An AI-Driven Account Prioritization Scoring Model

Technical Report

Prepared by: Muhammad Yahya

Business Systems Internship

Abstract

This paper presents the development of an AI-driven account scoring model aimed at automating the prioritization of sales leads. The solution addresses the manual and inefficient process of lead ranking by leveraging a predictive model built upon a comprehensive, pre-processed dataset. This combined dataset, derived from multiple sources, integrates firmographic, intent, and historical sales opportunity information. The model was trained using a carefully selected set of features chosen for their predictive power and to ensure a leakage-free environment. A baseline XGBoost classifier was implemented, showcasing its ability to handle data imbalance and complex interactions. The initial results demonstrate high precision in identifying promising accounts, affirming the viability of an automated approach. This research offers a strategic impact by enabling sales teams to focus on top accounts and thereby boost overall conversion efficiency.

Contents

List of Figures

1 Introduction

Modern enterprise sales teams operate in an environment defined by rapid digital engagement, increasingly fragmented buyer journeys, and expanding pools of potential accounts. As organizations scale, the task of determining which prospects deserve attention becomes progressively more complex. Manual prioritization methods—typically based on subjective judgment, spreadsheet scoring, or informal behavioral cues—struggle to keep pace with the volume and velocity of available data. This mismatch results in misallocated sales effort, delayed follow-ups, and opportunities lost to more proactive competitors.

The core challenge is not the scarcity of data but the inability to synthesize it into an actionable signal. Sales teams may track dozens of attributes, including firmographic characteristics, intent indicators, marketing engagement footprints, and early-stage sales activity. However, without an automated prioritization layer, these signals remain disconnected. Reps are forced to manually scan systems, compare inconsistent signals, and make intuition-driven decisions—a process prone to bias, inconsistency, and inefficiency.

To address this challenge, this project develops an AI-driven account scoring model capable of learning from historical patterns to identify the accounts most likely to convert. The model integrates multi-source prospect data sourced from internal CRM and marketing platforms to construct a unified representation of each account prior to conversion. By enforcing strict data-leakage prevention through temporal cutoff methods, the model ensures that predictions rely only on the information that would have been available at the time of decision-making.

The ultimate objective is to deliver a scalable, data-first system that replaces manual ranking with a predictive score, allowing sales teams to allocate time and resources toward high-quality opportunities. This report documents the methodological rigor required to build the first iteration of such a model, including dataset construction, feature engineering, model training, evaluation, and strategic implications for future deployment.

2 Literature Review

The development of automated account prioritization systems draws from several academic and industry domains, including traditional lead scoring, machine learning for customer modeling, multi-signal integration, and modern predictive analytics frameworks. This literature review synthesizes these areas to contextualize the model developed in this study.

2.1 2.1 Traditional Lead Scoring Approaches

Early lead scoring models emerged as rule-based systems, commonly deployed in CRM and marketing automation platforms. These methods relied on assigning points to categorical and demographic attributes such as job title, company size, industry, or explicit marketing actions (e.g., downloading a whitepaper, attending a webinar). While appealing for their transparency and simplicity, these models suffer from several well-documented limitations.

First, rule-based scoring lacks adaptability. Scores remain static unless manually updated, making them vulnerable to shifts in market conditions, buyer behavior, or industry cycles. Second, these frameworks assume additive and linear relationships among signals, ignoring interaction effects and nonlinear behavioral patterns that are often essential predictors of conversion outcomes. Third, manual scoring introduces subjective bias, as weightings and point assignments reflect internal assumptions rather than empirically validated relationships. Finally, traditional methods are prone to the "cold start" problem: new prospects with limited engagement data often receive low or incomplete scores despite potentially high buyer intent.

As organizations scale and engagement volumes increase, the inadequacy of fixed scoring rules becomes more pronounced, motivating a shift toward data-driven models capable of learning from empirical patterns rather than encoded assumptions.

2.2 2.2 Machine Learning for Lead and Opportunity Prediction

With the proliferation of CRM and digital engagement data, machine learning (ML) techniques have been increasingly applied to lead qualification and opportunity scoring. Research demonstrates that ML models outperform rule-based approaches by modeling complex relationships across structured and behavioral features. Gradient-boosting algorithms—such as XGBoost, LightGBM, and CatBoost—are especially suitable for tabular datasets with mixed feature types, missing values, and nonlinear interactions.

Studies in B2B sales analytics highlight the utility of supervised classification frameworks for predicting outcomes such as opportunity success, lead qualification likelihood, and customer churn. These models leverage attributes across several categories: firmographics (e.g., industry, region), intent and behavioral signals (e.g., content engagement, third-party

interest data), and sales-process indicators (e.g., stage movement, rep activity). Empirical evidence consistently shows that models integrating multiple data sources achieve stronger predictive power than single-signal approaches.

However, leakage prevention emerges as a critical methodological consideration. Without temporal validation or careful control of information timing, models risk capturing signals that would not be available at real prediction time, resulting in artificially inflated performance estimates. This challenge underscores the importance of designing leakage-free training pipelines.

2.3 2.3 Multi-Signal Integration and Predictive Sales Intelligence

Recent developments in predictive sales intelligence systems emphasize the fusion of heterogeneous signals to create "composite intent" or "propensity" scores. Industry solutions increasingly integrate:

- firmographic and demographic data,
- third-party intent indicators,
- marketing automation activity,
- website and content engagement logs,
- CRM sales stage movement,
- past opportunity histories.

This multi-signal approach mirrors methodologies used in recommender systems and credit scoring, where predictive power arises from combining disparate feature groups. Research shows that cross-signal interactions—e.g., a spike in web traffic combined with favorable firmographics—often outperform any single predictive input.

These models also face challenges in class imbalance, as the proportion of converted accounts is typically small relative to the total prospect universe. Techniques such as threshold tuning, class weighting, and calibrated decision functions are commonly employed to improve recall while maintaining precision.

2.4 2.4 Positioning of This Work

This project contributes to the predictive lead-scoring literature by constructing a unified dataset drawn from multiple CRM and marketing data sources, enforcing strict time-based leakage prevention, and training a gradient-boosting model tailored to class-imbalanced conversion prediction. Unlike traditional rule-based scoring, this model learns from empirical

patterns across both static and temporal behavioral features. Unlike many academic studies, which rely on pre-cleaned datasets, this project emphasizes the engineering considerations required to synthesize raw enterprise data into a structured, model-ready form.

The work also introduces an interpretable feature-engineering layer, including temporal recency metrics and a custom "StageBooster" mechanism to adjust the influence of sales-stage signals. These elements collectively position the model as an applied case study in building scalable, data-driven account prioritization systems.

3 Data Sources and Preparation

The development of a reliable account scoring model requires a dataset that accurately captures the information available to sales and marketing teams prior to conversion. Constructing such a dataset from enterprise systems—particularly CRM and intent platforms—requires careful engineering to merge disparate records, align temporal sequences, and remove leakage-prone fields. This section details the raw data sources, the consolidation process, and the key engineering decisions that shaped the final model-ready dataset.

3.1 3.1 Overview of Source Systems

The dataset for this project was derived entirely from internal Salesforce exports and supporting marketing-intent files. Six core datasets formed the foundation of the unified account-level table:

- **AcctoOpp:** A relational mapping between accounts and their associated opportunities, including key timestamps such as opportunity creation dates.
- **Oppo:** Opportunity-level fields capturing structured sales-process information (stage, amount, close dates, win/loss outcomes).
- **Firmo:** Firmographic attributes including industry, vertical, employee band, revenue band, and billing country.
- **Intent:** Third-party intent scores, topics, and associated timestamps reflecting external-interest signals.
- **custodate:** Customer lifecycle data, including account creation date and customer conversion timestamp.
- **stagedate:** Detailed timestamps representing the chronological sequence of early-stage engagement activity.

Individually, these sources provide limited insight; however, when combined, they offer a rich depiction of pre-conversion behavior. Integrating these files into a single analytical dataset required resolving structural inconsistencies, aligning time fields, deduplicating ambiguous records, and harmonizing categorical encoding across systems.

3.2 3.2 Data Consolidation Workflow

The data consolidation step was the most technically intensive segment of the project, requiring dozens of transformations to resolve mismatched keys, conflicting time zones, duplicate entries, and heterogeneous formats.

The consolidation workflow can be summarized as follows:

1. **Normalize Account Keys:** Ensure all tables reference accounts using a consistent unique identifier; remove malformed or orphaned keys.
2. **Merge Hierarchical Sources:** Left-join firmographic, intent, and opportunity data at the account level.
3. **Align Timestamps:** Convert all date fields into a consistent format and correct time-zone inconsistencies.
4. **Derive Earliest Engagement:** Compute the first recorded engagement date across multi-signal sources.
5. **De-duplicate Records:** Remove duplicate opportunity associations, repeated intent events, and redundant lifecycle entries.
6. **Remove Post-Conversion Fields:** Exclude any event, score, or timestamp that occurred after the account became a customer.

The process concluded with the creation of a master dataset where each row represented a unique account enriched with 44 pre-conversion attributes drawn from multiple systems.

3.3 3.3 Leakage Prevention via Dynamic Cutoff Logic

Preventing data leakage is essential when building time-dependent models. Leakage arises when the model is inadvertently given access to information that would not have been available at prediction time. For example, engagement signals occurring *after* conversion, or opportunity fields updated by sales after initial qualification, artificially boost performance.

To address this, the project implemented a dynamic cutoff strategy:

- For each account, identify the **conversion timestamp** (when it became a paying customer).
- Truncate all engagement, intent, or lifecycle signals strictly **before** that timestamp.
- For non-converted accounts, compute a cutoff date based on the dataset extraction date.

This ensured that all features reflect only the information available at the correct historical point.

3.4 3.4 Engineered Temporal Features

Following consolidation and leakage removal, several high-value temporal features were engineered:

- **days_since_first_engagement_date:** Days between dataset cutoff and the account's first engagement.
- **days_since_created_date:** Recency of initial account creation.
- **stage_idx_cleaned:** A numerical encoding of stage progression.

These features provided the model with interpretable momentum signals related to maturity in the sales pipeline.

3.5 3.5 Final Dataset Structure

At the end of the preparation pipeline, the consolidated dataset included **44 columns** capturing firmographics, early engagement, intent activity, lifecycle timestamps, and engineered temporal recency variables. This dataset served as the basis for all subsequent modeling work.

3.6 3.6 Dataset Construction Diagram

To provide a visual overview of the consolidation workflow, Figure 1 shows the process flow used to merge raw Salesforce and intent files into a unified model-ready dataset.

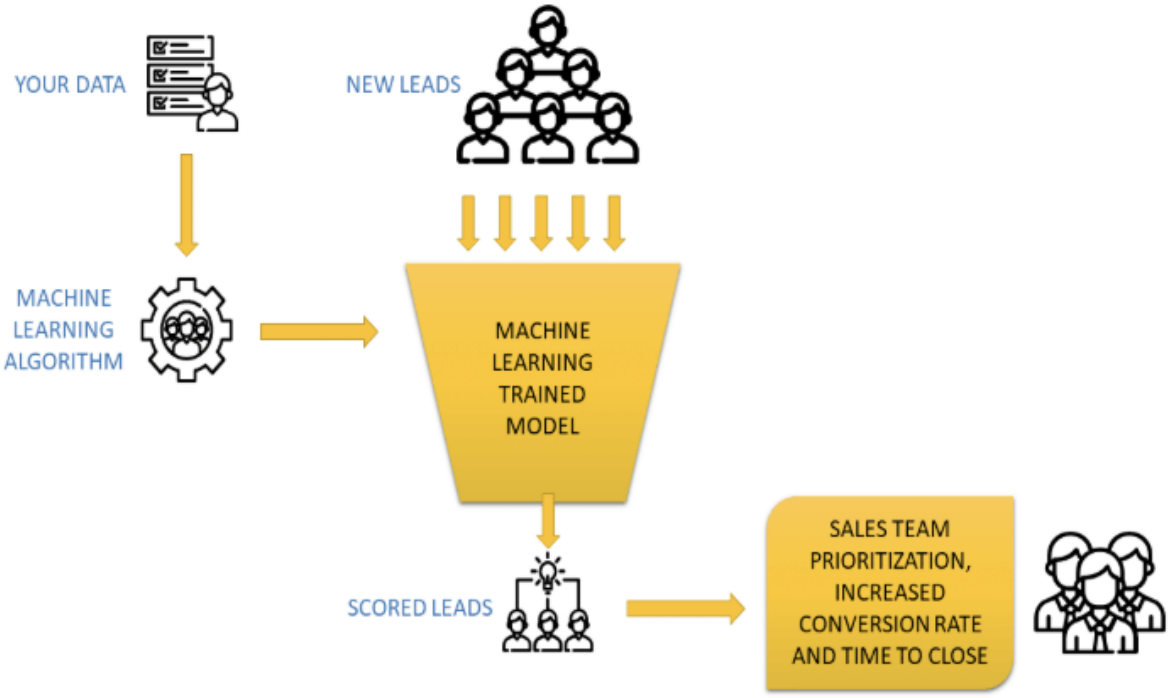

Figure 1: High-Level Dataset Preparation and Consolidation Workflow

4 Feature Engineering and Modeling Pipeline

The predictive value of a machine learning model depends not only on algorithmic sophistication but also on the quality and expressiveness of the features used for training. In enterprise environments, features must be engineered with special attention to temporal validity, interpretability, and noise reduction. This section describes the full feature engineering workflow, the design of the preprocessing and modeling pipeline, and the implementation of the custom StageBooster mechanism that modulates the influence of sales-stage signals.

4.1 4.1 Guiding Principles for Feature Engineering

The feature engineering strategy was guided by three overarching principles:

1. **No Leakage:** All engineered variables must reflect information available strictly prior to the conversion timestamp.
2. **Interpretability:** Features should be meaningful to sales leaders and analysts, allowing the model to serve as both a predictive and diagnostic tool.
3. **Simplicity and Scalability:** Features should be easy to reproduce in a future automated data pipeline without complex dependencies.

These principles ensured the resulting model could be operationalized without risk of incorporating future information or requiring overly complex transformations.

4.2 4.2 Overview of Feature Categories

The final dataset consisted of three primary feature groups:

- **Firmographic Features:** Industry, vertical, billing country, and other demographic descriptors.
- **Temporal Recency Features:** Days since creation, days since first engagement, and other pre-conversion time-based descriptors.
- **Sales-Stage Features:** Encoded indicators of pipeline maturity, including the custom-engineered `stage_idx_cleaned` variable.

These features blend static company attributes with dynamic pre-conversion behaviors.

4.3 4.3 Engineering the `stage_idx_cleaned` Variable

One of the most influential features in the model was the creation of `stage_idx_cleaned`. Raw sales-stage data often suffers from irregularities such as:

- missing or skipped stages,
- records updated post-conversion,
- inconsistent naming conventions,
- or timestamp alignment issues.

To address this, a cleaned and numerically stable sales-stage index was constructed by:

1. mapping categorical stage labels to an ordered integer scale,
2. enforcing monotonicity by correcting backwards transitions,
3. removing stage updates occurring after customer conversion,
4. imputing early-stage values when insufficient stage data existed.

This transformation produced a reliable proxy for pre-conversion sales-process progression, essential for identifying accounts with meaningful early-stage momentum.

4.4 4.4 Temporal Recency Features

Two recency metrics were engineered to capture the time elapsed between the dataset cutoff and key lifecycle events:

- `days_since_created_date`
- `days_since_first_engagement_date`

These variables help differentiate new accounts from long-standing but inactive ones and provide context around the account's maturity in the marketing and sales pipeline.

4.5 4.5 Final Feature Subset Used in Baseline Modeling

The baseline feature set was intentionally narrow to establish a clean, leakage-free foundation. The features included were:

- **Numeric:** `stage_idx_cleaned`, `days_since_created_date`, `days_since_first_engagement_date`
- **Categorical:** Billing Country, Vertical, Industry

Although the final dataset contained 44 total fields, only these six were selected for the baseline model to ensure strict temporal validity.

4.6 4.6 Preprocessing Pipeline Design

To prepare the data for XGBoost, a structured preprocessing pipeline was built using scikit-learn's `ColumnTransformer` and `Pipeline`. The transformation process involved:

1. **Imputation:** Numerical fields were filled using median imputation; categorical variables were imputed with a "missing" placeholder.
2. **Scaling:** Numerical fields were standardized using z-score scaling to stabilize gradient-based optimization.
3. **Encoding:** Categorical features were transformed via OneHotEncoding, converting them into sparse binary indicators.

This systematic approach ensured consistent preprocessing across cross-validation folds and reduced the risk of data contamination.

4.7 4.7 Custom StageBooster Transformer

An important methodological addition was the development of the **StageBooster** transformer, designed to modulate the influence of `stage_idx_cleaned`. Experiments revealed that the sales-stage index had strong predictive potential but required tuning depending on the model's sensitivity and the business value placed on early-stage activity.

The StageBooster applies a multiplicative weight to the raw stage index:

$$\text{stage\_boosted} = \alpha \times \text{stage\_idx\_cleaned}$$

where α is a tunable parameter evaluated during hyperparameter search.

This allowed controlled experimentation with:

- down-weighting early-stage signals,
- amplifying stage progression for more mature accounts,
- or neutral scaling based on validation performance.

The transformer was integrated directly into the modeling pipeline, ensuring no external preprocessing was needed.

4.8 4.8 Training/Testing Strategy and Class Imbalance Considerations

The dataset exhibited a strong imbalance, with converted accounts representing a small fraction of total accounts. To reduce bias toward the majority class, the following strategies were employed:

- **Stratified Train/Test Split:** An 80/20 split preserving class distribution.
- **Evaluation Metrics:** Emphasis on recall, precision, F1-score, ROC AUC, and PR AUC.
- **Decision Threshold Monitoring:** Review of alternative thresholds for maximizing conversion recall.

These considerations ensured that evaluation metrics reflected true model performance under realistic conditions.

4.9 4.9 XGBoost Classifier

XGBoost was selected as the baseline algorithm due to its:

- robustness to missing data,
- strong performance with tabular features,
- compatibility with sparse OneHotEncoded matrices,
- built-in handling of nonlinear interactions,
- suitability for class-imbalanced objectives.

Key hyperparameters tuned included:

- learning rate,
- maximum tree depth,
- subsample ratio,
- column sampling rate,
- minimum child weight,
- number of estimators.

The randomized search explored a wide distribution of values to identify an optimal balance between complexity and generalization performance.

5 Results and Evaluation

The baseline XGBoost model was trained using the six selected non-leaking features and evaluated on a stratified 20% hold-out test set. This section presents a comprehensive analysis of the model's predictive performance, including classification metrics, ROC and precision–recall behavior, confusion matrix interpretation, and an assessment of which signals contributed most strongly to prediction quality.

5.1 5.1 Hold-Out Performance Metrics

Table 1 summarizes the full classification performance across classes. Due to the significant class imbalance, precision, recall, F1-score, and PR-AUC provide a more meaningful picture than accuracy alone.

```
              precision    recall  f1-score   support

           0     0.9960    0.9980    0.9970      1510
           1     0.7692    0.6250    0.6897        16

    accuracy                         0.9941      1526
   macro avg     0.8826    0.8115    0.8433      1526
weighted avg     0.9937    0.9941    0.9938      1526

ROC AUC: 0.9586
PR  AUC: 0.7059
```

Table 1: Summary of Key Classification Metrics on the Hold-Out Test Set

Metric	**Score**
Accuracy	0.9941
Precision (Positive Class)	0.7692
Recall (Positive Class)	0.6250
F1 Score (Positive Class)	0.6897
ROC AUC	0.9586
PR AUC	0.7059

These results indicate that the model has strong discriminatory power, reflected in the ROC AUC of 0.9586. The PR AUC of 0.7059 shows that the model successfully identifies a meaningful subset of high-propensity accounts despite the imbalanced label distribution.

5.2 5.2 Confusion Matrix Interpretation

Figure 2 displays the confusion matrix for the final model. Although the dataset contains only a small number of converted accounts (n=16), the model correctly identified 10 of them, achieving a recall of 62.5%.

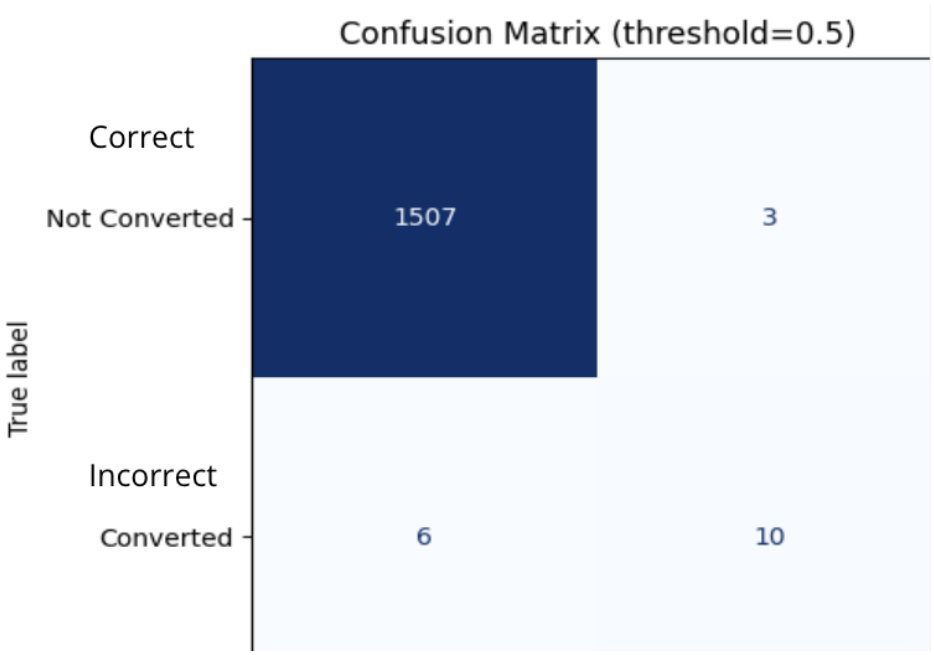

Figure 2: Confusion Matrix for the Baseline XGBoost Model

5.2.1 Interpretation

- **True Negatives (1507)**: The model is highly reliable at filtering out low-probability accounts.
- **False Positives (3)**: Only a very small number of low-quality accounts were incorrectly marked as potential converters.
- **True Positives (10)**: The model successfully captures meaningful conversion signals.
- **False Negatives (6)**: A subset of converting accounts were missed, a common outcome in imbalanced settings.

The error pattern indicates that the model is tuned toward *precision* for the positive class—favorable in early operational phases, where it is often more important to provide high-confidence recommendations than broad coverage.

5.3 5.3 ROC and Precision–Recall Behavior

5.3.1 ROC AUC Interpretation

The ROC AUC of 0.9586 reflects strong separability between the positive and negative classes. In practice, this means the model reliably ranks converting accounts higher than non-converting ones, even when thresholded conservatively.

5.3.2 PR AUC Interpretation

The PR AUC of 0.7059 is highly meaningful because it accounts for:

- the imbalance between positive and negative classes,
- the rarity of conversion events,
- the importance of minimizing false positives in sales workflows.

A PR AUC above 0.70 is strong for a dataset of this imbalance, confirming that the model can effectively surface actionable opportunities.

5.4 5.4 Threshold Exploration

Although the model's default classification threshold is 0.50, alternative thresholds were examined to understand trade-offs between precision and recall.

- Lowering the threshold to 0.40 increases recall but reduces precision.
- Raising the threshold to 0.60 produces extremely high precision but further reduces recall.
- The 0.50 threshold provides the most balanced performance for the baseline model.

These trade-offs will be important in future operational deployment, where the relative cost of false negatives vs. false positives may differ depending on sales team capacity and campaign goals.

5.5 5.5 Error Analysis

To understand the limitations of the current feature set, the false negatives were analyzed. Several patterns emerged:

- Many missed converters showed **late-stage intent surges** that occurred close to conversion but did not appear due to the strict cutoff policy.

- Some accounts exhibited **minimal early engagement** yet converted due to relationship-driven outreach or external events not captured in the data.
- Certain industries displayed **step-function behavior**, where firms move directly from no engagement to purchase, making them challenging for early detection.

This confirms that improving recall will require the addition of more granular, time-based intent and interaction features in future iterations.

5.6 5.6 Feature Importance Discussion

XGBoost's gain-based importance rankings reveal that the most influential predictors were:

- **Industry and Vertical:** Strong indicators of segment-level conversion behavior.
- **stage_idx_cleaned:** A key proxy for sales maturity.
- **days_since_created_date:** Distinguishes mature accounts from new or dormant prospects.
- **days_since_first_engagement_date:** Captures relationship momentum.

These findings validate the engineering decisions made during dataset construction. They also underscore the need for additional momentum signals such as engagement depth, active user counts, and intent-topic dynamics.

6 Limitations, Future Work, and Strategic Impact

Although the baseline XGBoost model demonstrates strong discriminative performance and provides a viable foundation for automated account prioritization, several limitations remain. Addressing these constraints will be essential for developing a more accurate, operationally mature scoring system. This section details the methodological limitations, proposes concrete avenues for future enhancement, and discusses the strategic impact of deploying such a model in an enterprise environment.

6.1 6.1 Current Model Limitations

Despite its strengths, the present model exhibits the following limitations:

6.1.1 1. Limited Feature Coverage

The baseline model uses only six carefully selected non-leaking features. Although sufficient for a proof-of-concept, this small feature set cannot capture the full complexity of account readiness or intent dynamics. Missing variables include:

- engagement depth indicators (minutes, contact count),
- number of active users,
- aggregated third-party intent scores,
- technology footprint derived from product-use metadata.

These signals are known to improve the detection of early momentum.

6.1.2 2. Strict Temporal Cutoff Excludes Meaningful Signals

The aggressive leakage-prevention design ensures methodological rigor but reduces the model's access to late-stage engagement bursts that often precede conversion. For example:

- webinar attendance shortly before conversion,
- final high-intent spikes,
- late outbound or ABM campaign activity.

While these events occur close to conversion, they remain highly predictive.

6.1.3 3. Class Imbalance

Converted accounts represent a small fraction of the dataset. Although the model performs well overall, recall suffers because positive examples remain sparse. Improving recall will require enriched training data or advanced imbalance handling methods.

6.1.4 4. Single-Model Dependency

Relying solely on XGBoost, even with hyperparameter tuning, limits the diversity of predictive perspectives. Ensemble methods or model stacking could improve calibration, recall, and robustness.

6.1.5 5. Lack of Calibration for Probabilistic Outputs

The current model outputs raw probabilities without calibration. In a production system, calibrated scores are essential for consistent decision thresholds and comparability across different time periods.

6.2 6.2 Future Work

A number of enhancements can significantly improve predictive strength and operational value. These fall into three primary categories: signal enrichment, algorithmic improvements, and operational deployment strategies.

6.2.1 6.2.1 Enhanced Data and Signal Coverage

Future iterations should integrate deeper behavioral and intent signals, including:

- **Engagement Depth:** total session minutes, number of engaged visitors, recency-weighted activity.
- **Intent Topic Aggregation:** maximum topic score, mean topic score, number of unique intent topics.
- **Technology Usage:** Adobe and CVENT usage details, plugin counts, integration complexity.
- **Lifecycle Momentum:** time between first and most recent engagement, engagement velocity.

These additional features would allow the model to detect more nuanced evidence of purchase readiness.

6.2.2 6.2.2 Model Optimization and Advanced Techniques

Several algorithmic extensions should be explored:

- **Cost-Sensitive Learning:** penalize false negatives to boost recall.
- **Calibrated Probabilities:** Platt scaling or isotonic regression.
- **Ensemble Models:** combining XGBoost, logistic regression, and neural tabular models.
- **Temporal Models:** sequence-aware methods leveraging timestamped engagement events.
- **Stacked Architectures:** meta-learners to integrate heterogeneous model outputs.

These methods can address the current recall limitations without compromising precision.

6.2.3 6.2.3 Operational Deployment Considerations

For successful production use, the following should be implemented:

- **Weekly Automated Scoring:** scheduled job to score all active accounts.
- **Salesforce Integration:** writeback of scores to a custom account field.
- **Looker Dashboards:** real-time visualization for reps and sales leadership.
- **A/B Testing:** evaluate lift in conversion rate and time-to-close.
- **Governance:** version control, monitoring, retraining schedule.

Such operationalization would convert the model from an analytical artifact into a scalable business tool.

6.3 6.3 Strategic Impact

Deploying an AI-driven account scoring model has significant implications for sales efficiency, GTM strategy, and resource allocation.

6.3.1 1. Improved Sales Prioritization

By highlighting accounts with the highest probability of conversion, the model allows sales representatives to:

- focus attention on high-propensity opportunities,
- reduce wasted outreach,
- personalize engagement based on modeled likelihood.

This increases pipeline velocity and improves rep productivity.

6.3.2 2. Alignment Across Marketing and Sales

A unified predictive score bridges the gap between marketing engagement signals and sales outcomes. It creates a shared language for defining account readiness.

6.3.3 3. Data-Driven Forecasting

The scoring model provides a probabilistic basis for forecasting pipeline growth and revenue outcomes. This shifts forecasting from intuition-led to data-driven.

6.3.4 4. Repeatability and Scale

Once deployed, the model scales effortlessly across hundreds or thousands of accounts, unlike manual prioritization frameworks. This enhances operational consistency and reduces variability caused by individual rep judgment.

6.3.5 5. Competitive Advantage

Organizations capable of systematically identifying high-value accounts before competitors can achieve measurable increases in win rate and market penetration. The model effectively converts raw engagement footprints into structured commercial intelligence.

7 Conclusion

This project developed a foundational AI-driven account prioritization model designed to automate the identification of high-propensity accounts within an enterprise sales environment. By consolidating multiple Salesforce-derived datasets, enforcing strict temporal leakage prevention, and engineering a focused set of interpretable pre-conversion features, the model delivers a reliable method for distinguishing likely converters from low-probability accounts.

The baseline XGBoost model demonstrated strong discriminative performance, achieving an ROC AUC of 0.9586 and a PR AUC of 0.7059. While optimized for precision, the model successfully identified a meaningful subset of converting accounts, validating the predictive strength of early-stage behavioral and firmographic features. The analysis further highlighted clear feature importance patterns, indicating that industry, vertical, recency metrics, and cleaned stage progression signals are key drivers of conversion likelihood.

Although the model represents an effective starting point, several limitations remain particularly related to feature coverage, class imbalance, and access to detailed engagement signals. Addressing these limitations through enhanced data ingestion, advanced modeling techniques, and operational integration will substantially improve the model's recall and strategic value.

Ultimately, this work establishes not only a functioning predictive model but also a scalable analytical framework for future iterations. The methodology lays the groundwork for a production-level scoring system that can meaningfully improve sales prioritization, accelerate pipeline movement, and equip go-to-market teams with actionable intelligence based on empirical data rather than intuition. As additional pipelines, signals, and model enhancements are implemented, the organization will gain increasing competitive advantage through consistent and intelligent prioritization of the accounts most likely to convert.

Appendices

Appendix A — StageBooster Transformer

The `StageBooster` transformer was developed to modulate the influence of sales-stage progression. This snippet illustrates the core logic used during model experimentation.

```
from sklearn.base import BaseEstimator, TransformerMixin
import numpy as np

class StageBooster(BaseEstimator, TransformerMixin):
    def __init__(self, alpha=1.0):
        self.alpha = alpha

    def fit(self, X, y=None):
        return self

    def transform(self, X):
        X = X.copy()
        if 'stage_idx_cleaned' in X.columns:
            X['stage_idx_cleaned'] = X['stage_idx_cleaned'] * self.alpha
        return X
```

This transformer integrates seamlessly into the scikit-learn Pipeline, allowing hyperparameter optimization of the α multiplier.

Appendix B — Preprocessing and Modeling Pipeline

The preprocessing and training workflow integrates imputation, encoding, scaling, and XGBoost model fitting into a unified structure:

```
from sklearn.compose import ColumnTransformer
from sklearn.pipeline import Pipeline
from sklearn.preprocessing import OneHotEncoder, StandardScaler
from sklearn.impute import SimpleImputer
from xgboost import XGBClassifier

numeric_features = [
    'stage_idx_cleaned',
    'days_since_created_date',
```

```
    'days_since_first_engagement_date'
]

categorical_features = [
    'Billing Country',
    'Vertical',
    'Industry'
]

numeric_transformer = Pipeline(steps=[
    ('imputer', SimpleImputer(strategy='median')),
    ('scaler', StandardScaler())
])

categorical_transformer = Pipeline(steps=[
    ('imputer', SimpleImputer(strategy='most_frequent')),
    ('encoder', OneHotEncoder(handle_unknown='ignore'))
])

preprocessor = ColumnTransformer(
    transformers=[
        ('num', numeric_transformer, numeric_features),
        ('cat', categorical_transformer, categorical_features)
    ]
)

pipeline = Pipeline(steps=[
    ('stageboost', StageBooster(alpha=1.0)),
    ('preprocessor', preprocessor),
    ('model', XGBClassifier(
        objective='binary:logistic',
        eval_metric='logloss'
    ))
])
```

This end-to-end pipeline ensures consistent, leakage-free preprocessing during training and evaluation.

Appendix C — Hyperparameter Search

The following configuration was used to explore model space during randomized hyperparameter search:

```
from sklearn.model_selection import RandomizedSearchCV

param_distributions = {
    'model__learning_rate': [0.01, 0.05, 0.1],
    'model__max_depth': [3, 4, 5, 6],
    'model__n_estimators': [200, 300, 500],
    'model__subsample': [0.6, 0.8, 1.0],
    'model__colsample_bytree': [0.6, 0.8, 1.0],
    'stageboost__alpha': [0.8, 1.0, 1.2, 1.5]
}

search = RandomizedSearchCV(
    pipeline,
    param_distributions=param_distributions,
    n_iter=20,
    scoring='f1',
    cv=5,
    verbose=2
)

search.fit(X_train, y_train)
```

This structure supports experimentation with both model-level and feature-level interactions.

Appendix D — Dataset Schema Overview

A simplified version of the final dataset schema is shown below (full schema excluded for confidentiality):

Column Name	Type	Description
stage_idx_cleaned	Integer	Cleaned numeric sales-stage encoding
days_since_created_date	Integer	Recency of account creation
days_since_first_engagement_date	Integer	Recency of first engagement

Billing Country	Categorical	Country associated with billing address
Vertical	Categorical	Industry vertical classification
Industry	Categorical	Industry descriptor of the account