BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE, PILANI

Work Integrated Learning Program Division

Post Graduate Program
in
Artificial Intelligence and Machine Learning

CAPSTONE PROJECT
on
INSURANCE RENEWAL PREDICTION

Submitted in partial fulfilment of the requirements
for the award of
Post Graduate Program in Artificial Intelligence and Machine Learning

By
Akash Mohanty (2024AIML133)
Ishika Gupta (2024AIML088)
Namratha S Hegde (2024AIML118)
Thrilok Attota (2023AIML120)
Yaswanth Reddy Dasari (2024AIML010)

Under the supervision of
Mr. Tapas Chakraborty

Project work carried out at
BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE
Pilani (Hyderabad) INDIA

Monday, 01 September 2025 – Sunday, 26 October 2025

In partial fulfilment of the requirements of the Capstone Project, embodies the work carried out by the students under my supervision.

Tapas Chakraborty

Place: **Signature of Mentor**

Date: **Name: Tapas Chakraborty**

Acknowledgements

We, the team behind this capstone project titled **"Insurance Renewal Prediction"**, would like to express our sincere gratitude to everyone who contributed to the successful completion of this work.

First and foremost, we extend our heartfelt thanks to our mentor, **Mr. Tapas Chakraborty**, for his continuous guidance, support, and valuable feedback throughout the project. His mentorship helped us approach the problem with clarity and apply machine learning concepts effectively to achieve meaningful results.

We are deeply thankful to **BITS Pilani – Work Integrated Learning Program Division** for providing us with this opportunity and for offering a strong academic framework that enabled us to apply theoretical knowledge to a real-world business challenge. The structured learning environment and access to quality academic resources played a vital role in the completion of this project.

We would also like to express our gratitude to all faculty members of the **Artificial Intelligence and Machine Learning** program for their valuable inputs and for building a strong foundation in data science and machine learning methodologies that supported our work.

Finally, we wish to acknowledge the efforts and collaboration of all our team members—**Akash, Ishika, Namratha, Thrilok, and Yaswanth**. The teamwork, dedication, and shared commitment from each member were key to successfully completing this project.

We are grateful for this enriching experience, which has enhanced our technical understanding and strengthened our problem-solving abilities.

Table of Contents

1. Overview

In the rapidly evolving insurance industry, **customer retention** has emerged as a critical business priority. With increasing competition and policy-switching options, insurers face mounting challenges in maintaining long-term customer relationships. Recent global reports highlight the severity of this issue — *Insurance Times (2023)* reported that **44.4% of home insurance customers** switched providers in 2022, reflecting large-scale churn in mature markets. Similarly, *Insurance Asia (2024)* observed that higher premiums and reduced coverage have led to elevated switching rates across property and casualty insurance segments.

The financial implications of churn are substantial. According to the *Forbes Business Council (2022)*, acquiring a new customer can cost **five to twenty-five times more** than retaining an existing one, emphasizing the economic value of renewal-focused strategies. At the same time, *Reuters (2025)* reported that India's LIC experienced a notable rise in quarterly profits driven by **renewed policies**, underscoring how retention directly supports profitability in the insurance sector.

This project aims to build a predictive framework that forecasts whether an insurance policyholder will renew their policy for the next term. The analysis leverages a dataset sourced from Kaggle's "Insurance Renewal Prediction" repository, which includes customer demographics, payment behavior, underwriting scores, and premium-related attributes. By identifying patterns associated with renewal decisions, the project seeks to provide actionable insights that can help insurers proactively engage customers at risk of non-renewal. While this study focuses on customer-level behavioral data, broader factors such as insurer pricing and policy decisions also play a significant role in influencing renewal trends.

Ultimately, the goal is to demonstrate how **data-driven analytics** can be used to reduce churn, optimize marketing efforts, and enhance long-term customer loyalty — translating predictive intelligence into measurable business impact for the insurance industry.

2. Sources / Useful Links

2.1. Dataset Source

Insurance Renewal Prediction Kaggle

2.2. Industry References

- Forbes Business Council. (2022, December 12). Customer Retention Versus Customer Acquisition. Retrieved from https://www.forbes.com/councils/forbesbusinesscouncil/2022/12/12/customer-retention-versus-customer-acquisition/
- InsuranceNewsNet. (2023, March 27). P/C insurers faced with accelerating churn and switching. Retrieved from P/C insurers faced with accelerating churn and switching behaviors
- Insurance Times. (2023, May 23). Home insurance market seeing considerable customer churn. Retrieved from Home insurance market seeing considerable customer churn
- Insurance Asia. (2024, June 20). Insurers face customer churn as prices rise in select products. Retrieved from Insurers face customer churn as prices rise in select products
- Reuters. (2025, August 7). India's LIC posts quarterly profit rise on higher premiums from renewed policies. Retrieved from LIC Q1 results: Profit up 5% to ₹10,987 cr on strong policy renewals
- The Guardian. (2024, June 11). 'It's game over after AXA raised my home insurance price by 70%.' Retrieved from It's game over after Axa raised my home insurance price by 70%
- The Guardian. (2025, July 5). Car insurance renewal price rises hit customers. Retrieved from Car insurance: Which? warns over hefty renewal price rises
- Life policies worth Rs1 trillion lapse Policies worth Rs1 trillion lapse

3. Problem Statement

Given a new customer's policy and behavioral information, we need to predict whether the policyholder will renew their insurance policy for the next term. The objective is to build a machine learning model that can classify each policy record into one of two categories — Renewed or Not Renewed — based on historical customer and policy attributes.

The solution should enable insurers to identify potential non-renewals in advance, allowing them to take proactive actions such as targeted communication, premium offers, or retention campaigns to improve customer loyalty and reduce churn.

4. Solution

The dataset used in this project was sourced from Kaggle's "Insurance Renewal Prediction" repository. It contains customer and policy-level data, including demographic details, payment behavior, underwriting scores, and premium information. Each record represents a policyholder, and the target variable indicates whether the policy was renewed (1) or not renewed (0).

Since the dataset is moderately sized and labeled, we approached the problem as a supervised binary classification task. The initial phase involved data preprocessing and feature engineering, where missing values, categorical variables, and outliers were systematically handled. Exploratory Data Analysis (EDA) was performed to understand key patterns and relationships between features and renewal outcomes.

In the modeling phase, various machine learning algorithms were explored to learn from historical customer data and predict renewal likelihood. The models were evaluated using metrics such as accuracy, precision, recall, F1-score, and ROC-AUC to ensure both predictive performance and business reliability.

This framework enables insurers to forecast renewal decisions for new policyholders and identify customers at high risk of churn, thereby supporting proactive retention initiatives and improving business profitability.

5. Which Type of ML Problem is This?

Insurance Renewal Prediction is a supervised machine learning classification problem.
Each data point in the dataset represents a customer's policy record, including demographic, payment, and policy-related features. The target variable indicates whether the customer renewed the policy (1) or did not renew (0).

Since the output variable has two distinct categories, this is a binary classification task. However, the dataset is highly imbalanced, with a much larger number of renewals compared to non-renewals. This imbalance poses a modeling challenge, as a naïve model predicting every case as renewed could still achieve high accuracy while failing to identify customers who are likely to churn.

Therefore, while the problem remains a binary classification task, the imbalance nature of the data requires careful consideration of model evaluation metrics and sampling techniques to ensure that the minority class (non-renewals) is properly represented during training and evaluation.

6. Evaluation Metrics

Evaluating model performance is crucial to ensure that predictions are both statistically sound and aligned with business priorities. In this project, the dataset is highly imbalanced, with approximately 94 % of records representing renewals (class 1) and 6 % representing non-renewals (class 0). Because of this imbalance, metrics such as accuracy can appear deceptively high even for weak models. Therefore, multiple complementary metrics are used to assess how well the model distinguishes between the two classes.

6.1. Accuracy

Accuracy measures the proportion of correctly classified records among all predictions. Although it provides a quick overview of model performance, accuracy alone is not reliable for imbalanced data. For instance, a model that labels every policy as renewed would still achieve about 94 % accuracy, yet it would completely fail to identify non-renewing customers—the group of greatest business importance. Hence, accuracy is reported only as a basic indicator and not used for model selection.

6.2. Confusion Matrix

A confusion matrix summarizes predictions versus actual outcomes, revealing where errors occur.
In this binary classification task, it is represented as:

Predicted / Actual	0 (Not Renewed)	1 (Renewed)
0 (Predicted Not Renewed)	True Negative (TN)	False Negative (FN)
1 (Predicted Renewed)	False Positive (FP)	True Positive (TP)

From a business standpoint:

False Negatives (FN) → customers predicted as renewed but who actually did not renew → the most costly error.

Minimizing FN (i.e., maximizing Recall for Class 0) is therefore a top priority.
The confusion matrix helps visualize and quantify these trade-offs between correctly identified renewals and missed churners.

6.3. Precision, Recall and F1-Score

Because the dataset is imbalanced, these three metrics provide a clearer evaluation of model behavior. Precision (Class 0) measures, of all customers predicted as non-renewals, how many truly did not renew. Recall (Class 0) measures, of all actual non-renewals, how many were correctly identified—this is the key metric for this project. The F1-Score is the harmonic mean of precision and recall, offering a balanced measure of model quality. A high recall for class 0 ensures that the model successfully captures most potential churners, even if this slightly reduces precision.

6.4. ROC–AUC Score

The Receiver Operating Characteristic – Area Under the Curve (ROC–AUC) measures the model's ability to distinguish between renewal and non-renewal cases across decision thresholds. An AUC closer to 1 indicates stronger class separation. This metric complements recall by providing an aggregate measure of classification performance.

6.5. Summary

While all metrics are reported for completeness, Recall for Class 0 (Non-Renewal) is treated as the most important evaluation criterion. Maximizing this metric ensures that the majority of customers likely to churn are correctly identified, enabling timely retention actions. Even if overall accuracy or precision decreases slightly, improving recall provides the greatest business value by minimizing lost renewals and sustaining profitability.

7. Business Objectives and Constraints

The primary business objective of this project is to predict whether an existing insurance policyholder will renew their policy for the next term. By identifying customers who are likely not to renew, the company can take proactive measures such as offering discounts, sending personalized reminders, or improving customer engagement. This directly supports higher customer retention and improved profitability.

The model must generate actionable insights that can be integrated into the insurer's customer relationship management (CRM) or marketing systems, enabling the business team to target at-risk customers effectively.

Machine-learning models enable insurers to identify at-risk customers early by recognizing behavioral and transactional patterns within available policy data. In this project, features such as payment history, underwriting scores, number of premiums paid, income, and sourcing channel are used to understand the renewal behavior of customers. These insights help insurers focus retention strategies on customers showing higher churn risk and improve overall renewal performance.

However, it is also important to note that several real-world factors influencing policy renewals—such as changes in premium pricing, previous claim experience, customer satisfaction, and policy service quality—are not included in the current dataset. These insights, learned from industry research and blogs, indicate that while our model focuses on customer-level behavioral data, insurer-driven and experiential factors also play a critical role in renewal decisions.

The cost of misclassification is high, particularly when the model predicts a renewal for a customer who actually does not renew (False Negative). Missing these customers represents lost revenue and potential long-term customer value. Therefore, recall for the non-renewal class must be prioritized over overall accuracy.

The solution should maintain interpretability, allowing business stakeholders to understand which factors most influence renewal decisions, and scalability, so it can be applied to larger customer bases or extended to other insurance products in the future.

8. Data Understanding and Pre-Processing

8.1. Data Description

The dataset used for this project is sourced from Kaggle's *Insurance Renewal Prediction* repository. It contains information about existing insurance policyholders, including customer demographics, payment behavior, and policy details. Each record represents one customer's policy and includes both numerical and categorical features.

Key attributes include:

- **perc_premium_paid_by_cash_credit:** Proportion of premium payments made through cash or credit.
- **age_in_days:** Customer's age expressed in days.
- **income:** Annual income level of the policyholder.
- **Count_3–6_months_late, Count_6–12_months_late, Count_more_than_12_months_late:** Indicators of delayed premium payments in different time periods.
- **application_underwriting_score:** Underwriting risk score assigned during policy approval.
- **no_of_premiums_paid:** Total number of premiums paid to date.
- **sourcing_channel:** Channel through which the policy was sold (e.g., agent, online, partner).
- **residence_area_type:** Type of customer's residential area (urban or rural).
- **premium:** Policy premium amount.
- **renewal:** Target variable where *1 = Renewed* and *0 = Not Renewed*.
  The dataset contains approximately **79,853 records**, with a significant imbalance between the two target classes — around **94 % renewals** and **6 % non-renewals**.

8.2. How is the Data Pre-Processed?

The dataset used in this project was already available in a structured tabular format, containing customer- and policy-level information suitable for direct analysis. Therefore, the preprocessing mainly involved standard data cleaning and transformation steps to ensure quality and consistency before model training, rather than extensive feature engineering or raw data preparation.

The following steps were applied:

1. Handling Missing Values:
    a. Columns such as application_underwriting_score and payment delay counts contained missing values, which were imputed using suitable statistical or logical methods to preserve data integrity.
2. Encoding Categorical Variables:

   a. Categorical features (sourcing_channel and residence_area_type) were label-encoded or one-hot encoded to make them compatible with machine-learning.
3. Feature Scaling:
   a. Continuous variables like income and premium were normalized or standardized to ensure all features contributed proportionally during model training.
4. Feature Selection:
   a. Correlation and importance analyses were performed to eliminate redundant or weakly contributing variables.
5. Train–Test Split:
   a. The dataset was divided into training and testing sets to evaluate model generalization.

It is also important to note that while the dataset captures customer-level behavioral patterns, several key business and insurer-side factors influencing policy renewal are not included. These missing aspects include:

- The customer's previous claim history and settlement experience,
- Their satisfaction with the insurance company's past service,
- Any premium price changes or policy adjustments,
- The presence of competitor offers or discounts, and
- Special offers or loyalty benefits extended to the customer.

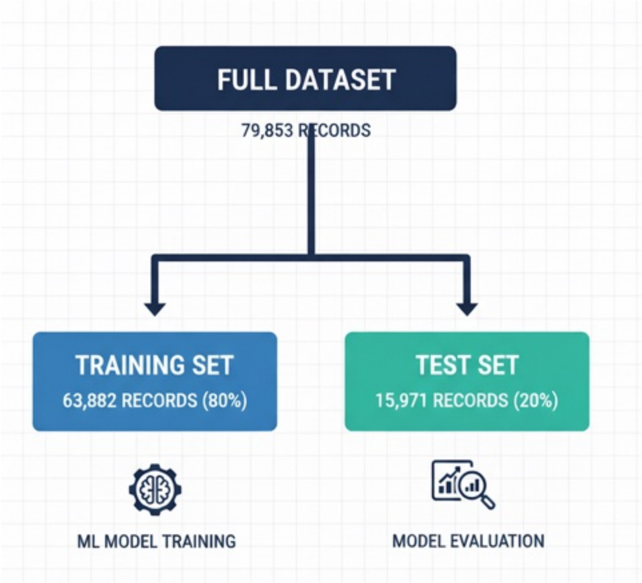
The absence of such variables limits the model's ability to fully represent real-world renewal dynamics. However, the available features still provide meaningful insight into behavioral trends and enable practical prediction of renewal outcomes based on customer and policy attributes.

9. Train and Test Ratio

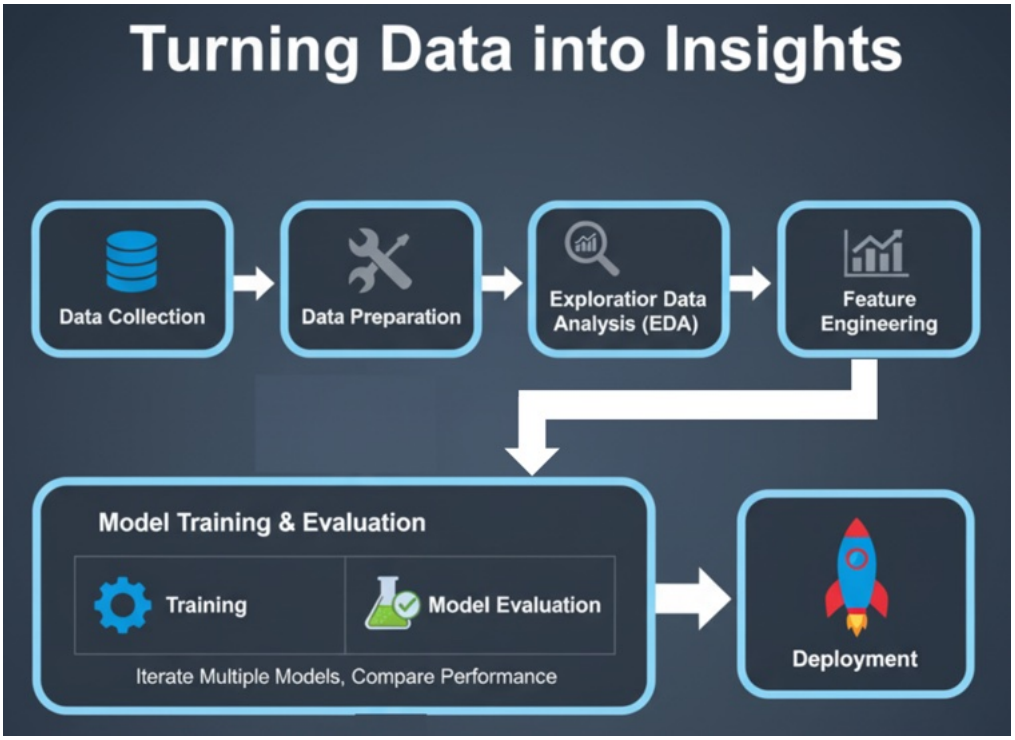
The dataset provided on Kaggle consists of two separate files — one for training and one for testing. The training file contains approximately 79,853 records, each with both feature columns and the target variable (renewal), while the test file contains around 34,224 records, but does not include the target column.

Since the test data does not provide the actual renewal outcomes, it cannot be directly used for model evaluation. Therefore, during this project, the training dataset was further divided into training and validation subsets to evaluate model performance. A typical 80:20 train–test split was applied to the Kaggle training data, ensuring that the model was trained on one portion and tested on another unseen portion to measure generalization accuracy.

Once the model achieved satisfactory performance, it was used to predict the renewal class (0 or 1) for the Kaggle test dataset, which does not include the target variable. These predictions can be used for submission to Kaggle or for business inference on unseen customer records.



10. Project Lifecycle and Workflow


Turning Data into Insights
Data Collection
Data Preparation
Exploratior Data Analysis (EDA)
Feature Engineering
Model Training & Evaluation
Training
Model Evaluation
Iterate Multiple Models, Compare Performance
Deployment

The overall process followed in this project aligns with the standard machine learning workflow used in data-driven problem solving. The end-to-end lifecycle—from data collection to deployment—is shown in the figure below. Each stage ensures that the data is properly understood, processed, and modeled to generate actionable business insights.

Project Lifecycle Stages:

10.1. Data Collection: Gathering the raw customer and policy-level data from the Kaggle *Insurance Renewal Prediction* repository.

10.2. Data Preparation**:** Cleaning, formatting, and organizing the data to handle missing values, inconsistent records, and categorical encoding.

10.3. Exploratory Data Analysis (EDA): Performing statistical and visual analysis to understand data patterns, identify correlations, and detect outliers or biases.

10.4. Feature Engineering: Deriving meaningful variables (such as payment ratios or risk indicators) that can improve model performance.

10.5. Model Training and Evaluation: Building multiple machine learning models, tuning hyperparameters, and comparing metrics to select the best-performing model.

10.6. Deployment: Using the trained model to predict renewal outcomes for unseen customers and generate actionable insights for business decision-making.

11. Feature Overview

The dataset consists of multiple attributes that describe customer demographics, payment behavior, underwriting scores, policy characteristics, and distribution channels. These features collectively provide the basis for predicting whether a policyholder will renew their insurance policy. To understand the data structure, the variables are grouped into the following categories:

Category	**Features**	**Dtype**
Customer Demographics	age_in_days, Income	Numeric
Payment Behavior	perc_premium_paid_by_cash_credit, Count_3-6_months_late, Count_6-12_months_late, Count_more_than_12_months_late, no_of_premiums_paid	Numeric
Risk & Underwriting Factors	application_underwriting_score	Numeric
Policy Attributes	premium	Numeric
Distribution & Geography	sourcing_channel, residence_area_type	Categorical
Target Variable	renewal	Binary (Binary classification problem)

These attributes cover key dimensions of the customer lifecycle. Payment-related features capture consistency in premium payments, underwriting scores reflect the customer’s risk profile, and demographic and channel-based variables help segment customers for targeted retention strategies. The target variable, *renewal*, is binary in nature, representing whether a customer renewed (1) or did not renew (0) their policy.

11.1. Missing but Important Features

While the available dataset provides valuable behavioral and demographic insights, it lacks several real-world variables that significantly influence a customer's decision to renew an insurance policy. The absence of these factors limits the model's ability to capture the full business context behind renewal or churn behavior.

The following are some of the key missing but important features:

- **Claim History:** Frequent or high-value claims may reduce renewal likelihood due to the insurer's risk-based pricing or customer dissatisfaction with claim settlements.

- **Type of Insurance:** Renewal behavior often varies across product lines such as health, motor, and life insurance. Product-specific patterns and claim cycles can heavily influence retention rates.

- **Policy Add-ons and Riders:** Additional policy benefits, such as riders or coverage extensions, typically improve customer stickiness and increase the probability of renewal.

- **Customer Satisfaction and Service Experience:** Poor claim support, unresolved complaints, or negative service experiences are major contributors to non-renewal.

- **Competitor Switching and Market Alternatives:** Availability of cheaper or more comprehensive plans from competing insurers often leads to policy cancellations or non-renewals.
- **Change in Premium or Policy Terms:** An increase in premium amount or modification of coverage terms during renewal can affect customer decisions.

- **Customer Tenure and Loyalty Benefits:** Longer association with the insurer and access to loyalty rewards or discounts usually strengthen renewal probability.

- **Interaction and Engagement History:** Frequency and quality of communication with the insurer—such as reminders, personalized offers, or service calls—can significantly influence renewal decisions.

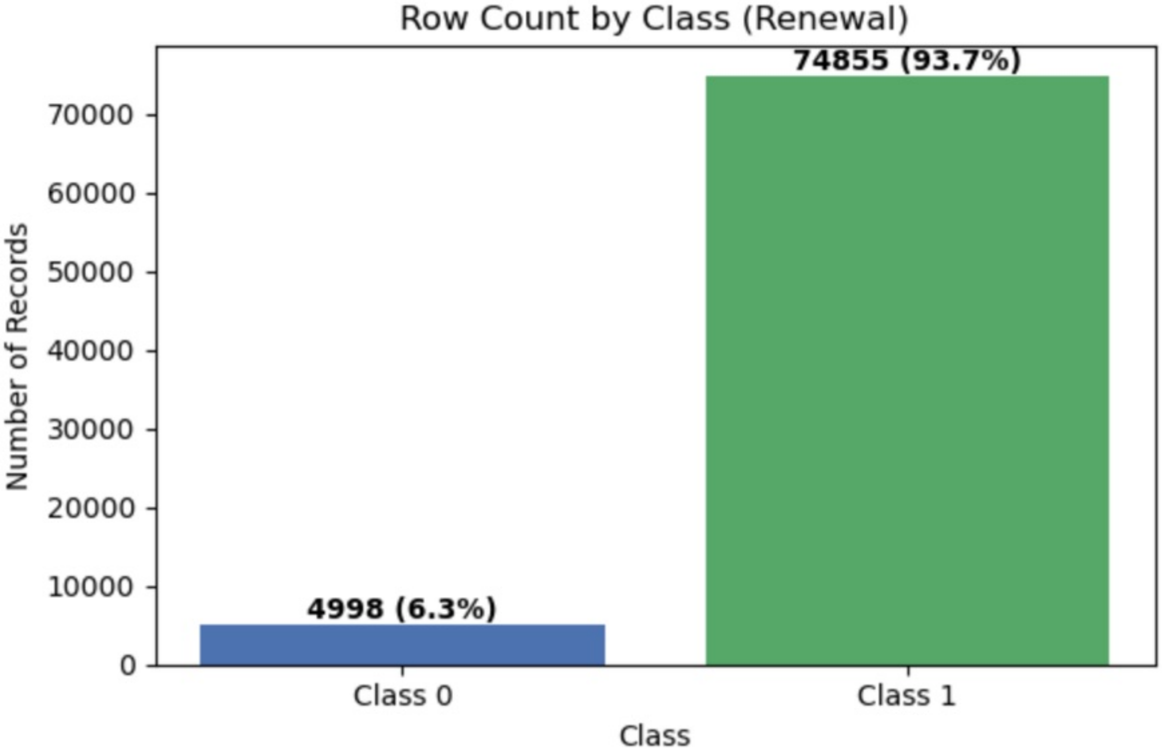
Including these factors in future datasets would substantially improve the model's predictive power. A more comprehensive dataset combining customer demographics, policy structure, claims experience, and market competition would provide a realistic view of the drivers influencing policy renewals.

11.2. Data Limitations and Business Implications

During the feature review, two critical constraints were identified that directly affect the predictive and business potential of the dataset.

- **Limited Feature Scope:** The dataset contains only basic demographic and payment-related attributes such as *age_in_days*, *income*, *premium*, and *late payment counts*. However, several important churn drivers—such as **policy type**, **claim history**, **claim delays**, **complaint records**, and **loyalty or cross-sell indicators**—are missing. These variables often play a major role in influencing renewal decisions. Their absence limits the model's ability to capture real-world behavioral and operational factors that drive customer churn.

- **Severe Class Imbalance:** The dataset is highly imbalanced, with approximately **93% of customers renewing** and only about **7% not renewing** their policies. Such skewness can cause models to favor the majority class, achieving deceptively high accuracy while failing to identify potential churners. This imbalance weakens the model's ability to address the core business objective—early detection of customers at risk of non-renewal.



Together, these limitations highlight the need for more comprehensive and balanced data. While the available dataset is sufficient for building a baseline predictive model, integrating richer business features and addressing class imbalance would significantly enhance both prediction accuracy and business applicability.

12. Initial Data Inspection and Quality Checks

Check Performed	**Description / Purpose**	**Observation / Result**
Dataset Size and Structure	Verified the total number of records and feature columns.	The dataset contains approximately **79,853 records** and multiple numerical and categorical variables.
Target Variable Validation	Confirmed that the target variable *renewal* is binary and correctly encoded.	Validated: *renewal = 0 (Not Renewed)*, *1 (Renewed)*. Distribution is highly imbalanced (~6 % non-renewals, ~94 % renewals).
Feature Type Verification	Checked that continuous and categorical variables are correctly typed.	Numeric: *age_in_days, income, premium*, etc. Categorical: *sourcing_channel, residence_area_type*.
Missing Values Check	Identified missing data across all features.	Missing values found in *application_underwriting_score* and late-payment features (*Count_3–6_months_late*, *Count_6–12_months_late*, *Count_more_than_12_months_late*). These were later imputed during preprocessing.
Duplicate Records Check	Ensured no duplicate customer entries exist in the dataset.	No duplicate records detected after validation.
Invalid or Inconsistent Values Check	Checked for spelling, case, or formatting inconsistencies in categorical variables.	No inconsistencies were found; all categorical values were properly standardized.

13. Data Preprocessing and Transformation

Following the initial data inspection and quality validation, several preprocessing and transformation steps were carried out to prepare the dataset for further analysis and modeling. The objective of this stage was to refine the data so that all features were relevant, interpretable, and suitable for input into machine learning algorithms.

13.1. Dropping Irrelevant Column

The dataset originally contained an *id* column, which served only as a unique identifier for each policyholder. Since this field carried no predictive information related to policy renewal, it was removed from the dataset. Retaining such an identifier could lead to false correlations or spurious patterns being learned by the model, thereby affecting the integrity of predictions.

13.2. Converting Age from Days to Years

The feature *age_in_days* was transformed into *age_in_years* to enhance interpretability and business relevance. Representing age in years is more meaningful when segmenting customers into intuitive groups, such as young, middle-aged, or senior policyholders. This transformation also helps in creating age buckets (e.g., 20–30, 30–40) if required for further analysis or visualization.

13.3. One-Hot Encoding of Categorical Features

The categorical variables *sourcing_channel* and *residence_area_type* were converted into numerical format using one-hot encoding. This technique transforms categorical labels into binary indicators (0/1) for each category, ensuring that the model can process them while preserving their distinct meanings. One-hot encoding also avoids introducing any unintended ordinal relationships between categories—for example, "Channel A" and "Channel B" are treated as separate, non-ranked variables.

These preprocessing steps ensured that the dataset was clean, interpretable, and compatible with machine learning models such as Logistic Regression and XGBoost. The processed data was then used for exploratory data analysis and subsequent model development.

14. Missing Value Analysis

During the data preparation phase, two key sets of features were identified with missing values — the late payment indicators and the *application_underwriting_score*. Each required a distinct imputation approach based on the observed missingness pattern and the feature's role in prediction.

14.1. Late Payment Features

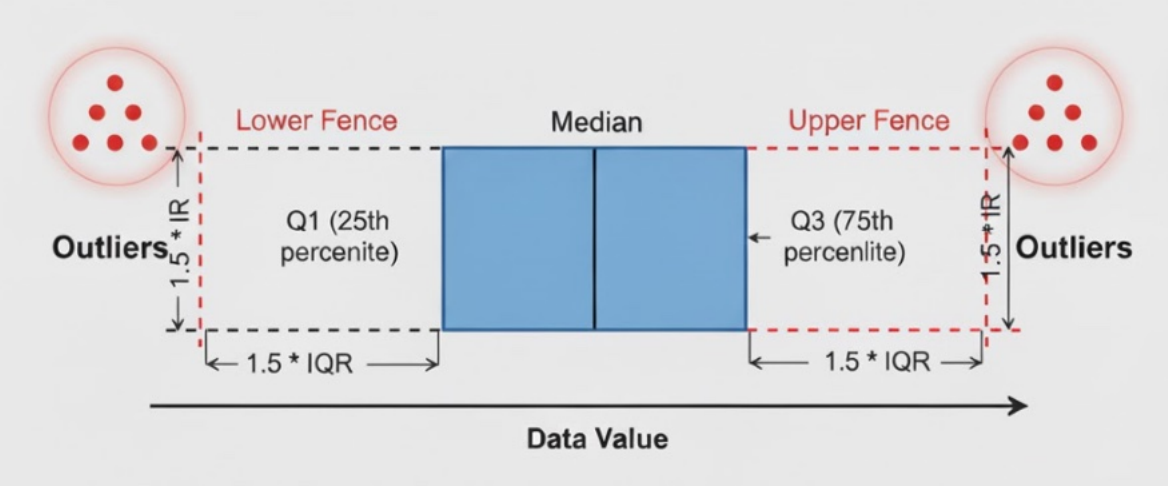
The features *Count_3–6_months_late*, *Count_6–12_months_late*, and *Count_more_than_12_months_late* showed missing values that consistently occurred together across the same records. This pattern indicated **systematic missingness**, suggesting that the absence of data corresponds to **no late payments** for those customers. Instead of treating the missingness as random or dropping these records, the missing values were imputed with **0**, representing "no delayed payments." Given that late payment behavior is a strong predictor of renewal likelihood, this imputation method helped retain variance while ensuring interpretability and preventing distortion of predictive relationships.

14.2. Application Underwriting Score

The *application_underwriting_score* variable represents a continuous risk score reflecting the insurer's evaluation of customer risk, typically influenced by features such as income, premium, age, and the number of premiums paid. Using simple mean or median imputation could have distorted the natural variability and weakened correlations with related features. To preserve these relationships, **K-Nearest Neighbors (KNN) imputation** was used. This approach estimates missing values based on similarity with other records, maintaining both distributional integrity and inter-feature relationships.

Through these targeted imputations, the dataset was made complete while retaining key behavioral and statistical properties essential for model accuracy.

15. Outlier Analysis



Method Used

Interquartile Range (IQR) method applied to all numeric features.

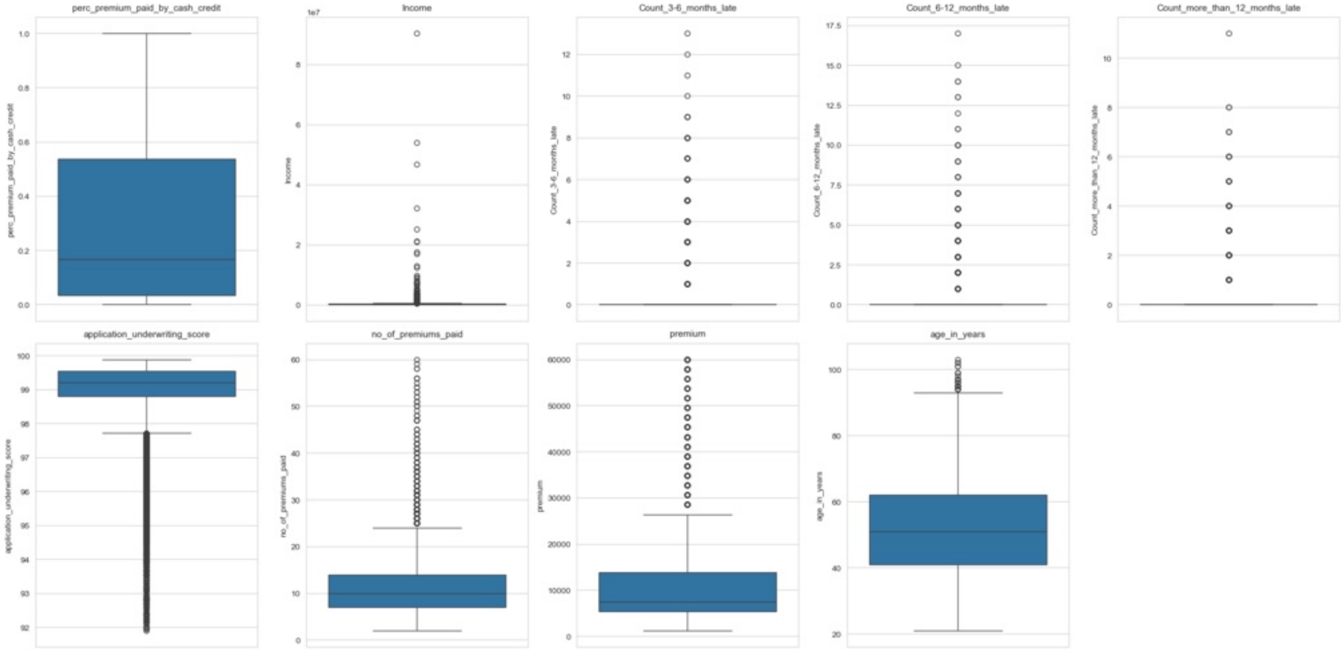
Rationale

Detect values that lie beyond the expected distribution range.

Approach

Calculated Q1, Q3, and IQR for each feature

Flagged observations outside $\boldsymbol{Q1–1.5 \times IQR}$ **and** $\boldsymbol{Q3+1.5 \times IQ}$



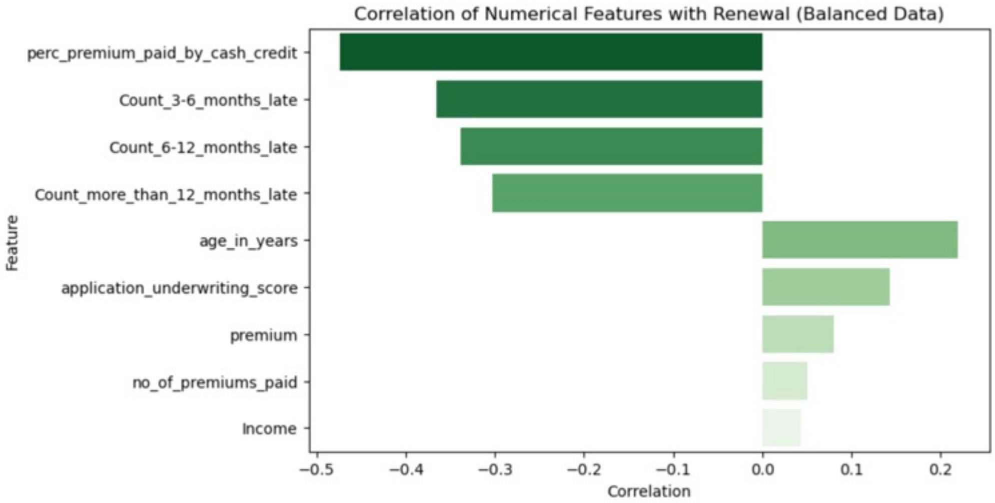
15.1. Late Payment Features

IQR Bounds: 0 to 0

The late payment variables — *Count_3–6_months_late*, *Count_6–12_months_late*, and *Count_more_than_12_months_late* — exhibited heavily zero-inflated distributions, with the majority of customers having no delayed payments and a smaller subset showing positive late payment counts.

Applying the Interquartile Range (IQR) method produced lower and upper bounds equal to zero, which technically flagged every positive value as an outlier. However, these positive counts are **not errors** but valid behavioral indicators of payment delinquency. Further correlation analysis revealed that these variables have a **strong negative relationship with policy renewal**, confirming them as critical predictors for churn identification.

A key observation from this analysis was that if all outlier values had been imputed with zero**, these columns would effectively collapse to all-zero distributions**, removing their predictive power. Therefore, the decision was made to **preserve the raw distribution**, retaining positive counts to maintain variance and predictive value. Blanket zero-imputation was deliberately avoided to prevent information loss.



15.2. Application Underwriting Score

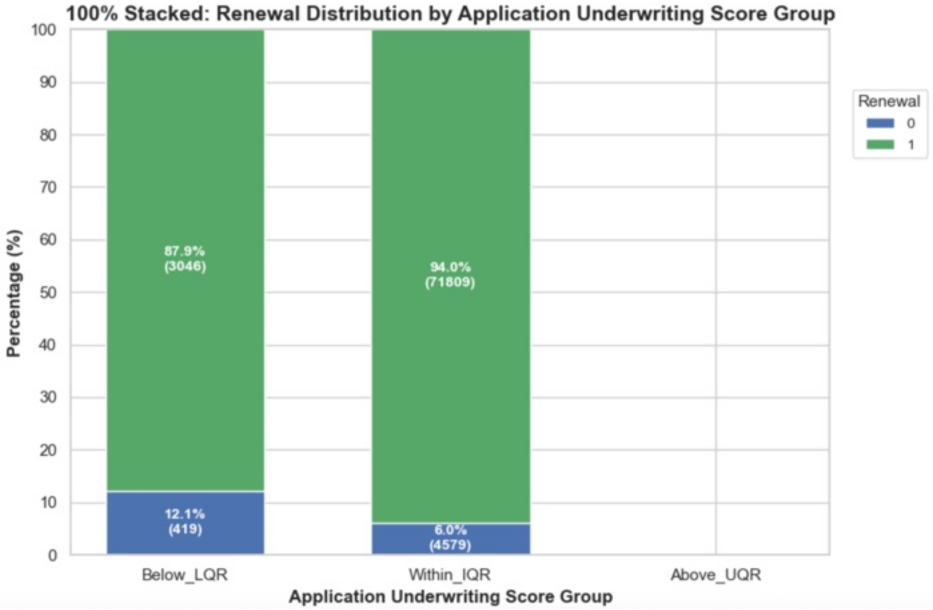
IQR Bounds: 97.72 to 100.64.

Outliers Detected: A total of **3,381 rows** (corresponding to **454 distinct values**) were classified as outliers, including lower underwriting scores such as *91.9*, *91.96*, and *92.03*.

Renewal Rate Comparison: The renewal rate among outlier records was **87.84%**, compared to **94.00%** for non-outliers. This indicates that customers with lower underwriting scores were approximately **6.16% more likely not to renew** their policies.

Interpretation: Lower underwriting scores reflect higher-risk customers, which aligns with their increased likelihood of non-renewal. These outliers, though statistically distant, carry meaningful business insights into customer risk and retention behavior.

Decision: The outliers were **retained** in the dataset, as they provide a strong predictive signal for non-renewal and represent valid business scenarios rather than erroneous data points.



15.3. Age (in Years)

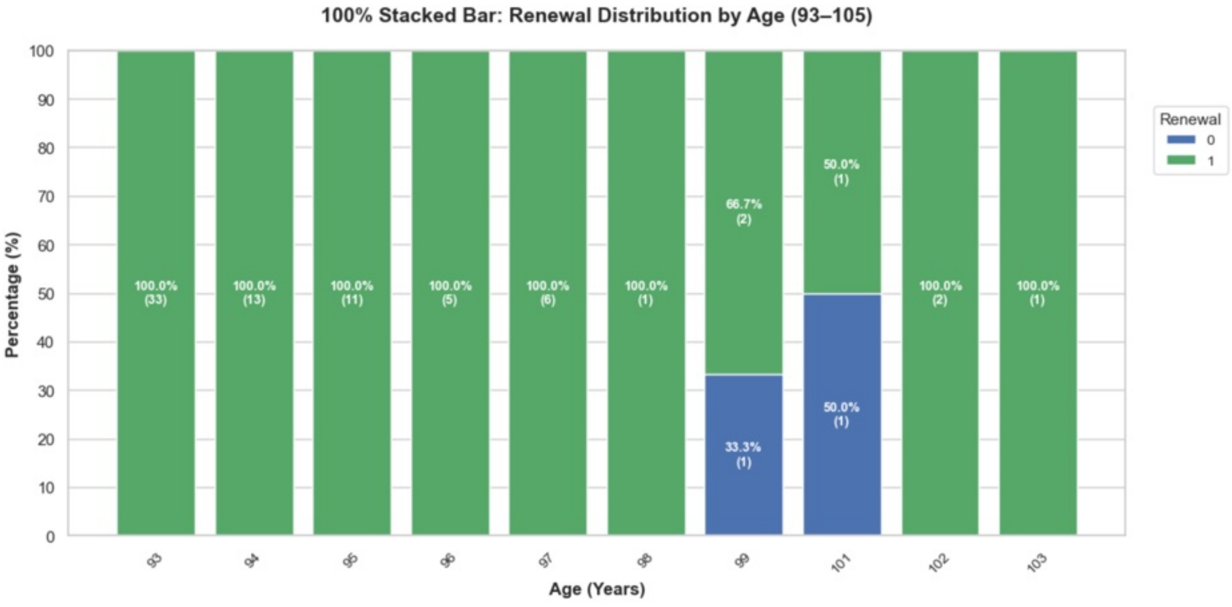
IQR Upper Bound: 93.5 years

Outliers Detected: 44 records (ages 94–103)

Renewal Rates: Outliers: **95.45%**, Non-outliers: **93.74%**

Interpretation: These high-age records represent realistic elderly customers rather than data errors. The renewal distribution by age (shown below) indicates that **older customers have a slightly higher likelihood of renewing** compared to younger ones. Since the renewal trend does not decline with age, these outliers were retained.

Note: In real-world insurance scenarios, some policy types may have upper age limits for new enrollment or renewal. Since the dataset lacks *policy type* information, such interactions could not be analyzed further.

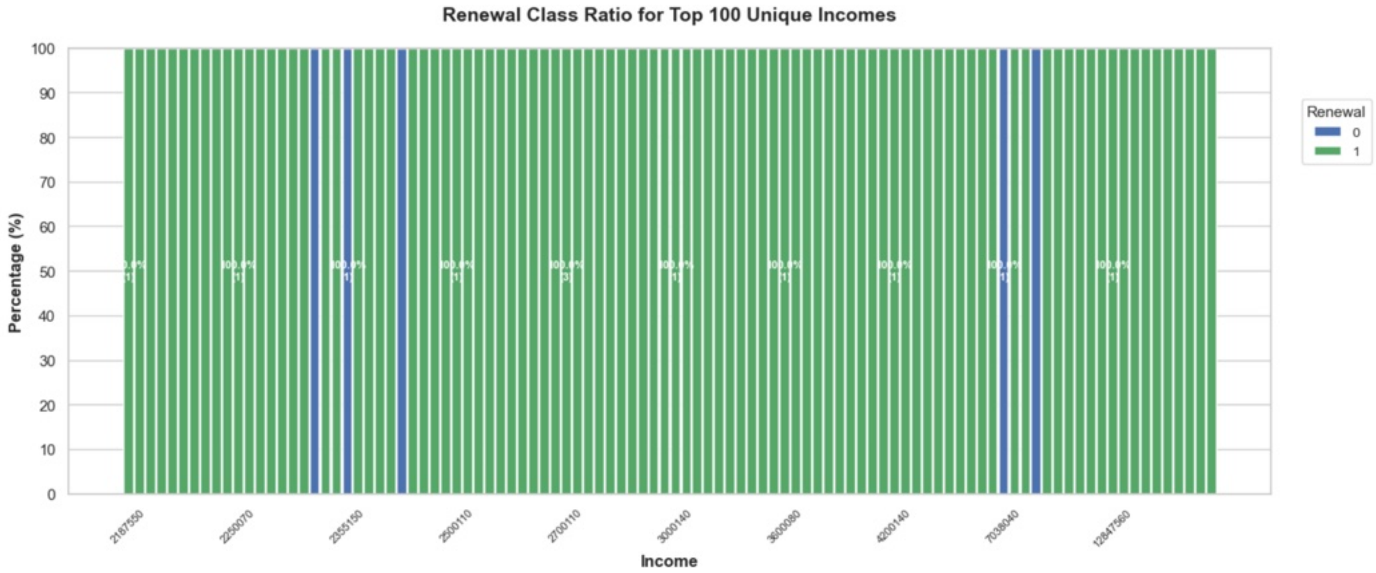


15.4. Income

IQR Bounds: –108,110 to 468,210
Outliers Detected: 3,428 rows (2,233 unique values), extending up to 470,040
Renewal Rates: Outliers: **95.22%**, Non-outliers: **93.67%**

Interpretation: The outliers represent very high-income customers. Renewal rates indicate that **higher-income policyholders are more likely to renew** compared to lower-income groups, suggesting greater financial stability and customer loyalty.

Decision: Outliers were **retained**, as they correspond to valid, business-relevant segments that positively influence renewal predictions.



Observation from Plot: As shown above, the **renewal ratio for the top 100 unique income levels** is consistently close to 100%, with minimal churn. This confirms that customers with higher incomes tend to maintain long-term relationships with the insurer, reinforcing income as a valuable predictive feature.

15.5. Premium

IQR Upper Bound: 26,400
Outliers Detected: 4,523 rows (16 distinct values)
Pattern Observed: Structured premium brackets ranging from **28,500 to 60,000**, in uniform increments.
Renewal Rates: Outliers: **95.49%**, Non-outliers: **93.64%**

Interpretation: These outliers correspond to valid high-value insurance policies rather than data anomalies. Renewal rates remain strong among these records, suggesting that customers with higher premiums are more consistent in policy renewal.

Decision: Outliers were **retained**, as they represent legitimate premium segments with no adverse effect on renewal behavior.

15.6. Number of Premiums Paid

IQR Upper Bound: 24.5 premiums
Outliers Detected: 1,426 rows (34 distinct values), extending up to 60 premiums
Renewal Rates: Outliers: **93.13%**, Non-outliers: **93.75%**

Interpretation: These outliers represent long-tenured and loyal policyholders who have paid premiums over an extended period. Renewal patterns among these customers are consistent with the overall population, indicating stable behavior.

Decision: Outliers were **retained**, as they provide valuable information about customer longevity and do not negatively affect renewal prediction.

16. Logical Consistency Checks

16.1. Late Payments vs. Premiums Paid

A total of **242 records (~0.3%)** showed more late payments than the total number of premiums paid. For example, one customer had only *2 premiums paid* but *6 late payments* recorded. This is logically inconsistent since a policyholder cannot default more times than the number of payments made. Such cases likely indicate **data entry or recording errors**.

16.2. Premiums Paid vs. Age

Some records showed that the number of premiums paid exceeded the customer's **age in years**, which is unrealistic under a typical annual premium cycle assumption. These cases likely reflect **reporting or data entry inconsistencies** and warrant further verification in source systems.

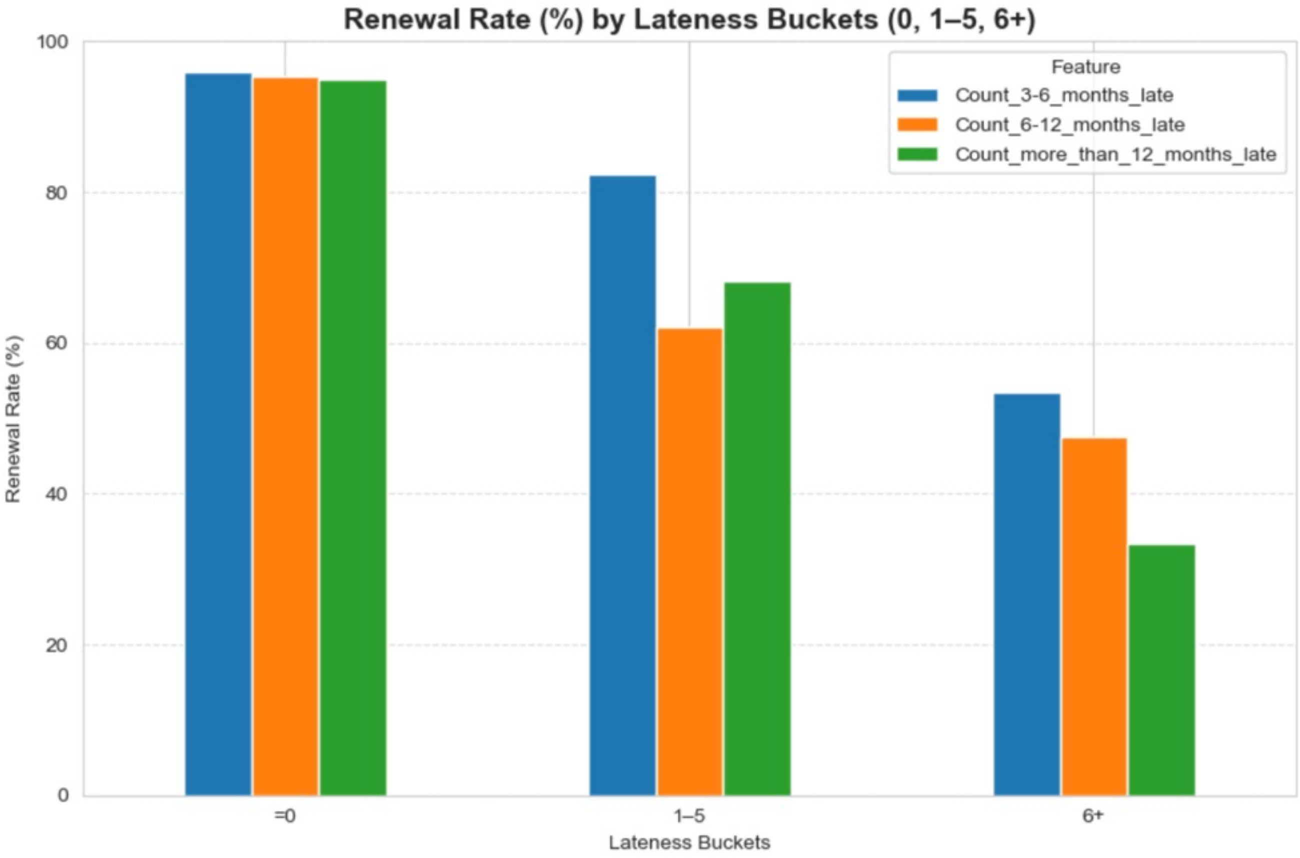
16.3. Extreme Late Payment Behavior

A further consistency check was conducted to examine customers with unusually high late payment counts across any lateness category.

Findings:
A total of **183 customers (~0.23%)** recorded **six or more late payments**, representing extreme delinquency behavior. Among them, **92 customers renewed (Class 1 – 50.3%)** and **91 did not renew (Class 0 – 49.7%)**, resulting in a renewal rate of approximately **50%**, far below the dataset average of around **95%**.

Interpretation:
These cases are inconsistent with normal business behavior — customers with multiple late payments would rarely be expected to renew. The **renewed records (Class 1)** among this group likely represent **noise or data irregularities**, as their behavior contradicts both intuition and the overall data pattern. For churn modeling, **removing these 92 renewed noise records (~0.12%)** improves data clarity and ensures better model alignment with business reality, where high late payment frequency typically indicates higher churn risk.

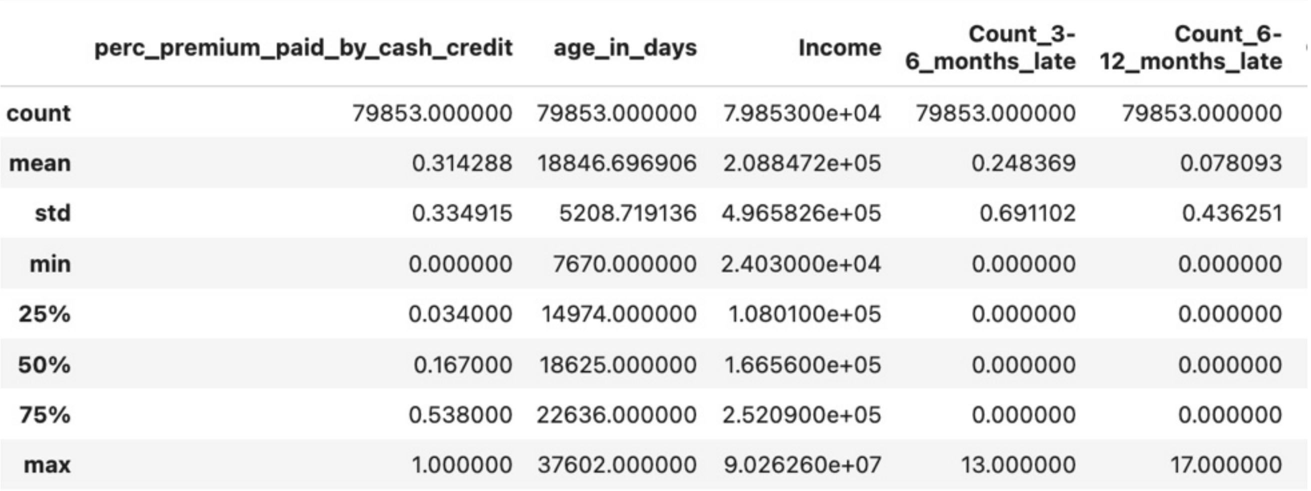
KNN Neighbor Validation:
A **K-Nearest Neighbors (KNN)** analysis (excluding lateness and target variables) confirmed this anomaly. The **average neighbor lateness for these renewed outliers was below 0.5**, showing they are **statistically dissimilar to their peer group**. This strengthens the conclusion that these *renewed (Class 1)* observations behave as **noise** rather than meaningful data points.



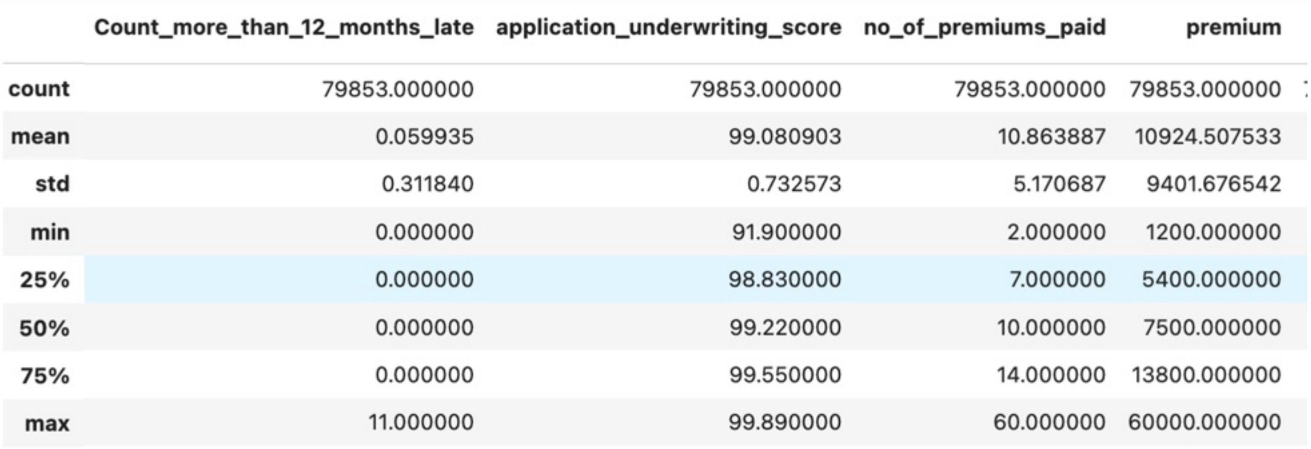
17. Exploratory Data Analysis (EDA)

After completing data cleaning, outlier treatment, and logical consistency verification, the dataset was confirmed to be reliable, internally consistent, and business-sound. With the data quality ensured, the next step was to perform **Exploratory Data Analysis (EDA)** to uncover key trends, relationships, and behavioral patterns influencing policy renewal decisions. This stage aimed to translate validated data into actionable insights for model design and business interpretation.

17.1. Summary Statistics



	perc_premium_paid_by_cash_credit	age_in_days	Income	Count_3-6_months_late	Count_6-12_months_late
count	79853.000000	79853.000000	7.985300e+04	79853.000000	79853.000000
mean	0.314288	18846.696906	2.088472e+05	0.248369	0.078093
std	0.334915	5208.719136	4.965826e+05	0.691102	0.436251
min	0.000000	7670.000000	2.403000e+04	0.000000	0.000000
25%	0.034000	14974.000000	1.080100e+05	0.000000	0.000000
50%	0.167000	18625.000000	1.665600e+05	0.000000	0.000000
75%	0.538000	22636.000000	2.520900e+05	0.000000	0.000000
max	1.000000	37602.000000	9.026260e+07	13.000000	17.000000



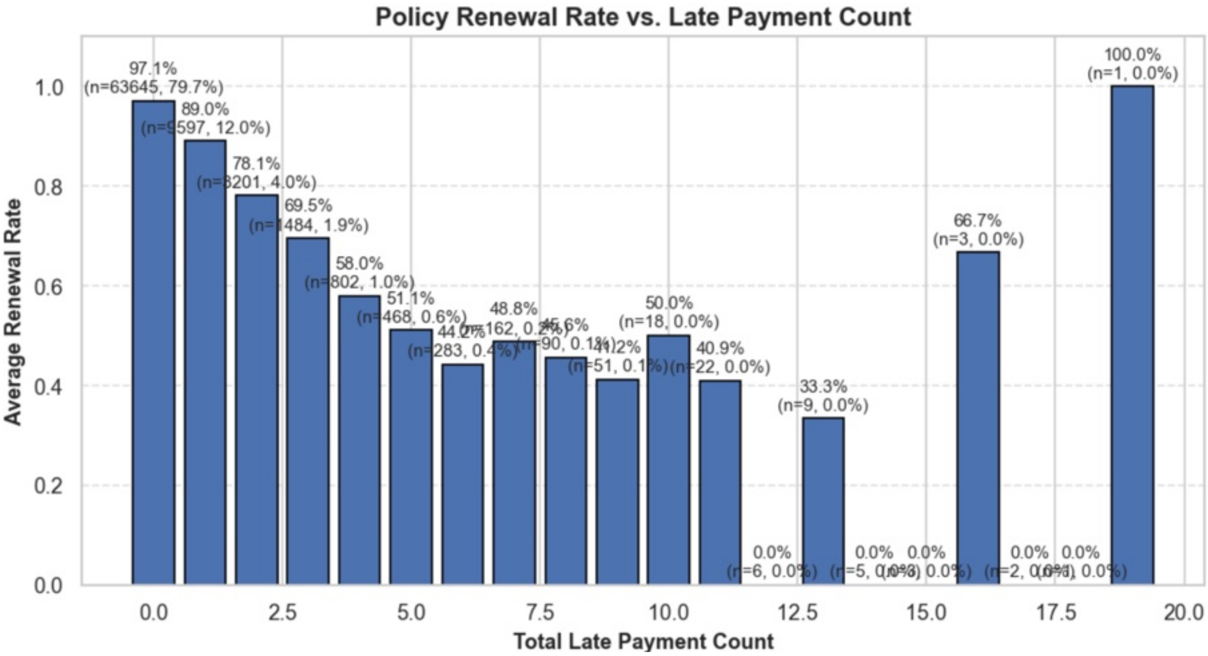
	Count_more_than_12_months_late	application_underwriting_score	no_of_premiums_paid	premium
count	79853.000000	79853.000000	79853.000000	79853.000000
mean	0.059935	99.080903	10.863887	10924.507533
std	0.311840	0.732573	5.170687	9401.676542
min	0.000000	91.900000	2.000000	1200.000000
25%	0.000000	98.830000	7.000000	5400.000000
50%	0.000000	99.220000	10.000000	7500.000000
75%	0.000000	99.550000	14.000000	13800.000000
max	11.000000	99.890000	60.000000	60000.000000

This dataset contains ~79.9k records and a mix of numeric and categorical fields relevant to renewal prediction. Key descriptive statistics were computed to verify scale, skew and extrema: premium and income are right-skewed with large upper tails (median premium ≈ 7,500; mean ≈ 10,925; median income ≈ 166,560; mean ≈ 208,847), payment-lateness variables are heavily zero-inflated (most customers have zero late counts but a small tail exists up to 13–17 events), the application underwriting score is tightly clustered near 99 (IQR ≈ 98.83–99.55), and the typical customer has paid ≈10 premiums to date (mean ≈ 10.86).

17.2. Relationship Between Late Payments and Renewal

A clear negative relationship was observed between **late payment frequency** and **policy renewal**.
Customers with higher counts of late payments across any time window (*3–6 months*, *6–12 months*, or *>12 months*) showed a significantly **lower renewal rate** compared to those with no or few delays.

This confirms that **payment discipline is a strong behavioral indicator of customer retention**.
In other words, **higher late payment counts are strongly associated with non-renewal**, aligning with business intuition — customers who habitually delay or miss payments are more likely to discontinue their policies.

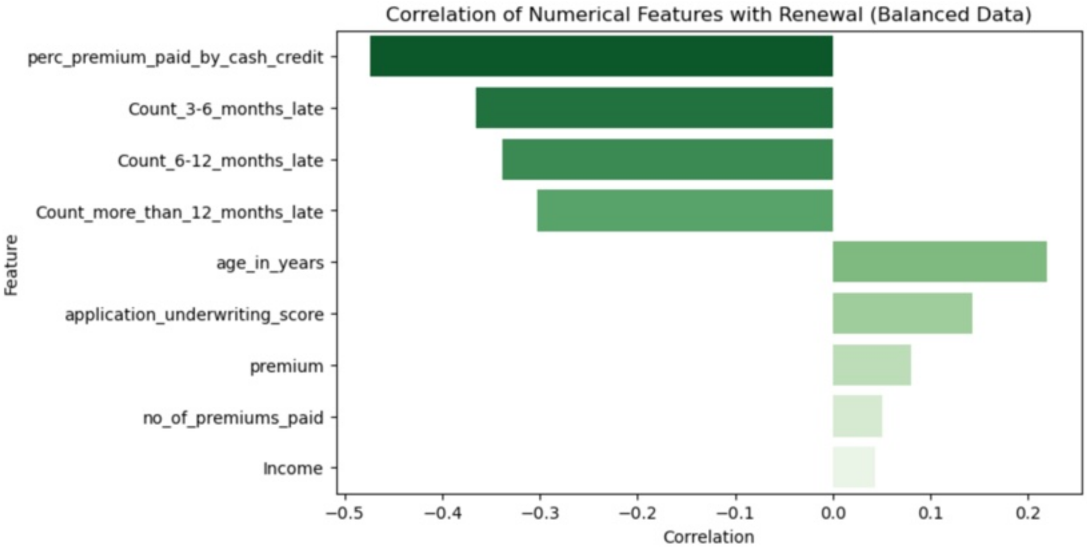


Interpretation: Policy renewal rate shows a clear and consistent decline as late payment counts increase—from over 97% among customers with no late payments to around 50% beyond six late payments. However, the last few bars (16–20 late payments) correspond to fewer than five records each and display erratic renewal rates (e.g., 0% or 100%). These are statistically insignificant and likely represent noise or data entry anomalies rather than meaningful customer behavior. Such extreme cases were not removed but excluded from modeling insights to prevent bias.

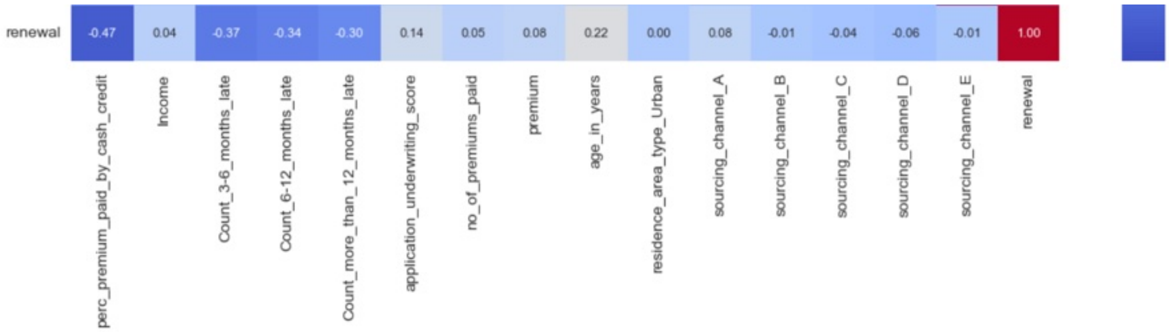
17.3. Correlation of Numerical Features with Renewal

The figure below shows the correlation of numerical features with the target variable (*renewal*).
As expected, **late payment features** and **percentage of premium paid by cash/credit** exhibit strong **negative correlations** with renewal, indicating that customers with more late payments or inconsistent payment modes are less likely to renew.

In contrast, **age**, **application underwriting score**, and **premium amount** show **positive correlations** with renewal, suggesting that older, higher-scoring, and higher-value customers tend to maintain their policies.



17.4. Correlation of Features with Target Variable



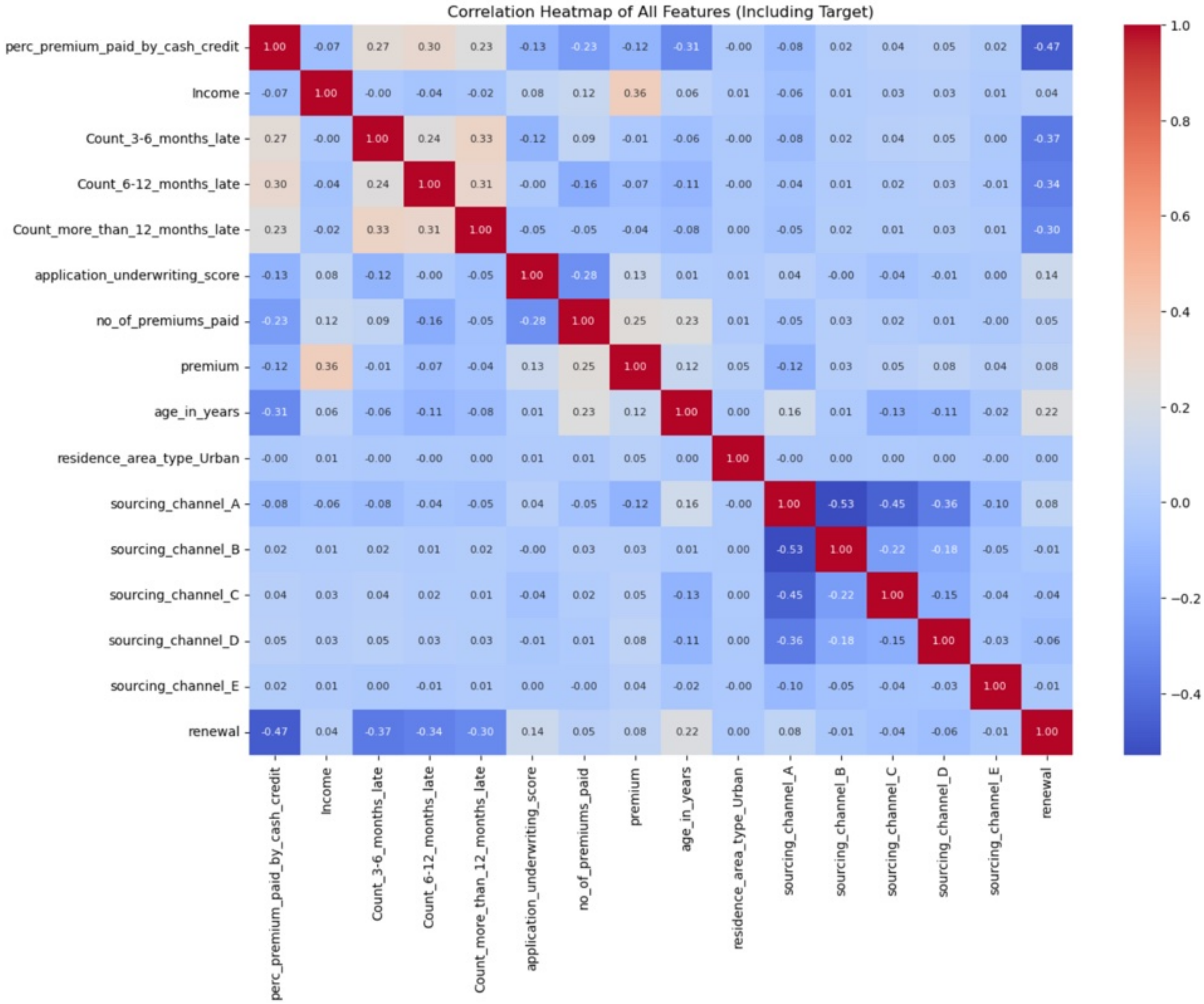
The heatmap below highlights that late-payment features
Count_3–6_months_late,
Count_6–12_months_late, and
Count_more_than_12_months_lateand
perc_premium_paid_by_cash_credit show the **strongest negative correlations** with renewal (ρ ≈ –0.30 to –0.47). This confirms that frequent or extended delays in premium payment are clear indicators of non-renewal risk.

In contrast, age_in_years and application_underwriting_score display weak positive correlations (ρ ≈ 0.14–0.22), suggesting that older, higher-scoring customers are slightly more likely to renew. Features such as income, sourcing_channel, and residence_area_type have near-zero correlations, implying minimal direct influence when considered independently.

Interpretation:

Late-payment behavior dominates as the primary signal for churn, while demographic and distribution-related variables contribute only marginally. These insights guided subsequent feature selection and modeling focus.

17.5. Correlation Heatmap of All Features



This subsection analyzes the relationships among all independent variables to assess possible redundancy or multicollinearity. Understanding how features relate to each other helps ensure a robust and interpretable model.

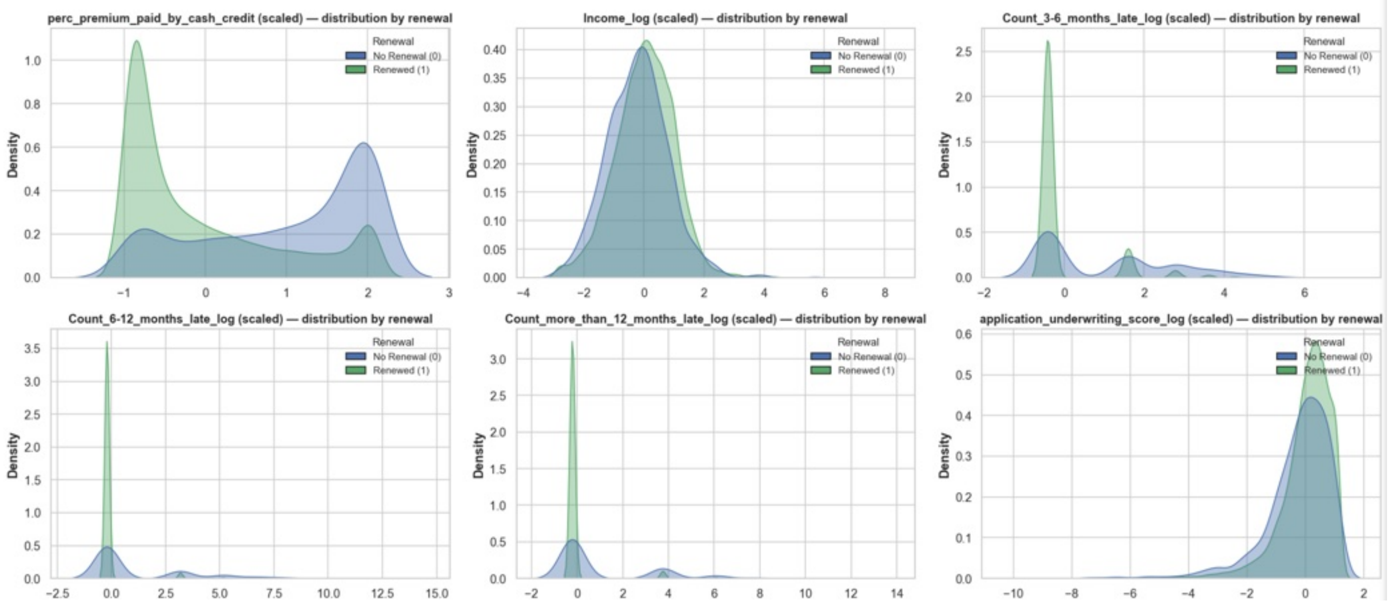
The correlation matrix indicates that the **three late payment variables** (Count_3–6_months_late, Count_6–12_months_late, and Count_more_than_12_months_late) show **low to moderate correlation** with one another (ρ ≈ 0.3–0.5). This suggests that while these variables measure related customer behaviors, they each capture somewhat distinct aspects of payment delay patterns.

perc_premium_paid_by_cash_credit also shows **weak positive correlation** (ρ ≈ 0.3–0.4) with the late payment features, reflecting that customers who rely more on cash or credit payments may be slightly more prone to late payments.

Other features such as income, premium, application_underwriting_score, and age_in_years display **minimal inter-correlation** (ρ < 0.2), indicating that they provide independent signals and do not overlap in representation.

Interpretation:
Overall, no strong multicollinearity was observed among the predictors. The weak to moderate correlations between lateness and payment behavior features suggest complementary information rather than redundancy. Hence, all features were retained for model development, with further dimensionality reduction considered during feature selection if required.

17.6. Density Plots of Key Features by Renewal Status



The kernel density plots below visualize the distribution of major numerical features for renewed and non-renewed policies. These plots were used to assess how well individual features can distinguish between customers who renew and those who do not.

Across most features, the distributions of the two classes are **heavily overlapping**, indicating that renewers and non-renewers share similar demographic and financial characteristics. Simple thresholds or univariate rules would therefore be ineffective in predicting renewal outcomes.

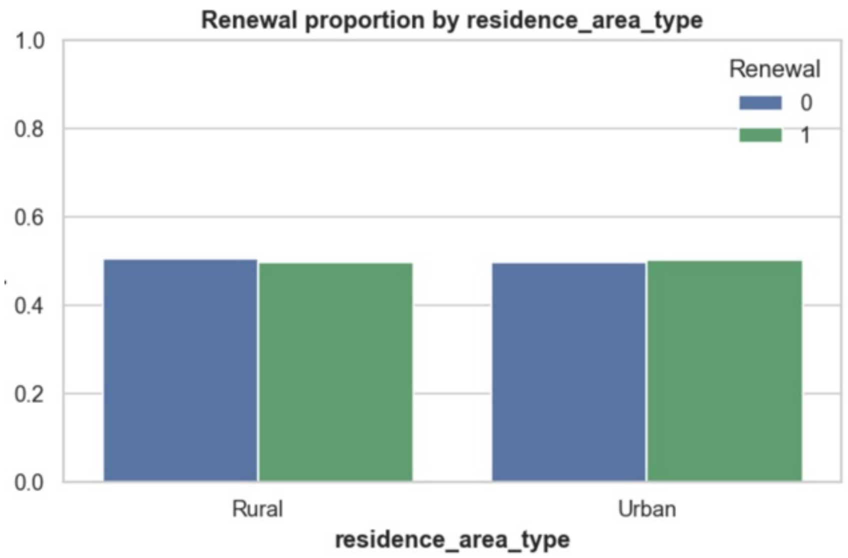
The **only feature showing a visible separation** is the **percentage of premium paid by cash or credit**. Non-renewers display a distinct peak at higher values, suggesting that a greater reliance on cash or credit payments is associated with a higher likelihood of non-renewal. This aligns with business intuition that customers who do not maintain consistent or automated payment methods are more prone to churn.

Other variables, including **income**, **underwriting score**, and **late payment counts**, exhibit almost identical distributions between classes, reinforcing that these factors do not independently drive renewal behavior. Their impact, if any, likely emerges only when considered in combination with other variables during modeling.

Interpretation:
Overall, the density plots indicate that **payment mode** provides the clearest individual signal for renewal prediction, while most other variables require interaction effects or non-linear modeling to uncover meaningful patterns.

17.7. Residence Area Type Analysis

The plot below compares renewal proportions between **Urban** and **Rural** customers. The renewal rates are nearly identical across both categories, indicating that **residence area type has very limited predictive power** for renewal behavior.



Exploratory Findings:

- Renewal proportions differ by less than 1% between Urban and Rural groups.
- This suggests that geographic segmentation at this level does not meaningfully influence customer retention.

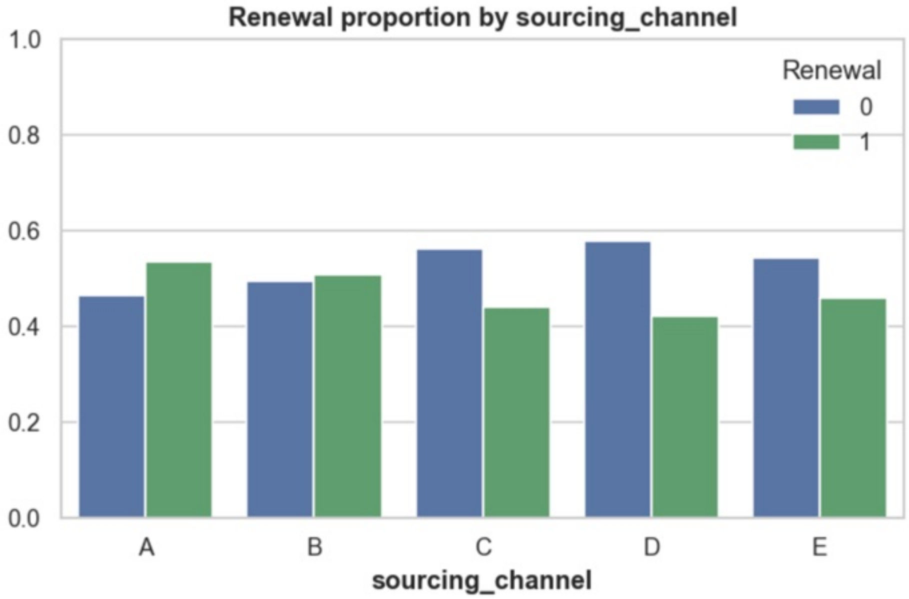
Multivariate Interaction Check: To ensure that subtle interactions were not being missed, residence type was tested in combination with other key features such as **premium**, **age**, **income**, **late payment counts**, **underwriting score**, and **sourcing channel**.

- Across all combinations, the difference in renewal rates remained negligible (within ±3%).
- This confirms that residence type does not significantly interact with other predictors to explain renewal outcomes.

Conclusion:
For **tree-based models** (e.g., Random Forest, XGBoost), this feature can be retained since such models automatically reduce the influence of weak predictors.
However, for **statistical or linear models**, it can be safely excluded to minimize noise and improve model stability without loss of information.

17.8. Sourcing Channel Analysis

The chart below illustrates renewal proportions across different **sourcing channels (A–E)** used to acquire customers. While renewal patterns vary slightly between channels, the overall differences are modest, suggesting only **weak to moderate predictive power** for this feature.



Exploratory Findings:

- Channels **C, D, and E** show **slightly lower renewal proportions**, indicating that policies sourced through these channels may experience marginally higher churn.
- Channels **A and B** perform marginally better, but the difference in renewal rates across all five channels is **within ±10%**, indicating limited standalone impact.

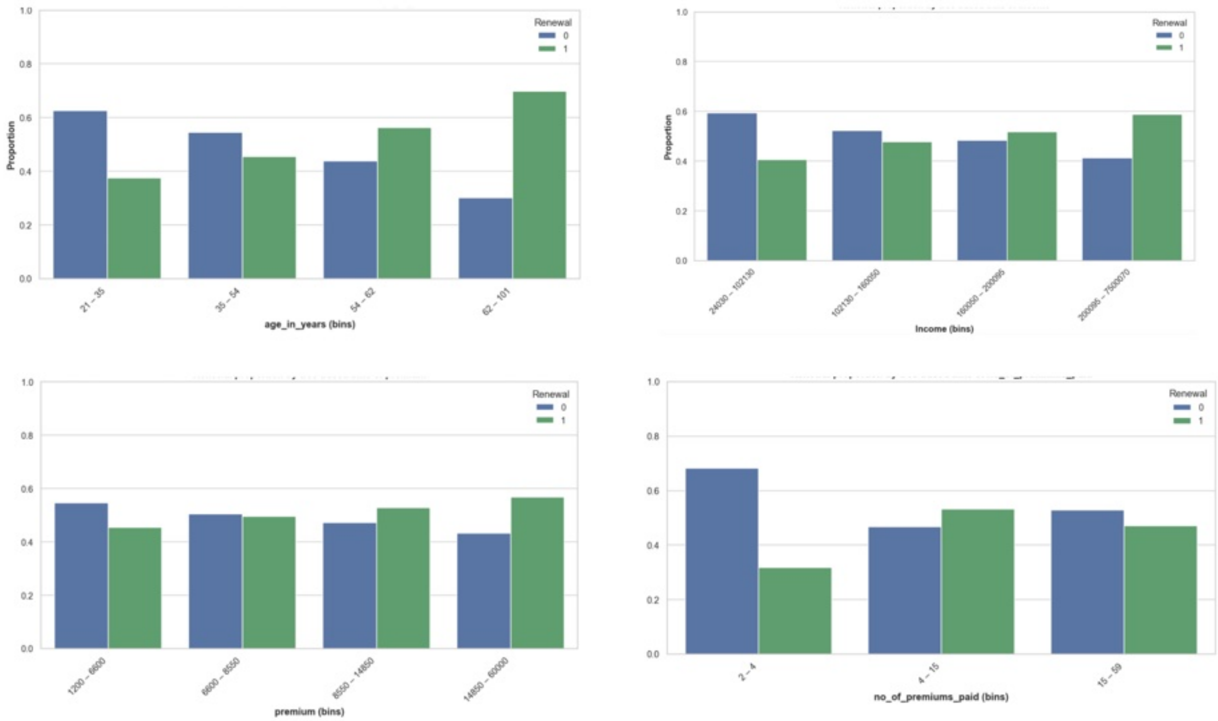
Interpretation:
The sourcing channel may act as a **proxy for acquisition quality or customer type** — for example, policies sold through direct or digital channels may differ in engagement and retention behavior compared to agent-based channels. However, these patterns are not strong enough to serve as independent predictors and likely gain relevance only through interaction effects (e.g., with premium amount or underwriting score).

Modeling Implications:

- For **tree-based models**, the feature should be retained, as interactions and non-linear effects will be automatically handled.
- For **linear or regularized models**, this feature may need careful encoding (e.g., one-hot or frequency encoding) to avoid overfitting to minor variations.

17.9. Renewal Trends by Customer Segments

The dataset was segmented by **age**, **income**, **premium**, and **premium payment ratio** groups to examine renewal behavior across different customer profiles. Renewal proportions were calculated using balanced data to account for class imbalance.



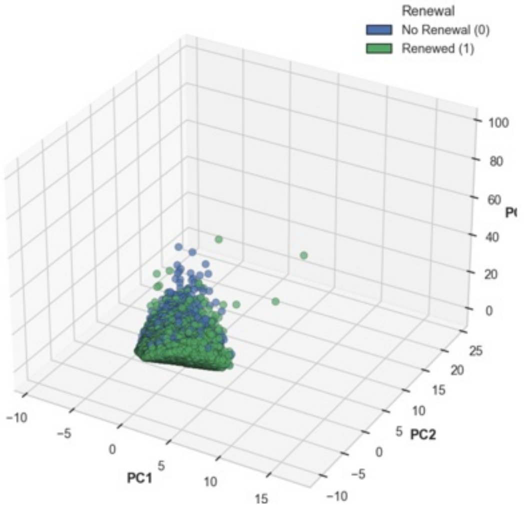
Observations:

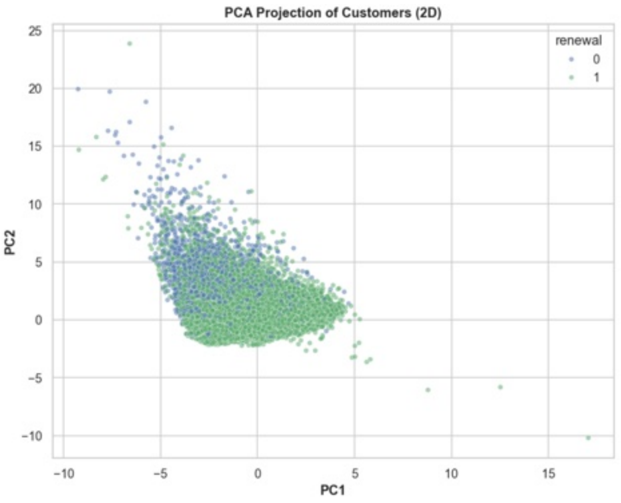
- **Age:** Renewal rate **increases steadily with age**, with customers aged **60+ showing the highest renewal proportion**, suggesting stronger loyalty or lower churn intent among older customers.
- **Income:** Higher-income groups renew slightly more often than low-income groups, but the difference is modest (≈5–7%).
- **Premium Amount & Payment Ratio:** Both features exhibit near-uniform renewal rates (~60%) across groups, indicating **limited standalone predictive power**.

Interpretation:
While **age** shows a mild positive association with renewal likelihood, overall demographic and financial factors display **weak separation between renewing and non-renewing customers**. This emphasizes that **behavioral features**, particularly those linked to **late payments and financial discipline**, play a far greater role in influencing renewal decisions.

17.10. PCA Projection – Customer Renewal

To visually assess class separability, Principal Component Analysis (PCA) was applied to the scaled feature set, and both **2D and 3D projections** were plotted using the top principal components.





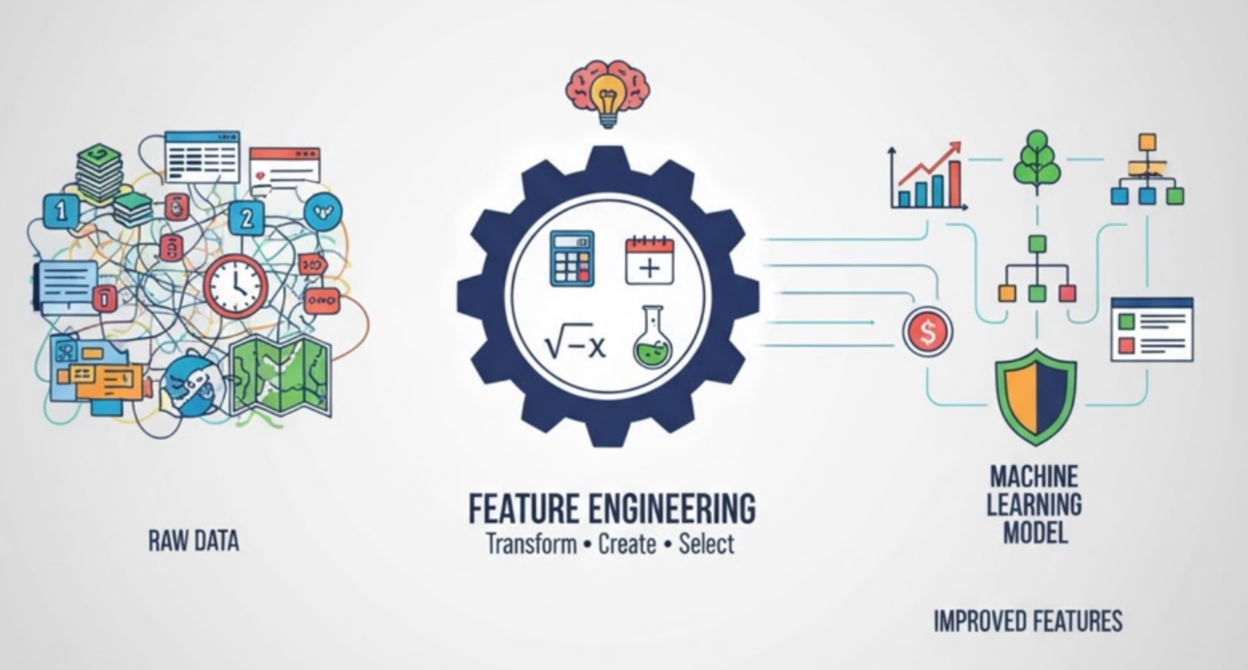
Observations:

- The PCA plots show **significant overlap** between **renewers (1)** and **non-renewers (0)**.
- The **dominant variance directions** in the data do not align with the renewal classification boundary.
- This confirms that **renewal behavior is not linearly separable** and is instead driven by **complex, non-linear feature interactions**.

Implications:

- **Linear models** relying purely on variance (e.g., Logistic Regression without engineered interactions) are unlikely to perform well.
- **Tree-based ensemble models** such as **Random Forest, XGBoost, or Gradient Boosting** are better suited, as they can capture **non-linear dependencies and higher-order interactions** between behavioral and demographic variables.

18. Feature Engineering



Based on exploratory analysis and business understanding, new derived variables were created to capture meaningful financial and behavioral relationships not directly available in the original dataset. These engineered features aimed to strengthen the model's ability to detect subtle patterns linked to policy renewal behavior.

18.1. Premium–Income Related Features

These variables were designed to represent customer affordability and long-term payment commitment relative to their income levels.

- **premium_to_income** — Ratio of one installment (premium) to income.
- **total_premium_paid** — Total premiums paid to date.
- **total_premium_to_income** — Cumulative premium payments as a proportion of income.

Feature	Correlation with Renewal	Insight
premium_to_income	–0.01	No significant relationship; affordability does not influence renewal decisions.
total_premium_paid	+0.02	Very weak positive relationship; effectively neutral in predictive strength.
total_premium_to_income	~0.00	No measurable impact; cumulative payment-to-income ratio is not a renewal driver.

18.2. Late Payment Related Features

Given the strong behavioral influence of payment timeliness on renewal, aggregate and ratio-based lateness features were created to capture delinquency intensity.

- **total_late_payments** — Total late payments across all time windows (3–6m, 6–12m, >12m).
- **late_payment_ratio** — Proportion of total late payments relative to total premiums paid.

Feature	Correlation with Renewal	Insight
total_late_payments	–0.35	Moderate negative correlation; more late payments are associated with lower renewal likelihood.
total_late_payments	–0.34	Moderate negative correlation; frequent late payers are less likely to renew their policies.

18.3. Transformations Applied

To make the dataset model-ready, several feature transformations were applied to improve interpretability, maintain numerical stability, and ensure compatibility with different machine learning algorithms.

Transformations Implemented:

- **Converted age_in_days → age_in_years** to enhance interpretability and enable easier demographic segmentation.
-
- **Applied one-hot encoding** to categorical features (sourcing_channel, residence_area_type) to convert them into numeric binary indicators suitable for most models.
-
- **Normalized continuous variables** (Income, Premium, Age, Application Underwriting Score) to ensure consistency of scale for algorithms sensitive to feature magnitude, such as Logistic Regression and Neural Networks.

Conclusion:
These transformations made the raw data more interpretable, numerically stable, and model-friendly. The resulting dataset supports both **statistical models**, which require scaled and encoded inputs, and **tree-based models**, which can effectively handle non-linear patterns without extensive normalization.

18.4. Dataset Versions and Logical Derivations

During the data preparation and feature engineering stages, six distinct dataset versions were created to enable controlled experimentation and avoid information leakage. Each dataset reflects a deliberate design choice beginning with basic cleaning, extending through feature engineering, and concluding with logical filtering and feature reduction. The intent behind this design is to study how data quality, feature complexity, and derived transformations influence model performance and stability.

18.4.1. Clean Datasets:

Represent logically consistent and noise-free data. Records violating domain or business logic were removed for instance, cases where the total number of late payments exceeded the total number of premiums paid, or where a policy was marked renewed despite exhibiting extreme late-payment behavior. These datasets ensure internal consistency and prevent contradictory observations from influencing model learning.

18.4.2. Reduced Datasets:

Derived from the full feature-engineered datasets by analyzing the correlation of engineered features with the target variable renewal. Out of the five engineered features created, three exhibited **very little correlation** with the target and were removed. This reduction retains only meaningful engineered predictors while eliminating weak derived variables that contribute minimal information.

18.4.3. Reduced_No_Cat Da tasets:

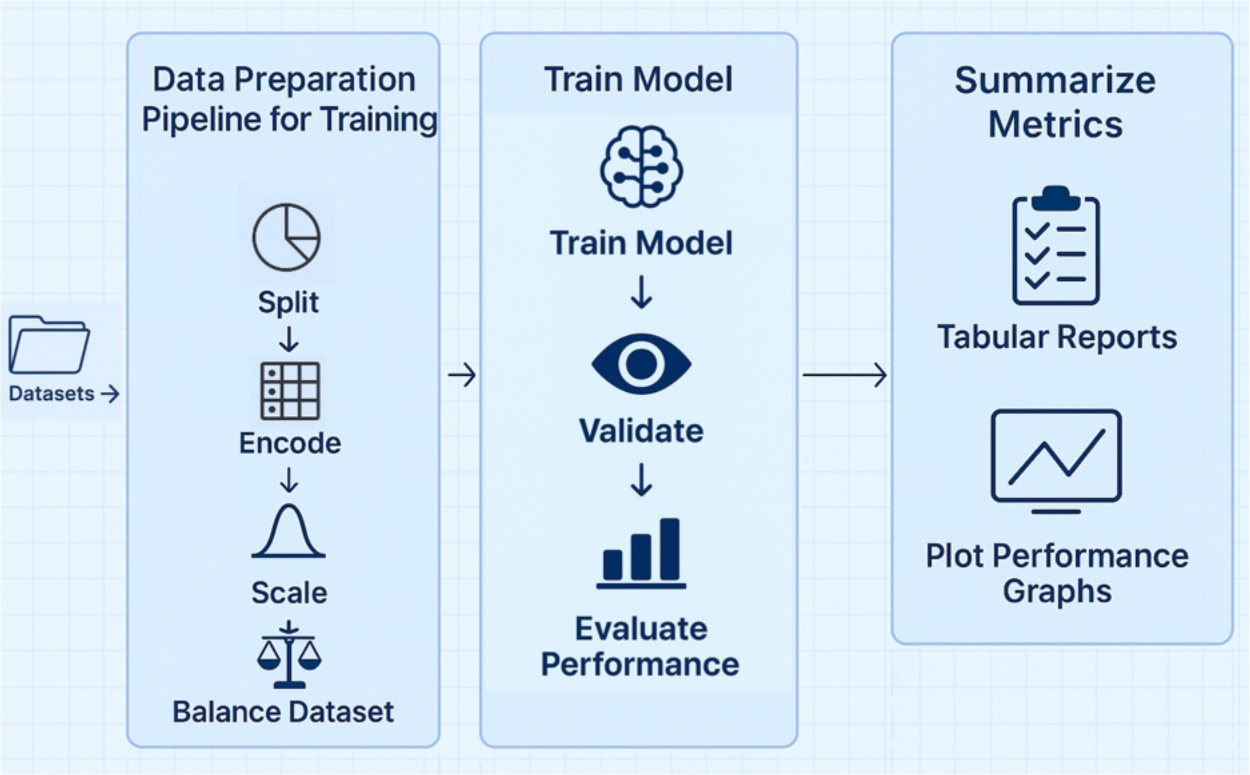
Further refinement of the reduced datasets, achieved by dropping **categorical features that displayed very little correlation** with the target variable. This step was performed to focus the modeling process on variables with stronger predictive relationships, rather than for numerical compatibility. The resulting datasets contain the most relevant numeric predictors and are used to test whether removing weak categorical attributes improves overall performance.

Dataset Name	**Description**
base_full	Original feature-engineered dataset containing all variables. Used as the baseline before any cleaning or feature reduction.
clean_full	Cleaned version of the full dataset with logical inconsistencies and invalid records removed. Used to test the effect of data quality improvement.
base_reduced	Dataset derived from *base_full* by removing engineered features that showed very low correlation with the target variable (renewal).
clean_reduced	Cleaned and feature-reduced dataset combining logical consistency with dimensional efficiency. A more reliable version for modeling.
base_reduced_no_cat	Derived from *base_reduced* by removing categorical variables that had **very little or no correlation with the target**. Used to study the impact of excluding low-signal categorical features.
clean_reduced_no_cat	Derived from *clean_reduced* after removing categorical variables with **negligible correlation to the target**. Combines cleaning, reduction, and removal of non-informative categories for focused model evaluation.

18.5. Flipping Target Labels (Non-Renewal as Positive Class)

In the original dataset, the renewal variable was encoded as 1 for renewed policies and 0 for non-renewals. However, from a business perspective, non-renewal (churn) represents the critical event that the model must detect accurately. To align the model's objective with the business goal and ensure metric consistency, the target labels were flipped—so that the positive class (target = 1) now corresponds to non-renewal. This convention makes recall, precision, and F1-score directly interpretable as measures of how effectively the model identifies potential churners. While most machine learning libraries allow specifying the positive label parameter (e.g., pos_label=0) within evaluation functions, relying on such per-metric configurations can lead to confusion and inconsistency, especially when comparing results across multiple models or during automated evaluation loops. Redefining the label at the dataset level eliminates this ambiguity, simplifies model training and reporting, and ensures that all downstream analyses consistently treat churn as the positive outcome. This approach improves clarity, reduces coding errors, and aligns the modeling workflow with the project's primary business objective—early identification of customers at risk of non-renewal

19. Modeling Workflow Overview



19.1. Objective

The objective of the data preparation pipeline is to convert raw datasets into consistent, model-ready input while preventing data leakage, ensuring fair validation, and handling class imbalance effectively. The datasets contained both numeric and categorical variables with a highly skewed target distribution (renewal rate ≈ 6%). To maintain consistency across experiments, all preprocessing was performed using a modular, reusable pipeline that was uniformly applied across all models — including Logistic Regression, Decision Trees, K-Nearest Neighbors, XGBoost, SVM, Neural Networks, and Ensemble methods.

19.2. Pipeline Stages

Each stage in the pipeline was executed sequentially and designed to preserve strict data isolation between training, validation, and testing. This ensured unbiased model performance and reproducible experimentation.

19.2.1. Dataset Split

Function: prepare_dataset_df()
Each dataset was divided into:

Training set (70%) – used to model traing (Default 80)
Validation set (10%) – used for decision threshold tuning (Default 0)
Test set (20%) – used for final model evaluation (Default 20)

Rationale:
Splitting the dataset before any transformation prevents preprocessing steps (e.g., scaling or encoding) from learning information from the validation or test sets. This separation ensures that only the training set influences fitted transformations, eliminating any risk of data leakage.

19.2.2. Encoding of Categorical Variables

Method: One-Hot Encoding
The encoder was fitted only on the training set categories. The same mapping was then applied to validation and test sets.

Rationale:

- Ensures unseen categories in validation or test are handled safely.
- Prevents leakage of category frequency or distributional information.

Outcome:
Eliminated bias from categorical features while retaining interpretability.

19.2.3. Scaling of Numeric Features

Method: StandardScaler()
Scaling was fitted on the training data using its mean and standard deviation, and the same parameters were applied to validation and test sets.

Rationale:
Features such as income, age, and premium vary widely in magnitude. Scaling ensures that features contribute equally during model optimization and gradient updates. Fitting on the training set only prevents exposure of test data statistics, ensuring realistic generalization performance.

19.2.4. Handling Class Imbalance

Function: `resample_train()`

Supported Methods: Random undersampling, Random oversampling, SMOTE, ADASYN, SMOTE-Tomek, Edited Nearest Neighbors (ENN), and Variational Autoencoder (VAE)-based resampling.

Resampling was performed **exclusively on the training set**, leaving the validation and test sets untouched.

Rationale:
Balancing only the training set allows models to learn effectively from both classes, while maintaining the natural class distribution in validation and test sets for accurate performance evaluation. VAE-based synthetic sampling was applied in cases of nonlinear separability, while SMOTE and ADASYN were used for moderate imbalance.

Avoided Issues:

- Overfitting to synthetic validation/test samples
- Artificial inflation of recall or precision due to resampled data in evaluation splits

19.2.5. Model Training and Threshold Optimization

Functions:

```
choose_threshold_for_recall_from_scores()
choose_threshold_for_precision_from_scores()
choose_threshold_max_f1_from_scores()
```

Models were trained on the training set, and probability outputs were generated for the validation set.
Thresholds were tuned using the validation set to achieve a desired target recall (e.g., 0.8) or precision (e.g., 0.6).
These optimized thresholds were then applied to the test set for final evaluation.

Rationale:
A fixed 0.5 threshold is not appropriate for highly imbalanced data, as it often biases predictions toward the majority class. Dynamic tuning provides fairer evaluation across models.

Benefit:
Improved control over recall–precision trade-offs, allowing consistent comparison across algorithms.

19.2.6. Standardized Metric Computation and Visualization

Functions:

```
compute_metrics_from_probs()
capture_results_clean()
print_results_summary_clean()
_plot_pr_roc_inline()
```

Metrics were computed across all dataset splits (Train / Validation / Test), including:

- Precision
- Recall
- F1-Score

- ROC–AUC Score
- Optimal threshold
- PR and ROC curves

All metrics were stored in a unified DataFrame, ensuring consistency in reporting and visualization.

Rationale:
Uniform evaluation metrics across models enable transparent performance comparison and reliable interpretation of trade-offs between precision and recall.

19.3. Outcome

All datasets were processed through the same standardized pipeline, ensuring reproducibility and fairness across models. The pipeline eliminated data leakage, preserved statistical independence of evaluation sets, and maintained consistent scaling across all experiments. Models achieved the desired target recall or precision without bias from preprocessing artifacts. Variance between train, validation, and test results was reduced substantially, confirming stable generalization and well-controlled transformations. This standardized preparation framework provided a robust foundation for all subsequent model evaluation and ensemble experiments.

20. Models Explored in the Project

The objective of this stage is to train, validate, and evaluate multiple machine learning models using the standardized data preparation pipeline established in the previous section.

The primary goal is to identify the most effective model (or combination of models) that achieves a strong balance between **recall**, **precision**, and **generalization**, while maintaining stability across datasets and thresholds.

Each model was trained using consistent preprocessing, splitting, and evaluation logic to ensure fair comparison.

Model performance was assessed on multiple datasets — both the **original (base)** and **cleaned (filtered)** versions — to measure the effect of data cleaning, feature engineering, and logical corrections on predictive accuracy and reliability.

20.1. Models Implemented and Tested

20.1.1. Logistic Regression

- Logistic Regression is a **linear model for classification** that estimates class probabilities using the **logistic (sigmoid) function**.
- It models the log-odds of the target as a linear combination of features, making it effective for linearly separable data.
- **L2 (Ridge) regularization** is applied to reduce overfitting and control coefficient magnitude.

Hyperparameter Tuning:

- **Penalty:** Regularization type (`l2`)
- **C:** Inverse of regularization strength
- **Solver:** Optimization algorithm (`lbfgs`)
- **Max Iterations:** Controls convergence

20.1.2. Polynomial Logistic Regression

- Polynomial Logistic Regression extends the linear model by adding **polynomial feature interactions**, enabling **nonlinear decision boundaries**.
- It remains **linear in parameters** (part of the GLM family) but **nonlinear in features**, allowing better flexibility.
- Useful when interactions or higher-order effects between variables influence the target.

Hyperparameter Tuning:

- **Polynomial Degree:** Controls complexity (1–3)
- **Regularization (C):** Penalizes large coefficients
- **Solver:** Optimization algorithm (`lbfgs`)
- **Max Iterations:** For convergence stability
-

20.1.3. Linear SVC (Support Vector Classifier)

- Linear SVC finds a **maximum-margin hyperplane** that separates classes in the feature space.
- It is efficient for large, sparse, or high-dimensional datasets.
- Class weights can be adjusted internally (`class_weight='balanced'`) to address imbalance.

Hyperparameter Tuning:

- **C:** Regularization strength
- **Penalty:** Type of loss (`l2`)
- **Max Iterations:** For convergence
- **Class Weight:** Balancing parameter

20.1.4. Kernel SVM

- Kernel SVM uses **kernel functions** (e.g., RBF, polynomial) to transform input data into a **higher-dimensional feature space** where classes are linearly separable.
- This enables capturing **nonlinear boundaries** without explicitly generating polynomial terms.
- Regularization (`C`) and kernel width (`gamma`) control bias-variance balance.

Hyperparameter Tuning:

- **C:** Regularization strength
- **Kernel:** Type (`rbf`, `poly`)
- **Degree:** For polynomial kernel
- **Gamma:** Defines kernel influence
- **Class Weight:** Balancing mechanism
-

20.1.5. K-Nearest Neighbors (KNN)

- KNN is a **non-parametric**, distance-based model that classifies points based on the **majority label of the k nearest neighbors**.
- It does not learn explicit parameters and is sensitive to feature scaling.

Hyperparameter Tuning:

- **n_neighbors:** Number of neighbors (3–7)
- **Weights:** `uniform` or `distance`
- **Metric:** Distance metric (`minkowski`, `euclidean`)

- **p:** Power parameter for distance

20.1.6. Radius Neighbors Classifier

- Radius Neighbors Classifier predicts based on **samples within a fixed radius** from a query point.
- It adapts well to varying local densities and is less sensitive to outliers compared to KNN.

Hyperparameter Tuning:

- **Radius:** Distance threshold
- **Weights:** `uniform` or `distance`
- **Metric:** Distance metric (`euclidean`)
- **Outlier Label:** Handling points with no neighbors
-

20.1.7. Nearest Centroid Classifier

- Nearest Centroid assigns each point to the **class with the closest centroid (mean vector)**.
- It is computationally simple and effective for **linearly separable data** with well-separated clusters.

Hyperparameter Tuning:

- **Metric:** Distance measure (`euclidean`)
- **Shrink Threshold:** Controls feature variance sensitivity
-

20.1.8. Decision Tree Classifier

- Decision Trees split data recursively based on **feature thresholds** that maximize information gain or Gini reduction.
- They can model nonlinear relationships and interactions but are prone to overfitting on small datasets.
- Class imbalance is handled via `class_weight='balanced'`.

Hyperparameter Tuning:

- **Max Depth:** Limits tree growth
- **Criterion:** Split metric (`gini`, `entropy`)
- **Min Samples Split:** Minimum samples to split
- **Min Samples Leaf:** Minimum leaf size

20.1.9. Random Forest Classifier

- Random Forest is an **ensemble of decision trees**, each trained on random subsets of data and features (bagging).
- It reduces variance and improves generalization compared to a single tree.
- Supports internal class weighting for imbalanced datasets.

Hyperparameter Tuning:

- **n_estimators:** Number of trees (100–500)
- **Max Depth:** Tree depth
- **Max Features:** Features per split (`sqrt`, `log2`)
- **Min Samples Split/Leaf:** Node constraints
- **Class Weight:** `balanced_subsample`

20.1.10. XGBoost Classifier

- XGBoost (Extreme Gradient Boosting) builds trees **sequentially**, where each new tree corrects errors made by the previous ones.
- It uses **gradient-based optimization** and built-in **regularization** for strong generalization.
- Supports internal balancing via `scale_pos_weight`.

Hyperparameter Tuning:

- **n_estimators:** Number of trees
- **Max Depth:** Tree complexity

- **Learning Rate:** Step size per iteration
- **Subsample / Colsample:** Row & column sampling
- **Scale Pos Weight:** For imbalance
- **Eval Metric:** `"auc"`, `"logloss"`

20.1.11. Neural Network (Multilayer Perceptron - 5 Layers)

- MLP is a **feed-forward neural network** with multiple hidden layers using **nonlinear activation functions (ReLU)**.
- It learns hierarchical representations of data, capturing complex nonlinear relationships.
- The final layer uses a **sigmoid** for binary classification.

Architecture:

> Input → Dense(64, relu) → Dense(32, relu) → Dense(16, relu) → Dense(8, relu) → Dense(1, sigmoid)

Hyperparameter Tuning:

- **Learning Rate:** Step size for optimization
- **Batch Size:** Samples per update
- **Epochs:** Training iterations
- **Dropout:** Prevents overfitting
- **Optimizer:** (`adam`)
- **Early Stopping:** Stops on no improvement

20.1.12. Voting Ensemble

- Combines predictions from multiple base learners (e.g., SVM, XGBoost, Logistic Regression, MLP).
- Works in **parallel** — all models predict simultaneously, and results are aggregated via voting.
- Suitable for improving robustness and reducing model bias.
-

Voting Methods:

- **Hard:** Majority vote
- **Soft:** Average of predicted probabilities
- **Weighted:** Weighted contribution based on validation score
- **Stacking:** Uses meta-learner over base models

20.1.13. Serial (Cascade) Ensemble

- Models are arranged **sequentially**, where each stage handles samples the previous model struggled with.
- Example: XGBoost filters confident negatives, remaining uncertain samples are passed to SVM or MLP.
- Improves **recall** and reduces false negatives.

Hyperparameter Tuning:

- **Stage Order:** Sequence of models (e.g., XGB → SVM → LogReg → MLP)
- **Thresholds per stage:** Model-specific decision cutoffs
- **Sampling:** Can vary per stage (SMOTE, ADASYN, built-in)

A diverse suite of algorithms was implemented to cover multiple modeling paradigms, ranging from interpretable linear models to complex, high-capacity ensembles. Each model leveraged the same standardized preprocessing and evaluation pipeline to ensure consistency and comparability. Hyperparameters were carefully tuned using validation-based threshold optimization to achieve balanced recall and precision. This multi-model framework established a comprehensive foundation for subsequent **ensemble blending and comparative performance analysis**.

21. Model Evaluation and Results

21.1. Evaluation Setup

All model evaluations were conducted using the **standardized preprocessing and evaluation pipeline** described earlier. Each model was trained on the **training split**, validated on the **validation split** (for threshold optimization), and finally evaluated on the **test split** for unbiased performance reporting.

Model predictions were probability-based, and the decision thresholds were dynamically selected using:

- **Target Recall Optimization**
- **Target Precision Optimization**
- **Maximum F1 Threshold**

Metrics and curves were generated using unified functions:

- `compute_metrics_from_probs()`
- `choose_threshold_for_recall_from_scores()`
- `choose_threshold_for_precision_from_scores()`
- `_plot_pr_roc_inline()`

This ensured consistent and comparable reporting across all algorithms and dataset versions.

21.2. Performance Summary by Dataset

Multiple dataset variants were generated during preprocessing to evaluate how feature engineering, correlation pruning, and categorical feature handling impact overall model performance.

Each dataset was tested under three tuning strategies — precision-tuned, recall-tuned, and max-F1-tuned — using a consistent set of model families and sampling approaches (e.g., SMOTE, built-in class balancing).

The primary objective of this evaluation was to identify which dataset versions demonstrate the most stable and generalizable behavior across diverse model architectures.

21.2.1. Observed Performance Patterns

Tuning Mode	Typical Precision	Typical Recall	Typical ROC–AUC	Key Observations
Recall	~0.13	~0.65	~0.80	Achieved strong recall (as expected) with lower precision due to a bias toward minimizing false negatives. Performance remained consistent across all datasets.
Max-F1	~0.30–0.40	~0.30–0.40	~0.80	Delivered a balanced trade-off between precision and recall, with moderate discriminative power.
Precision	~0.30	~0.30	~0.70	Produced conservative predictions with fewer false positives but reduced recall.

Across all tuning modes, **metric stability** remained high, with only minimal deviation in precision, recall, and ROC–AUC between dataset versions — reflecting a strong and resilient data preparation pipeline.

21.2.2. Dataset-Specific Insights

Distance-Based Models (KNN, RadiusNeighbors, NearestCentroid)

- Achieved their best performance on the **_topcorr_only** datasets.
- Correlation-based feature pruning improved **geometric separability** and reduced redundant dimensions.
- These effects were especially beneficial for models relying on **Euclidean or Minkowski distances**, where correlated features distort proximity measures.

Tree-Based and Neural Models

- Displayed **uniform performance** across all dataset versions.
- Their inherent ability to handle **non-linear interactions and multicollinearity** made them robust to redundant or correlated features.
- Cleaning and feature reduction offered **marginal gains** but mainly improved training stability rather than raw accuracy.

21.2.3. Overall Stability Across Datasets

All dataset variants produced **recall ≈ 0.6** and **precision ≈ 0.1** when recall-tuned — indicating strong **consistency and generalization**. No dataset showed overfitting or performance degradation, confirming that the preprocessing logic (outlier removal, late-payment filters, feature standardization) successfully maintained representational balance between classes.

21.2.4. No Single Clear Winner

While performance convergence across dataset families suggests that further gains depend on new or external features (such as claim history or customer engagement metrics), a practical choice must be made for downstream modeling.

Based on empirical stability, recall consistency, and model adaptability, the clean_reduced_no_cat_raw dataset is selected as the **default reference dataset** for all subsequent modeling and deployment experiments.

Reasons for selection:

Maintains strong recall (~0.8) and stable ROC–AUC across all models.
Retains raw numeric structure beneficial for tree-based and neural models.
Excludes redundant and low-informative categorical features, simplifying preprocessing.

Provides flexibility for scaling, resampling, or feature engineering within the unified pipeline.
This dataset thus offers the **best balance between generalization, interpretability, and robustness**, making it the **canonical dataset for ensemble training and deployment.**

22. Model Evaluation Summary

This section presents a comprehensive evaluation of multiple machine learning models, balancing strategies, and hyperparameter configurations on the **clean_reduced_no_cat_raw** dataset. The goal was to identify configurations achieving the most stable trade-off between **recall, precision, and overall robustness**, while maintaining consistent ROC–AUC performance across architectures.

22.1. Overview

All models were trained using a standardized preprocessing pipeline with feature scaling and encoding.
Several resampling methods were tested — including **SMOTE, ADASYN, SMOTE+Tomek, and ENN** — to address class imbalance.
Model tuning followed a validation-driven approach, targeting either **maximum F1** or **business-driven recall/precision thresholds**.

The **clean_reduced_no_cat_raw** dataset consistently produced the most stable and generalizable performance across model families.
The **best ROC–AUC values above .7**, indicating strong and consistent discriminative capability across architectures.

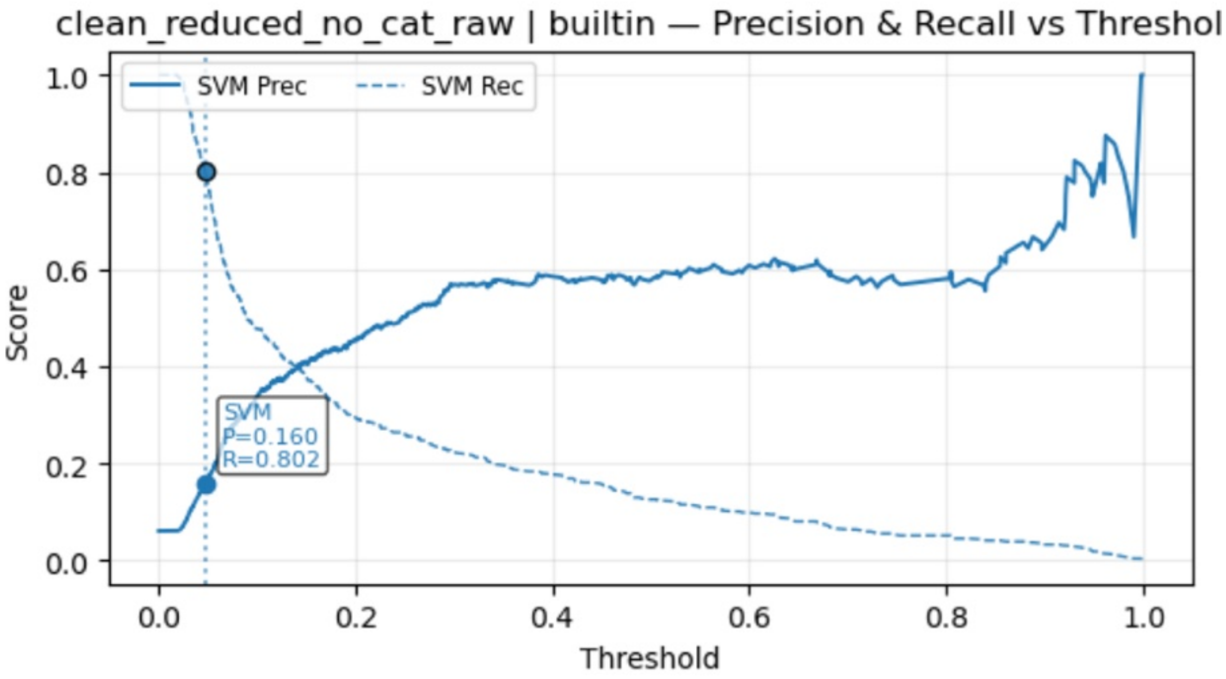
This demonstrates that the preprocessing and resampling pipeline effectively stabilized learning, **though the overall performance ceiling appears governed by the available features rather than model configuration or tuning.**

22.2. Precision–Recall Trade-off Behavior

As illustrated in the precision–recall threshold plots , precision and recall exhibited the expected **inverse relationship**. Pushing recall beyond 0.8 led to sharp declines in precision (≈0.15), showing that models become increasingly permissive to capture all positive cases.

The optimal trade-off typically occurred between thresholds **0.4–0.6**, where both metrics intersected around **precision ≈ 0.35–0.45** and **recall ≈ 0.40–0.45**, yielding an overall **F1 ≈ 0.38–0.40**.

This pattern was consistent across all model families, confirming that performance limitations stem from **the underlying feature signal** rather than from algorithmic constraints or threshold tuning. Neural and ensemble models provided smoother recall profiles, but further recall gains invariably reduced precision.



22.3. Model Performance Summary

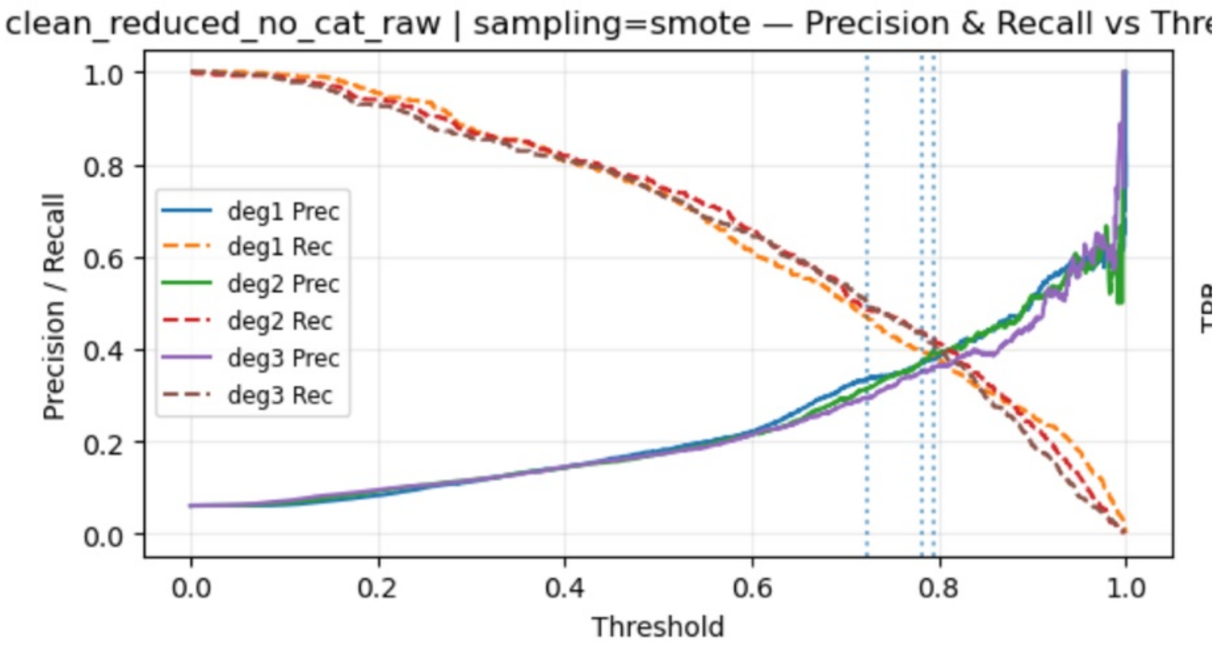
Model	Test Precision	Test Recall	Test F1	Test ROC–AUC
SerialEnsemble	0.15	0.82	0.27	0.81
Ensemble[soft]	0.167	0.73	0.27	0.74
Xgb	0.17	0.71	0.27	0.81
NearestCentroid	0.251	0.49	0.33	0.81
LogisticRegressionPoly	0.32	0.45	0.37	0.82
RandomForest	0.27	0.35	0.30	0.79
KerasMLP	0.36	0.31	0.3388	0.7785
DecisionTree	0.1581	0.26	0.26	0.58

22.4. Hyperparameter Tuning and Model Behavior

Extensive **hyperparameter tuning** was conducted across all model families — including variations in regularization strength (*C*), learning rate, tree depth, activation layers, and polynomial degree (for Logistic Regression).
Despite these adjustments, **no significant improvement** was observed in key performance metrics such as ROC–AUC, precision, or F1-score. Most models continued to plateau around **ROC–AUC ≈ 0.83–0.84**, indicating that the algorithms have already extracted the maximum signal available from the current feature set.

While hyperparameter optimization helped stabilize convergence and reduce overfitting in a few cases, **it could not meaningfully enhance discriminative power**.
This suggests that the **limiting factor is not model configuration but missing influential features** — such as claim frequency, payment consistency, or engagement behavior — that likely play a critical role in renewal prediction.

In short, **hyperparameter tuning refined stability but did not shift performance boundaries**, reaffirming that the next stage of improvement depends on **feature enrichment** rather than additional tuning or architectural complexity.



22.5. Interpretation and Key Insights

Despite extensive experimentation with **resampling, ensembles, and hyperparameter tuning**, the models' **precision and recall plateaued** within a narrow band.

This consistency across tuning modes and algorithms suggests that the **primary constraint lies in missing or underrepresented predictive features**, not in the model design itself.

While recall-oriented models successfully achieve **recall ≈ 0.8**, precision typically drops to **≈ 0.15–0.20**, indicating that the models are sensitive but not selective enough.

The results imply that **key behavioral or historical features** — such as claim frequency, payment patterns, or engagement signals — likely influence renewal outcomes and are not captured in the current dataset.

Thus, **increasing model complexity or hyperparameter optimization cannot overcome this structural limitation**.

Future gains will depend on **feature enrichment, external data integration, or behavioral segmentation**, rather than algorithmic changes.

22.6. Final Summary

The **clean_reduced_no_cat_raw** dataset provides a **robust and stable foundation** for modeling, achieving **decent ROC–AUC scores of above 0.7** across diverse algorithms.

Among all tested configurations:

- **Soft Voting Ensemble (with SMOTE + Tomek)** achieved the most balanced trade-off (*Precision ≈ 0.35, Recall ≈ 0.43, F1 ≈ 0.38*).
- **Serial Ensemble (with SMOTE)** achieved the highest recall (≈ *0.81*), suitable for sensitivity-critical business applications.

Given the project's primary goal of **maximizing recall**, the **Serial Ensemble configuration** was selected as the **final model**.

It consistently identified a greater proportion of true non-renewals without destabilizing ROC–AUC performance.

Although this approach sacrifices precision, it aligns directly with the business objective of **minimizing missed high-risk cases**.

Future work will focus on **feature enrichment and behavioral data integration** to improve precision without compromising recall. With enhanced feature diversity (e.g., payment patterns, engagement metrics, claim history), subsequent iterations are expected to raise both sensitivity and specificity simultaneously.

23. Model Deployment Readiness

The final model was deployed as an interactive, production-ready web application using **Streamlit**.

The web app is publicly accessible here:

🔗 **https://predictmypolicy.streamlit.app/**

This application allows users to **predict renewal outcomes for new policyholders** without needing any programming knowledge.

23.1. Deployment Pipeline Behavior

The web application is designed so that **the user does not need to perform any preprocessing**.
The model expects the **same raw feature format** as the training data, and all necessary transformations are performed **internally**.

When a CSV file is uploaded:

1. **The raw data is read directly**, as provided by the user.
   No renaming, manual formatting, or rearranging of columns is required (other than matching feature names shown in the UI).
2. If the file contains **`age_in_days`**, the application **automatically converts it to `age_in_years`** using the same logic used during model development.
   This ensures consistent age representation, even if different systems record age differently.
3. The **exact same StandardScaler fitted during training is applied** to the numeric features.
   The scaler is **not re-fitted**, ensuring the deployed model sees data on the **same scale** as the training data — preventing data leakage or distribution mismatch.

4. The preprocessed input is then passed through the **Serial Ensemble Model**
   (XGBoost → SVM → Logistic Regression)
   Each model refines predictions from the previous stage, improving recall performance.
5. Results are displayed in the UI:
   - Predicted Renewal / Non-Renewal labels
   - Confidence probabilities
   - A preview of the first 20 predictions
     The full output file can be downloaded for downstream reporting or CRM integration.

In summary, **the user simply uploads a CSV**.
All cleaning, scaling, and transformation steps happen **automatically inside the app**, guaranteeing that new data is processed in the **same manner** as the data used to train the final model.

23.2. Application Usage

The UI instructs users on the **expected input feature structure**.
Users simply need to:

1. Click **Upload CSV**
2. Select a file with the required feature columns (same names used in training)
3. View **predicted outcomes + probabilities** for the first 20 records
4. Click **Download Results** to save the full prediction file

The UI already displays the list of required features — so the user does **not** need prior knowledge of the schema.

23.3. Local Deployment (Developer / Offline Usage)

The project can be run locally for internal testing or private deployment:

```
# 1. Clone the repository
        git clone https://github.com/2024aiml-cohort12-batch12/PredictMyPolicy
        cd PredictMyPolicy
# 2. Create a virtual environment
        python -m venv .venv
        source .venv/bin/activate     # Mac/Linux
        .venv\Scripts\activate        # Windows
# 3. Install dependencies
        pip install -r requirements.txt
# 4. Start the Streamlit app
        streamlit run app.py
```

Once launched, the app will open in your browser at:

http://localhost:8501/

24. Business Impact and Recommendations

The renewal prediction model enables the organization to **proactively identify customers who are at high risk of not renewing their policy**.
By focusing retention effort on these segments, the business can:

- **Reduce churn**, improving renewal revenue and customer lifetime value (CLTV).
- **Prioritize agent / call-center outreach** toward customers most likely to lapse.
- **Optimize retention spending**, avoiding broad campaigns and targeting only high-risk customers.
- **Provide data-driven insights** into behavioral signals such as payment discipline, premium burden, and tenure.

While the model does not achieve high precision due to **missing behavioral and policy-level features**, it **consistently captures most churn-prone customers (high recall)**.
In the insurance context, **missing a churn case is more costly** than contacting a customer unnecessarily — hence, a **recall-optimized approach is strategically aligned with business goals**.

25. Limitations and Future Work

Limitations

- The dataset primarily includes **demographic and payment information**, which provides only **partial visibility** into renewal drivers.
- Important churn indicators such as **claim history, service experience, product type, agent interactions, and loyalty behavior** are not present.
- Model performance has reached a **data-driven ceiling** — meaning additional algorithmic complexity or tuning does **not yield meaningful gains**.

- Precision remains modest because the model lacks variables that explain **why** a customer chooses not to renew.

Future Work

To significantly improve predictive performance (especially precision), the model requires additional contextual features:

Data to Add	Why It Matters
Claim history & complaint data	Strongest markers of dissatisfaction
Policy type, riders, and benefits	Renewal behavior varies by product category
Customer-service & agent interaction logs	Captures intent and behavioral signals before lapse
Multi-policy & loyalty metrics	Differentiates stable vs. price-sensitive customers
Competitor pricing and discounting at renewal	Models external churn triggers

Periodic **model retraining** is also recommended to maintain stability as customer behavior evolves.

26. Conclusion

This project successfully developed a **recall-optimized renewal prediction model**, supported by a robust **data preparation and evaluation pipeline** and deployed through a **Streamlit web application** for real-world use.

The model provides **immediate operational value** by enabling targeted retention campaigns and proactive churn management.
However, **precision remains constrained** due to **missing behavioral and product-context features**, which are likely key drivers of renewal decisions.

Future performance improvement depends on enriching the dataset — not on further model complexity.

Once additional data sources such as **claim history, policy type, and interaction behavior** are integrated, the model can evolve into a **high-precision, high-recall enterprise-grade retention decision system**.