Propensity-to-Convert Modeling Report

Prepared for: Kissterra Recruiting Team **Date:** January 2026

1. Executive Summary

- **Goal:**
  - The goal of this project was to develop a machine learning model to predict the probability of lead conversion. This probability score serves as the engine for optimizing Real-Time Bidding (RTB) and ranking high-intent users ("Whales") to maximize ROI.
- **Approach:**
  - We developed a production-ready pipeline comparing a Logistic Regression baseline against a tuned **XGBoost** champion model. Crucially, we implemented a **Temporal Cohort Split** (training on past data and testing on future data) to simulate real-world production conditions and strictly prevent look-ahead bias.
- **Outcome:**
  - **2.5x Efficiency Lift:** The champion model captures **24.6%** of all conversions by targeting just the top 10% of leads (Recall@10%). This represents a significant lift over the random baseline (10%).
  - **Financial Accuracy:** Recognizing that raw model outputs were overconfident, we applied **Isotonic Calibration**. This reduced the probability error (Log Loss) by ~40%, ensuring that predicted probabilities align with actual conversion rates for accurate financial bidding.
  - **Projected ROI:** A financial simulation on held-out data demonstrates an **incremental profit of $28,200 per campaign** (+167% profitability) compared to a random bidding strategy.
- **Risks:**
  - **Data Leakage:** Strict removal of post-event features (e.g., `time_to_contact`, `post_click_revenue`) that are not available at the moment of lead arrival.
  - **Class Imbalance:** Managed via scale-weighting and evaluated using precision-recall metrics (PR-AUC) rather than accuracy.

2. Data Exploration & Understanding

- **2.1 Dataset Overview:** The dataset comprises **60,000 lead records** with 27 features, spanning 299 days (Jan 1st, 2025 – Oct 27th, 2025).

    - **Structure:** It includes a mix of numerical attributes (e.g., `bid`, `budget`), categorical metadata (`channel`, `device`), and timestamps.
    - **Temporal Continuity:** The continuous time range enables us to simulate a realistic production environment using a **Time-Based Train/Test Split** rather than random shuffling.
- **2.2 Data Quality:**
    - **Duplicate Handling:** We identified 24 duplicate `lead_id` entries (<0.04%). Investigation confirmed these are valid **Retargeting Events** (same user, different time/channel) rather than data errors. **Decision:** All records were retained to capture the full marketing funnel history; removing them would bias the model against multi-touch paths.
    - **Missing Values (Cold Start):** `cvr_hist` is missing in ~6% of records, representing new inventory. **Strategy:** Impute with the global mean to preserve data density.
    - **Type Casting:** Temporal features (`created_at`) were cast to datetime objects to facilitate feature engineering (Hour, Day of Week).
- **2.3 Leakage Analysis:** We performed a target-stratified missingness analysis and identified three types of data leakage. **Decision:** The following features will be **excluded** to prevent look-ahead bias:
    - **Direct Leakage:** `converted_at` and `conversion_delay_minutes` (100% missing for non-converters). These are textbook labels, not predictors.
    - **Disguised Leakage:** `post_click_revenue`: While non-null (0 for non-converters), it is a deterministic proxy for the label.
    - **Operational Leakage:** `time_to_contact_min`, `call_attempts`, and `crm_status`. These are downstream events that occur *after* the bid is placed and are unavailable at the moment of prediction.
- **2.4 Feature Analysis:**
    - **Class Imbalance:** The overall conversion rate is 13.53% (~1:6.4 ratio).
        - *Implication:* Standard Accuracy is a misleading metric (a dummy model achieves 86.5%). We will prioritize **Recall@10%** and **PR-AUC** for evaluation.
    - **Channel Signal:** Strong variance was observed by channel, with **Google Search (~18%)** significantly outperforming **Google Display (~9%)**, identifying `channel` as a high-value predictor.
    - **Skewed Features:** `bid` and `prior_site_visits` exhibit heavy right-skew distributions with extreme outliers. **Strategy:** Apply **Log-Transformation (`np.log1p`)** to normalize variance.
    - **Feature Validation (`lead_score_raw`):** Investigated for potential leakage. The weak correlation (0.11) and overlapping distributions confirm this is a valid upstream feature (likely 3rd-party enrichment) safe for modeling.

- **2.5 Temporal Analysis: Temporal Stability:** A time-series analysis reveals a stable process with no significant drift.
  - **Volume:** Daily lead volume fluctuates naturally around the mean without anomalous drops.
  - **Conversion Rate:** Maintains a low standard deviation (0.0245) around the mean.
  - **Conclusion:** Historical data is a reliable predictor of future behavior.
- **2.6 EDA Conclusions & Assumptions:**
  - **What patterns surprised me?**
    - **Channel Variance:** The 2-3x performance gap between channels (Search vs. Display) was higher than expected, suggesting `channel` is a potential source of bias we must monitor.
    - **Zero-Inflation:** `prior_site_visits` is heavily concentrated at 0, suggesting that binning this feature (New vs. Returning) may yield a better signal than raw counts.
  - **Which features raised red flags?**
    - **Direct Leakage:** `converted_at` and `conversion_delay` are absent for non-converters.
    - **Implicit Leakage:** `post_click_revenue` (0 for non-converters) and `crm_status` (e.g., "won") are deterministic outcomes, not predictors.
    - **Look-Ahead Bias:** Operational metrics like `time_to_contact` are generated post-bidding and must be excluded.
  - **Modeling Assumptions:**
    - **Data Availability:** We assume all metadata (`bid`, `device`, `campaign_id`) is available via API at the exact moment of `created_at`.
    - **Evaluation Realism:** Due to the existence of retargeting (duplicates), we assume that a **Grouped Split** (keeping a user's history together) is required to prevent leakage between train and test sets.

3. Data Preprocessing & Feature Engineering

- **3.1 Feature Selection & Pruning** Instead of relying on "blind" automated selection methods (like RFE) at this baseline stage, we applied a Domain-Driven Feature Selection process. We rigorously filtered features based on three criteria: Availability, Quality, and Redundancy.
  - **Action:** Removed 7 features identified as downstream events (e.g., `crm_status`, `post_click_revenue`) or direct labels (`converted_at`).

    - **Result:** Reduced feature space from 28 to **21 predictive features**, ensuring the model only trains on information available at the moment of bidding.
- **3.2 Cardinality Management** We analyzed high-cardinality categorical features to prevent the "Curse of Dimensionality
    - **`creative_id`:** Found to have **60,000 unique values** (100% cardinality). This confirms it acts as a unique identifier rather than a grouping feature. **Action:** Dropped.
    - **`adgroup_id`:** High cardinality (1,800 unique values). Including this would create an extremely sparse matrix. **Action:** Dropped for the baseline model to reduce noise.
    - **`campaign_id`:** Medium cardinality (280). **Action:** We will retain the Top-50 most frequent campaigns and bundle the rest into an "Other" category to capture the main signal without exploding feature space.
    - **`kw_group`:** Low cardinality (6). **Action:** Retained for One-Hot Encoding.
- **3.3 Transformations:** We engineered features to capture specific marketing behaviors and normalize distributions:
    - **Log-Transformation:** Applied `log1p` to `bid` and `budget`. This compresses the long tail of high-value outliers, helping the linear baseline model converge and improving tree-split efficiency.
    - **Bid Aggressiveness (`bid_to_budget`):** A custom ratio feature representing how "aggressive" a strategy is (high bid relative to budget). This captures intent that raw values alone might miss.
    - **Interaction Effects:** Created `channel_device` (e.g., "google_search_mobile") to capture the combined effect of platform and user context, which often correlates with conversion intent.
    - **Temporal Features:** Extracted `is_business_hours` (9-17) and `is_weekend`. B2B leads often convert better during business hours, while B2C patterns may spike on weekends.
    - **Zero-Inflation Handling:** Converted `prior_site_visits` (highly skewed towards 0) into a binary `is_new_visitor` flag to simplify the signal for the model.
- **3.4 Dimensionality Reduction (Categorical Encoding)** To manage the trade-off between information loss and the "Curse of Dimensionality":
    - **Top-K Encoding:** For `campaign_id` (280 categories), we retained the **Top-50** most frequent campaigns and grouped the tail (230+) into an "Other" category. This preserves the signal from major campaigns while preventing matrix sparsity.
    - **One-Hot Encoding:** Applied to low-cardinality features (`channel`, `state`, `kw_group`), resulting in a final feature space of **120 predictors**.

    - **Leakage Prevention in Encoding:** We deliberately chose **Frequency-Based Grouping** (Top-K) for `campaign_id` rather than Target Encoding (Mean Encoding). While Target Encoding can capture more signal, it carries a high risk of overfitting and leakage in time-series data if not handled with complex cross-validation. Our approach prioritizes robustness.
- **3.5 Missing Value Imputation (Cold Start)** We addressed the ~6% missingness in `cvr_hist` (Cold Start inventory) using a hybrid approach:
    - **Binary Flagging:** Created an explicit `cvr_hist_missing` feature to allow the model to learn the specific risk/value associated with new inventory.
    - **Mean Imputation:** Filled missing values with the global mean rather than zero. Zero-imputation would bias the model against new campaigns (treating them as "failed"), whereas mean-imputation provides a neutral baseline for exploration.
- **3.6 Feature Impact Expectations** Based on our domain analysis, we hypothesize that `channel`, `bid_to_budget`, and `is_new_visitor` will be high-impact drivers. *A quantitative analysis of the most helpful features is provided in Section 4 (Model Evaluation) under Feature Importance.*

4. Outlier Handling & Data Quality

- **4.1 Definition & Methodology:** We define an outlier as an extreme value that deviates significantly from the norm yet remains biologically/physically possible. To identify these**,** we compared the **99th percentile (P99)** to the **Maximum value** for key features. Features where `Max > 1.5 * P99` were flagged.
    - *Result:* Significant right-skew was observed in `bid` and `budget`, confirming the presence of extreme values (e.g., `bid_to_budget` ratio spiking to 1.08).
- **4.2 Business Interpretation: What do extreme values mean?** Rather than data errors, these outliers represent "Whales"**:**
    - **High Spenders:** Campaigns targeting high-value enterprise customers.
    - **High Intent:** Leads with exceptional historical CTR/CVR, indicating premium traffic.
    - **Aggressive Strategy:** The high `bid_to_budget` ratios likely reflect intentional "blitz" strategies during testing phases.
- **4.3 Risk Analysis: Risks of Outlier Removal** We identified three critical risks associated with removing these data points:

    - **Selection Bias:** Removing high-spend campaigns would cause the model to systematically under-value our most profitable segments ("Blindness to Whales").
    - **Deployment Mismatch:** If the model is not trained on high bids, it will yield unreliable predictions when encountered in production.
    - **Revenue Loss:** High-bid leads often correlate with higher Lifetime Value (LTV); filtering them out directly impacts the estimated ROI.
    - **Final Decision:**
        - **Business Outliers (Whales): KEPT.** We chose NOT to remove valid extreme values (e.g., high bids) to avoid Selection Bias and revenue loss from premium segments. We rely on Log-Transformations to manage their scale.
        - **Impossible Values (Data Errors): CAPPED.** We identified 21 records with `lead_score_raw` outside the valid [0-100] range (e.g., -5, 105). Unlike whales, these violate business logic. We applied Clipping to cap them at the boundaries (0 and 100), correcting the error while preserving the "high/low quality" signal.

5. Model Selection & Evaluation

- **5.1 Data Splitting Methodology** To faithfully simulate a production environment, we rejected a standard random split in favor of a Lead-Level Temporal Split.
    - **The Problem:** Standard random splitting introduces two risks:
        - **Look-ahead Bias:** Using future data to predict the past.
        - **Identity Leakage:** Since duplicates exist (retargeting), a random split might place a user's first visit in the Training set and their second visit in the Test set, allowing the model to "memorize" the user rather than learn generalized patterns.
    - **The Solution:** We ordered all unique leads by their first arrival time (T_0) and strictly partitioned them into three sets:
        - **Training (70%):** Jan 1st – July 29th (Used for model fitting).
        - **Validation (15%):** July 29th – Sept 12th (Used for hyperparameter tuning & threshold selection).
        - **Testing (15%):** Sept 12th – Oct 27th (Held-out "future" data for final evaluation).
- **5.2 Split Verification** We verified the integrity of the split to ensure no sampling bias occurred:
    - **Consistency:** The conversion rate remained stable across splits (Train: 13.6%, Test: 13.4%), confirming that the time periods are comparable.
    - **Independence:** Validated **0% overlap** of `lead_id` between sets.

- **5.3 Evaluation Metrics Strategy:** Given the class imbalance (13.5%) and the business goal (bidding), we defined a hierarchy of metrics:
    - **Primary Metric: PR-AUC (Average Precision):** unlike ROC-AUC, PR-AUC focuses heavily on the minority class (converters). It is the most honest metric for imbalanced datasets where "False Positives" have a real cost (wasted budget).
    - **Calibration Metric: Log Loss:** Since the model's output will be used to calculate expected value (E[V] = P(Conv) X Value), the probabilities must be calibrated. Log Loss penalizes confident errors, ensuring the model doesn't just rank well but outputs realistic probabilities.
    - **Operational KPI: Recall @ Top 10%:** A business-centric metric. It answers: *"If we only bid on the top 10% of leads, what fraction of total conversions would we capture?"* This translates model performance directly into potential revenue efficiency.
- **5.4 Baseline Model: Logistic Regression** We established a linear baseline to benchmark performance and verify signal stability.
    - **Performance:** The model achieved a **ROC-AUC of ~0.71** and **Recall@10% of ~23%** on the held-out Test set.
    - **Stability:** Metrics showed negligible variance between Train (0.716) and Validation (0.700) sets, confirming that the model generalizes well over time and isn't overfitting.
    - **Lift:** The baseline demonstrated a **2.3x lift** over random guessing (23.4% capture rate in the top decile vs. expected 10%), proving that the engineered features contain significant predictive power even in a linear setting.
    - **Initial Feature Insights (Linear):** The coefficients validated key business assumptions:
        - **Intent Matters:** `kw_group_brand` was the strongest positive driver, while `low_intent` was the strongest negative.
        - **Friction Kills:** `form_fields_count` showed a negative correlation, confirming that longer forms reduce conversion rates.
        - **Lead Scoring Works:** `lead_score_raw` had a strong positive signal, validating the external scoring vendor.
- **5.5 Initial Non-Linear Model: Initial XGBoost Results (Overfitting Diagnosis)** We trained an initial XGBoost model using standard parameters (`max_depth=5`, `scale_pos_weight=6.34`).
    - **Result:** The model exhibited significant **overfitting**. While it achieved a high ROC-AUC of **0.84** on the training set, performance dropped to **0.69** on the test set.
    - **Comparison:** The initial non-linear model failed to beat the simple linear baseline (0.71).

  - **Diagnosis:** The aggressive class weighting combined with the tree depth allowed the model to memorize specific historical patterns that did not persist into the future (Temporal Decay). This necessitates a **Regularization Phase**.
  -
- **5.6 Hyperparameter Tuning Insights** Responding to specific assignment questions:
  - **Impactful Parameters:** Our experiments showed that `max_depth` and `min_child_weight` had the most significant impact on model performance. Restricting `max_depth` to 3 (down from 5) was the key factor in eliminating the overfitting observed in the initial model.
  - **Diminishing Returns:** We observed a clear plateau. The gap between the best configuration and the average of the top-5 configurations was negligible (<0.003 PR-AUC). Further tuning yielded marginal gains that did not justify the computational cost or the risk of overfitting the validation set.
  - **Stability vs. Performance:** Introducing row and column subsampling (`subsample=0.9`) successfully reduced the variance across validation folds (lower standard deviation). We accepted a minor trade-off in absolute PR-AUC (~0.002 drop) in exchange for this increased stability, which is preferable for a production deployment.
  - While we utilized `TimeSeriesSplit` for hyperparameter tuning, we acknowledge that standard temporal splitting allows for repeat visitors (same `lead_id`) to appear across folds. However, the integrity of our final results is guaranteed by the **Strict Temporal Holdout (Test Set)** defined in Section 5.1, which enforced a hard separation of unique users. The final metrics reported on the Test set are therefore free of any leakage bias
- **5.7 Final Tuned Model Performance (The "Fixed" Model)** Following the hyperparameter analysis, we retrained the XGBoost model using the optimal configuration (`max_depth=3`, `subsample=0.9`, `min_child_weight=3`).
  - **Generalization Success:** The tuning successfully eliminated the overfitting. The large divergence between Train and Test performance was closed, resulting in a stable model.
  - **Performance Metrics:** On the held-out Test set, the tuned model achieved:
    - **ROC-AUC: 0.700** (Slightly lower than baseline, indicating noise reduction).
    - **PR-AUC: 0.265** (Improved over the initial overfit model).
    - **Recall@10%:** 24.6% (The highest capture rate among all experiments).

    - **Outcome:** Unlike the initial model, which "memorized" the past, this regularized model effectively identified the high-quality segments, outperforming the baseline in the top decile.
- **5.8 Final Model Selection & Trade-off Analysis** *Responding to specific assignment questions:*
    - **Model Choice & Trade-offs:** We deliberately chose two contrasting architectures to balance predictive power with explainability:
        - **Logistic Regression (Baseline):** Chosen for its transparency and speed. The trade-off is its inability to capture non-linear feature interactions (e.g., specific device-hour combinations).
        - **XGBoost (Challenger):** Chosen for its ability to model complex, non-linear boundaries. The trade-off is higher computational cost and "Black Box" opacity (mitigated via SHAP).
    - **Final Decision:** Post-tuning, we faced an interesting dilemma:
        - **Logistic Regression** achieved a slightly higher global ROC-AUC (0.707 vs 0.700), indicating strong linear signal dominance.
        - **Tuned XGBoost** achieved a superior **Recall@10% (24.6% vs 23.4%)**.
    - **Conclusion:** Since our business goal is Bidding Optimization (targeting the highest probability leads), capturing more converters in the top decile is more valuable than global ranking accuracy. Therefore, we selected the Tuned XGBoost as the final production model, accepting the minor complexity trade-off for the 1.2% gain in recall among top leads.
- **5.9 Probability Calibration (Isotonic Regression)** *Since the model's output will drive financial bidding decisions (Bid = Predicted_Prob X Value), accurate probabilities are as critical as accurate ranking.*
    - **The Issue:** The uncalibrated XGBoost model significantly **overestimated** probabilities (predicting ~90% for segments that only converted at ~45%), which would lead to severe overbidding and budget waste.
    - **The Solution:** We applied **Isotonic Regression** using the held-out Validation set to map the model's raw scores to true probabilities.
    - **Results:**
        - **Log Loss:** Reduced dramatically from **0.6052** to **0.3692** (Lower is better).
        - **Reliability:** The calibration curve aligned significantly closer to the diagonal, correcting the overconfidence bias.
    - **Impact:** This ensures that predicted probabilities now reflect true conversion rates, enabling a safe and profitable bidding strategy.

6. Business Interpretation & Deployment

Interpretability & Business Recommendations: To ensure the "Black Box" XGBoost model is transparent and actionable, we utilized SHAP (SHapley Additive exPlanations) to derive the marginal contribution of each feature.

- **6.1 Key Drivers of Conversion**
  - **Search Intent (Primary Driver):** The model relies heavily on keyword categorization. Users entering via "Brand" terms have the highest propensity to convert, while "Generic Low Intent" terms act as a strong negative signal.
  - **User Experience (Friction):** We observed a distinct negative relationship between `form_fields_count` and conversion probability. Leads faced with longer forms are statistically less likely to convert.
  - **Lead Quality:** The external `lead_score_raw` validated its utility, showing a strong positive correlation with the model's predictions.
- **6.2 Strategic Recommendations** Based on these findings, we propose the following actions:
  - **Bidding Strategy:** Implement the calibrated XGBoost model to aggressively bid on top-decile leads (Top 10%), specifically targeting the `kw_group_brand` segment where conversion probability is maximized.
  - **UX Optimization:** A/B test a shorter landing page form. The SHAP analysis suggests that reducing the number of fields will directly improve conversion rates (`form_fields_count` impact).
  - **Budget Allocation:** Shift budget away from `generic_low_intent` keywords, as the model identifies them as "budget wasters" with consistently low propensity scores.
  - **Projected Business Impact (Simulation):** A financial simulation on the held-out test set demonstrates the model's ROI. Given a fixed budget to bid on 2,000 leads:
    - A random selection strategy yields **$16,800** in profit.
    - The XGBoost-guided strategy yields **$45,000** in profit.
    - **Result:** The model delivers an **incremental value of $28,200 per campaign**, marking a **167% increase in profitability** compared to the baseline.
  - **Sensitivity Note:** This simulation assumes a constant CPA of $5 and LTV of $100. In a real-world scenario, we would perform a **Sensitivity Analysis**, varying these parameters (e.g., CPA $2-$10) to determine the profitability break-even point for the model.
- **6.3 Deployment Strategy & Risk Management:** *Responding to specific assignment questions regarding deployment.*
  - ***Operational Implementation:*** *To integrate this model into a live bidding system, we propose a* ***Real-Time Scoring API****:*

        - ***Workflow:*** *When a user lands on the site (or a bid request is received), the backend sends the lead attributes (device, time, referral source) to the Model API.*
        - ***Decision:*** *The model returns a probability (P_{conv}).*
        - ***Action:*** *The bidding engine calculates the expected value (Bid = P_{conv} X LTV). If Bid > Cost, we place the bid. This replaces the current static rule-based logic.*
        - ***Production Pipeline:*** *For the live API, the current Pandas-based preprocessing steps would be refactored into a `sklearn.Pipeline` with `ColumnTransformer`. This ensures that the same transformations (e.g., Log/Imputation) applied during training are automatically applied to incoming inference requests, eliminating training-serving skew.*
- **6.4 Risk Assessment (Failure Modes)** We identified three critical failure modes that require mitigation:
    - **Feedback Loops:** If we stop bidding on "Low Intent" leads entirely, we stop gathering data on them. Over time, the model will "forget" why they were bad (Blind Spot). *Mitigation:* Reserve 5% of traffic for random exploration.
    - **Feature Drift:** Marketing campaigns change rapidly. If a new campaign launches with a new `campaign_id` the model hasn't seen, it may default to a safe (low) score, missing opportunities.
    - **Tracking Failures:** If the conversion pixel breaks, the model will see 0 conversions and drastically downgrade all predictions within hours.
- **6.5 Production Monitoring Plan** To ensure stability, we will monitor three layers of metrics:
    - **Data Integrity:** Alert if the percentage of missing values in `lead_score` spikes or if unknown `campaign_ids` exceed 5%.
    - **Model Drift:** Track the distribution of predicted probabilities daily. A sudden shift (e.g., mean probability dropping from 0.13 to 0.05) triggers an alert.
    - **Business KPI:** Monitor the actual **Calibration Curve** in production weekly. If the predicted 20% probability cohort starts converting at only 10%, immediate retraining is required.

7. Conclusion & Final Verdict

- **7.1 Project Summary:** We successfully developed a machine learning framework to optimize the bidding strategy for lead acquisition. By transitioning from a linear baseline to a **Tuned & Calibrated XGBoost model,** we achieved a robust solution that balances predictive accuracy with probability reliability.

- **7.2 Key Achievements**
  - **Precision:** The final model captures **24.6% of all converters** within the top 10% of leads, offering a 1.2% lift over the baseline in the most critical segment.
  - **Reliability:** Isotonic calibration reduced probability error (Log Loss) by **~40%**, ensuring that bid prices accurately reflect expected value.
  - **Value:** A simulation on held-out data projects an **incremental profit of $28,200 per campaign** (+167% ROI) compared to random bidding.
- **7.3 Final Recommendation** We recommend **deploying the Calibrated XGBoost model** into production for a live A/B test. The model should effectively filter out low-intent traffic (identified via SHAP analysis) and aggressively target the high-value "Brand" and "US-based" segments, driving immediate efficiency gains for the marketing budget.