

AI Lead Intent Score Assignment :

Dear Candidate,

Congratulations on advancing to the second round! Your first-round PDF submission showcased your plan for an AI Lead Scoring Engine. Now, we want you to implement a simplified version of your solution to demonstrate your skills in Machine Learning, AI Techniques, LLM Understanding, Coding, Fast API, and API Knowledge. Below are the task details. Submit your deliverables within 48 Hours via email to "ceo@cleardeals.co.in".

AI Lead Scoring Dashboard Prototype

Objective

Build and deploy a web-based lead scoring dashboard that implements your first-round plan (or a simplified version). The dashboard should predict lead intent using a machine learning model and a rule-based LLM Re ranker, delivering scores via a Fast API endpoint. Deploy to Netlify or GitHub Pages.

Scope

- **Problem:** Brokers waste time on low-intent leads, slowing conversions.
- **Goal:** Deliver an "Intent Score" (0–100) to prioritize high-intent prospects, targeting a 2–3x conversion lift.
- **Task:** Implement a dashboard with a form to input lead data, a table to display scored leads, and a backend to compute scores using a machine learning model and a rule-based LLM Re ranker.
- **Unique Idea:** Use a machine learning model of your choice (e.g., gradient-boosted model like XGBoost or scikit-learn's GradientBoostingClassifier) for

initial scoring and a rule-based LLM Re ranker to adjust scores based on comments (e.g., keywords like "urgent").

- **Time:** 48 Hours, Solo Work.
- **Submission:** Deployed app URL, GitHub repository link, and a single PDF (max 3 pages). Submit within 48 Hours via email to "ceo@cleardeals.co.in".

Requirements

1. Data

- **Dataset:** Source a dataset (~10000 rows) either from Kaggle or UCI ML Repository or create a Synthetic Dataset (~10000 rows) with meaningful Relationships and Patterns between columns. Ensure the Dataset includes fields such as:
 - Phone Number (e.g., "+91-9876543210")
 - Email (e.g., "john.doe@test.com")
 - Credit Score (e.g., 300–850)
 - Age Group (e.g., "18–25", "26–35", "36–50", "51+")
 - Family Background (e.g., "Single", "Married", "Married with Kids")
 - Income (e.g., 100,000–10,00,000 INR)
- Include at least 5 features relevant to predicting lead intent, aligned with your first-round plan or inspired by the examples.
- This 5 should at least be there but you should consider more fields as well.
- For Dataset do not take randomly created data. Take data from Kaggle or UCI ML Repository and also you can create a synthetic dataset as well but it should make sense; the columns should have relationships and patterns.
- Save the dataset as a CSV file in your GitHub repository.
- Use this dataset for training and testing.

2. Frontend (HTML, CSS, JavaScript)

- Build a single-page application (SPA) with a responsive UI using React, Vue.js, vanilla.Js or next.js.
- Include two sections:
 - **Form:** Inputs for your chosen dataset fields (minimum 5, plus a Comments text field) and a consent checkbox ("I consent to data processing").
 - **Table:** Display leads with at least 3 fields (e.g., Email, Initial Score, Reranked Score) and Comments.
- Use Tailwind CSS for styling (Optional)
- Ensure mobile responsiveness. (Optional)

3. Backend (Fast API)

- Develop a Fast API backend with a POST endpoint (/score) to:
 - Accept lead data (JSON payload).
 - Compute an initial intent score using your chosen ML model.
 - Apply a rule-based LLM Re ranker to adjust the score based on Comments.
 - Store leads temporarily in memory (e.g., Python dictionary).
- Return JSON with initial and reranked scores.
- Implement input validation (e.g., valid email, non-negative numeric fields).

4. Machine Learning Model

- Implement a machine learning model of your choice (e.g., gradient-boosted model like XGBoost or scikit-learn's GradientBoostingClassifier, logistic regression, or similar) to predict lead intent (binary: high/low intent, scaled to 0–100).
- Use at least 5 features from your dataset, aligned with your first-round plan or inspired by the example fields.
- Train on the sourced or synthetic dataset (~10000 rows) with meaningful relationships and patterns. (~10000 rows).

- Save the model (e.g., using joblib, .pkl File) and load it in the Fast API app.
- Justify your model choice in the PDF.

5. LLM-Inspired Re ranker

- Implement a rule-based Re ranker to simulate LLM behavior:
 - Parse Comments for keywords (e.g., "urgent" → +10, "not interested" → -10).
 - Adjust the ML model's score (cap at 0–100).
- Document Re ranker logic in the PDF (e.g., keyword list, scoring rules).

6. Deployment

- Deploy the frontend to Netlify or GitHub Pages.
- Host the Fast API backend on a free platform (e.g., Render, Fly.io) or use Netlify Functions for serverless integration.
- Ensure the app is accessible via a public URL with API latency <300 ms.

7. Compliance

- Ensure the consent checkbox is mandatory in the form.
- Use only dummy data; no real PII collection.
- In the PDF, describe one compliance measure from your first-round plan (e.g., DPDP-ready, India-based PII storage) and its implementation (e.g., mock consent).

8. Optional Features (Bonus)

- Add a table filter to sort leads by reranked score.
- Persist leads in browser local Storage for page refresh.
- Display a score distribution chart using Chart.js.

Deliverables

1. **Deployed App:** Public URL of the running app.

2. **GitHub Repository:** Public repo with:

- Organized code (/frontend, /backend, /model, /data).
- Sourced or Synthetic Dataset CSV, trained model file.
- README.md with instructions to run locally (e.g., pip install -r requirements.txt, npm install).

3. **PDF Report** (max 3 pages):

- Sections: Solution Overview, Architecture, ML Model, LLM Re ranker, Compliance, Challenges & Mitigations.
- Explain how you implemented your first-round plan (or adapted it).
- Justify your ML model and dataset choices.
- Address one data quality challenge and one compliance issue from your first-round plan, with mitigation results.
- Discuss one technical metric (e.g., precision) and one business metric (e.g., conversion lift) to judge success.
- Describe one implementation challenge and how you overcame it.
- Include your Name, LinkedIn, and GitHub at the top.

4. **Submission:** Email deliverables to "ceo@cleardeals.co.in" within 48 Hours. In the email, add your PDF should be in this format: 'Your_Name.zip', 'Your_Name.pdf' and make sure in PDF your public URL should be there.

Evaluation Criteria

- **Functionality:** App implements ML model and Re ranker per your plan.
- **ML/AI Skills:** Effective model training, feature selection, Re ranker logic.
- **Coding Skills:** Clean, commented code with robust Fast API integration.
- **API Knowledge:** Efficient API design, latency

Guidelines

- Use free tools (e.g., Netlify, Render, scikit-learn, Fast API, Tailwind CSS).

- Use a Sourced or Synthetic dataset with fields of your choice, inspired by the examples.
- Test thoroughly to ensure API latency <300 ms.
- Avoid complex models (e.g., deep learning) to stay within time limits.
- Reuse ideas from your first-round PDF where applicable.

Notes

- Clock starts upon receiving this email.
- Work solo, remotely.

Best of luck !!

Best regards,

PropTech ClearDeals PVT. LTD.