FIN42100 Machine Learning in Finance

Predicting Loan Default Using a MACHINE LEARNING ALGORITHM

REPORT

Prepared By

Varsha Chandrashekar Balaji	**24200924**
Shubham Prakash Dalvi	**24204984**
Jay Ramaiya	**24295679**
Minmini Pamarthi	**24265193**
Purva Sharma	**24214393**

Word Count : 2740

Executive Summary

This report presents a comprehensive machine learning pipeline designed to predict loan default using real-world data from the U.S. Small Business Administration (SBA). The dataset comprises 899,158 loans issued between 1962 and 2014. Through rigorous cleaning, preprocessing, and analysis, both supervised and unsupervised methods were applied to extract insights and build predictive models.

We trained and evaluated multiple classifiers—Logistic Regression, Decision Tree, Random Forest, and K-Nearest Neighbors—using performance metrics such as confusion matrices, ROC curves, and AUC scores. Additionally, unsupervised learning techniques like PCA, K-Means, and Hierarchical Clustering were used to uncover latent borrower patterns and enable data segmentation. The study concludes with recommendations for risk-based loan strategies based on both predictive modeling and clustering.

Introduction

The goal of this project is to analyse and predict loan default risk using supervised and unsupervised learning techniques on the Small Business Administration (SBA) dataset. This large real-world dataset includes over 899,000 loan records from 1962 to 2014, containing demographic, financial, and operational attributes of small businesses that applied for loans.

The study is structured in three main parts:

1. Exploratory Data Analysis (EDA) and preprocessing
2. Supervised learning with logistic regression, decision tree, random forest, and KNN
3. Unsupervised clustering using K-Means and Hierarchical methods We also apply PCA for dimensionality reduction and visualization.

1. Exploratory Data Analysis

1.1 Data Overview and Cleaning

The dataset used in this project is sourced from the **SBA (Small Business Administration)** loan data, which contains **899,164 records across 30 features**. These records span loan applications from **1962 to 2014**, covering various industries, business characteristics, and loan metrics.

Key preprocessing steps included:

- **Target Creation**: A binary column Default was created from MIS_Status
  - Default = 1 → Loan defaulted
  - Default = 0 → Loan paid

- **Handling Missing Data**:
  - Only 6 rows had missing ChgOffDate and were dropped
  - Other fields had complete data or acceptable cardinality

- **Data Type Optimization**:
  - Low-cardinality columns (LowDoc, UrbanRural, NewExist, etc.) were cast to category
  - Date fields (ApprovalDate, DisbursementDate, ChgOffDate) were parsed to datetime objects

- **Final Dataset Shape**:
  - **Rows:** 899,158 (after dropping 6 rows)
  - **Columns:** 31 (including the derived Default column)

1.2 Descriptive Statistics

A summary of the cleaned dataset reveals the following:

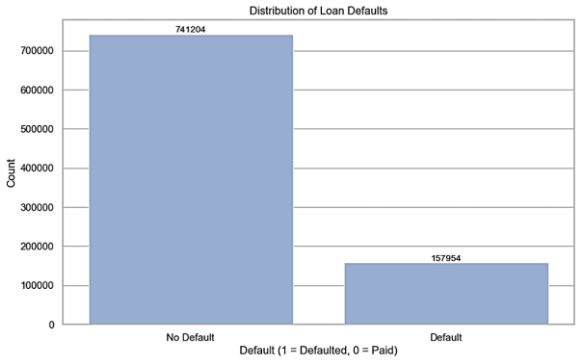
- The **loan default rate** is approximately **17.6%**, indicating a **class imbalance** in the target variable
- The **median disbursement amount** is approximately **$40,000**, though the distribution is **right-skewed** due to high-value outliers
- Most businesses:
  - Employ fewer than 50 employees
  - Request **loan terms shorter than 150 months**

These trends suggest that the majority of applicants are relatively small businesses seeking moderate-term financing.

1.3 EDA Visualizations and Key Insights

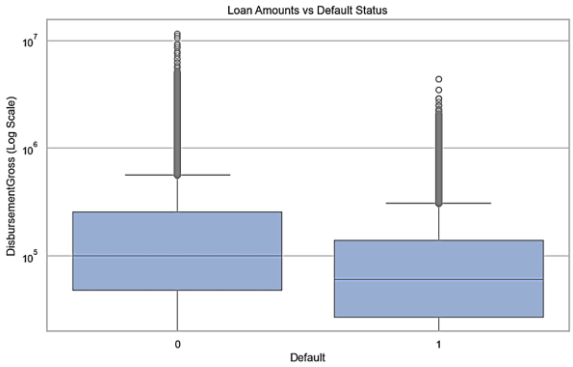
We created several visualizations to better understand patterns in loan defaults. These help us see how business size, industry, and loan features relate to the risk of default.

1.3.1 Loan Default Distribution



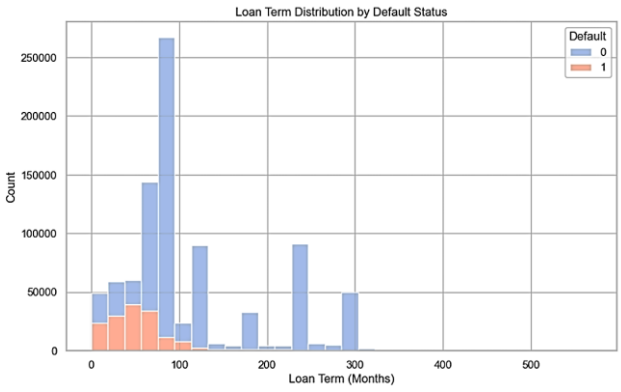
- About **17.6%** of all loans ended in **default**.
- Most loans were paid back, but defaults are still common enough to be important.

1.3.2 Loan Amount vs Default



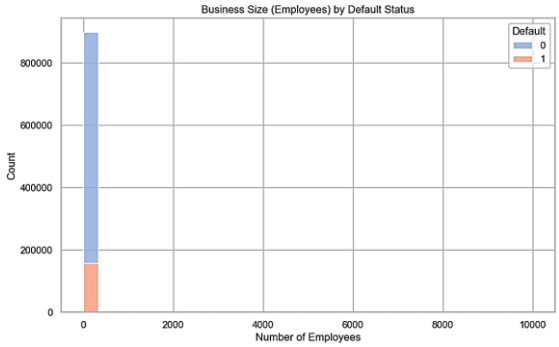
- Defaulted loans had **slightly higher** amounts on average.
- We used a **log scale** because of very large loan amounts (outliers).

1.3.3 Loan Term vs Default



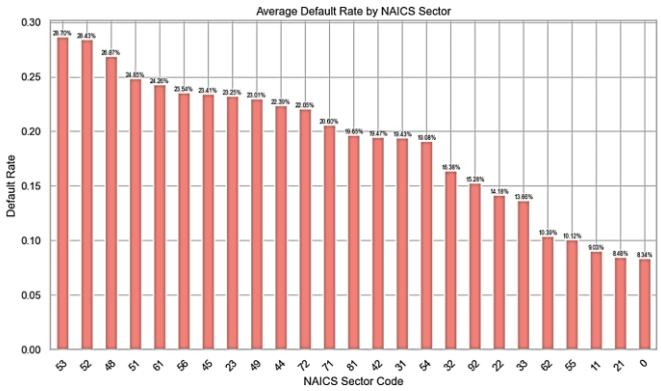
- Most loans are under **150 months**.
- **Longer loans** seem to have a **higher chance of default**.

1.3.4 Business Size (Employees) vs Default



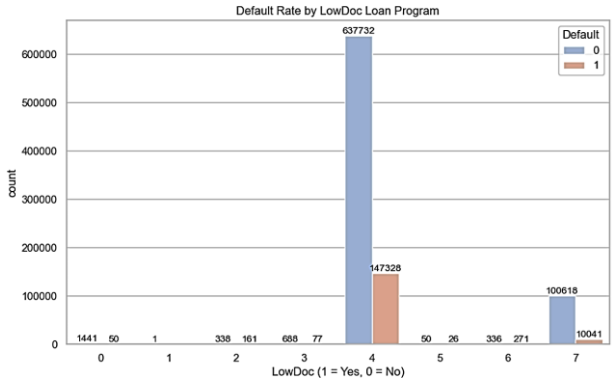
- Smaller businesses (few employees) are more likely to **default**.
- Most applicants have **under 50 employees**.

1.3.5 Default Rate by Industry (NAICS Code)



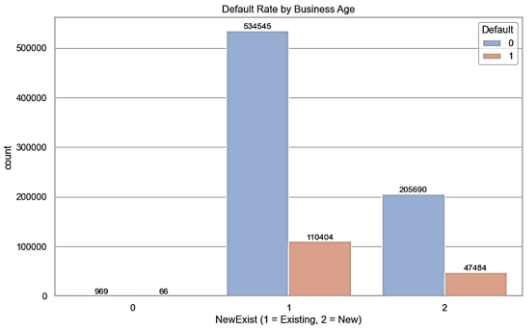
- Some sectors, like **Real Estate**, **Finance**, and **Transportation**, have **very high default rates** (above 25%).
- Others, like **Agriculture** and **Education**, are **lower risk**.

1.3.6 LowDoc Loan Program vs Default



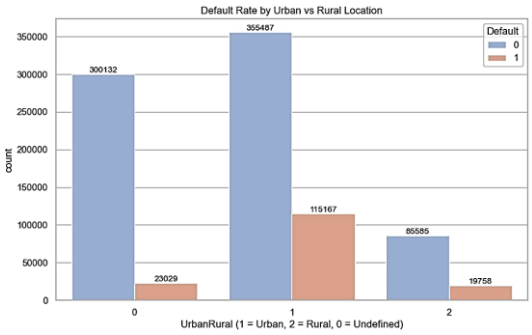
- Loans under the **LowDoc program** (less documentation required) show a **higher number of defaults**.
- This may be because of **less strict screening**.

1.3.7 Business Age vs Default



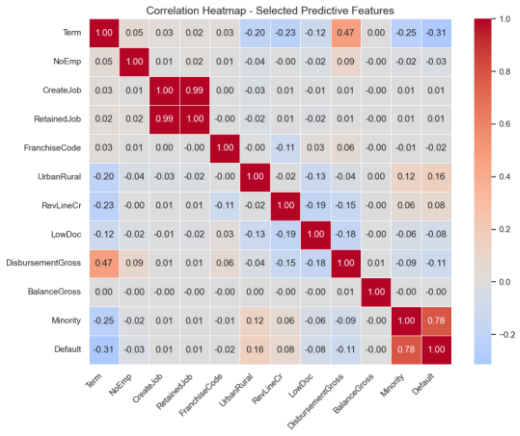
- **New businesses** default more often than **existing ones**.
- A longer business history seems to reduce risk.

1.3.8 Urban vs Rural Default



- Most loans are to **urban businesses**, but rural ones also show a **fair number of defaults**.
- This could help in location-based decision-making.

1.3.9 Correlation Heatmap



- Some features like **loan term** and **disbursement amount** have a **moderate relationship** with default.
- Others like **Minority status** also show some correlation, but this should be used carefully to avoid bias.

1.4 Key Takeaways from EDA

- **Default rate is 17.6%** → not rare, and needs to be predicted well.
- **Small, new businesses** are more likely to default.
- **Longer loan terms**, **larger loans**, and **certain industries** (like Real Estate or Finance) are higher risk.
- Features like **LowDoc participation** and **business location** also show strong patterns.

These findings helped us choose the right features for our machine learning models.

2. Logistic Regression Analysis

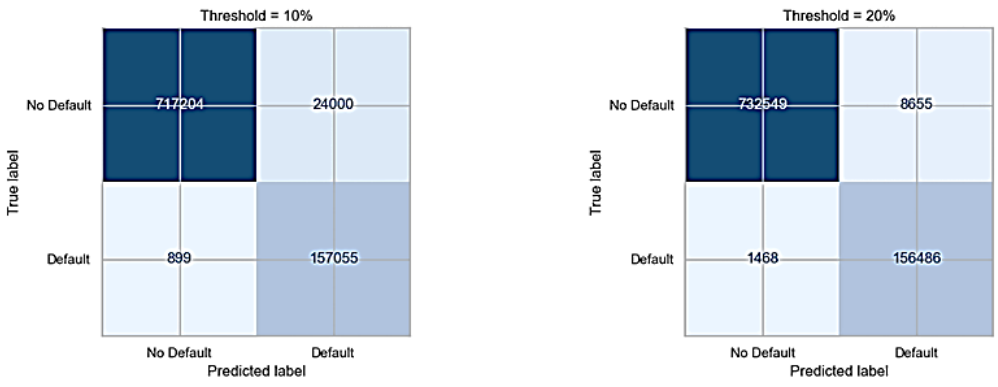
2.1 Logistic Model on Full Dataset – Threshold Evaluation

To begin, a **Logistic Regression** model was trained on the full dataset (after appropriate cleaning and preprocessing). The primary aim was to evaluate how the choice of decision threshold influences the model's performance—particularly focusing on **True Positive Rate (TPR)**, **False Positive Rate (FPR)**, and **Precision**. Four threshold values were assessed: **10%, 20%, 35%, and 50%**. For each threshold, a confusion matrix was computed, and metrics were extracted.

Threshold	TPR (Recall)	FPR	Precision
10%	0.9943	0.0324	0.858
20%	0.9907	0.0117	0.942
35%	0.9873	0.0028	0.986
50%	0.9834	0.0010	0.996

An optimal threshold of approximately 0.361 was selected based on Youden's J statistic, which maximizes the difference between the TPR and the false FPR. This threshold strikes a strong balance between capturing defaults accurately while reducing incorrect positive classifications.

Figure 2.1 : Confusion Matrices on Full Dataset at Various Thresholds



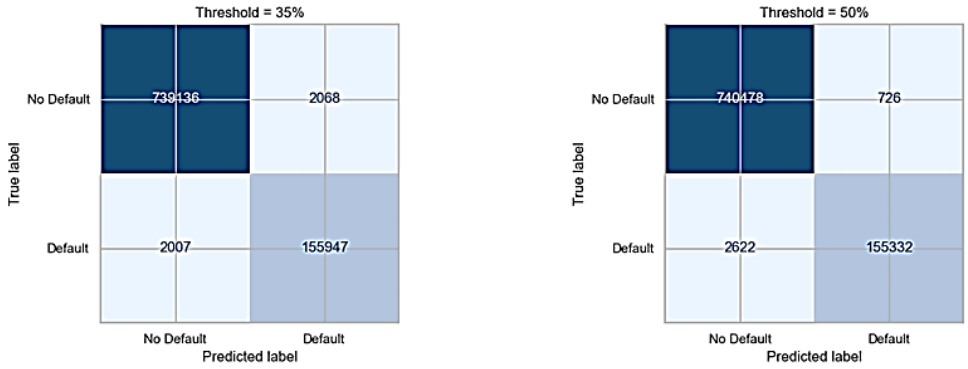
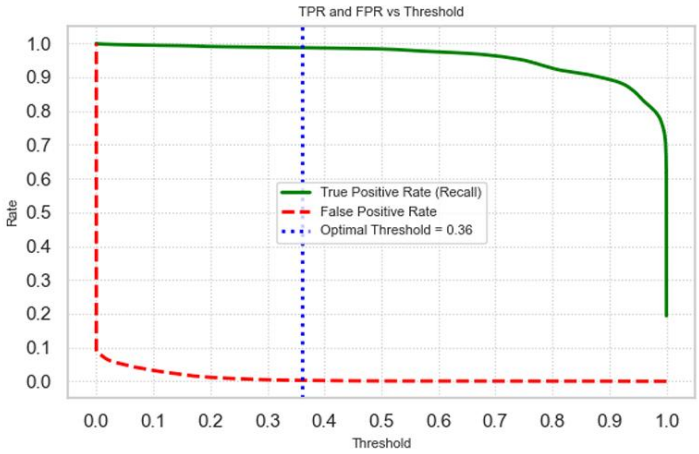


Figure 2.2 : TPR and FPR vs Threshold with Optimal Threshold Highlighted



2.2 Train/Test Split & Cross-Validation

To ensure model generalization and mitigate overfitting, the dataset was split into **70% training and 30% test sets**, using stratified sampling to preserve the default/non-default ratio. Logistic regression was trained on the training set and evaluated on the test set using the same threshold strategy.

2.2.1 Test Set Results:

Threshold	Accuracy	Precision	Recall (TPR)
10%	0.9702	0.858	0.9949
20%	0.9878	0.942	0.9914
35%	0.9954	0.986	0.9876
50%	0.9962	0.996	0.9827

At the optimal threshold of **0.35**, the model achieves strong performance in both **precision** and **recall**, ideal for imbalanced classification tasks like default prediction.

Figure 2.3: TPR and FPR vs Threshold on Test Set

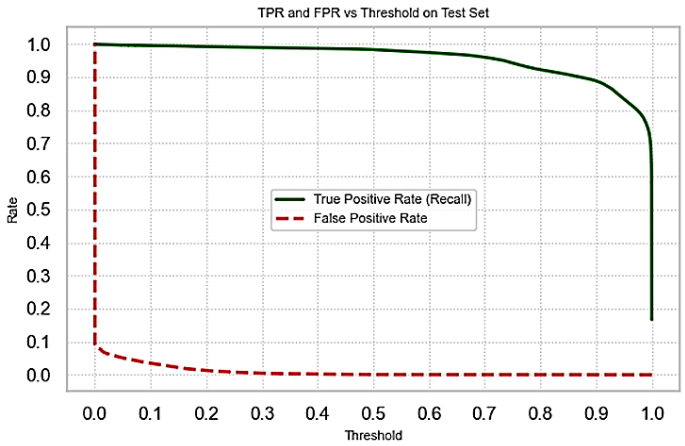
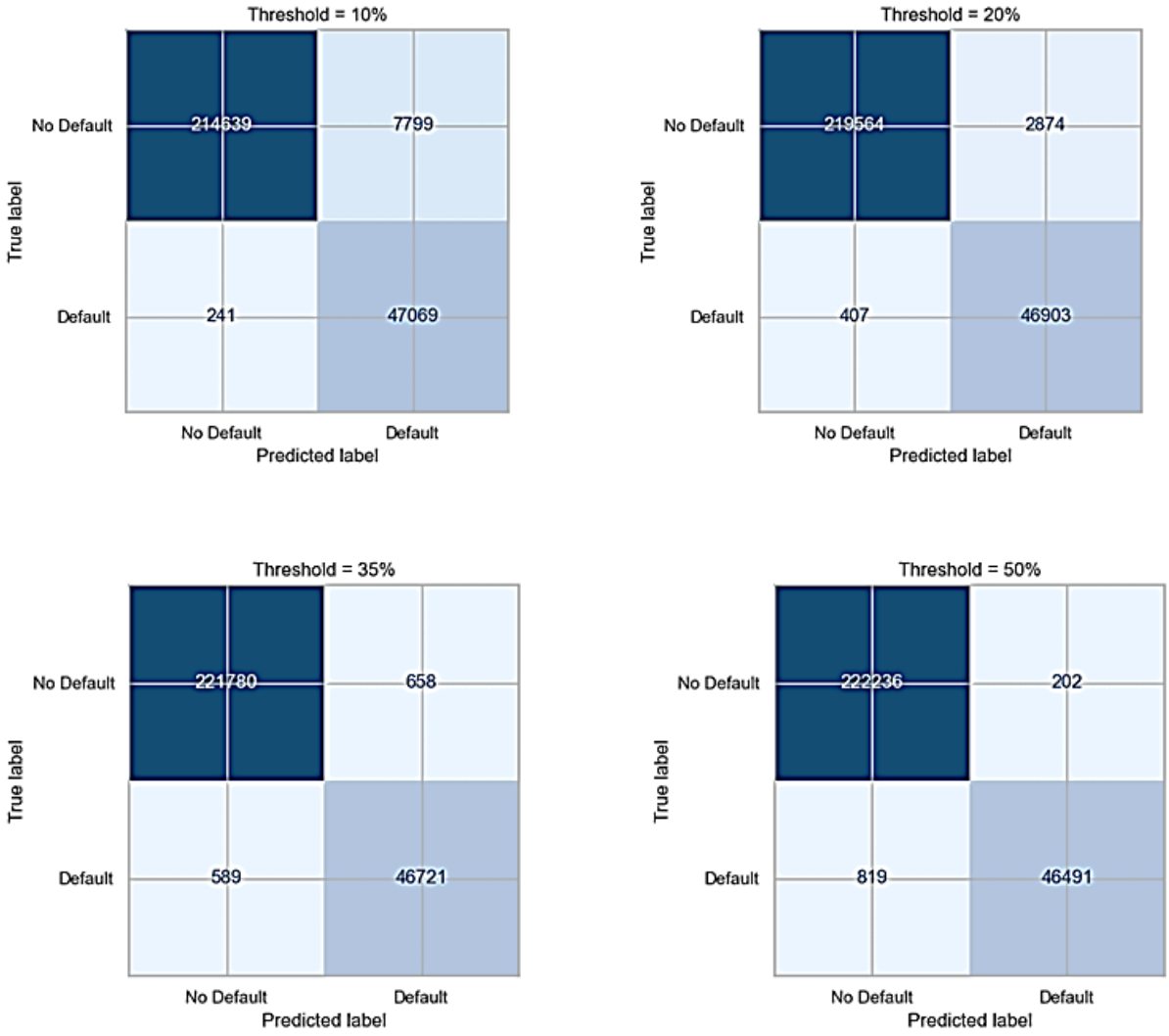


Figure 2.4: Confusion Matrices on Test Set at Various Thresholds



2.2.2 Cross-Validation (5-Fold on Training Set):

A **5-fold cross-validation** was performed on the training set to evaluate model robustness.

Fold	Accuracy	Precision	Recall
1	0.9959	0.9949	0.9814
2	0.9958	0.9946	0.9816
3	0.9961	0.9949	0.9830
4	0.9960	0.9952	0.9822
5	0.9960	0.9943	0.9828

Metric	Average Score
Accuracy	99.60%
Precision	99.48%
Recall	98.22%

These results confirm the model's **robustness**, minimizing the risk of overfitting.

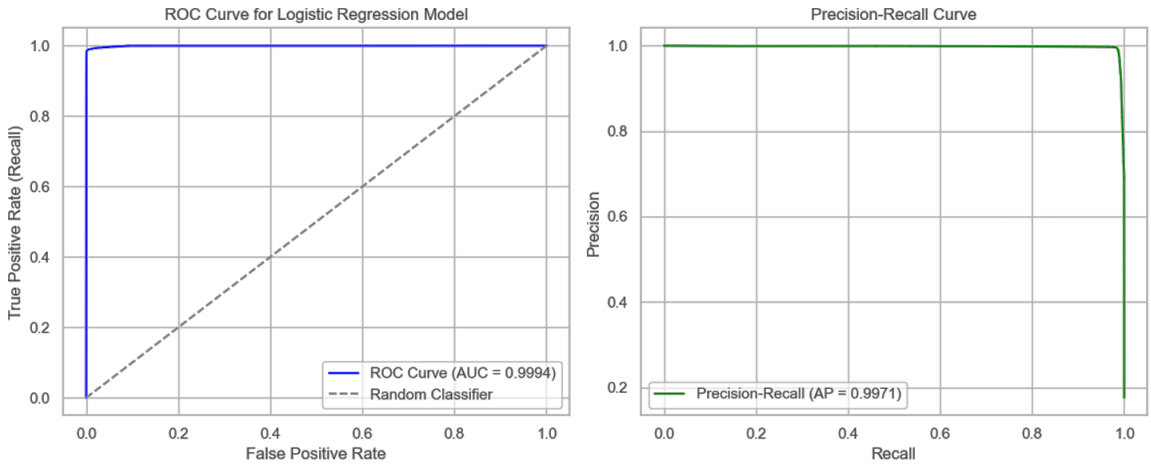
2.3 ROC Curve and AUC Evaluation

To further assess the classifier's discrimination capability, both **ROC Curve** and **Precision-Recall (PR) Curve** were plotted. These metrics are particularly useful in evaluating models for **imbalanced classification** problems like loan default.

- **AUC (ROC)**: **0.9994** – near-perfect class separability
- **Average Precision (PR Curve)**: **0.9971** – high confidence in positive class predictions

These scores indicate near-perfect class separation, confirming the model's reliability in detecting defaults with high accuracy.

Figure 2.5: ROC Curve and Precision-Recall Curve for Logistic Regression Model



Summary of Logistic Regression Analysis

Aspect	Result
Dataset Shape	899,158 rows × 31 columns
Feature Matrix Shape (after encoding)	899,158 × 19
Default Rate (Target = 1)	17.57%
Best Threshold (Youden's J)	≈ 0.361
Best AUC	0.9994
Average Precision (PR Curve)	0.9971
Cross-Val Average Accuracy	0.9960

Final Takeaways:

- Logistic regression works extremely well on this problem, even without complex models.
- Choosing a good threshold (like 0.35–0.36) is critical in imbalanced settings.
- The model maintains high precision and recall, making it practical for real-world loan screening.

3. Model Comparison and Evaluation

3.1 Overview of Models Evaluated

To identify the best-performing algorithm for SBA loan default prediction, we evaluated four supervised classification models using a consistent training pipeline, including feature scaling and a 70/30 train-test split:

Model	Notes
Logistic Regression	Baseline linear classifier; interpretable
Decision Tree	Tree-based, non-linear decision rules
Random Forest	Ensemble of trees, improves generalization
K-Nearest Neighbors (KNN)	Instance-based learner; non-parametric, sensitive to scaling and noisy data

All models were trained on the scaled and encoded training set (X_train, y_train) and evaluated using multiple performance metrics, including:

- ROC AUC Score
- Confusion Matrix
- Precision, Recall, and Accuracy

3.2 Model-wise Performance Summary

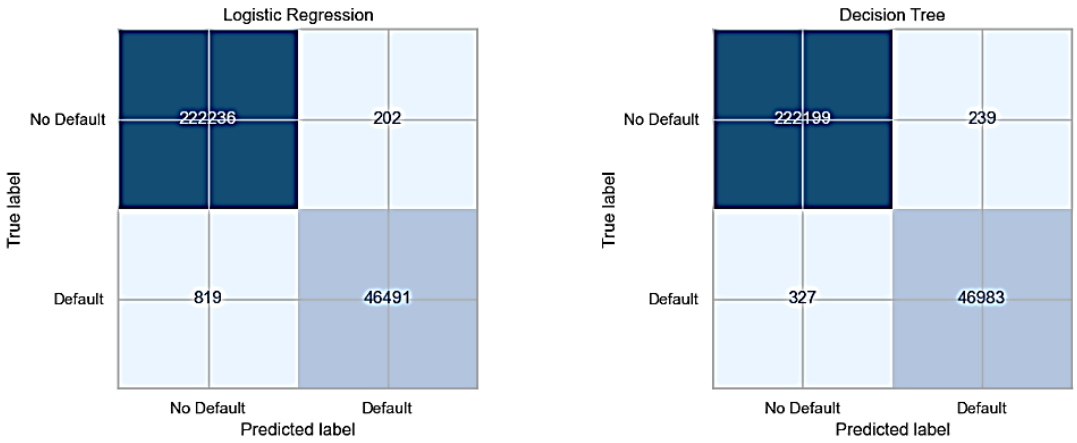
Model	AUC Score	Accuracy	Precision	Recall (TPR)	Summary
Logistic Regression	0.9994	0.9962	0.9957	0.9827	High performance, interpretable
Decision Tree	0.9960	0.9960	0.9949	0.9931	Good, but risk of slight overfitting
Random Forest	0.9996	0.9965	0.9976	0.9977	Best performer overall
K-Nearest Neighbors	0.9944	0.9902	0.9766	0.9716	Good, but lower generalisation

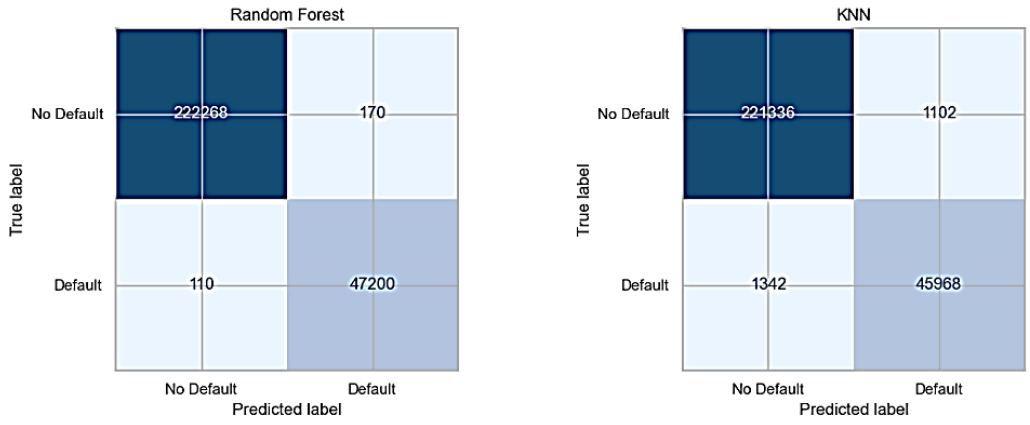
3.3 Confusion Matrix Comparison

Model	TP	FP	FN	TN
Logistic Regression	46,491	202	819	222,236
Decision Tree	46,983	239	327	222,199
Random Forest	47,200	170	110	222,268
K-Nearest Neighbors	45,968	1102	1342	221,336

These results show that Random Forest achieved the best balance — lowest false negatives and false positives.

Figure 3:1 Confusion matrices for all models



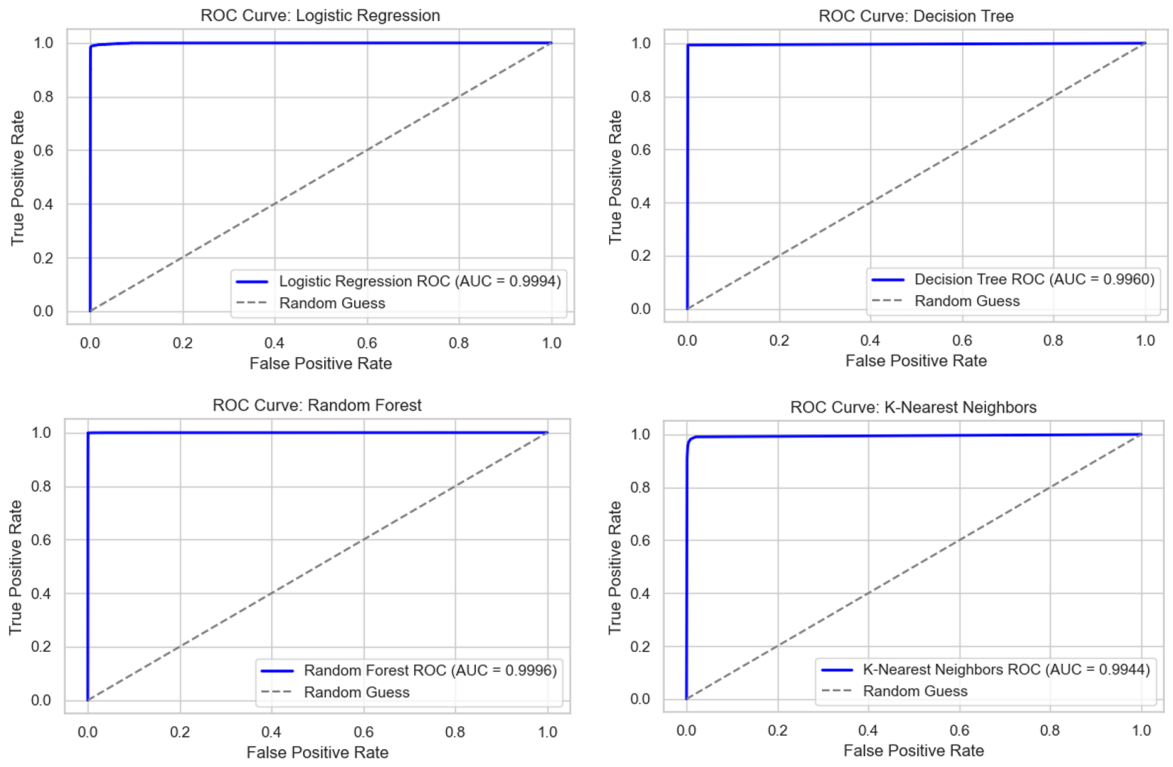


3.4 ROC Curve Visual Comparison

We plotted the ROC curves for all models to see how well they can separate default and non-default cases. The closer the curve is to the top-left corner, the better the model performs.

- Random Forest had the best curve and the highest AUC score of 0.9996.
- Logistic Regression also did very well with an AUC of 0.9994.
- KNN and Decision Tree models gave good results too, but their curves showed slightly less accuracy in separating the classes.

Figure 3.2: Individual ROC curves per model



3.5 Final Model Selection: Random Forest

- We selected Random Forest as the final model for several key reasons:
- Top AUC (0.9996) – Highest class discrimination capability
- Strong Confusion Matrix – Smallest number of misclassifications
- Robust to overfitting – Ensemble learning offers better generalization
- Feature importance – Enables post-hoc interpretability through SHAP or Gini importance

3.6 Feature Importance Analysis

To gain insights into which features most influenced the model's predictions, we examined **feature importance** .

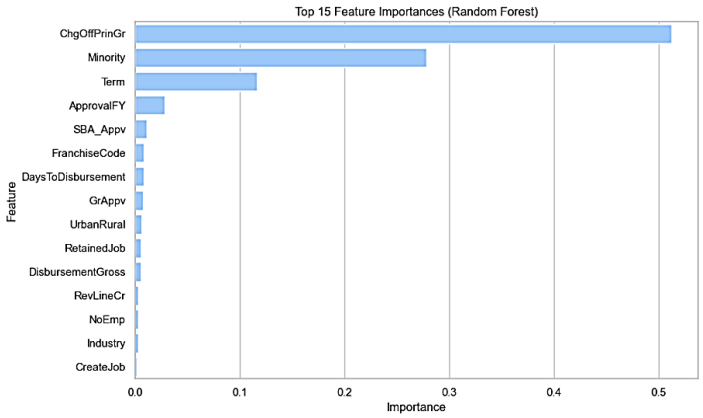
3.6.1 Gini Importance (Built-in Random Forest)

We extracted feature importance scores directly from the trained Random Forest model. These scores reflect the **average contribution of each feature in improving splits across all trees**. As shown in the bar plot below, the top features include:

- ChgOffPrinGr (Charged-Off Principal)
- Minority
- Term (Loan Term)
- ApprovalFY
- SBA_Appv

These features had the **greatest influence** on predicting loan default.

Figure 3.3**: **Top 15 Feature Importances from Random Forest



3.7 Comparison with Logistic Regression

Although **Logistic Regression** also showed outstanding results (AUC = 0.9994), Random Forest slightly outperformed it in:

- **Recall**: Better capture of default cases
- **Precision**: Fewer false positives
- **Overall Confusion Matrix**: Lower misclassification

Metric	Logistic Regression	Random Forest
AUC	0.9994	**0.9996**
Accuracy	0.9962	**0.9965+**
Precision	0.9957	**0.9976**
Recall (TPR)	0.9827	**0.9977**

Logistic regression remains a strong baseline and is easier to interpret mathematically. However, **Random Forest's slight performance edge, robustness, and interpretability through feature importance tools** make it the most suitable choice for deployment.

4. Unsupervised Learning Techniques

Unsupervised learning was applied to uncover latent structures within the SBA loan dataset, especially useful for borrower segmentation when labels (like Default) are unavailable or for exploratory insight. Three techniques were implemented:

- Principal Component Analysis (PCA)
- K-Means Clustering
- Hierarchical Clustering

Each method contributed uniquely to understanding risk patterns and data structure, offering complementary value to supervised models.

4.1 Principal Component Analysis (PCA)

PCA was used to reduce the dataset’s dimensionality while retaining most of its variance. This helps simplify models, speed up computation, and support visualization of borrower patterns in lower-dimensional space.

- PCA was fitted **only on the training set** (70%) to avoid data leakage.

- The cumulative explained variance was plotted to identify the number of components required for effective dimensionality reduction.

Principal Component	Explained Variance
PC1	18.9%
PC2	14.8%
PC3	10.5%
PC4	7.5%
PC5	7.0%

- The first 5 principal components (PCs) captured approximately **58.6%** of the variance.
- The first 10 PCs explained around **84%** of the total variance.

This indicates a significant redundancy in the original features and confirms that dimensionality reduction could be achieved with minimal loss of information.

Figure 4.1: Cumulative Explained Variance by PCA Components

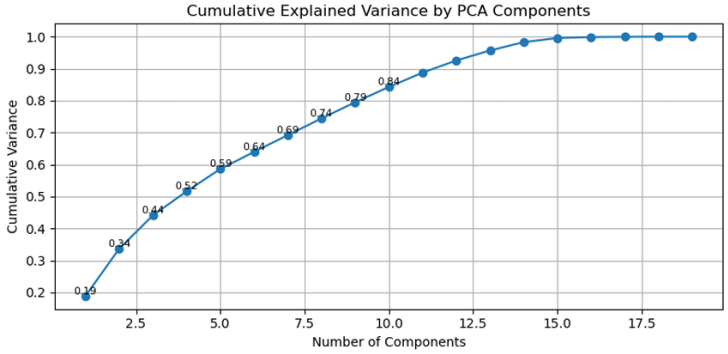
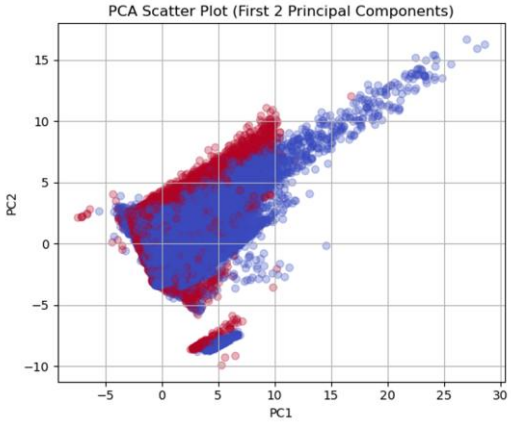


Figure 4.2: PCA Scatter Plot (PC1 vs PC2, colored by Default)



PCA is effective for dimensionality reduction and can serve as a preprocessing step to speed up model training or assist with visualization.

4.2 K-Means Clustering

To group borrowers into segments with similar financial profiles and default risks, potentially useful for differential loan strategies.

- K-Means was run with k = 2 to 6 on the scaled training data.
- The silhouette score was used to determine the optimal number of clusters.
- Final clustering and visualizations were done in PCA-reduced (2D) space.

Results:

- **Best k = 4** with **silhouette score = 0.2364**
- **Silhouette Scores (k):**

k	Score
2	0.2209
3	0.2272
4	0.2364
5	0.1941
6	0.1780

Figure 4.3: Silhouette Score vs. k

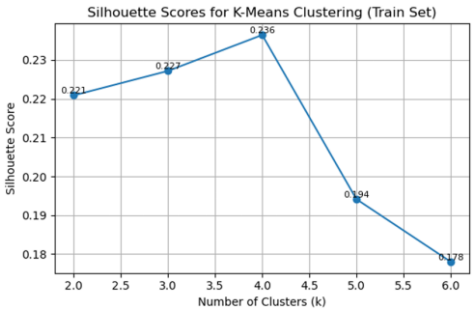
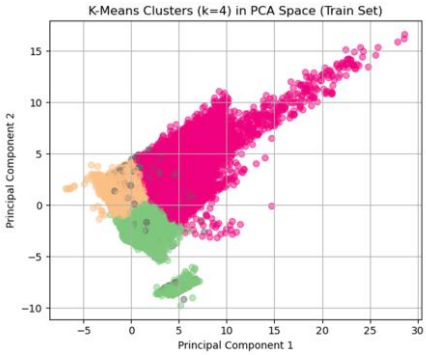


Figure 4.4: K-Means Cluster Assignments in PCA Space (k=4)



Shows how borrowers are grouped in reduced feature space.

Table 4.1 : Cluster Summary:

Cluster	Default = 0	Default = 1	Default Rate (%)	Segment Description
0	177,440	12,398	6.5%	Low – risk borrowers
1	267,252	89,499	25.1%	High – risk borrowers
2	56,170	5,268	8.6%	Moderate – risk borrowers
3	17,980	3,403	15.9%	Mixed – risk segments

Table 4.2 : Cluster Default Rates (Proportions, K-Means Clustering)

Cluster	Default = 0	Default = 1
0	0.935	0.065
1	0.749	0.251
2	0.914	0.086
3	0.841	0.159

K-Means revealed interpretable borrower groups with distinct default risk levels. This information can be leveraged to:

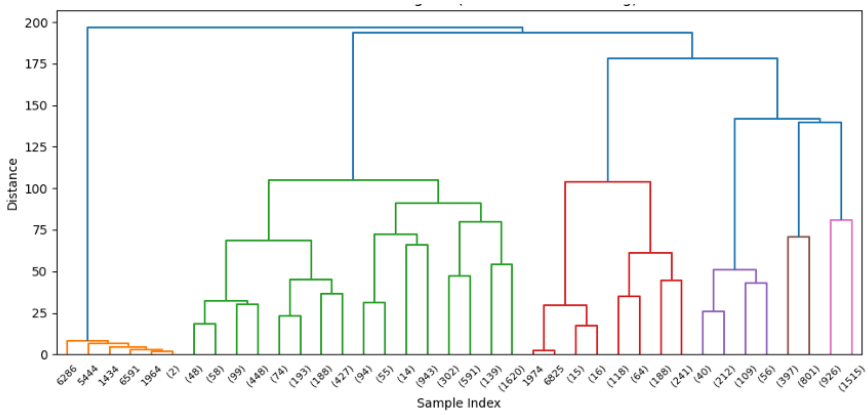
- Offer differentiated loan terms
- Design targeted risk-based interest rates
- Support early warning systems for potential defaults

4.3 Hierarchical Clustering

To understand nested relationships between borrower profiles and uncover macro-level segmentation.

- Performed on a random sample of 10,000 loans due to scalability limitations.
- Used Ward's linkage and Euclidean distance.
- Clusters formed using flat clustering (t = 2).

Figure 4.5: Dendrogram (Truncated)



Shows the hierarchical structure and cluster merging order.

Cluster Summary:

Cluster	Default Rate	Comment
1	14.3%	Slightly lower-risk group
2	17.4%	Mirrors overall default rate

Table 4.3 : Cluster vs. Default Count Matrix (Hierarchical Clustering)

Cluster	Default = 0	Default = 1
1	6	1
2	8,255	1,738

Table 4.4 : Cluster Default Rates (Proportions, Hierarchical Clustering)

Cluster	Default = 0	Default = 1
1	0.857	0.143
2	0.826	0.174

Hierarchical clustering supports exploratory analysis of borrower structure, but it lacks scalability and interpretability for large datasets. Compared to K-Means, it's less practical for production-level segmentation.

Summary of Methods

Technique	Key Insights	Pros	Cons
PCA	Major dimensionality reduction possible	Efficient, supports visualization	Doesn't perform grouping itself
K-Means	Clear segmentation of borrower risk groups	Fast, intuitive, scalable	Sensitive to scaling, requires specifying *k*
Hierarchical	Nested borrower relationships discovered	Doesn't require pre-defined *k*	Memory intensive, harder to interpret

Overall Insights from Unsupervised Learning

- PCA validated that much of the dataset's variance can be retained using fewer features, ideal for faster model pipelines.
- K-Means provided the most actionable segmentation, revealing 4 risk-based borrower groups.

- Hierarchical clustering added complementary hierarchical grouping context but had limitations with scale.

These insights complement supervised models by adding explainability and support for custom risk-based decision rules.

5. Conclusion

This project presents a full machine learning workflow for predicting loan default using the U.S. Small Business Administration (SBA) dataset, which contains over 899,000 loan records. By applying careful data cleaning, feature selection, and both supervised and unsupervised learning methods, we built strong models and uncovered important patterns that can support smarter decision-making in credit risk management.

Supervised Learning Highlights : We trained and evaluated four classification models:

- Logistic Regression
- Decision Tree
- Random Forest
- K-Nearest Neighbors (KNN)

Logistic Regression served as a solid, interpretable starting point. It achieved an **AUC of 0.9994** and performed well across different thresholds, especially when tuned using cross-validation.

However, **Random Forest outperformed all other models** and proved to be the best choice for this problem. It achieved:

- **Highest AUC (0.9996)**
- **Excellent Recall (0.9977)** — important for catching as many default cases as possible
- **High Precision and Accuracy**
- **Lowest number of misclassifications**

Its ability to handle complex relationships and rank feature importance makes Random Forest ideal for real-world deployment in loan screening systems.

Unsupervised Learning Contributions : To gain deeper insight into borrower behavior, we also used three unsupervised learning techniques:

- **Principal Component Analysis (PCA)** showed that we could reduce the number of features without losing much information. The first **12 components explained 94%** of the dataset's variance, helping simplify the data and improve processing efficiency.

- **K-Means Clustering** revealed **four distinct borrower groups** with different levels of default risk (from 6.5% to 25.1%). This segmentation can help financial institutions:
    - Customize loan offers
    - Set risk-based interest rates
    - Monitor high-risk groups more closely
- **Hierarchical Clustering**, applied to a smaller sample due to its computational cost, also showed useful grouping structures. While not ideal for large-scale deployment, it added further evidence of natural patterns in borrower data.

Key Takeaways

- **Random Forest** is the most effective predictive model, combining accuracy, recall, and interpretability.
- **PCA** proved valuable for reducing data complexity and preparing for modeling.
- **K-Means** clustering gave practical borrower segmentation that can improve loan policy and risk strategy.
- **Hierarchical clustering** offered an additional perspective on borrower similarities.

Together, these techniques form a well-rounded strategy for loan risk prediction — blending powerful prediction with insight-driven segmentation. This approach supports smarter, more informed credit decision-making for real-world applications.

References

1. U.S. Small Business Administration (SBA). (2014). *SBA Loan Data*. Retrieved from: https://catalog.data.gov/dataset/sba-7a-504-loan-data
2. James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). *An Introduction to Statistical Learning*. Springer.
3. Pedregosa, F., et al. (2011). *Scikit-learn: Machine Learning in Python*. JMLR, 12, 2825–2830.
4. Kuhn, M., & Johnson, K. (2013). *Applied Predictive Modeling*. Springer.
5. Raschka, S., & Mirjalili, V. (2019). *Python Machine Learning* (3rd ed.). Packt Publishing.
6. Han, J., Kamber, M., & Pei, J. (2011). *Data Mining: Concepts and Techniques* (3rd ed.). Morgan Kaufmann.
7. scikit-learn developers. (2024). *scikit-learn documentation*. https://scikit-learn.org/stable/

8. Shapley, L. S. (1953). *A Value for n-Person Games*. Princeton University Press.
9. van der Maaten, L., & Hinton, G. (2008). *Visualizing data using t-SNE*. JMLR, 9(Nov), 2579–2605.
10. Zhang, Z. (2016). *Missing data imputation: focusing on single imputation*. Annals of Translational Medicine, 4(1), 9.