Project 3 – Initial Proposal; Mahin Naveen

Title: Evaluating Natural-Language Reformulation for Tabular Machine Learning with Large Language Models

I. Introduction and Problem Statement

Traditionally, tabular data and natural-language data are handled using very different modeling approaches. Classical algorithms such as logistic regression, random forests, and XGBoost remain the strongest tools for structured datasets, while transformer models now dominate text, vision, and general purpose reasoning tasks.

Recently, there has been growing interest in whether large language models (LLMs) can actually learn from tabular data when the rows are rewritten into short natural language descriptions. If this method works reliably, it could hopefully give a new way to use LLMs on classical ML problems without feature engineering or specialized architectures.

The goal of this project is to run a careful, controlled evaluation of this idea. I will study whether converting rows of tabular data into natural language sentences allows LLMs to perform classification or regression competitively with classical baselines. I will also compare different prompting strategies (zero-shot, few-shot, chain-of-thought) and test whether small-scale LoRA fine tuning improves performance. The central question I want to answer is:

"Can natural-language reformulation meaningfully close the performance gap between LLMs and classical ML models on standard tabular tasks?"

II. Data Sources

I will use three well-known, publicly available benchmark datasets to keep the study consistent and reproducible:

- UCI Adult Income Dataset
    - Binary classification (income > $50k).
    - Clean and widely studied; good for comparing model behavior on demographic/ categorical features.

- UCI Bank Marketing Dataset
    - Binary classification (term-deposit subscription).
    - Mix of categorical and numerical fields; suitable for testing LLM sensitivity to different feature types.

- California Housing Dataset (from the StatLib/UCI repository via scikit-learn)
    - Regression task predicting median house value.
    - Provides contrast with the two classification datasets.

All three datasets are public, labeled, easy to preprocess, and accepted in standard ML benchmarking.

III. High-Level Methods and Techniques

- Classical Baselines (Tabular Models)
  These models will establish strong performance references:
    - Logistic Regression
    - Random Forests
    - XGBoost
    - Support Vector Machines (if appropriate for the dataset)

- Natural-Language Reformulation + LLM Prediction
  Each row will be rewritten as a short natural-language description.
    - Example (Adult Income):

"A 39-year-old individual with a Bachelor's degree works 40 hours a week and is employed as a tech support worker."

    - This format will be designed carefully to stay consistent across datasets.

- LLM Prompting Approaches
    - Zero-shot prompting: Provide only the reformulated instance and ask the LLM for a prediction.
    - Few-shot prompting: Include several example rows in natural-language form.
    - Chain-of-thought prompting: Ask the LLM to reason step-by-step about the decision.

- Light Fine-Tuning (LoRA)
    - For at least one smaller open-source LLM, I will run a light LoRA fine-tuning pass on the reformulated examples and compare improvements over prompting-only performance.

- Evaluation and Analysis
    - Accuracy, F1 score (classification)
    - RMSE or MAE (regression)
    - Calibration metrics such as Brier score
    - Error breakdowns by feature types (e.g., categorical vs. continuous)
    - Qualitative analysis of LLM reasoning, especially when it produces mistakes

IV. Products to Be Delivered

The final deliverables will include:

- A complete Jupyter notebook containing
    - data preprocessing steps
    - classical model training and evaluation
    - natural-language reformulation logic

    - prompting experiments with multiple LLMs
    - optional fine-tuning experiments
    - graphs, tables, and comparisons of all models
    - documentation and comments throughout
- The initial 2-page proposal committed to the Project 3 Git repository.
- A final written report (up to 10 pages) covering:
    - the motivation and problem statement
    - dataset descriptions
    - methods and experimental setup
    - results, comparisons, and discussion
    - limitations and suggestions for future work
- Recorded 10-minute presentation video summarizing the project.
- Use of AI Document listing any AI-assisted code generation or debugging steps, following the required formatting.