competent consumer grade GPU, the modelling functions run in between each of the main models added significant time to process the entire notebook, taking anywhere from **7-15 minutes.**

2. **Iterative Tuning Required**:
    - To achieve optimal performance, Challenger Model 3 required several rounds of **iterative tuning** of hyperparameters such as depth, learning rate, and class weights. While the grid search allowed us to find suitable parameters, each iteration was time-consuming, underscoring the intensive nature of gradient boosting models.
    - This iterative refinement process might not be feasible in scenarios where time is limited, or when continuous model updates are required in response to dynamic data changes.
3. **Over-reliance on last_fico_range_low**:
    - The model displayed a strong **dependency on last_fico_range_low**, with this feature consistently dominating both SHAP values (up to 3/-3) and feature importance plots. This indicates that while CatBoost is capable of handling many features, the model's performance was heavily reliant on a single numerical variable, overshadowing the contribution of other potentially valuable categorical variables.
    - This dependency can limit model robustness, as any shifts or inaccuracies in the data related to last_fico_range_low could significantly impact the predictions. Additionally, the heavy reliance on this variable suggests that other categorical features, which may offer diverse predictive insights, were underutilised.When removing last_fico_range_low other features were easier to understand but still only reached up to 0.6/-0.6 for SHAP values.





4. **Limited Impact of Categorical Variables**:
    - Despite CatBoost's capacity to handle categorical variables effectively, the categorical features in Challenger Model 3 had **lower than expected importance**. Features such as sub_grade and home_ownership, while intuitive, contributed less to the overall prediction, with most predictive power concentrated in numerical features.
    - This outcome suggests that either the categorical features were not highly correlated with default likelihood or that the model's dependence on last_fico_range_low and other financial metrics limited the impact of categorical data.

Summary

While CatBoost provided strong performance metrics, particularly in terms of recall and model interpretability, it required considerable processing time and fine-tuning, making it resource-intensive. The model's reliance on `last_fico_range_low` and limited influence of categorical features highlighted both the strengths and constraints of this approach in the context of our data.

For future model refinement, focusing on diversifying the feature set, especially by enhancing the value of categorical features, could improve model resilience. Additionally, testing alternative models, such as XGBoost or LightGBM, might offer more efficient training times with similar interpretability, providing a balanced approach between performance and operational feasibility.

Outcomes , Impacts & Further Considerations

Model Performance Analysis for True and False Default Rates

Confusion Matrix Comparison (Percentage Rates):

- **Baseline Model 2** (Confusion Matrix:
    - True Non-Default Rate (TN): **91.76%**
    - False Positive Rate (FP): **8.24%**
    - False Negative Rate (FN): **23.19%**
    - True Default Rate (TP): **76.81%**
- **Challenger Model 3** (Confusion Matrix:
    - True Non-Default Rate (TN): **86.56%**
    - False Positive Rate (FP): **13.44%**
    - False Negative Rate (FN): **14.26%**
    - True Default Rate (TP): **85.74%**
- **Justification for Challenger Model 3:** Challenger Model 3 achieves a superior True Default Rate of 85.74%, outperforming Baseline Model 2's 76.81% rate. This higher rate of correctly identifying defaults significantly improves the model's ability to predict high-risk loans, reducing potential loan losses more effectively.

Deployment and Scalability Considerations

Given Lending Club's large customer base of over 4.8 million, any model deployment strategy must ensure real-time scoring capabilities, optimize computational efficiency, and potentially include ongoing manual monitoring:

Challenger Model 3 (CatBoost)

- **Computational Requirements**: Challenger Model 3, leveraging CatBoost, provides strong predictive accuracy by effectively managing both categorical and numerical data. However, its complexity increases computational demands for training and scoring, challenging real-time application scalability when processing millions of accounts.
  Potential Solutions:
  - **Batch Scoring**: Implement batch processing for Challenger Model 3, scoring customer profiles during scheduled intervals (e.g., nightly or weekly). This method suits high-risk segment evaluations and periodic assessments without overwhelming resources in real-time.
  - **Hybrid Scoring Strategy**: Use Challenger Model 3 for high-risk and batch processes, while applying Baseline Model 2 or a simpler logistic regression model for real-time, low-risk scoring. This balances CatBoost's predictive advantages with logistic regression's lightweight nature for faster scoring.
- **Manual Intervention**: Due to its iterative refinement and tuning needs, Challenger Model 3 may require periodic monitoring and updates, especially as borrower behaviours evolve. However, the accuracy in identifying true defaults and reducing loan losses can potentially yield significant savings, justifying the investment in operational oversight.

Baseline Model 2 (Logistic Regression)

- **Real-Time Compatibility**: Baseline Model 2, based on logistic regression, is ideal for real-time scoring due to its simpler, computationally efficient structure. Logistic regression's lower processing demand allows it to be quickly applied to Lending Club's cast member base (4.8m +), ensuring smooth scalability.
- **Advantages in Real-Time Scoring**: Despite lower precision compared to Challenger Model 3, Baseline Model 2 is effective for lower-risk cases. For high-risk segments, flagged instances could trigger a second-level assessment using Challenger Model 3 or additional risk models, maintaining a balance between accuracy and efficiency.

Summary of Deployment Strategy

For effective business-as-usual (BAU) application at Lending Club, a **hybrid deployment approach** would optimise both accuracy and operational scalability:

- **High-Risk Assessment**: Deploy Challenger Model 3 in batch mode for in-depth evaluations of high-risk applicants and periodic re-assessments, enhancing the ability to flag and mitigate default risks.
- **Real-Time Scoring**: Implement Baseline Model 2 for fast, scalable scoring across lower-risk segments, avoiding strain on computational resources and achieving reliable results for the majority of loan applications.
- **Continuous Monitoring**: Establish a monitoring framework to track model performance, with periodic re-tuning as needed to adapt to changes in economic conditions or borrower behaviour patterns.

ROI Impact

By leveraging a hybrid approach with Challenger Model 3 for high-risk identification and Baseline Model 2 for real-time scoring, Lending Club can achieve:

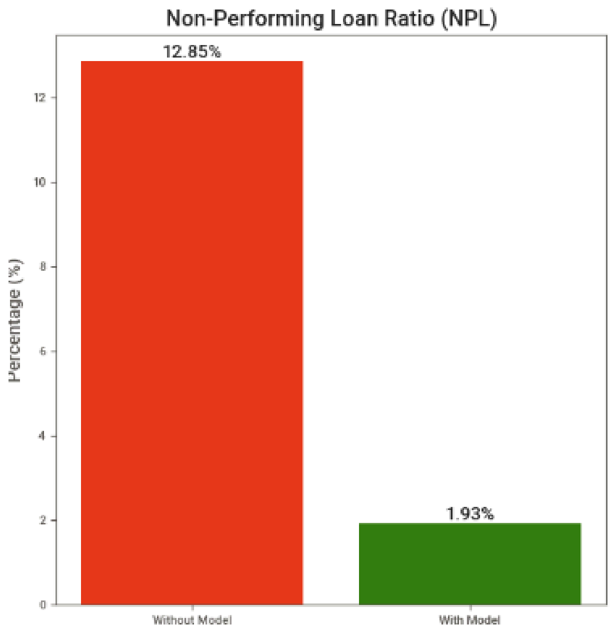
1. **Reduced Default Losses**: Challenger Model 3's ability to accurately flag high-risk loans reduces the number of non-performing loans, potentially saving millions in loan losses.
2. **Increased Net Interest Income**: A more accurate model preserves net interest income by minimising defaults and maintaining a healthier portfolio of performing loans.
3. **Operational Efficiency**: Hybrid scoring minimises computational strain, offering a cost-effective solution that scales with customer volume while preserving scoring accuracy across loan segments.

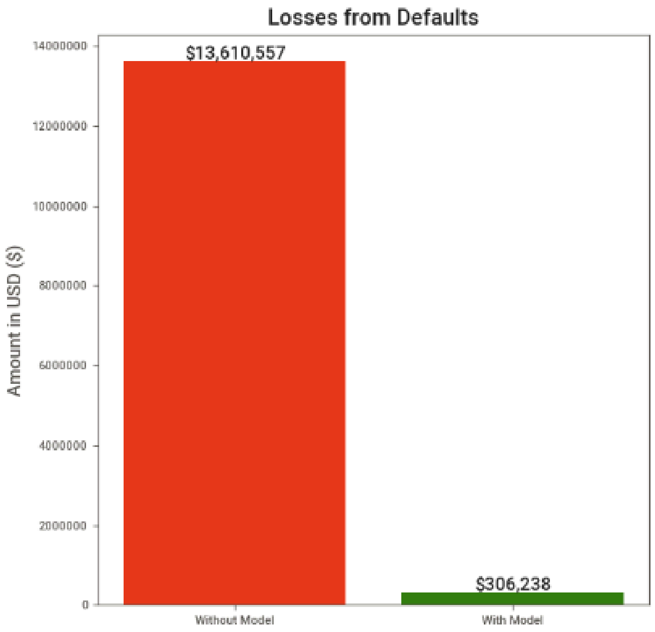
Summary of Challenger Model 3 Performance and Business Impact

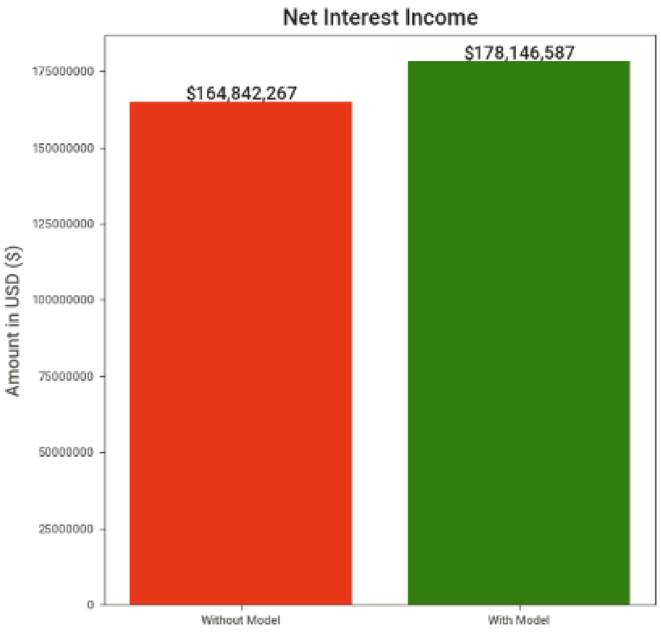
The visualisations showcase the substantial impact of Challenger Model 3 when applied in an extreme case scenario where all loans predicted as high-risk of default are rejected:

1. **Non-Performing Loan Ratio (NPL)**:
   - Without the model, the NPL is high at **12.85%**, indicating a significant portion of loans are likely to default.
   - With Challenger Model 3, the NPL drops dramatically to **1.93%**, highlighting the model's effectiveness in identifying and mitigating high-risk loans.
2. **Losses from Defaults**:
   - Losses from defaults without the model stand at **$13,610,557**, representing potential loan write-offs due to defaults.
   - By using the model to reject high-risk loans, these losses are minimised to **$306,238**, a reduction that could save the business millions of dollars in potential losses.
3. **Net Interest Income (NII)**:
   - Without the model, the Net Interest Income is **$164,842,267**. This income reflects the potential earnings assuming 100% repayment of all loans.
   - With the model, the NII rises to **$178,146,587**, showing an increase in revenue generated by more accurately targeting lower-risk loans. This boost in income illustrates the model's ability to improve profitability through better loan risk assessment.

Challenger Model 3 demonstrates significant business value by effectively reducing the Non-Performing Loan Ratio, decreasing potential losses from defaults, and increasing net interest income. This scenario shows the extreme impact of the model's risk prediction capabilities, which, even if applied less aggressively, could still bring substantial financial benefits. By leveraging Challenger Model 3, Lending Club can improve profitability, better manage risk, and drive a more sustainable lending strategy.







END OF REPORT

Thank You For Reading