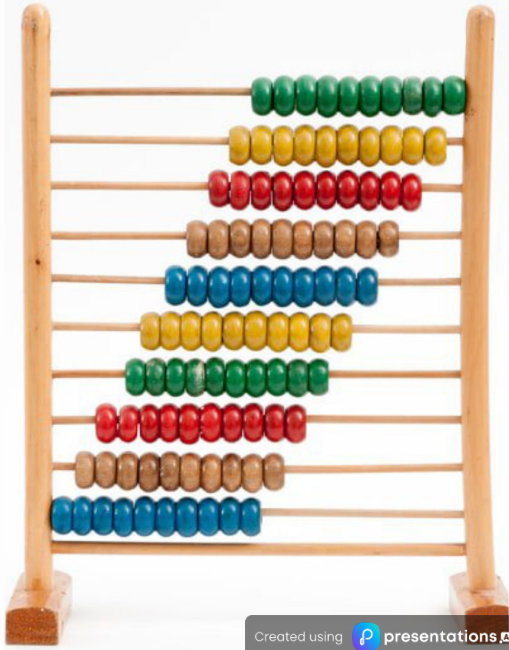
Lead Scoring System Using Machine Learning



KANCHERLA VENKAT SAI
Presenter Designation

Business Problem Statement

Highly competitive B2B sales environment.

Marketing and sales teams invest significant resources in acquiring leads.

Low lead conversion rates.

A large proportion of leads fail to become paying customers, resulting in wasted efforts and suboptimal ROI.

Lack of intelligent lead prioritization.

No existing mechanisms to prioritize high-potential leads based on historical data and behavioral attributes.

Need for data-driven lead scoring solution.

Business aims to help sales representatives identify and focus on leads most likely to convert, enabling better resource allocation and outreach.

Project Objective

Primary Goal: To develop a scalable, production-ready ML pipeline that:

Includes scoring leads based on conversion probability, prioritizing follow-up, and automating retraining.

Sub-Objectives: Ingest, clean, and transform raw lead data.

Prepare data for model training and ensure quality input for predictions.

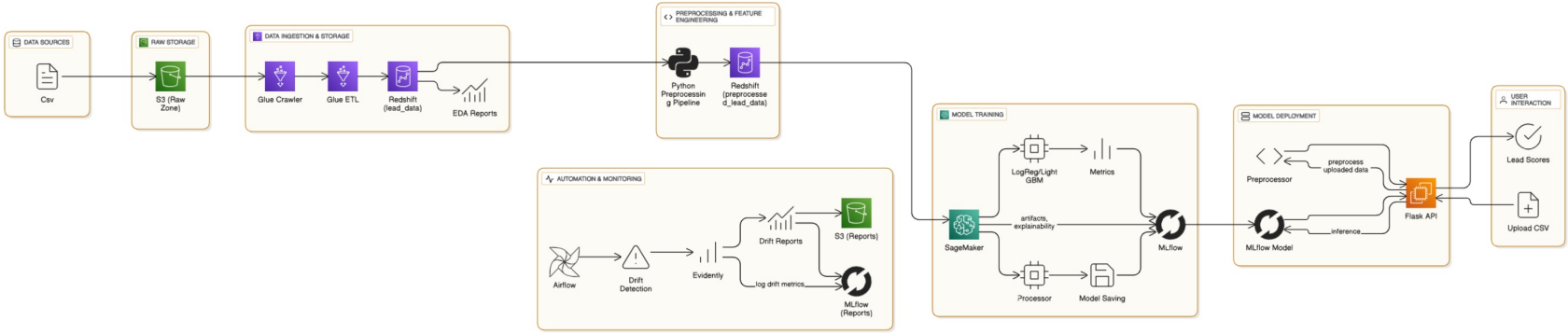
Train multiple classification models and select the best.

Evaluate various models to identify the most effective one for lead scoring.

Deploy a REST API for real-time predictions.

Facilitate instant lead scoring through an API interface.

System Architecture



Building an Efficient Data Pipeline with AWS

Integrating AWS services for optimal data processing and model training

Data collection from diverse sources

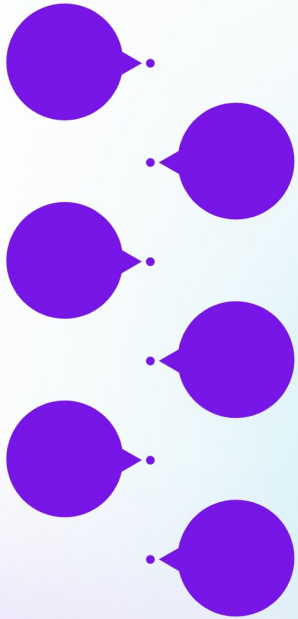
Gather data via website forms, marketing campaigns, and third-party tools, primarily in CSV format.

Schema detection with AWS Glue Crawlers

Automatically detect and catalog dataset schemas, enabling schema evolution tracking.

Data warehousing in Amazon Redshift

Load transformed data into Redshift for structured storage and fast querying.



Centralized storage in Amazon S3

Upload raw datasets to Amazon S3 buckets for scalable and secure data storage.

ETL processing using AWS Glue jobs

Clean, normalize, and merge datasets based on business rules through automated ETL workflows.

Dedicated tables for training and user data

Maintain separate Redshift tables for model training datasets and incoming user-uploaded data to support development and inference.

Data Preprocessing and Feature Engineering

Handle missing values and encode categorical features.

Use methods such as imputation or deletion for missing values and techniques like label encoding or one-hot encoding for categorical features.

Normalize and scale numerical variables.

Adjust numerical features to a common scale to improve model performance.

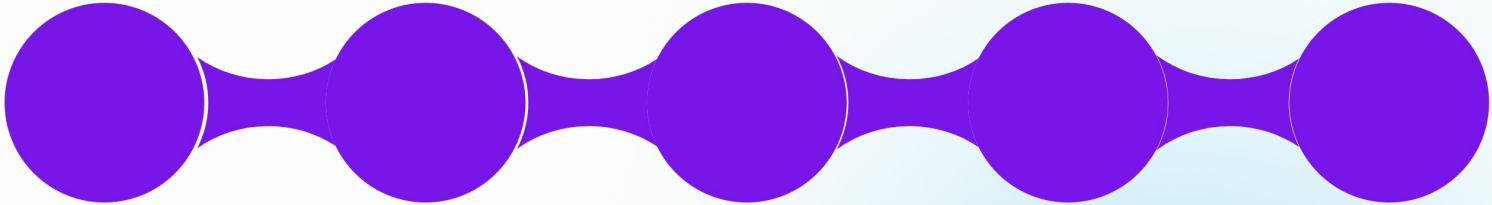
Drop irrelevant or constant columns and validate schema.

Remove features that do not contribute to the predictive power and ensure schema compatibility for inference.

Implement feature engineering techniques.

Create interaction features and flags to enhance model insights, such as engagement rate and activity indicators.

Model Training and Evaluation



Models Trained: Logistic Regression, Random Forest, XGBoost, LightGBM

These models were selected for their effectiveness in classification tasks.

Hyperparameter Tuning with GridSearchCV

Utilized GridSearchCV to optimize model parameters for better performance.

Cross-Validation with Stratified Folds

Ensured robust model evaluation through stratified cross-validation.

Evaluation Metrics: F1-Score, ROC-AUC, Precision, Recall

Models were evaluated using various performance metrics to ensure reliability.

Best Model Selection Based on F1 Score and Interpretability

The best model was chosen for its balanced performance and clarity in feature importance.

Automation and Monitoring

Apache Airflow Orchestration

Automates pipeline tasks like data ingestion, preprocessing, model training, drift detection, and retraining with scheduled DAGs.

Evidently AI Monitoring

Detects data and concept drift by comparing incoming data with training distributions and generates visual reports.

Automated Retraining

Triggers model retraining automatically through Airflow when drift thresholds are exceeded to maintain model performance.

MLflow Logging

Tracks logs and metrics for reproducibility, versioning, and traceability of each pipeline step.

Model Deployment

MLflow Model Registry for model and preprocessing pipeline versioning and stage management.

Validated Production model deployed via Flask API on AWS EC2 instance.

API applies preprocessing and returns real-time lead conversion predictions.

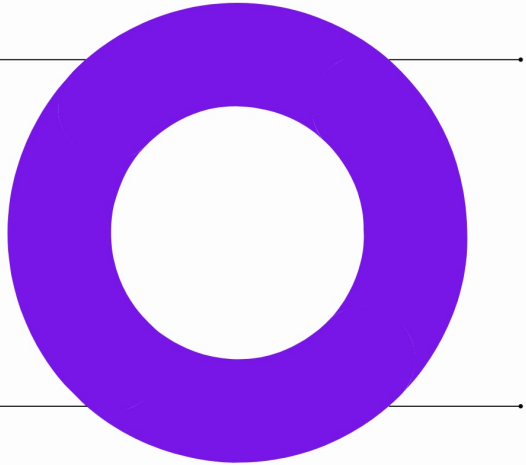
Nginx used as reverse proxy for reliable web traffic routing.

Enables integration with frontend dashboards or CRM systems for on-demand predictions.

Monitoring of prediction requests and responses ensures consistency and traceability.

User Interaction

Streamlined user engagement through web-based forms and machine learning evaluation



User interaction via web UI

Users can input data easily.

CSV file uploads allowed

Users can upload data files directly.

Designed for non-technical users

Requires no ML expertise.

Instant result display

Facilitates quick decision-making.

We welcome any questions regarding the Lead Scoring System project and look forward to discussing how it can benefit your organization.