Project Title: "AI-POWERED LEAD SCORING SYSTEM FOR SMARTER SALES CONVERSION"

Industry: Cross-Industry (Applicable to Sales, Marketing, and CRM domains)

Project Type: B2B/B2C Salesforce CRM Implementation with AI Integration

Target Users: Sales Teams, Marketing Teams, CRM Administrators, Business Managers

---

1.1 Problem Statement

Organizations generate thousands of leads daily through websites, campaigns, social media, and referrals. However, sales teams face difficulty in identifying which leads are most likely to convert. This results in wasted efforts on low-quality leads, inconsistent follow-ups, and delayed deal closures.

To address this challenge, the company wants to integrate an AI-Powered Lead Scoring System within Salesforce CRM to:

- Automate the scoring and ranking of leads using machine learning
- Prioritize high-intent leads for faster follow-ups
- Improve alignment between marketing campaigns and sales conversions
- Provide real-time dashboards to track lead quality, agent performance, and revenue impact

---

1.2 Use Cases

Lead Management

- Capture leads from multiple sources (web forms, campaigns, referrals, social media)
- Deduplicate and assign leads automatically to sales agents

AI Lead Scoring

- Analyze customer data (demographics, engagement history, campaign interactions)
- Assign a predictive lead score based on likelihood to convert
- Categorize leads into Hot, Warm, Cold for prioritization

Sales Engagement

- Automatically notify sales agents of high-scoring leads

- Recommend next best action (e.g., call, email, schedule a demo)
- Track response time and conversion likelihood

Deal Conversion

- Update deal stages dynamically based on lead score trends
- Provide probability-based forecasts for pipeline deals
- Capture win/loss reasons for model improvement

Reporting & Analytics

- Dashboard with Lead Quality Funnel (Hot vs. Warm vs. Cold)
- Agent performance based on high-intent lead handling
- Predictive revenue forecasts from AI scoring
- Continuous feedback loop to improve the ML model

AI-Powered Lead Scoring System — Detailed Roadmap (Phase-by-Phase)

Phase 1 — Problem Understanding & Industry Analysis

Goal: define why scoring is needed, what success looks like, and what data exists.

Steps

1. Run stakeholder interviews (Sales, Marketing, Manager, IT) — capture pain points and KPIs.
2. Write a short Problem Statement and Success Metrics (e.g., better lead prioritization, higher conversion %) — convert into measurable acceptance criteria.
3. Map the current Lead lifecycle (capture → qualification → assignment → follow-up → conversion) with a swimlane diagram.
4. Inventory lead data sources (web forms, CSV imports, event lists, ad platforms, manual entries) and sample fields available.
5. Create sample user stories (e.g., “As a Sales Rep I want hot leads assigned to me automatically so I can call them first”).
6. Explore options (Einstein Lead Scoring vs custom ML) and list pros/cons for your org.

Deliverables

- BRD / One-page project brief
- User stories & acceptance criteria
- Data inventory spreadsheet
- Recommendation: Einstein vs Custom ML decision note

Phase 2 — Org Setup & Configuration

Goal: prepare a secure Salesforce org and sandboxes with users, roles, and access relevant to lead scoring.

Steps

1. Choose edition & org setup:
    - Use a Developer org or Sandbox for dev/testing; plan target Production (Sales Cloud features required).
2. Company profile:
    - Set default currency, time zone, locale, and company information.
3. Business hours & holidays:
    - Configure business hours and holidays so automation (assignment / SLA) respects working hours.
4. Fiscal year:
    - Confirm fiscal settings so reporting aligns with finance.
5. Users & licenses:
    - Create sample users: Admin, Sales Manager, Sales Rep(s), Marketing. Assign appropriate licenses.
6. Profiles & permission sets:
    - Create profiles: System Admin, Sales, Marketing, ReadOnly.
    - Create Permission Sets: Lead_Scoring_Access, AI_Integrator_Access (grant access to scoring fields, dashboards, and integration-related metadata).
7. Roles & sharing:
    - Build role hierarchy (Sales Manager → Sales Rep).

    - Set Org-Wide Defaults: Leads = Private; configure sharing rules for role visibility and any queue access.
8. Login & security policies:
    - Admin login access enabled, set password policies, IP ranges if required.
9. Dev tools:
    - Connect VS Code + Salesforce CLI, create a sandbox/scratch org strategy.
    - Set up source control (Git) with a branching strategy (dev → test → main).

Deliverables

- Configured dev org and sandbox(s) with test users & roles
- Permission set matrix document
- "How to connect" notes for VS Code + SFDX

Phase 3 — Data Modeling & Relationships

Goal: design Lead data model (fields, types, relationships) to support scoring and auditability.

Steps

1. Define custom fields on Lead (API name / type / purpose):
    - AI_Score__c — Number(3,0) — normalized 0–100
    - Conversion_Probability__c — Percent/Number(5,2)
    - Lead_Priority__c — Picklist [High, Medium, Low]
    - Last_Scored__c — DateTime
    - Model_Version__c — Text(50)
    - Score_Factors__c — Long Text Area (explainability summary)
2. Create a Lead Scoring Log custom object:
    - Lead__c (Lookup), Score__c, Probability__c, Model_Version__c, Response_Payload__c, Scored_At__c, Status__c.
    - Purpose: traceability and debugging of scoring calls.
3. Record types & page layouts:
    - Record Types: Web, Event, Referral — adjust required fields per type.

    - Page Layouts: show scoring fields and Score History related list.
4. Schema Builder: document relationships (Lead → Lead_Scoring_Log).
5. Data validation rules: require minimum fields before scoring (e.g., Email or Phone present).
6. Sample data: prepare CSV with at least 200 sample leads across sources for testing.

Deliverables

- Field definition spreadsheet (label, API name, type, help text)
- ER diagram and sample dataset CSV
- Validation Rules doc

Phase 4 — Process Automation (Admin)

Goal: automate scoring invocation, routing & notifications using Flows / Process Builder and lightweight automation where possible.

Steps

1. Use a Record-Triggered Flow on Lead (after insert / after update) to:
    - Validate data completeness
    - Push lead payload to an integration mechanism (Platform Event or call Apex queueable)
2. Create Assignment Rules:
    - If AI_Score__c ≥ threshold → auto-assign to Sales Queue or specific owner.
    - If AI_Score__c < threshold → route to nurture queue (Marketing).
3. Build Scheduled Flow / Apex Batch to re-score leads in batch (e.g., when model updates).
4. Email Alerts & In-App Notifications:
    - Notify sales rep on new hot lead (push or email).
5. Error handling flow:
    - If scoring fails, set Scoring_Status__c = Failed, add entry in Lead Scoring Log, and notify admin.

Acceptance Checks

- New lead triggers a scoring process and populates AI_Score__c.
- Assignment rules reassign leads based on score automatically.

Phase 5 — Apex Programming (Developer)

Goal: implement reliable, testable server logic for callouts, batching, and async processing.

Steps & Components

1. Named Credential + Auth Provider configured (secure endpoint).
2. Apex classes:
    - AILeadScoringService — performs HTTP callouts using NamedCredential to send lead features and parse responses.
    - LeadScoringTriggerHandler — lightweight trigger handler to enqueue scoring jobs.
    - LeadScoringQueueable / LeadScoringBatch — for async/rescoring and backoff.
    - LeadScoringRetryManager — retry logic & circuit breaker state.
3. Logging:
    - Write scoring responses to Lead_Scoring_Log__c for each request (status/success/error).
4. Testing:
    - Use HttpCalloutMock to mock responses.
    - Achieve required test coverage, assert logging created and fields updated.
5. Exception & limits handling:
    - Check governor limits, implement chunking in batches, and catch HTTP failures.

Deliverables

- Apex classes, trigger, and test classes committed to Git
- Deployment-ready metadata

Phase 6 — User Interface Development

Goal: present score and context so reps can act — clean, actionable UI.

Steps

1. Create a Lightning App: “Lead Scoring”.
2. Lead Record Page:
    - Add a “Lead Score Card” LWC showing:
        - AI_Score__c as a meter / large number
        - Conversion_Probability__c
        - Lead_Priority__c
        - Short Score_Factors__c explanation
        - “Rescore” button (calls Apex to re-enqueue scoring)
3. List Views & Lightning Quick Actions:
    - “Hot Leads” list view filter: Lead_Priority__c = High
    - Quick action to add follow-up task from score card
4. Mobile friendliness:
    - Ensure compact layout shows core scoring fields to reps on mobile.

Acceptance

- Sales rep can see score/history and trigger manual rescore with one click.

Phase 7 — Integration & External Access

Goal: securely connect Salesforce to your AI model (or Einstein).

Steps

1. Choose architecture:
    - Option A: Einstein Lead Scoring (native) — minimal integration work.
    - Option B: Custom model hosted (Cloud Run / Lambda / Heroku / any REST API).
2. Secure connection:
    - Create Named Credential with OAuth or token auth.
    - Store keys/secrets in Salesforce Named Credentials or protected custom metadata.
3. API contract:

    - Define request payload schema and response schema (score, probability, factors, model_version).
4. Reliability & monitoring:
    - Implement retry/backoff in Apex
    - Log full request/responses to Lead_Scoring_Log__c (truncated if large)
    - Implement alerting for repeated failures
5. Compliance & privacy:
    - Ensure PII handling rules are followed — mask or avoid sending sensitive fields if not needed.
6. Alternative integration patterns:
    - Platform Events or Streaming API for high throughput
    - Use Change Data Capture to trigger external processing

Deliverables

- Named Credential & Remote Site / Auth Provider config
- API contract doc
- Integration test scripts

Phase 8 — Data Management & Deployment

Goal: ensure data quality and a reliable deployment pipeline.

Steps

1. Data loading:
    - Use Data Loader / Data Import Wizard to import sample leads.
    - Clean duplicates via Duplicate Rules and Matching Rules.
2. Backups:
    - Configure scheduled data exports or use AppExchange backup tool.
3. Deployment strategy:
    - Store metadata in Git.

    - Use SFDX / CI pipeline (GitHub Actions / Azure / Jenkins) to deploy to sandbox then prod.
    - Use change sets only for small items; prefer SFDX for automation.
4. Release checklist:
    - Run full Apex tests, validate Flows, do manual regression in a UAT sandbox.
5. Post-deploy checks:
    - Smoke test: create lead → scoring flow runs → lead assigned and notification sent.

Deliverables

- Data migration & cleansing scripts
- Deployment README & rollback plan

Phase 9 — Reporting, Dashboards & Security Review

Goal: create the reports and dashboards to prove impact; ensure data access is secure.

Steps

1. Reports to build:
    - Lead Conversion Rate by AI Score bucket
    - Avg Time to Contact for High vs Low scored leads
    - Lead Volume by Source vs Average Score
2. Dashboards:
    - “Sales Leaderboard” — Top Hot Leads, conversion funnel widgets.
    - Dynamic dashboards per user (view-as).
3. Security review:
    - Field Level Security for scoring fields
    - Session settings, login IP ranges, audit trail checks
    - Verify Lead_Scoring_Log__c access restricted to admins for sensitive payloads.
4. KPIs to measure (for mentor demo):
    - % of conversions from High priority leads

- Avg follow-up time for High priority leads

Deliverables

- Report & Dashboard pack
- Security checklist & audit results

Phase 10 — Final Presentation & Demo Day

Goal: demonstrate the end-to-end solution, hand off documentation and training.

Steps

1. Prepare demo script:
   - Show lead capture → auto scoring → priority assignment → rep notification → convert to opportunity.
2. Create final slide deck:
   - Problem, approach, architecture diagram, demo, results & next steps.
3. Handoff documentation:
   - Admin guide (how scoring works, how to change thresholds)
   - Developer guide (integration, key classes/flows)
   - Runbook for failures (who to contact, how to rescore)
4. Training:
   - Short video / 30-minute walkthrough for Sales team
   - Quick reference one-pager
5. Collect mentor / stakeholder feedback and log issues for follow-up.

Deliverables

- Demo deck & recorded walkthrough
- Handoff docs and training materials
- List of future enhancements (A/B test thresholds, improve model explainability)
