Pricing Optimization ML Framework

Kelsey Shaun

Nov 11, 2025

Executive Summary

Challenges:

1. The conversion label is defined at the quote level, while the pricing optimization goal operates at the product level (Category, Item).
2. Determine optimal markups that balance conversion probability and profit margin, given the observed strong negative relationship between markup levels and quote conversions.

Solutions:

1. **Quote Conversion Modeling**
   a. Developed a ML model to predict quote conversion probability using strong quote-level and product-level features.
   b. The XGBoost model with SHAP interpretation revealed that:
      i. **Compute and Networking Category is highly price-sensitive**, indicating that small changes in markup significantly affect conversion.
      ii. **Quote Type matters**: Quote-to-Order (QTO) conversions drop faster with higher markups than Normal Quotes, suggesting stronger price competition for QTO buyers.
      iii. **Region**: North American quotes exhibit higher conversion rates, reflecting regional pricing and competition patterns.
      iv. **Timing**: Conversions are time-dependent—quotes published later in the week (Friday to Sunday) show higher conversion rates.
2. **Pricing Optimization Framework**
   a. Built an Expected Margin Optimization Framework that integrates the tuned Conversion XGBoost model with a Grid Search engine.
   b. The framework simulates multiple product-level markup combinations, predicts conversion probabilities, and identifies the combination that maximizes expected margin at the quote level.

Results:

1. Model Performance: XGBoost model achieved **85.3%** accuracy and **AUC = 0.97** on the test set.
2. Optimization Outcome: When applied to a random test quote (simulating a new order):
   - Predicted Conversion Probability increased by **124%.**
   - Expected Margin improved by **197%.**

Learnings:

1. This project test and prove a concept that **product-level features can effectively serve as quote-level predictors, allowing models like XGBoost to capture how within-quote variations influence overall conversion probability.**
2. **SHAP analysis** based on XGBoost model provides transparency in understanding which product categories drive conversion changes.
3. The **Grid Search** approach successfully identifies optimal markup combinations that balance profitability with deal-winning likelihood.

Next steps:

1. **Model Enhancement:** Explore hierarchical or mixed-effects modeling once more item-level data becomes available to better capture nested relationships between quotes and products.
2. **Business Integration**: Collaborate with the sales and pricing teams to:
   - Investigate why current **intra-quote markup variations** are limited.
   - Incorporate **business rules and strategic constraints** into the optimization process for real-world deployment.

Tradeoff between Markup and Conversion Probability

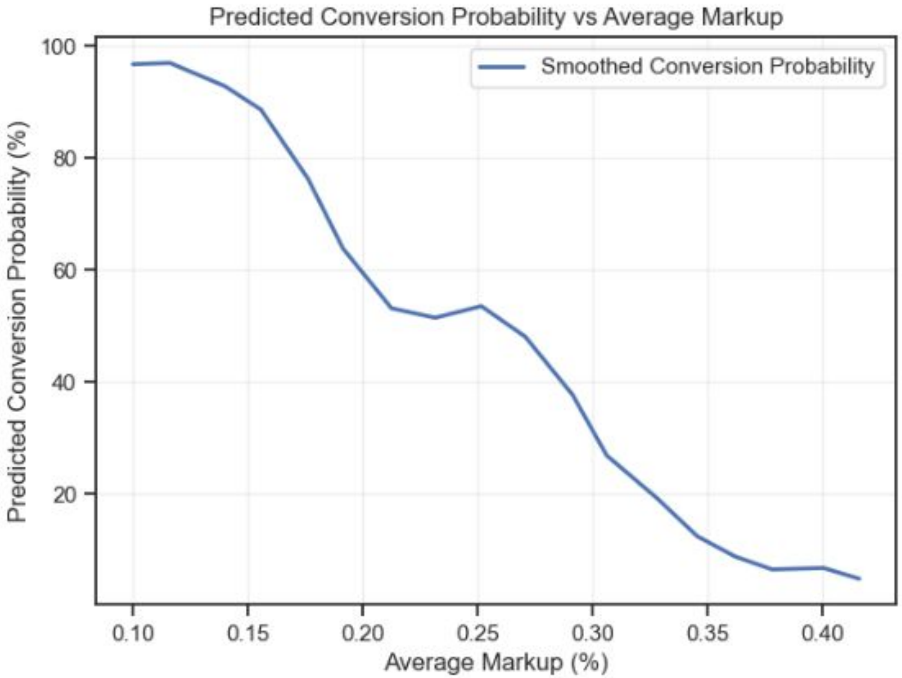
Pricing Tradeoff (logistic curve):

Higher markup brings more margin per deal, but fewer deals convert.

Expected Margin

= Total Cost×Markup×Conversion Probability

This also suggest models capable of capturing nonlinear relationships, such as tree-based, are likely to outperform regression models in conversion prediction



My Plan to Find Solution

Step 1 | Step 2

1. Exploratory Data Analysis (EDA)

- Uncover insights on
  - Markup (Weighted)
  - Product Hierarchy (Categories, Items)
- Learn quote characteristics: Type, Region, Time
- Understand current pricing pattern

2.1 Create Features for Conversion Model

- Inputs:
  - Product factors

    Category Markups (weighted)
  - Quote factors

    categories cost, mix, Items, quantity

    Quote type, region, time
- Output:

  Quote-level conversion probability

2.2 Conversion model Validation and Selection

3. Margin Simulation and Markup Optimize framework

- Inputs:
  - Search space for each Category Markup
  - Pricing Rules and limitations
- Output:
  - Optimal markup for each category
  - Expected Conversion rate & Margin

Conversion Predictors – Quote Factors

Quote Price, Value and Mix – Related Features

Summarized at the Quote Level

- Average Quote Markup (weighted)

$$\text{avg\_markup} = \frac{\sum_i (\text{unit\_cost}_i \cdot \text{qty}_i \cdot \text{markup}_i)}{\sum_i (\text{unit\_cost}_i \cdot \text{qty}_i)}$$

- Markup Standard Deviation
- Markup Range (Max – Min)
- Total Units
- Total Cost
- Average Unit Cost

Quote Characteristics – Related Features

- Quote Type
- Region
- Quote Publish Date - related
  - Month Indicators
  - Quarter Indicator
  - Day-of-Week Indicators
  - Month-End Indicator

Conversion Predictors – Product Factors

Category Price Sensitivity and Composition

Summarized per Quote, per Category

- **Category Average Markup (weighted)**

$$\text{avg\_markup} = \frac{\sum_i (\text{unit\_cost}_i \cdot \text{qty}_i \cdot \text{markup}_i)}{\sum_i (\text{unit\_cost}_i \cdot \text{qty}_i)}$$

- **Category Total Cost**
- Category Item Count
- **Category Cost Ratios:** indicates whether any category dominates a quote’s total cost

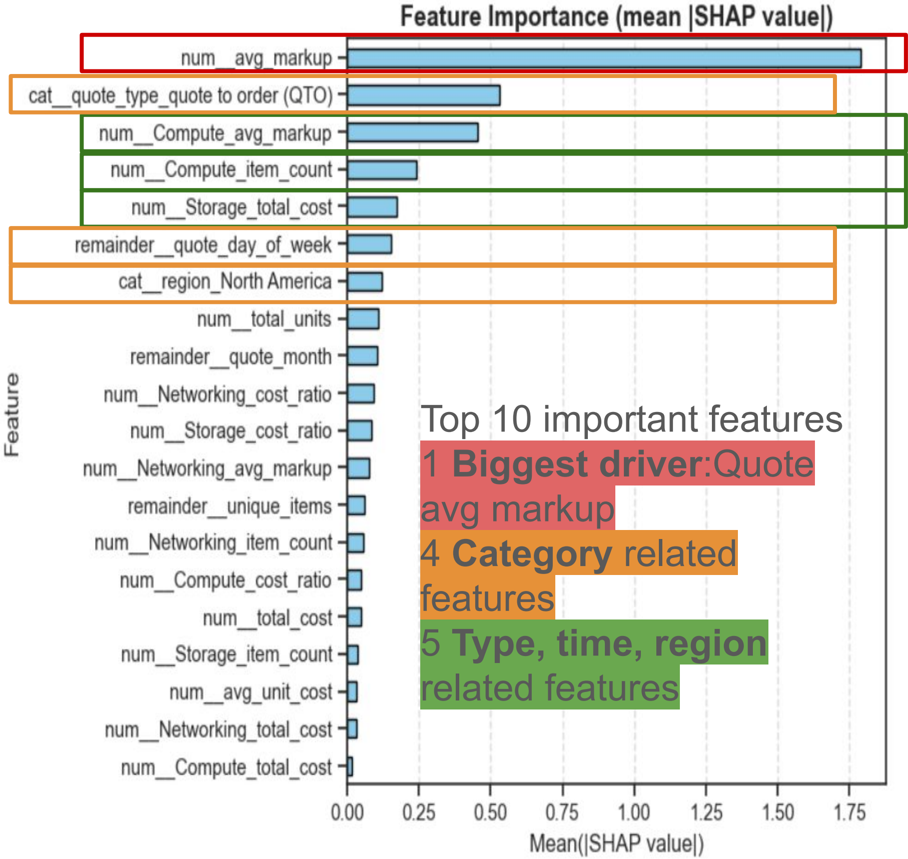
Quote Complexity – Related Features

- **Unique Categories**
- **Unique Items**
- **Units per Item**

Feature Importance (SHAP)

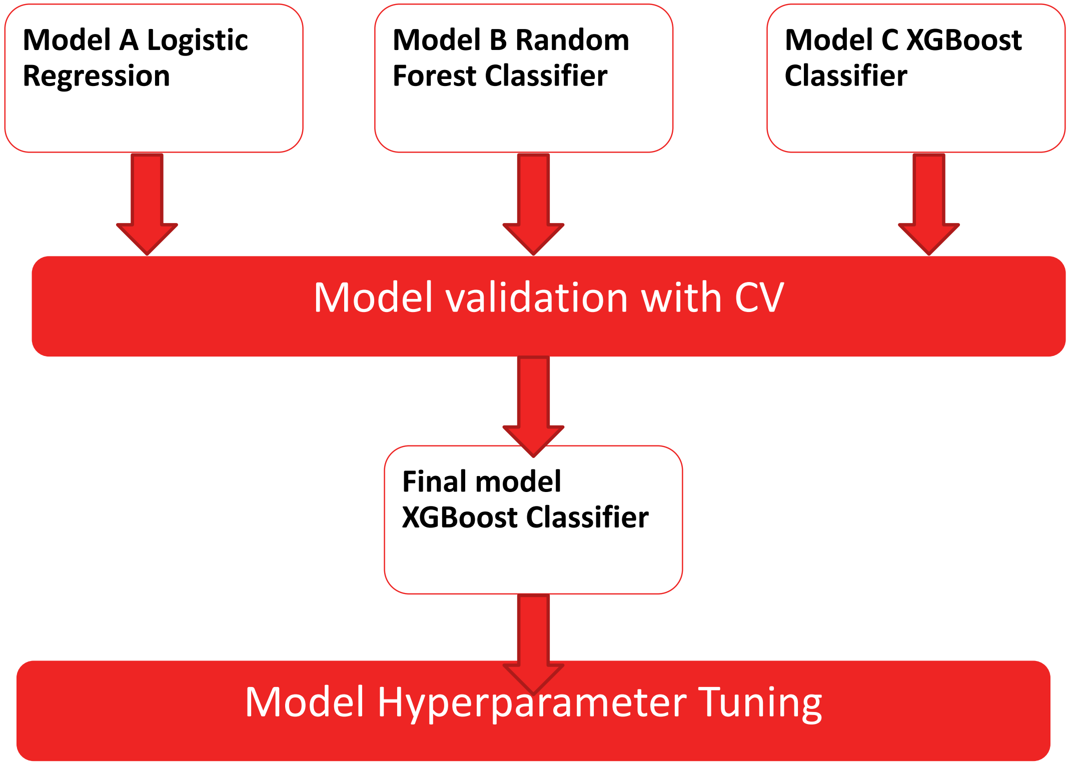
Conversion probability forecasting model (XGBoost)

Rank	Feature	Interpretation
1	num__avg_markup (1.79)	Biggest driver: quotes average markup have the strongest influence on conversion probability.
2	cat__quote_type_quote to order (QTO) (0.53)	Quote type (especially "quote to order") significantly affects likelihood. Possibly signals intent to purchase.
3	num__Compute_avg_markup (0.45)	Compute products, markup variations strongly affect conversion. This indicate pricing sensitivity in this category.
4	num__Compute_item_count (0.24)	Quote's Larger or smaller compute item counts influence decisions
5	num__Storage_total_cost (0.17)	Total storage cost matters. quotes with expensive storage components might be harder to convert.
6	remainder__quote_day_of_week (0.15)	Timing (e.g., weekday quote creation) has moderate influence.
7	cat__region_North America (0.12)	Geographic region impacts pricing and conversion. it indicate regional pricing/competition patterns.
8–20	Other features	Moderate or minor influences that help fine-tune predictions.



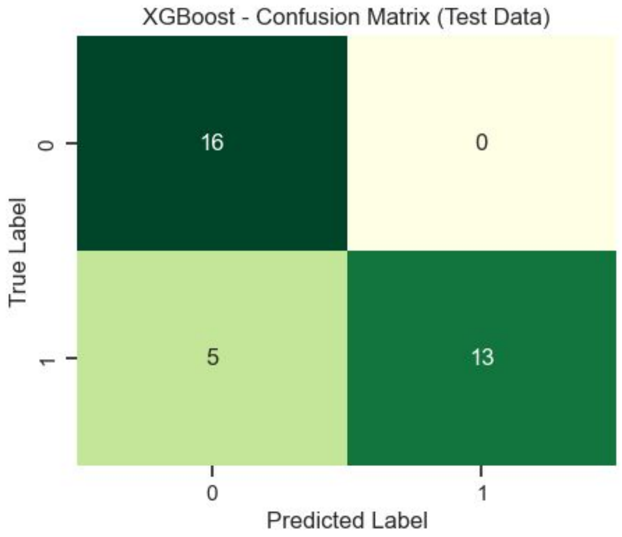
Proposed Model Design – Win model XGBoost Classifier

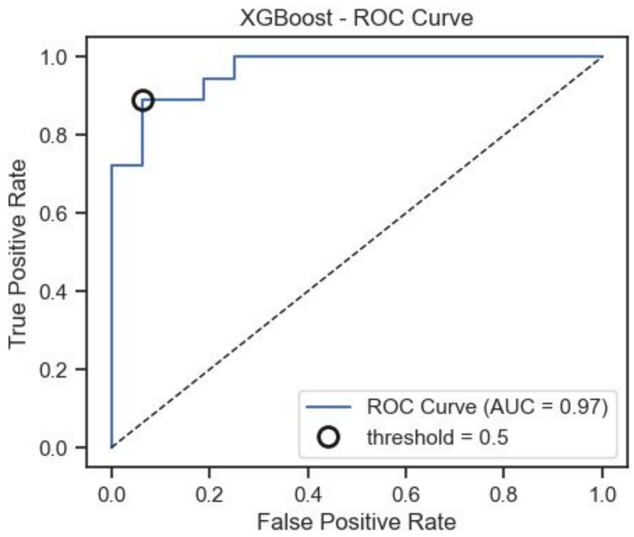
- Among the three, the **XGBoost model** achieves the highest accuracy on the validation data, demonstrating the strongest performance
- Due to the limited sample size, all three models show noticeable gaps between training and testing accuracy, indicating varying degrees of overfitting. As more data becomes available, these gaps are expected to narrow, improving model generalization.



XGBoost Conversion Model Evaluation

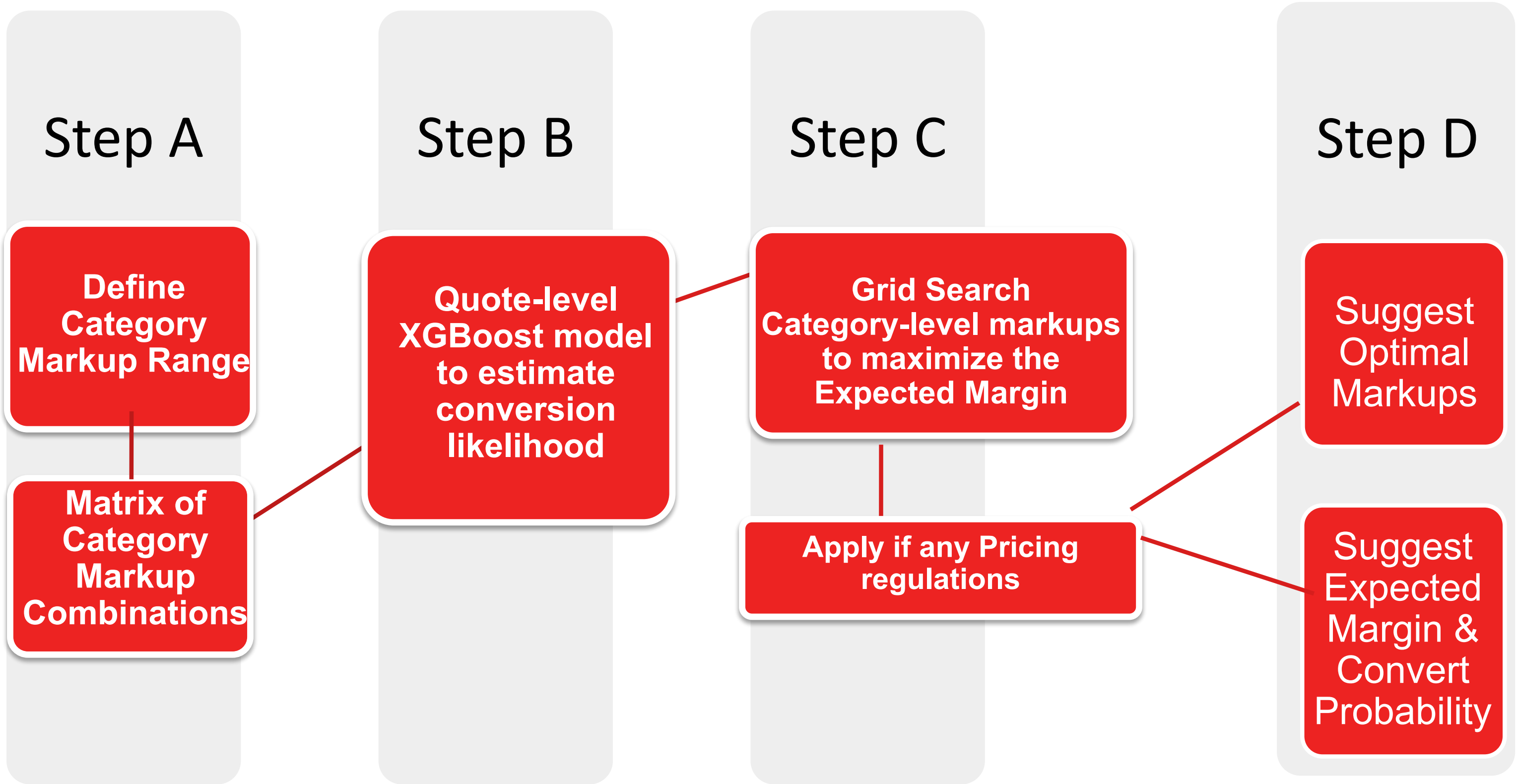
- XGBoost CV mean accuracy: 87.29% ± 4.95%
- XGBoost Train Accuracy: 98.50%
- XGBoost Test Accuracy: 85.29% AUC:97%





Due to the limited sample size, all three models show all models show overfitting.As more data becomes available, these training and testing gaps are expected to narrow, improving model generalization.

Proposed Margin Optimize ML framework



ML Framework Test: Expected Margin & Order Conversion

- Expected Margin: (Total Cost)×[(**Networking Markup**×Networking Cost Ratio)+(**Storage Markup**×Storage Cost Ratio)+(**Compute Markup**×Compute Cost Ratio)]×(**Predicted Quote Conversion Probability**)

$$\text{Expected Margin} = (\text{Total Cost}) \times \Big[(M_N \times R_N) + (M_S \times R_S) + (M_C \times R_C)\Big] \times P_{\text{conv}}$$

- Test Result with Q12017

RMSE	Convert or not	Networking Avg Markup	Storage Avg Markup	Compute Avg Markup	Predicted Conversion Probability	Expected Margin
Real Quote (Markups)	No	0.26	0.27	0.26	27.7%	$728
ML Framework Suggested Quote (Markups)		0.10	0.30	0.10	**90.3%**	**$ 1722**
Improvement					**+226%**	**+137%**

ML Framework Test: Expected Margin & Order Conversion Rate

Quote Q001: Higher Margin With Small Conversion Tradeoff

- The ML framework recommended **higher markups** on Networking and Compute.
- Predicted conversion probability dropped **13%**, but the margin more than **doubled (+118%)**.
- **Insight:** For high-probability quotes (≈98%), the model learns that you can safely raise prices while still keeping the win probability above an acceptable threshold.

This demonstrates the framework's ability to **optimize margin without materially harming conversion** when the baseline probability is already high.

	Networking Avg Markup	Networking Total Cost	Storage Avg Markup	Storage Total Cost	Compute Avg Markup	Compute Total Cost	Predicted Conversion Probability	Expected Margin
Real Quote (Markups) Give same 0.2 markup	0.12	$1280		0	0.10	$780.0	98.3%	$228
ML Framework Suggested Quote (Markups)	**0.30**	$1280		0	**0.25**	$780.0	**86.0%**	**$ 498**
Improvement							**-13%**	**+118%**

ML Framework Test: Expected Margin & Order Conversion Rate

Quote Q002: Dramatic Improvement in Both Conversion & Margin

- I set a default uniform 0.20 markup to mimic how a salesperson might manually assign markups, which resulted in a low 30% probability of winning.
- The ML-recommended strategy adjusted all three markups differently:
    - Slightly decreased Networking
    - Significantly increased Storage (0.40)
    - Lowered Compute (0.10)
- Result:
  Conversion probability jumped from **30%** → **93% (+208%)**
    - Expected margin increased **195%**

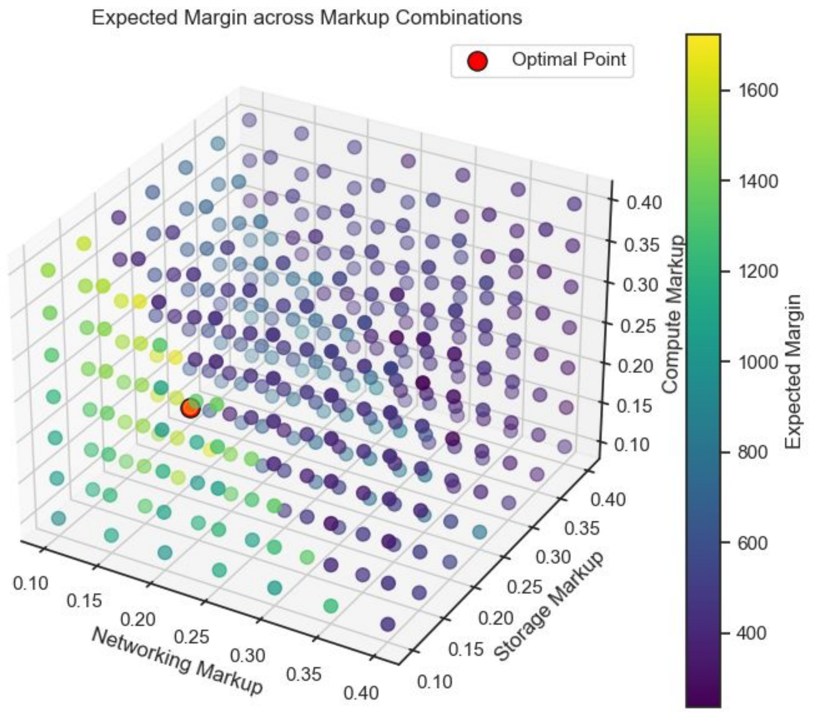
Insight:
The model correctly identifies **category-specific price sensitivity** and rebalances markups to increase the likelihood of winning while still raising the expected dollar margin.
This validates the idea that **category-level elasticities must be considered**, not a single uniform markup.

	Networking Avg Markup	Networking Total Cost	Storage Avg Markup	Storage Total Cost	Compute Avg Markup	Compute Total Cost	Predicted Conversion Probability	Expected Margin
Real Quote (Markups) Suppose same 0.2 markup	0.20	$4210	0.20	$1110.4	0.20	$634.5	30.2%	$360
ML Framework Suggested Quote (Markups)	**0.15**	$4210	**0.40**	$1110.4	**0.10**	$634.5	**93%**	**$ 1061**

Visualize ML Framework

- Visualized using Quote Q12017 from 2025
- Based on weighted markups

pricing sensitivity in compute and networking categories



Deployment Plan – From Prototype to Production

1. Data & Pipeline Setup

1. Connect to source systems to extract historical quotes, item details, and outcomes (convert/not).

2. Build an automated ETL pipeline run daily:

- Cleans data and applies the same feature engineering as the prototype.
- Stores model-ready tables in a secure analytics environment Azure.

2. Model & Optimization Service with API

1. Package the conversion model and Price Optimization FW into a versioned model service
2. Expose an API endpoint:
   - **Input**: quote items
   - **Output**: predicted conversion probability and expected margin, and recommended markup combination.
3. Monthly automated retraining using newly closed quotes + model performance check

3. Integrate Quoting Tools with API

1. Integrate the API into the quoting application

- When a sales builds a quote, the system calls the API to suggest optimized markups by category.
- Show side-by-side: current given markup vs. ML-recommended markup, expected margin, and conversion probability.

2. Log markup recommendations and final human overrides for future learning

4. Monitoring,Rollout, Iterating

1. Start with a pilot group of sales teams.

Run an A/B test (ML-assisted vs. business-as-usual) to measure uplift in win rate and margin.

2. Monitor

model accuracy, average markup, conversion rate, and realized margin.

3.Set up alerts

If performance drifts. Define ownership for reviewing results and updating business rules.

4. Full Roll Out

After successful pilot, roll out to all regions and continue iterating improve markups

Appendix: Python Notebook Content

1. Data ETL and Preparation
Handle missing values using quote-level reference data.

Detect and treat outliers.

Examine markup consistency within quotes to determine if quote-level aggregation is appropriate.

Review feature distributions and apply transformations for non-normal variables.

Convert and standardize date fields.

2. Exploratory Data Analysis (EDA)

Analyze feature distributions and summary statistics.

Examine relationships between key features and target (conversion).

Assess feature importance and correlation patterns.

3. Quote-Level Feature Engineering (Pipeline)

Category & Pricing Features:

Quote value metrics (total cost, total markup, etc.).

Category-based ratios (category share of total quote value, average category markup).

Mix & Diversity Features:

Number of unique categories, unique items, and unit diversity.

Time-Based Features:

Extract day, week, month, and quarter indicators from quote_publish_date.

4. Data Splitting and Preprocessing

Apply time-based train/test split.

Encode categorical variables.

Scale numerical variables using standardization.

5. Model Development and Cross-Validation

Compare multiple models:

- Logistic Regression
- LightGBM
- XGBoost

Evaluate via cross-validation metrics.

6. Model Fine-Tuning

Hyperparameter optimization using cross-validation.

SHAP Feature importance and model interpretability checks.

7. Model Evaluation

Confusion matrix and classification report.

ROC curve and AUC metrics.

Calibration and probability diagnostics.

8. Pricing Optimization Framework

Define markup matrix at category level.

Implement Grid Search to find optimal markups maximizing expected profit.

9. Testing and Visualization

Visualize conversion probability vs markup levels.

Display optimal pricing heatmaps and simulation results.

10. Future Work