Assessment 3: Real-World Applications of Artificial Intelligence

36121 Artificial Intelligence Principles and Applications

Group 3

Name	Student number
Niketh Damodaran	25159512
Brett McKeehan	24863134
Rakesh Hebbar	24710485
Tom Richards	26003335

TABLE OF CONTENTS

1. Executive Summary

1.1 Project Overview

This project creates an ethical AI-powered pricing recommendation system for a major Australian retail energy provider to maximise business value when renewing customer contracts. Using machine learning, the Contract Renewal Optimisation System (CROS) predicts customer responses to different renewal offers based on their demographics and circumstances so the provider can target discount strategies for each household.

1.2 Objectives

We aim to accurately predict customer responses to renewal offers, quantify the expected value of each strategy option and optimise offers based on either maximum profit or minimum churn. We target >60% prediction accuracy with our proof of concept (POC) and will provide visualisations that translate AI predictions into insights for business users. CROS must also comply with regulatory and ethical guidelines by excluding vulnerable customers. Success will be measured by improved portfolio profitability compared to historical renewals.

1.3 Key Findings

Our best-performing XGBoost model predicted customer renewal decisions with 73% accuracy across 869,000 anonymised historical cases. Back-testing confirmed CROS would have generated an extra $68 per customer across the 16-month data cycle by matching discount strategies to individual household characteristics – about $59 million in total value. The decision to maximise value resulted in churn only increasing by 0.4%.

2. Introduction

2.1 Background and Context

Customer retention is a key profit driver in Australia's $46.9 billion-a-year retail energy market (Fahey, 2024). Most customers enter fixed-term contracts that need to be renewed on expiry, normally after 12 to 24 months (ACCC, 2023). This is a delicate time, as customers might switch providers if they don't find the new offer value for money. This makes renewals a crucial decision point in which providers must weigh up potential profit against customer satisfaction (IPART, 2018).

2.2 Problem Statement

After fixed term expiry providers have three possible contract renewal options:

1. Maintain the existing discount (same contract)
2. Remove the discount
3. Lower the discount

Customers can respond by:

1. Accepting the new contract
2. Leaving for a different company (churn)
3. Calling back to negotiate a better deal.

Offering minimal or no discounts raises the risk of customers leaving or calling to negotiate. Keeping discounts unchanged makes customers stay put but limits profit growth. Yet while renewals are a juggling act, many providers take a simplistic approach with standard offers or basic customer segmentation. Some still use basic spreadsheet-based tracking systems that completely ignore individual customer characteristics or churn risk indicators (JEC, 2025).

The challenge is finding each customer's threshold i.e. the smallest discount that will secure their renewal. By learning from historical behaviour patterns, an AI solution can determine which discount strategy will squeeze the most value from each household's contract.

2.3 Project Scope

The POC project will:

- Build predictive models using 869,473 historical customer renewal records.
- Estimate the probability of each customer response (accept, churn or call back to negotiate) to three different offers.
- Determine the value expected from each option.
- Identify the best renewal offer for each customer.
- Visualise findings on an interactive dashboard.

Out of scope: Automated customer communications or CRM integrations not possible for the POC.

3. Project Planning

3.1 Project Goals and Objectives

The goal is to improve provider profitability by pairing customers with AI-optimised renewal offers. This will be achieved by:

- Delivering CROS POC prediction accuracy of at least 70%.
- Better average profit than existing renewal strategies.
- Comparing the expected value of treatment options.
- Building an interactive results dashboard for end users.
- Enforcing ethical safeguards to exclude vulnerable customers (i.e. those under financial hardship, seniors and concession holders) from profit-driven optimisation to comply with ACCC and Australian Energy Regulator guidelines.

3.2 Stakeholder Analysis

Provider management: Need ROI, accurate predictions. and a clear implementation roadmap. Require regular briefings.

Customer retention team: CROS end users and testers who need the dashboard and informed recommendations.

IT operations: Need assurances CROS won't cause security, maintenance or privacy issues, with architectural input to avoid integration headaches.

End customers: Get tailored offers from CROS. Responses tracked by the retention team.

Regulatory bodies: ACCC and Australian Energy Regulator must be approached on automatic exclusion of vulnerable customers in line with ethical AI implementation standards (NAIC, 2024).

3.3 Project Timeline and Milestones

Phase 1: Foundation (days 1-3)

- Problem definition, finalising scope
- Data acquisition, anonymisation
- Exploratory Data Analysis (EDA)

Phase 2: Model development (days 4-7)

- Feature engineering, selection
- XGBoost classifier development
- TensorFlow/Keras deep neural network implementation
- Class imbalance handling
- Model comparison
- Performance checks

Phase 3: Optimisation system (days 8-9)

- Expected value calculation system implementation
- Treatment optimisation algorithm development
- Integration of selected model into prediction pipeline

Phase 4: POC deployment preparation (days 10-17)

- Dashboard development with Streamlit
- System integration, testing, documentation (Appendix B)

3.4 Risk Assessment and Mitigation

Risk	Likelihood	Impact	Mitigation
Data quality	High	High	Data cleaning, outlier + missing value checks
Model performance	Medium	High	Fallback heuristic rules
GPU limitations	Medium	Medium	Efficient feature selection
Privacy	Medium	High	Anonymisation of customer data
Stakeholder resistance	Medium	Medium	Early pilot testing
Regulatory compliance	Low	High	Regular consultation with legal team
Integration	Medium	Medium	Early API agreement
Scope creep	High	Medium	Clear requirements documentation, strict change management process

4. Methodology/System Overview

4.1 System Architecture

CROS uses modular architecture which processes customer data to generate actionable renewal recommendations using six core components (figure 1).

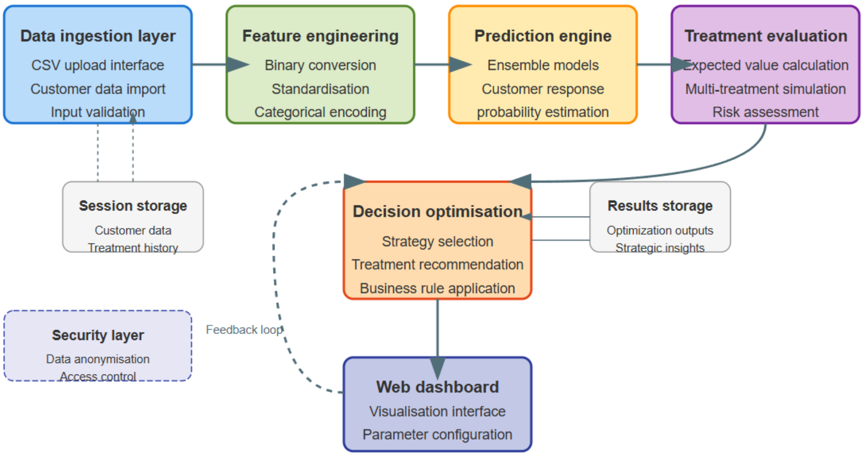


Figure 1: Contract Renewal Optimisation System (CROS) architecture

Data ingestion layer: Processes customer renewal data through a secure upload interface. Accepts CSV files containing anonymised customer information including state, usage patterns, contract history and age bracket.
Feature engineering pipeline: Converts raw customer data into model-ready features with binary conversion, numerical standardisation and categorical encoding. Generates temporal features from renewal dates to calculate discounts.
Prediction engine: Predict probabilities for customer responses (accept, churn or call back to negotiate) using XGBoost classifier and TensorFlow/Keras deep neural network models trained on historical renewals. Best model will be used in final POC.
Treatment evaluation: Determines expected value for three contract offers by applying outcome probabilities to corresponding profit or cost value.
Decision optimisation: Chooses the contract with highest expected value or lowest churn.
Web dashboard: Interactive Streamlit interface for business users to upload datasets, set parameters, choose optimisation target (value or customer retention) and visualise outcomes.

4.2 Tools and Technologies

- **Python 3.9.6:** Core language
- **Scikit-learn 1.6.1:** Machine learning library
- **Pandas 2.2.3:** Data analysis
- **NumPy 2.0.2:** Array operations
- **Streamlit 1.45.0:** Web interface
- **Plotly 6.0.1:** Data visualisation
- **Joblib 1.5.0:** Model persistence
- **Altair 5.5.0:** Statistical visualisations

Development and deployment:

- **Git:** Version control
- **GitHub:** Code repository here
- **Virtual environments:** Dependency management

4.3 Database Design and Management

Data management uses session-based storage instead of conventional database architecture (figure 2).

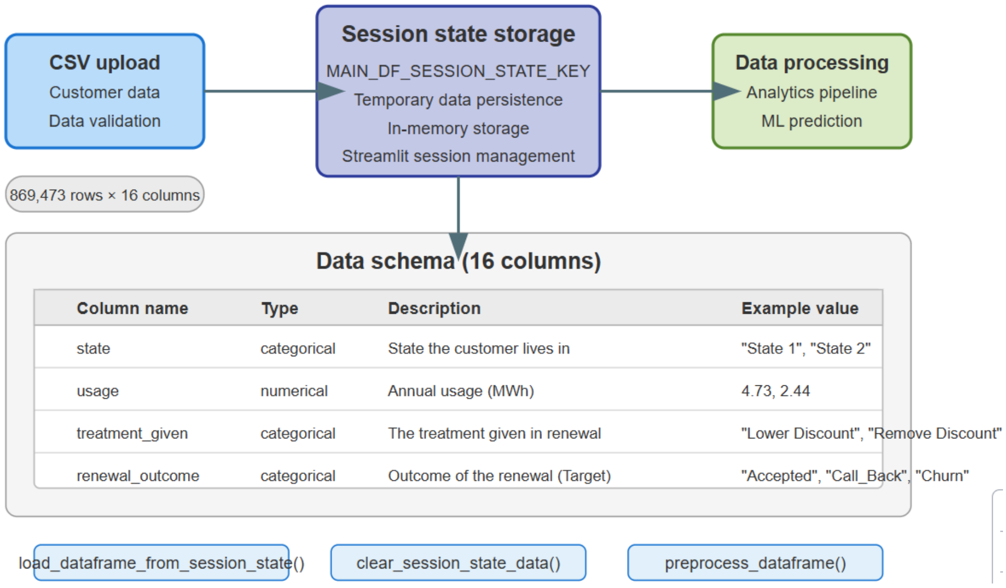


Figure 2: Data management and storage architecture

Session state storage: App makes use of Streamlit's MAIN_DF_SESSION_STATE_KEY session state feature to preserve data through user interactions.
Data structure: 16 columns make up the dataframe, which displays customer characteristics, treatment options and results (Appendix A).
Data flow: CSV loaded into session state, transformed in preprocessing and prediction pipelines.
Flow control: Load and save operations in database.py for state consistency.

4.4 Security Considerations

Anonymised data: Applied with randomisation to prevent customer identification for ethical AI implementation.
Access control: App cannot store data beyond the session environment, deleting it when the app terminates.
Input validation: Non-standard inputs identified and managed in preprocessing.
Controlled file operations: File type validation in data upload procedure.

5. Design and Implementation

5.1 System/Software Design

For cooperative development, the modular Python design divided functions:

- **Configuration module (config.py):** Includes data definitions, model paths, system parameters and optimisation for code mods.

- **Data handler (data_handler.py):** Data loading, preprocessing, binary conversion, categorical encoding and derived feature calculations like determining discount changes.
- **Model handler (model_handler.py):** Combines model loading and prediction processes with caching for performance.
- **Inference handler (inference_handler.py):** Carries out the logic for treatment evaluation and chooses the best contract strategy.
- **Visualisation handler (vizualisation_handler.py):** Plotly visualisations of recommendation distributions based on customer attributes.
- **Dashboard (app.py):** Interface with state management and user interaction.

5.2 Development Methodology

Agile development was adapted for the small-team environment (figure 3):

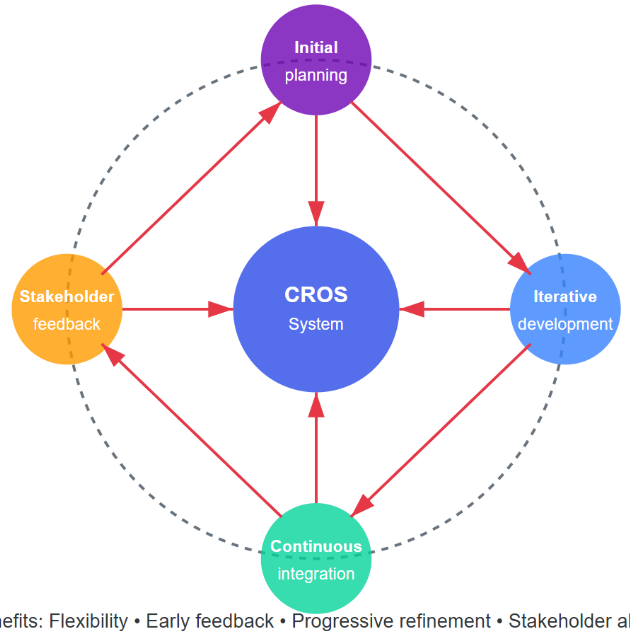


Figure 3: CROS agile project cycle

Planning: Defined core functionality and established POC features.
Iterative development: Prioritised functionality over optimisation for POC.
Continuous integration: Committed regularly to validate the system and spot interface issues.
Feedback loops: Regular progress check-ins with stakeholders.

5.3 Coding and Development Process

Development began with the EDA to understand customer feature relationships with renewal outcomes (Appendix C), followed by modular implementation organised into task-specific components. For the dashboard, session-state management kept context across interactions so business users could factor and refactor visualisations without losing their place (Appendix C).
XGBoost was tuned through grid search, with the optimal configuration using 100 estimators, maximum depth of 10, learning rate of 0.1 and 0.8 subsample ratio (figure 4). Sample

weights were not used as they hurt overall accuracy by forcing predictions toward the minority class.
Our TensorFlow/Keras deep neural network had two hidden layers (128 and 64 neurons) using ReLU activation at 30% dropout between layers, which helped prevent the model memorising majority class examples. It was compiled with Adam optimisation, trained over 20 epochs with batch size of 32.

Parameter	XGBoost	TensorFlow/Keras DNN
Architecture	100 estimators	2 hidden layers (128, 64 neurons)
Complexity control	Max depth=10	30% dropout rate
Learning parameters	Learning rate=0.1	ReLU activation
Sampling	Subsample ratio=0.8	Batch size=32
Training	Grid search	20 epochs
Optimiser	Gradient boosting	Adam
Sample weights	Not used	Not used

Figure 4: Model hyperparameter comparison

5.4 Testing Strategy and Implementation

Automated checks identified missing values and managed outliers, while class imbalance was handled with scikit-learn weighting techniques. The model handler included pipeline verification before predictions, while the inference handler used scenario-based testing in which contract options were evaluated with controlled inputs based on defined value calculation formulas (Appendix A). Data verification was also applied at each processing stage to guarantee proper flow between components.

6. Results and Analysis

6.1 Performance Evaluation

The EDA was conducted on 869,473 historical customer renewal records from August 2019 to December 2020. The 16-feature dataset (Appendix C) showed class imbalance in renewal outcomes (figure 5), with 64.1% of existing customers accepting the offer, 33.3% calling back to negotiate a better deal and just 2.6% churning.

	proportion
renewal_outcome	
Accepted	0.641034
Call_Back	0.333235
Churn	0.025731

Figure 5: Renewal outcome breakdown

The distribution of customer treatment types – remove discount (41.3%), lower discount (36.3%) and same contract (22.4%) – was balanced, indicating sufficient training samples across contract categories. A visualisation of offer made and renewal outcome revealed the treatments themselves garnered varied customer responses (figure 6).

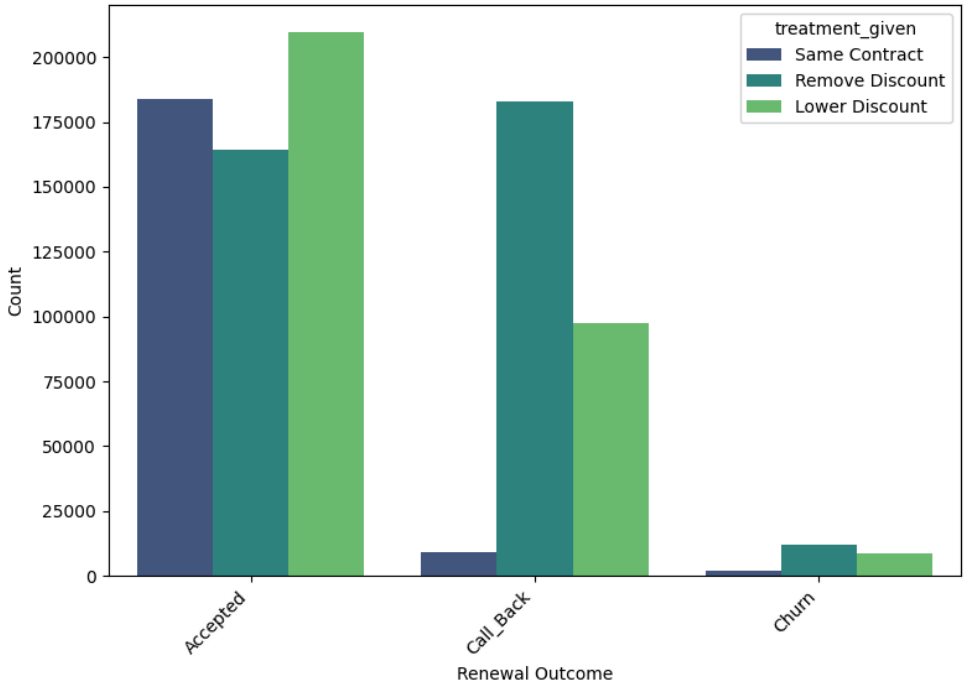


Figure 6: Renewal outcome v treatment given

Offering the same contract had the highest acceptance rate and lowest churn. Removing the discount altogether had the highest churn risk and highest call-back rate – while lowering the discount was a middle ground. A box plot showed higher discounts are strongly associated with direct contract acceptance, while lower discounts triggered callbacks or churn (figure 7). Obviously, the discount level makes a difference.

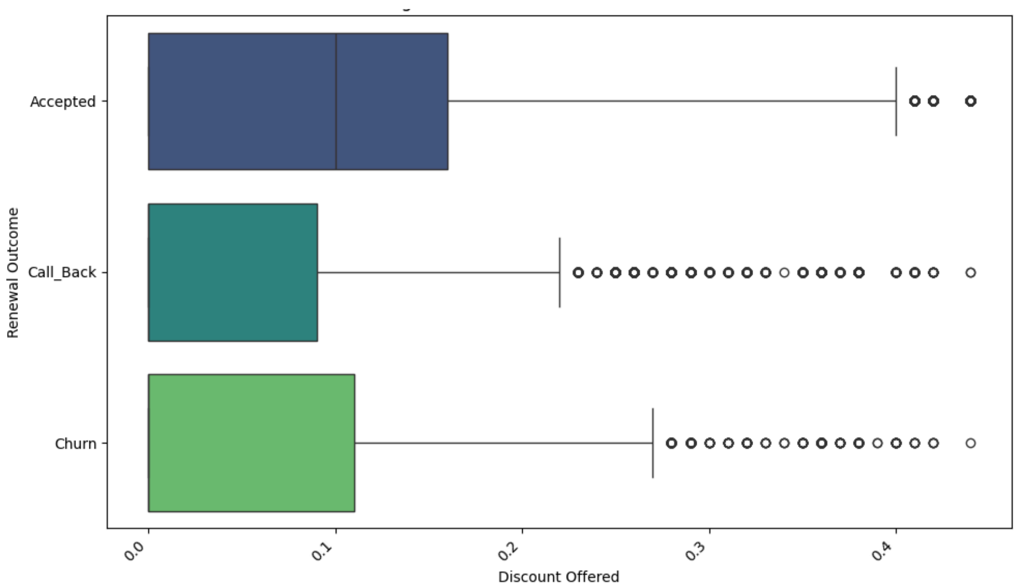


Figure 7: Box plot of average discount offered and renewal outcome

An XGBoost classifier and TensorFlow/Keras neural network were tested to predict customer responses to renewal offers. They were each trained on 75% of the data, with 25% for testing.

6.1.1 XGBoost classifier

The dataset imbalance was evidenced by varied performance across the three classes (figure 8). Precision was high (0.75) for the majority “accepted” class with 0.87 recall, meaning the model reliably predicted when a customer took an offer. For the “call back” class, precision (0.65) and recall (0.51) were moderate. The minority “churn” class had excellent precision (0.97) but very low recall (0.03), showing the model rarely flagged customers as likely to churn – but was accurate when it did.

Class	**Precision**	**Recall**	**F1 score**	**Support**
0 (accepted)	0.75	0.87	0.81	139,461
1 (call back)	0.65	0.51	0.57	72,396
2 (churn)	0.97	0.03	0.05	5512
Accuracy			0.73	217,369
Macro avg	0.79	0.47	0.48	217,369
Weighted avg	0.72	0.73	0.71	217,369

Figure 8: Table of XGBoost classification performance metrics

The model hit 73% overall accuracy, significantly outperforming the 33% baseline from random guessing in a three-class problem. Removing sample weights improved accuracy from 59%, though this hurt sensitivity to churn. Uneven recall across classes suggested the model favoured acceptance over churn, which made sense given the dataset class imbalance.

6.1.2 TensorFlow/Keras deep neural network

Precision was highest for the “churn” class (0.99, figure 9), but recall was low (0.03). For the majority “accepted” class, precision and recall were high (0.73 and 0.91 respectively). Again, it comes down to the imbalanced dataset. Attempts to improve recall of the “churn” class with the “sample_weights” hyperparameter were unsuccessful with accuracy decreasing to 49%.

Class	**Precision**	**Recall**	**F1 score**	**Support**
0 (accepted)	0.73	0.91	0.81	139,461
1 (call back)	0.68	0.41	0.51	72,396
2 (churn)	0.99	0.03	0.05	5512

Accuracy			0.72	217,369
Macro avg	0.80	0.45	0.45	217,369
Weighted avg	0.72	0.72	0.69	217,369

Figure 9: Table of sequential neural network performance metrics

Overall performance was slightly less balanced than XGBoost with weighted average recall and accuracy both 0.72. The class recall value disparity (0.03, 0.41, 0.91) indicated the model was tuned for general accuracy which aligns with the business objective of calculating expected value with probabilities of all classes.

6.1.3. Model selection

XGBoost was the best primary predictive engine. Not only did it boast superior performance with 73% accuracy compared to the neural network's 72%., its overall weighted metrics were stronger (figure 10). It also had better precision-recall balance with a higher weighted average F1 score (0.71 v 0.69).

Metric	XGBoost	TensorFlow/Keras
Overall accuracy	0.73	0.72
Weighted average recall	0.73	0.72
Weighted average F1 score	0.71	0.69
Class recall distribution	0.87, 0.51, 0.03	0.91, 0.41, 0.03

Figure 10: Comparison table of model performance

6.1.4. Financial impact analysis

Back-testing compared the AI approach with historical renewals. The model was tweaked to balance churn predictions with profits instead of prioritising churn recall, which helped the accuracy of expected value estimates. When projected across the entire 16-month dataset of 869,000 customers, CROS would have generated an extra $68 profit each – about $59 million combined – by matching individuals with the right discount strategy (figure 11). CROS's business value was validated.

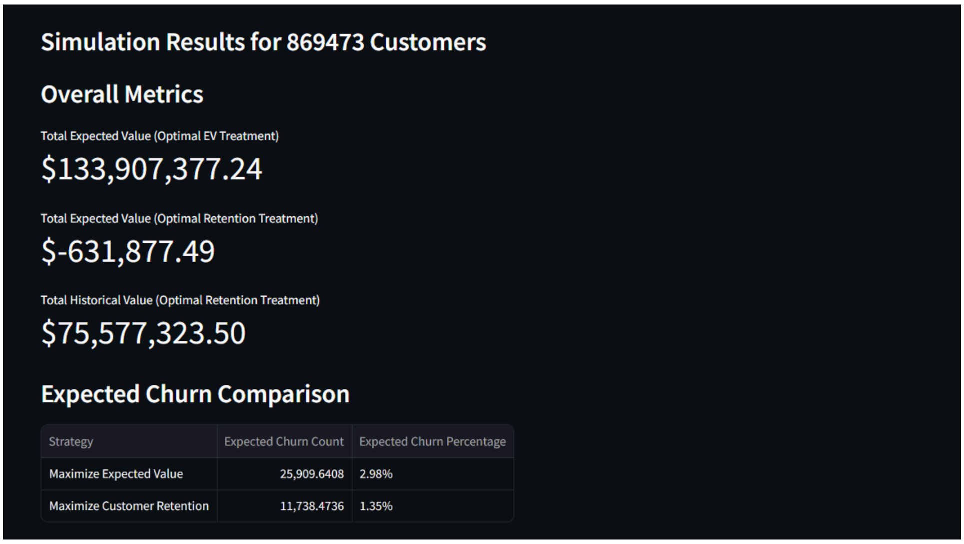


Figure 11: Streamlit dashboard showing $59 million in combined extra profit if AI solution was applied (total expected value-total historical value)

6.2 User Testing and Feedback

Feedback from business representatives was incorporated into development, like data anonymisation, value calculations and dashboard functionality. Future user testing will involve retention team members evaluating recommendations via the Streamlit app (Appendix B).

6.3 Challenges Faced and Solutions

Imbalanced data: With just 2.6% of historical renewals ending in churn, we tried class weighting in model training. Yet removing sample weights took XGBoost accuracy from 59% to 73%, which outweighed the benefits of higher recall for the minority "churn" class.
Model accuracy: Initial models over-predicted churn. We adjusted hyperparameters and removed sample weights for more balanced performance, improving expected value calculations.
Value calculation complexity: We developed a calculation framework to account for usage patterns and discount levels (Appendix A).

7. Deployment and Maintenance

7.1 Deployment Plan

As a POC, CROS deploys in three phases: 1) Limited pilot with retention specialists handling ~5% of renewals; 2) Controlled expansion to 25% of renewals with A/B testing; 3) Broader integration if POC is a success.
Technical implementation includes environment setup and user training.

7.2 Maintenance and Support Strategies

Quarterly model retraining with automated performance monitoring will address changing customer behaviours. Ethical reviews will ensure fair treatment of vulnerable customers, while a tiered support system handles technical issues and regulatory compliance.

7.3 Future Enhancements

Subject to POC take-up, future development should incorporate all available customer data beyond the 16-month lifecycle used for this project. Customisable Streamlit dashboards based on role would also be valuable for end users.

8. Conclusion and Recommendations

8.1 Summary of Findings

The CROS POC using XGBoost hit 73% accuracy in predicting customer renewal responses. The model performed strongly on the majority “accepted” class (87% recall) and moderately on callbacks (51% recall), but struggled to identify potential churn (3% recall). There were clear correlations between discount strategies and customer behaviour – offering the same contract led to the highest acceptance rate with minimal churn risk, while removing discounts raised the likelihood of both churn and callback frequency. Back-testing showed CROS would have delivered $68 in extra profit per historical customer while maintaining ethical safeguards that automatically exclude the vulnerable from profit-driven optimisation.

8.2 Recommendations for Future Work

To encourage wider business adoption, we recommend:

- Customer behaviour is likely to be influenced by competitor offers at renewal time. Enhancing the dataset with a feature for competition intensity will likely improve performance.
- API integration with existing CRM systems for automated renewal communications based on CROS recommendations.
- Add a feedback loop capturing actual customer responses to AI-recommended offers for prediction refinement over time.
- Expand use cases to other customer segments within the company’s bases (e.g., gas contracts, SME customers)

9. References

Australian Competition & Consumer Commission (2023) 'Inquiry into the National Electricity Market: December 2023 Report', Australian Government pp16-38. Accessed 06/05/25

Fahey, Jack (2024) 'Electricity Retailing in Australia - Market Research Report (2014-2029)', IBISWorld. Accessed 06/05/25

Independent Pricing and Regulatory Tribunal (2018) 'Review of the performance and competitiveness in the NSW retail energy market', Australian Government pp40-61. Accessed 07/05/25

Justice and Equity Centre (2025) 'Delivering more protections for energy consumers: changes to retail energy contracts', Australian Government pp 9-19. Accessed 07/05/25

National Artificial Intelligence Centre (2024) 'Voluntary AI Safety Standard'. Australian Government pp4-38. Accessed 08/05/25

10. Appendices

A. Project Documentation

CROS repository on GitHub

Link: https://github.com/dniketh/RenewalStrategy-Dashboard

Customer renewal value calculations

Purpose: Estimate financial impact of customer decisions (churn, call back, acceptance)

Current value without renewal

Baseline value (annual gross profit from customer prior to renewal)

$$Val_b = (1 - D_b) \times (U \times Vrev + Frev) - (U \times Vcost + Fcost)$$

Scenario analysis

1. Value if customer churns:

$$Val_{churn} = -Val_b$$

Note: Applies to all renewal offers

2. Value if customer calls back :

$$Val_{callBack} = 0$$

Note: Applies to all renewal offers

3. Value if customer accepts:

a) Same contract:

$$Val_{AccSameContract} = 0$$

b) Lower discount:

$$Val_{AccLowerDisc} = (D_b - D_o) \times (U \times Vrev + Frev)$$

c) Remove discount:

$$Val_{AccRemoveDisc} = (D_b) \times (U \times Vrev + Frev)$$

Definitions

$$D_b = Discount\ before\ renewal$$

$$D_o = Discount\ offered\ in\ renewal$$

$$U = Annual\ usage\ in\ MWh$$

$$Vrev = Variable\ revenue\ per\ MWh = \$247$$

$$Frev = Fixed\ revenue\ per\ year = \$436$$

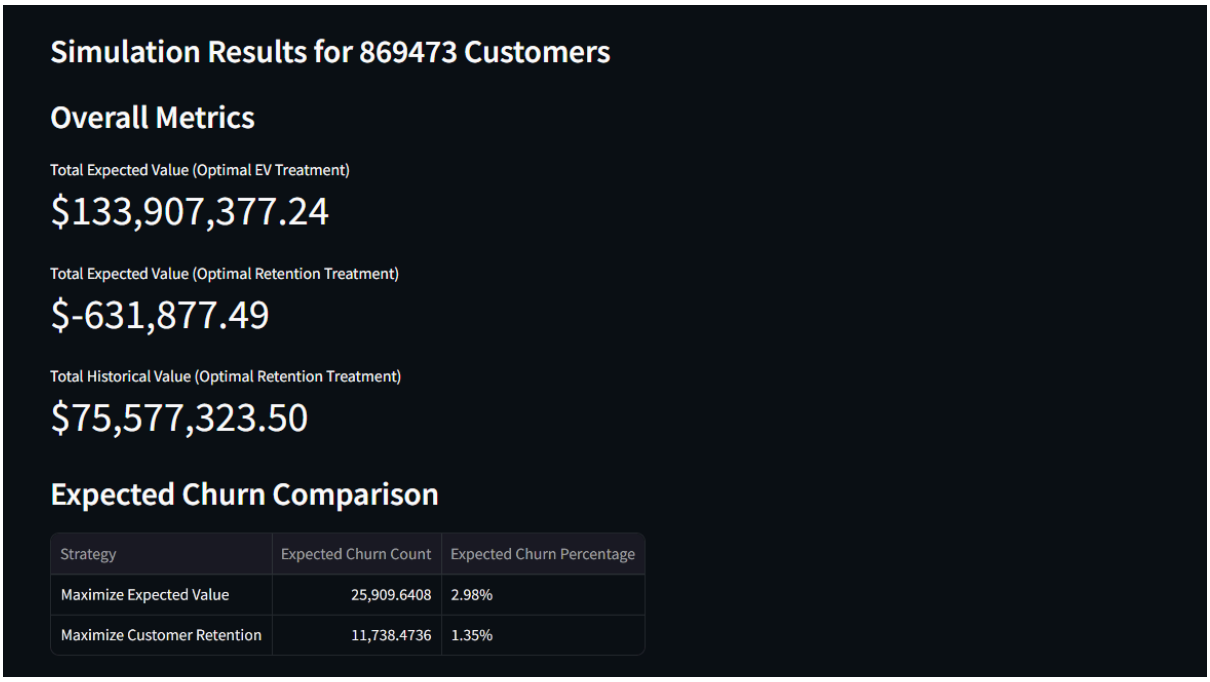
$$Vcost = Variable\ cost\ per\ MWh = \$209$$

$$Fcost = Fixed\ cost\ per\ year = \$269$$

B. User Guide

[Streamlit dashboard setup and dependencies]

Streamlit dashboard screenshots



Simulation Results for 869473 Customers

Overall Metrics

Total Expected Value (Optimal EV Treatment)

$133,907,377.24

Total Expected Value (Optimal Retention Treatment)

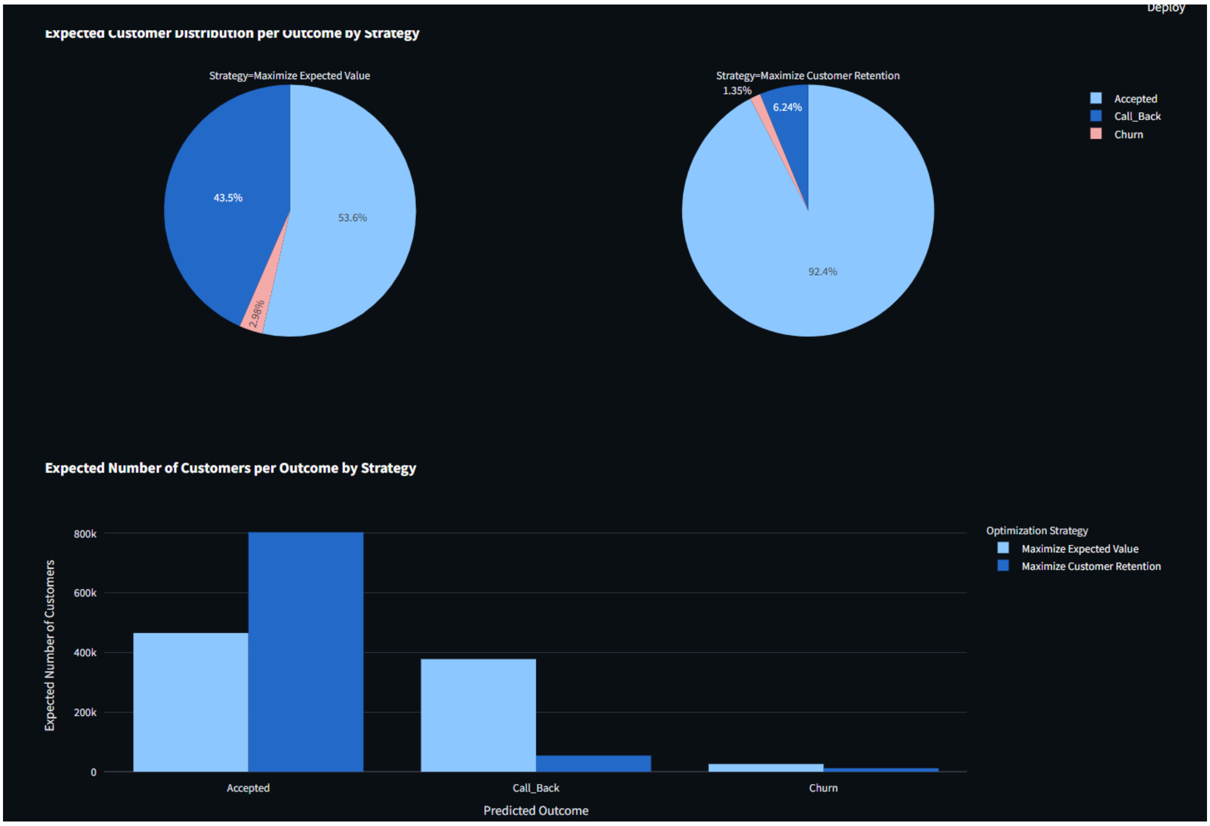
$-631,877.49

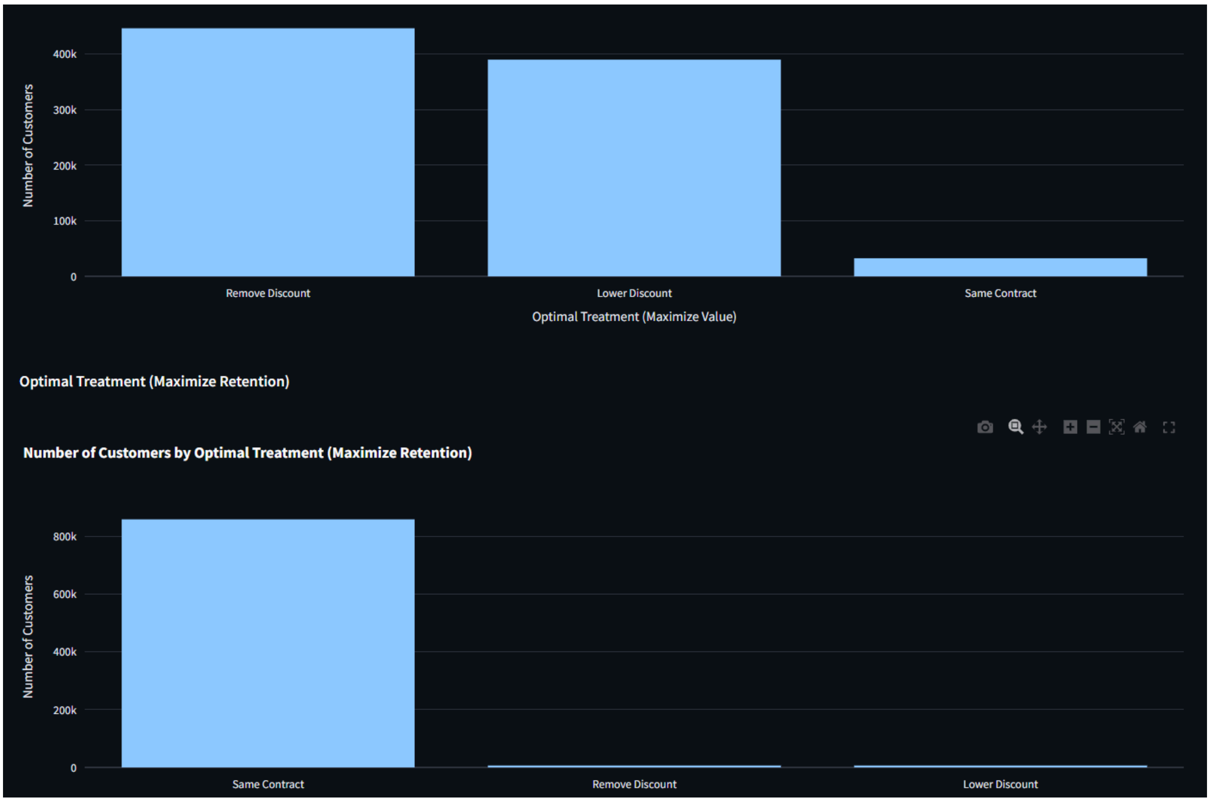
Total Historical Value (Optimal Retention Treatment)

$75,577,323.50

Expected Churn Comparison

Strategy	Expected Churn Count	Expected Churn Percentage
Maximize Expected Value	25,909.6408	2.98%
Maximize Customer Retention	11,738.4736	1.35%





C. Additional Data

Renewal data dictionary

Column name	Definition
state	State customer lives in
communication_preference	Preferred communication method (email or letter)
green	Does customer pay for green electricity (yes/no)
dual_fuel_customer	Does customer also have a gas account with this energy company (yes/no)
direct_debit_flag	Does customer pay via direct debit i.e. automatic payments (yes/no)
usage	Annual usage (MWh) of customer
cust_tenure	Number of years customer has been with energy retailer (max value of 5 indicates >=5 years)
years_on_disc	Number of years customer has been on a contract with discounted price (max value of 5 indicates >= 5 years)
age	Customer age in 10-year buckets
before_discount	Discount customer was on before renewal
before_channel	Channel customer was acquired through
renewal_date	Year and month renewal occurred
treatment_given	Renewal treatment given
discount_offered	Renewal discount offered
renewal_outcome	Renewal outcome (target column)
customer_id	Unique anonymised customer ID