📊 Full Report: AI-Powered CRM Intelligence System

# 1. Introduction

The rise of Customer Relationship Management (CRM) platforms has created a demand for intelligent automation to assist in lead conversion, opportunity win estimation, and customer retention.   
This project presents an AI-powered module built using XGBoost and Streamlit to deliver predictive insights across three core CRM areas: Lead Scoring, Opportunity Win Prediction, and (planned) Churn Risk Analysis.  
The prototype was designed to integrate with SQL-based CRM systems and offers a developer-friendly dashboard for training and prediction.

# 2. AI Modules and Business Objectives

This AI engine is structured around three actionable intelligence modules:  
• Lead Scoring – Predict the likelihood of converting a lead into a paying customer.  
• Opportunity Win Prediction – Forecast the probability that a deal/opportunity will be closed.  
• Churn Risk (Planned) – Identify accounts that are at risk of discontinuing services or leaving.  
Each model aims to enhance sales efficiency, revenue forecasting, and customer retention within any CRM system.

# 3. Dataset Structure and Key Variables

The prototype uses SQL inserts from three tables:  
• leads (1).sql  
• opportunities (1).sql  
• accounts (1).sql  
  
Key variables used in modeling include:  
• Leads: untouched\_since\_days, lead\_source, customer\_type, industry, area, designation, etc.  
• Opportunities: probability, amount, opportunity\_type, customer\_type, product\_sub\_category  
• Accounts: (for churn) account\_type, last\_activity, revenue, industry, city, state  
All data was transformed into a Pandas DataFrame before feeding into machine learning pipelines.

# 4. The Algorithm: XGBoost (Extreme Gradient Boosting)

XGBoost is a tree-based ensemble machine learning algorithm based on gradient boosting. It was developed by Tianqi Chen and is considered one of the most powerful classifiers for structured/tabular data.  
XGBoost builds models in stages and generalizes them by optimizing a loss function. In each stage, it fits a new tree to the residuals (errors) of the previous trees.

Key features of XGBoost:  
• Supports missing values internally – no need to impute manually.  
• Implements regularization (L1 and L2) to prevent overfitting.  
• Uses parallelization to speed up training.  
• Enables early stopping, boosting performance in real-world noisy data.  
• Provides feature importance scores.

How we used it:  
We used `XGBClassifier` for binary classification in both lead and opportunity prediction.   
The models were tuned using `RandomizedSearchCV` to explore hyperparameters such as:  
• learning\_rate: Controls the step size for updating weights.  
• max\_depth: Controls tree complexity.  
• n\_estimators: Total number of boosting rounds.  
• subsample and colsample\_bytree: Controls randomness and feature sampling.  
The performance was evaluated using ROC AUC score – a robust metric for binary classifiers.

# 5. App Architecture and Workflow

The project was implemented using the following stack:  
• Frontend: Streamlit (Python UI library)  
• Backend: XGBoost models wrapped in sklearn pipelines  
• Input: SQL insert files and CSV uploads  
• Database: SQLAlchemy support for MySQL/PostgreSQL integration

Workflow:  
1. User uploads CRM data or connects to SQL DB.  
2. Data is cleaned and features extracted.  
3. Model training is triggered via the UI.  
4. ROC AUC is computed for both baseline and tuned models.  
5. The better model is auto-selected and stored via joblib.  
6. Prediction tab allows on-the-fly scoring of new leads or opportunities.

# 6. Streamlit UI Features and Developer Mode

The UI allows toggling between “Lead Scoring” and “Opportunity Win” modes.  
Key features include:  
• Model training with accuracy comparison.  
• Uploads via CSV or SQL inserts.  
• Chart plotting: Before vs After tuning (ROC AUC)  
• Developer Mode toggle: reveals nulls, column types, feature lists, and imputation steps.

# 7. CRM Integration and Business Application

This prototype can be integrated into any CRM as:  
• A standalone AI dashboard (deployed via Streamlit).  
• An internal module calling `.joblib` models via REST API.  
• Backend microservice for real-time lead scoring.  
Business teams can use this to:  
• Route high-conversion leads to best agents.  
• Forecast sales pipeline health.  
• Alert sales team of at-risk opportunities.  
• Reduce churn using account activity scoring (planned).

# 8. Summary of Results

After hyperparameter tuning, models showed significant variance in performance. In some cases, the baseline model (XGBoost default settings) outperformed the tuned model.  
ROC AUC scores were plotted before and after tuning for visual comparison. The app auto-selects the best performing model to avoid accuracy loss.  
Lead scoring was more stable than opportunity prediction due to better feature variance in leads dataset.

# 9. Future Enhancements

• Churn prediction using accounts.sql  
• Cross-entity joins across leads, opportunities, accounts  
• Add support for automated model retraining with cron or background jobs  
• Dockerize and expose API for production use in CRM SaaS  
• Add user authentication and model explanation UI

© 2025 AI CRM Intelligence | Author: [Your Name] | Streamlit + XGBoost + SQL