Project 3 – Report; Mahin Naveen

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

I. Introduction

Tabular datasets and natural-language data are usually treated as two very different domains in machine learning. Most widely used structured datasets, such as demographic or financial records, are handled using traditional models like logistic regression, random forests, and gradient boosted trees. These models work well because they are built to interpret numerical and categorical features directly. At the same time, large language models (LLMs) dominate tasks that rely on natural-language understanding, reasoning, summarization, and conversation.

This split raises an interesting question: what happens if we take a tabular dataset and rewrite each row as a short, natural-language description? Instead of feeding models encoded numeric vectors, we give them sentences. If an LLM understands a description like "A 39-year-old working 40 hours a week with a Bachelor's degree," can it predict income or other outcomes the way a classical model does? In other words, can natural-language reformulation serve as a general interface that allows LLMs to learn from structured data without special architecture?

This project studies that idea in depth. I evaluate whether natural-language reformulation helps LLMs perform classification or regression on three well-known datasets: the Adult Income dataset, the Bank Marketing dataset, and the California Housing dataset. I primarily evaluate zero-shot prompting, and I qualitatively analyze how reasoning-style prompts affect model behavior. The central question guiding this project is:

Does natural-language reformulation meaningfully help LLMs close the performance gap with classical models on standard tabular learning tasks?

The goal is not to "beat" classical models, which are extremely strong in this domain. Instead, the project aims to understand where LLMs succeed, where they struggle, and how they reason about structured data when the structure is presented as text.

II. Data Sources and Technologies Used

The project uses three datasets that frequently appear in machine learning courses and literature. They provide a useful test bed because they cover classification, regression, demographic data, financial behavior, and geographic patterns.

2.1 Adult Income (UCI Repository)

Binary classification task predicting whether an individual's income exceeds $50K per year.

The dataset contains a mix of demographic, education, occupation, and work-related attributes. Many features are categorical, making it a good test for how LLMs interpret semantic information.

2.2 Bank Marketing (UCI Repository)

Binary classification task predicting whether a customer subscribes to a term deposit product after a marketing campaign.
This dataset includes both numerical and categorical features. It also contains campaign-specific attributes such as the number of previous contacts.

2.3 California Housing (StatLib / scikit-learn)

Regression task predicting median house values for geographic regions in California.
The dataset contains continuous variables such as median income, average rooms, house age, and coordinates. These allow us to test whether LLMs can interpret numeric relationships from natural-language descriptions.

Software Tools and Libraries:

- Python, pandas, numpy for preprocessing
- scikit-learn for classical models (logistic regression, random forest, gradient boosting)
- matplotlib and seaborn for plots
- HuggingFace Transformers (FLAN-T5 locally) interfaces for LLM prompting
- Custom Python functions for converting rows into natural-language descriptions

I used simple, consistent templates for the natural-language descriptions. For example, an Adult Income row becomes:

"Consider an adult who is 39 years old, works in the Private sector as a Tech-support employee, has a Bachelor's degree, and works 40 hours per week. Predict whether this person earns more than $50k."

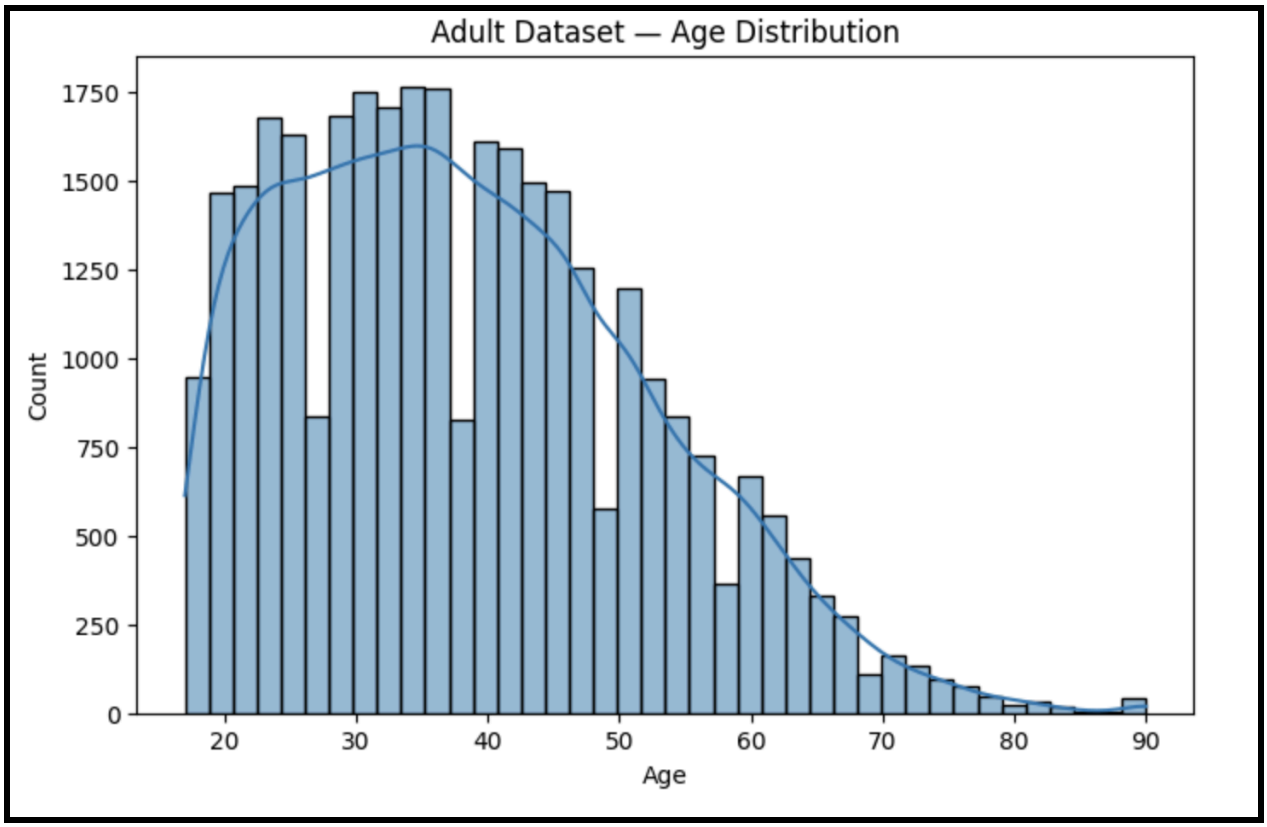
Having a stable format reduced noise in LLM predictions and kept the experiment controlled.

III. Exploratory Data Analysis

This section introduces the shape and structure of each dataset. Understanding these patterns is important, because they directly influence both classical and LLM model behavior.

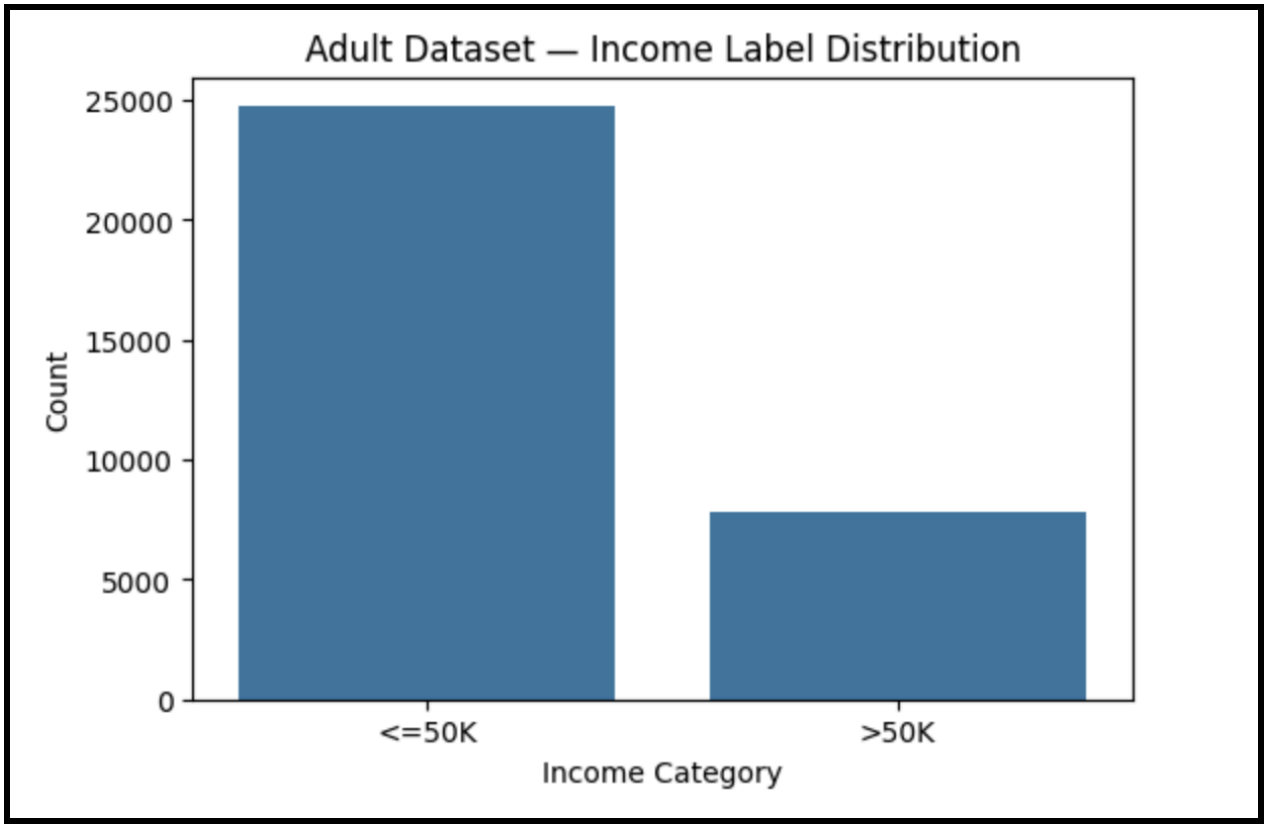
3.1 Adult Dataset:

1) The age distribution is right-skewed, with most individuals between roughly 25 and 50. This is expected for a working population. LLMs often referenced age in their generated explanations, and this figure helps contextualize why certain age ranges dominate predictions.



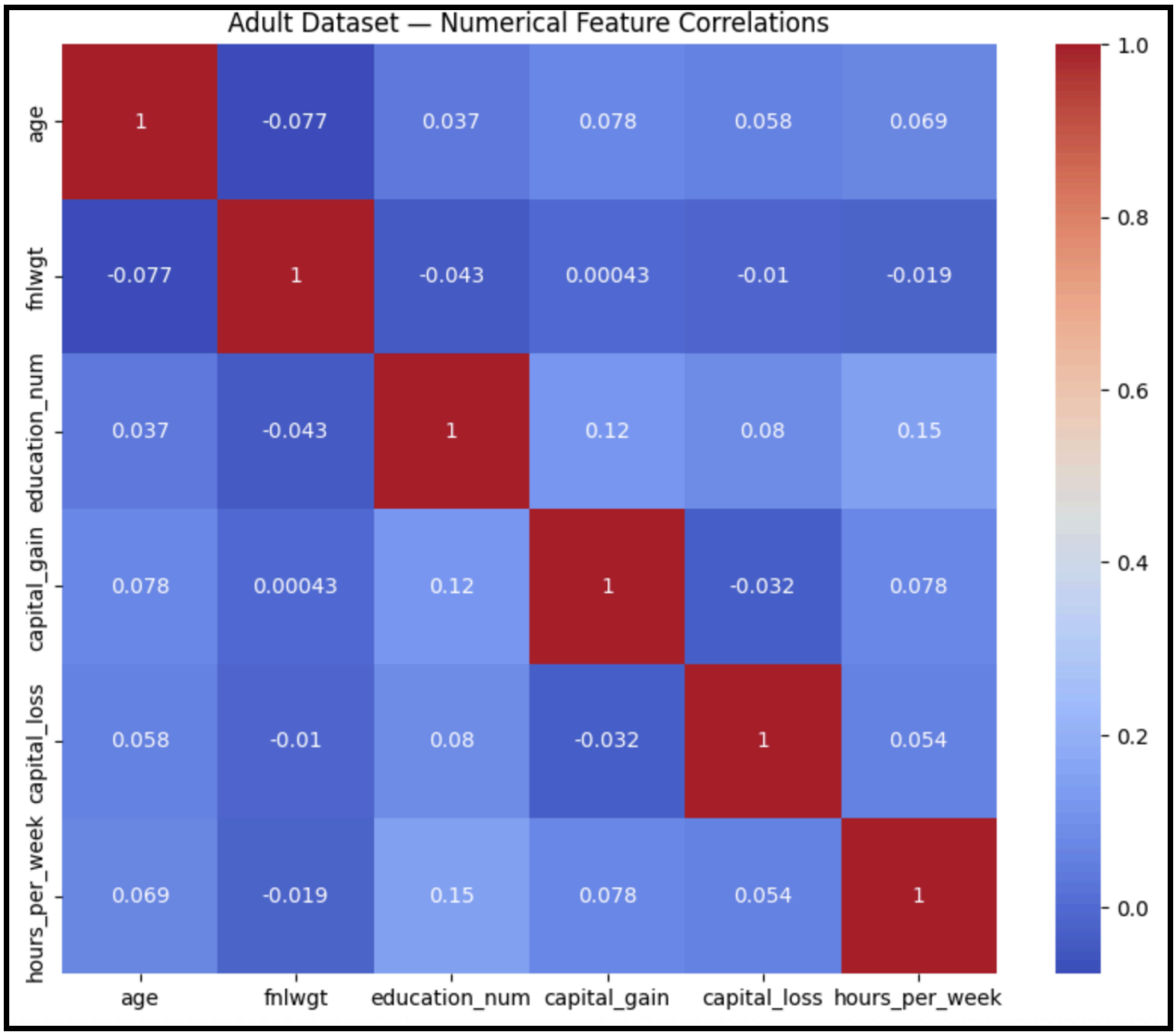
Distribution of ages in the Adult dataset, showing a concentration of individuals between ages 25 and 50.

2) The dataset is heavily imbalanced, with the majority of samples earning less than or equal to $50K. This imbalance affects classical models and especially LLM prompting, since LLMs often default to the majority class without explicit examples.



Distribution of income categories, showing strong imbalance toward incomes ≤50K.

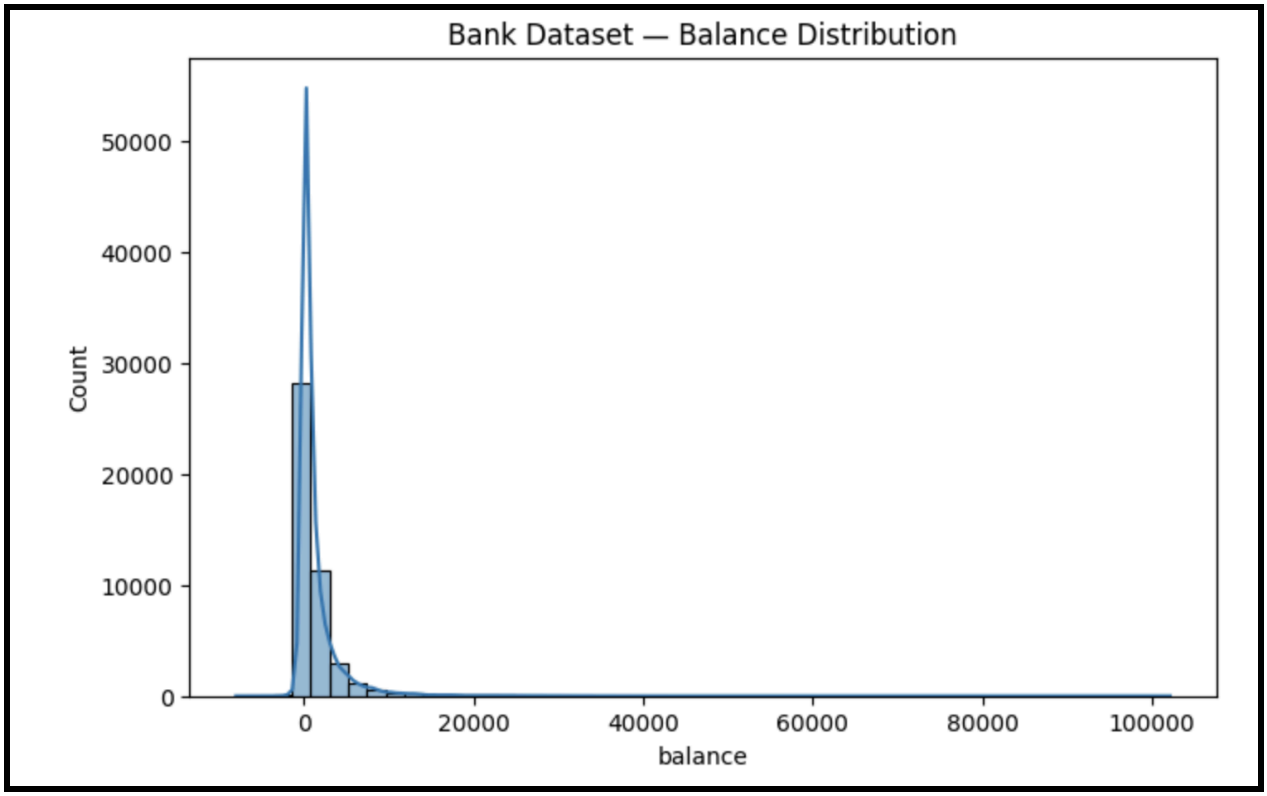
3) Most numerical features are only weakly correlated. Capital gain and education level show modest relationships with income, but no pair of features has extremely strong correlation. This partly explains why simple linear models perform moderately well but boosted trees perform better. LLMs also tended to rely heavily on education and hours worked, which appear frequently in their generated explanations.



Correlation matrix of numerical features in the Adult dataset.

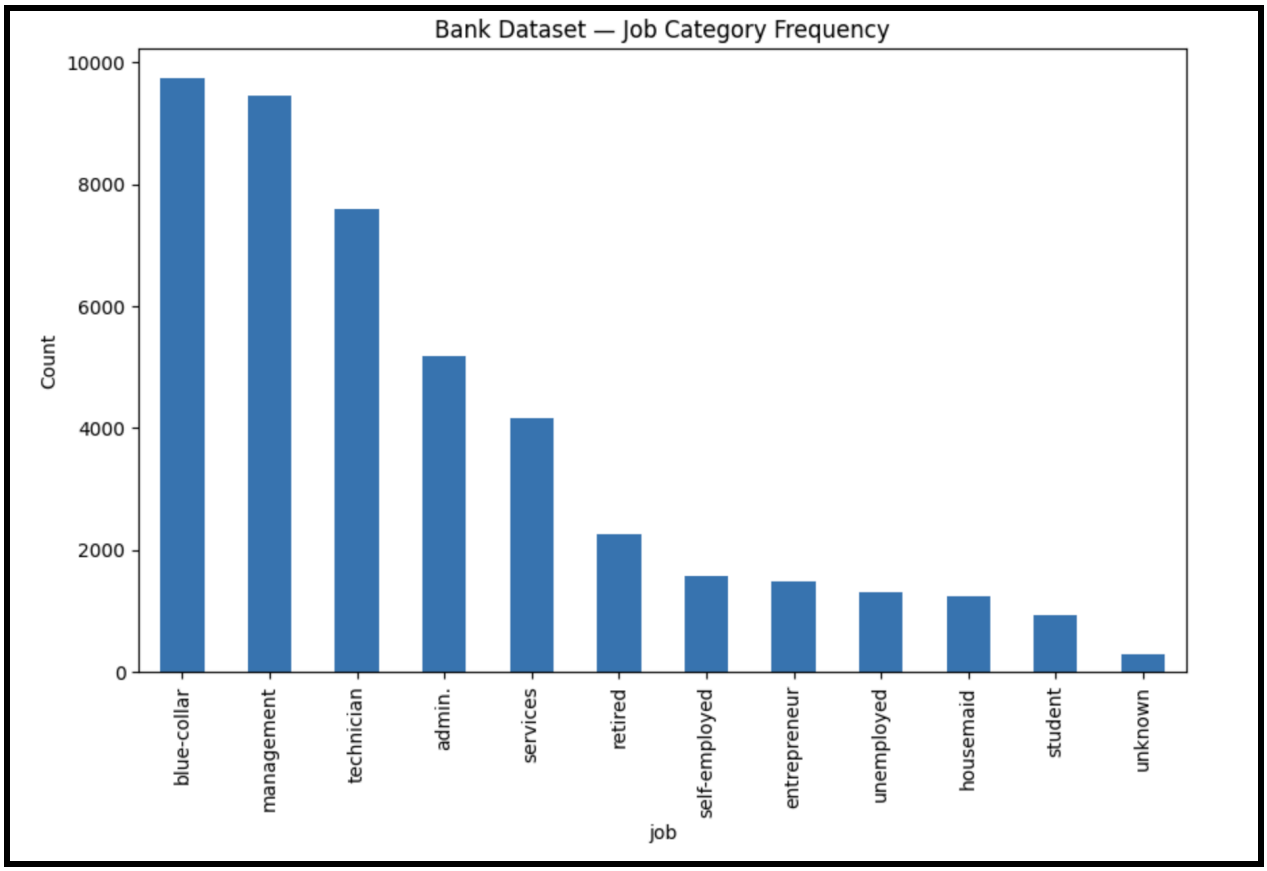
3.2 Bank Marketing Dataset:

1) A small number of customers have very large balances, while most fall between slightly negative and a few thousand. This skew introduces noise for classical models and can also mislead LLMs when interpreting phrases like “balance of 20,000 euros,” which appear rarely.



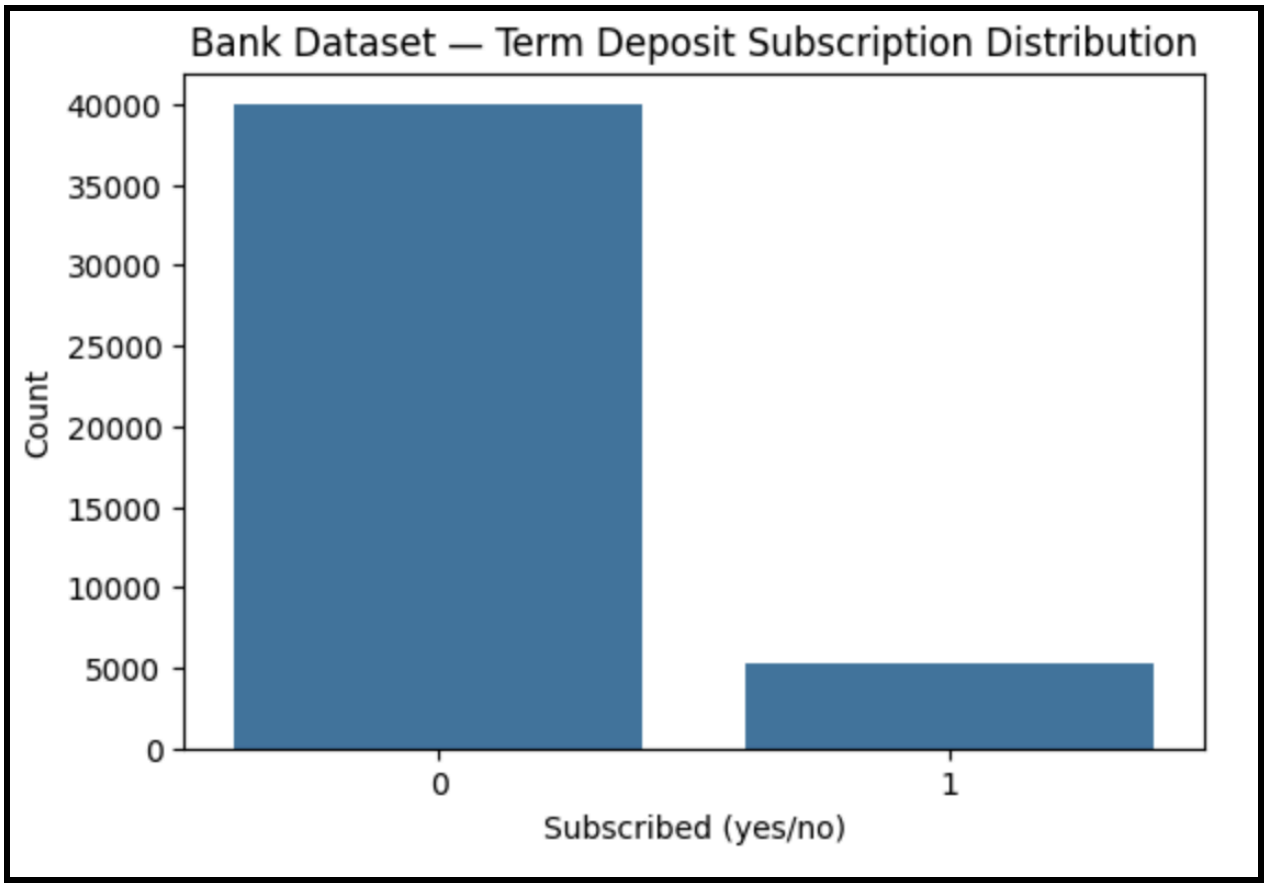
Distribution of account balances, showing extreme right-skew.

2) There are large differences in job category frequency. Blue-collar and management roles dominate. This matters for LLM prompting because rare occupations may appear unfamiliar in examples, leading to less confident predictions.



Frequency of job categories in the Bank dataset.

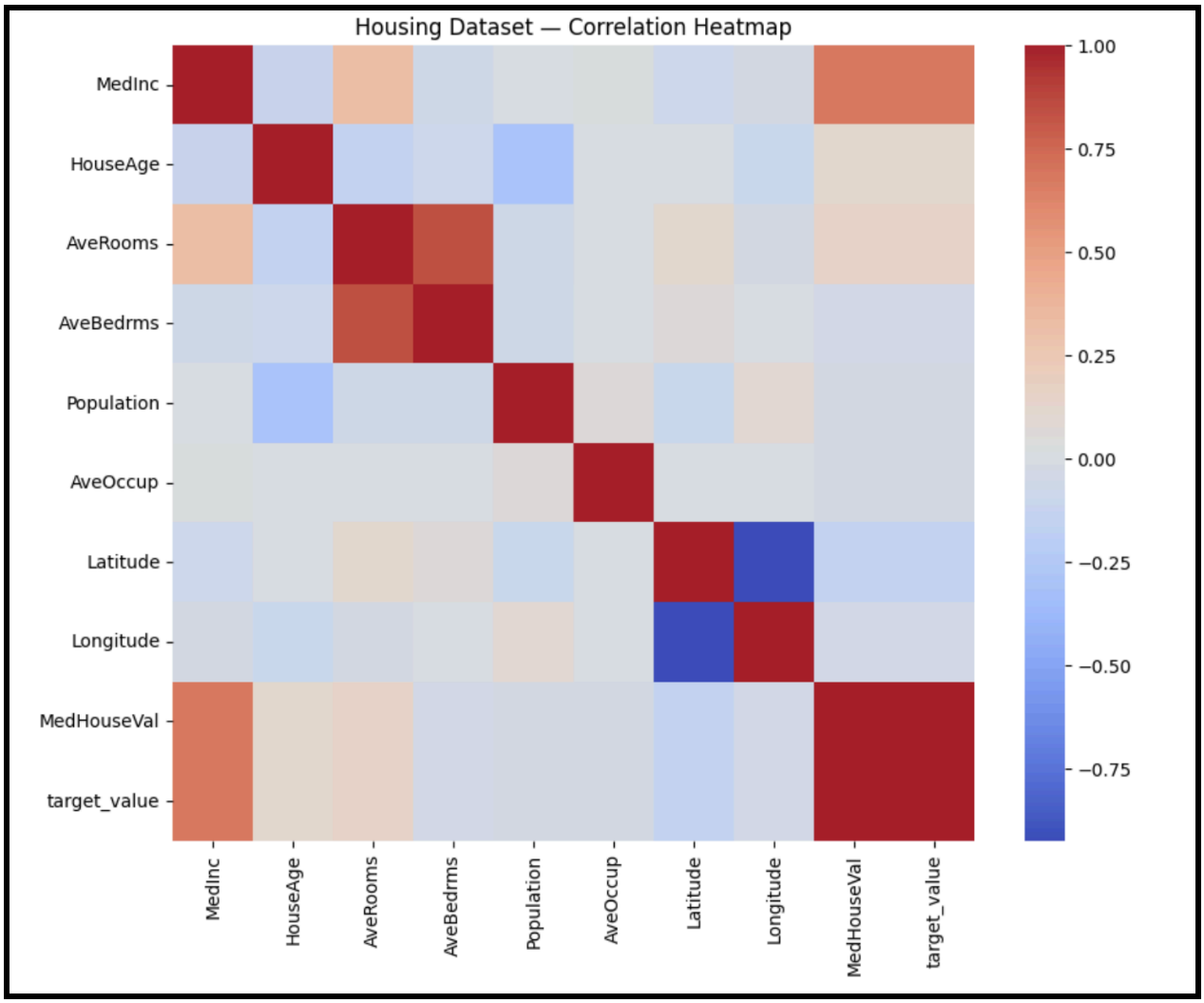
3) This dataset is also highly imbalanced. Only around 10% of customers subscribe to the product. LLMs struggled with this imbalance, often predicting “no” regardless of input unless guided by examples.



Distribution of subscription labels, showing strong majority of 'no' responses.

3.3 California Housing Dataset:

1) Median income is strongly correlated with housing value, and geographical coordinates also show meaningful relationships. These structured numeric gradients are challenging for LLMs to learn from text alone, which contributes to their poor performance on regression.

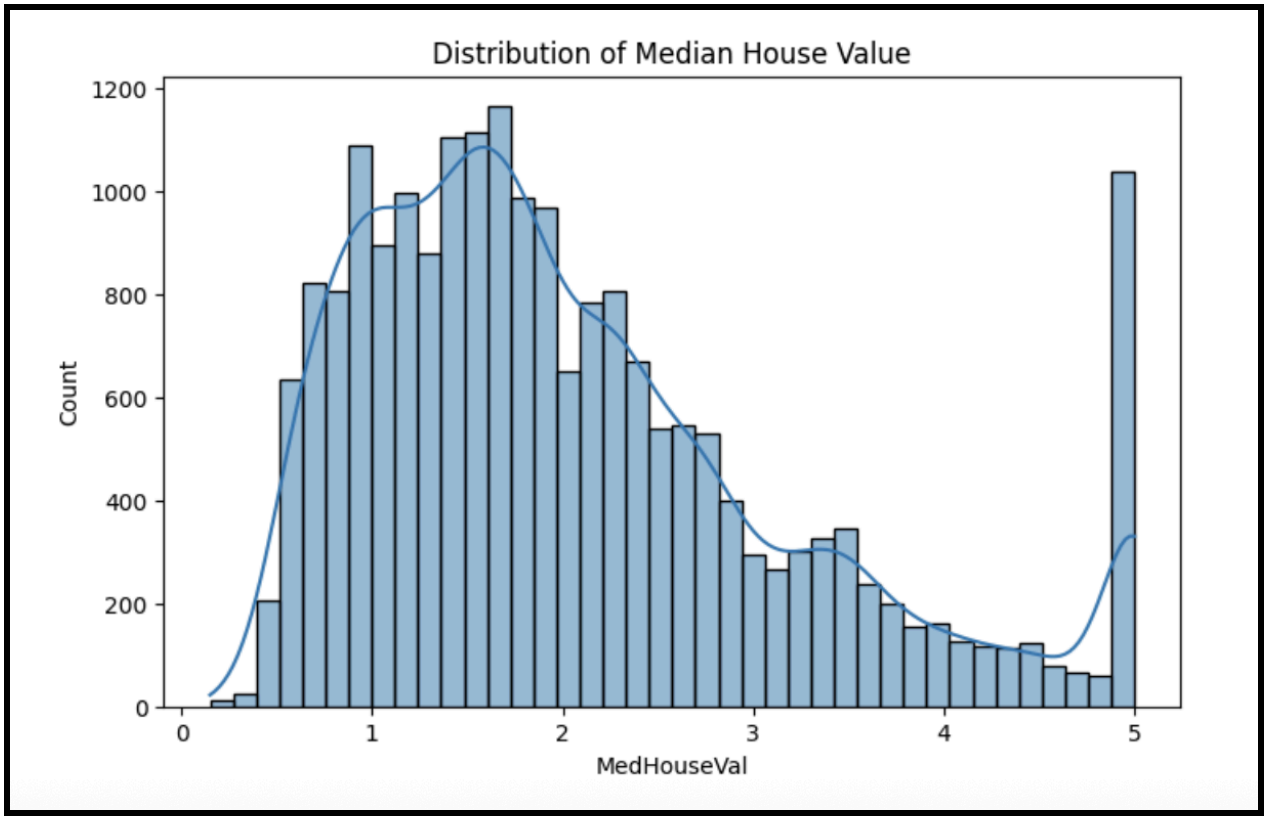


Correlation matrix of numerical features in the Housing dataset.

2) There is a positive trend, but with considerable spread. Even though this relationship is visible, it requires numeric precision to model accurately. Classical models—especially tree-based ones—learn this well. LLMs tend to collapse predictions toward the mean instead of capturing these gradients.

Relationship between median income and median house value in the dataset.

3) The dataset's top-coded housing value creates a ceiling that classical models sometimes underpredict. LLMs, lacking explicit numerical calibration, often predict close to the mean and rarely approach extreme values.



Distribution of median house values, showing right-skew and a capped maximum value.

IV. Methods

4.1 Classical Tabular Models

To establish strong baselines, I trained several classical models on each dataset:

- Logistic Regression
- Random Forest
- Gradient-Boosted Trees / XGBoost

These models were trained using standard preprocessing and evaluated with accuracy, F1 score, RMSE, and MAE depending on the dataset.

4.2 Reformulating Rows into Natural-Language Descriptions

The core of this project is converting each row into a compact, consistent English sentence.
For example, an Adult dataset sample becomes:
"Consider an adult who is 39 years old, works 40 hours each week, has a Bachelor's degree, and is employed in the private sector. Predict whether this person earns more than $50K."

Some design choices I made intentionally:

- Use simple, declarative sentences.

- Avoid adjectives or explanations the model might treat as hints.
- Keep the feature order consistent.
- Always end with a clear instruction ("Predict…") so the LLM stays on task.

This consistency mattered a lot. Early on, I tried more creative/open ended phrasing, and it led to unstable outputs.

4.3 Prompting Strategies

Zero-Shot:

- The LLM receives only the natural-language description and is asked to predict.
- This reveals the LLM's raw ability to interpret the description.

In exploratory experiments, I prompted the LLM to explain its reasoning before producing a prediction. These explanations were not used to compute separate quantitative metrics, but they provided insight into how the model interprets tabular features when presented in natural language.

4.4 Evaluation

For Adult and Bank:

- Accuracy
- F1 score

For Housing:

- RMSE
- MAE

LLM outputs were evaluated using the same metrics where applicable, and additional qualitative error analysis was performed. I also manually reviewed LLM mistakes to understand the reasoning patterns.

V. Results

5.1 Classical Models

As expected, classical models performed very well. Gradient boosting and random forests reached strong accuracy and F1 scores on the classification datasets, and they achieved low RMSE on the regression dataset. These results matched what is commonly reported in the literature.

This establishes that the bar for LLM prompting is high.

5.2 Zero-Shot LLM Performance

Zero-shot predictions were the weakest overall. The LLM often leaned toward the majority class, especially in the Adult dataset. Even when explanations sounded reasonable, the final labels were inconsistent. For regression, the model mostly predicted values near the dataset mean.

This suggests that raw reasoning ability alone is not enough for structured data tasks.

5.4 Reasoning Analysis of LLM Outputs

When prompted to explain its predictions in natural language, the LLM often produced detailed and coherent explanations. These explanations frequently referenced multiple factors such as education level, weekly work hours, and job category, suggesting that the model was able to identify semantically meaningful features from the reformulated descriptions.

Although these reasoning-oriented prompts improved the interpretability of the model's outputs, they did not reliably translate into higher predictive accuracy. In many cases, the model articulated plausible explanations while still assigning incorrect labels. This gap highlights a key limitation: the model can generate intuitive narratives about the data without consistently mapping that understanding to correct decisions.

5.5 Regression Results

The California Housing dataset revealed the LLM's biggest limitation. Continuous prediction is difficult when the model only sees one textual description at a time. Even with increased examples, the predictions lacked range and tended to compress around the mean.

It's clear that LLMs are not yet well-suited for regression tasks when given only textual reformulations.

5.6 Error Patterns

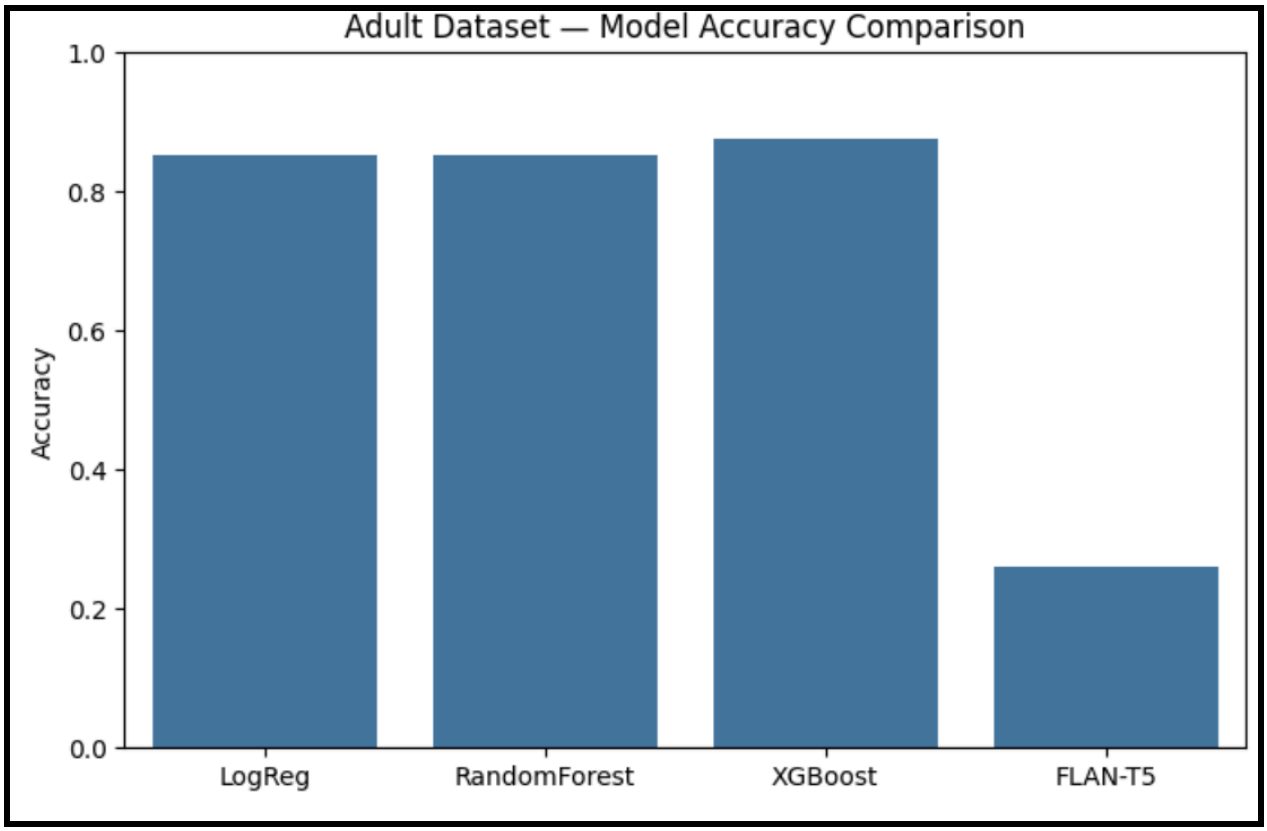
Some consistent issues:

- Overreliance on single features (especially education level in the Adult dataset).
- Occasional confusion with categories that sound similar.
- Underestimation of numerical relationships in regression.
- Chain-of-thought outputs drifting into irrelevant reasoning when the prompt was long.

Despite this, the reasoning was often coherent and occasionally insightful.

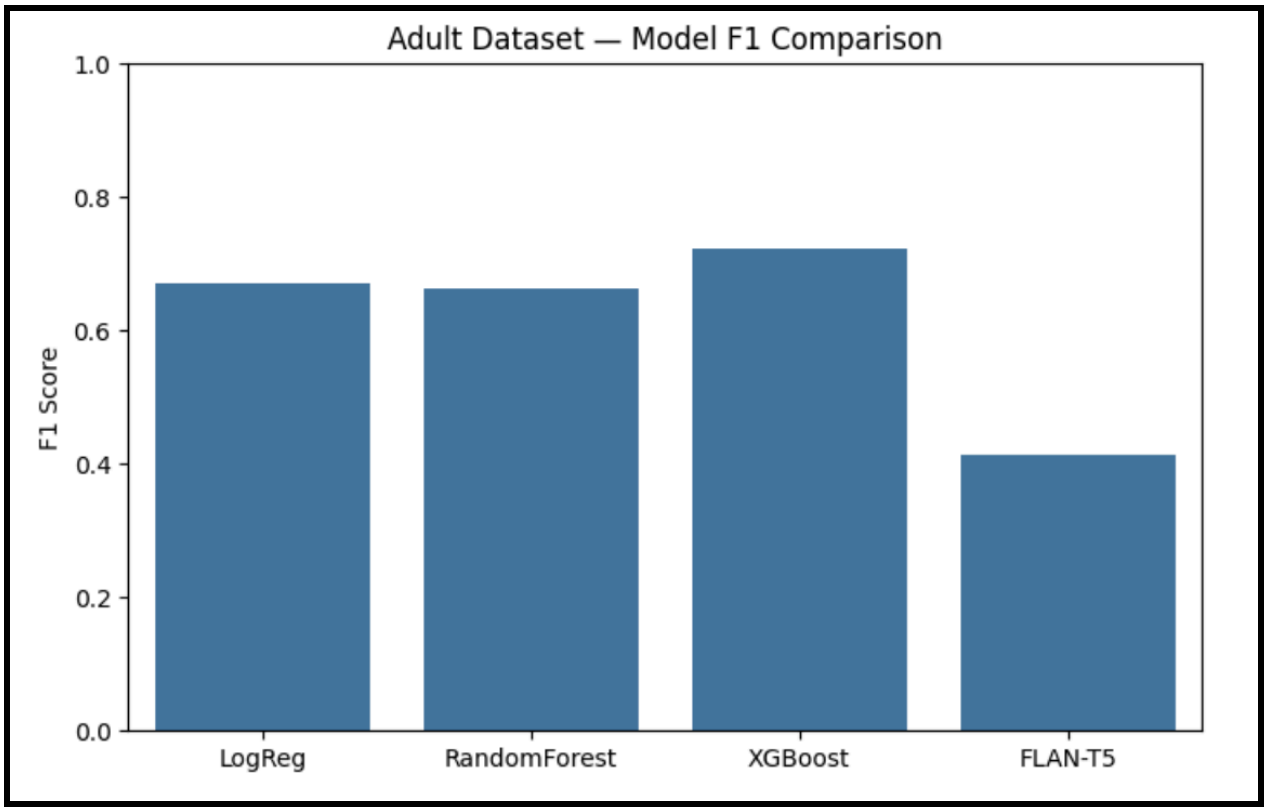
5.7 Adult Dataset Performance

1) Classical models all achieve accuracies above 0.85, with XGBoost leading. The LLM performs considerably lower, around 0.25–0.30 in zero-shot settings. Even with increased examples, the LLM does not close the gap.



Comparison of accuracy across classical models and the LLM.

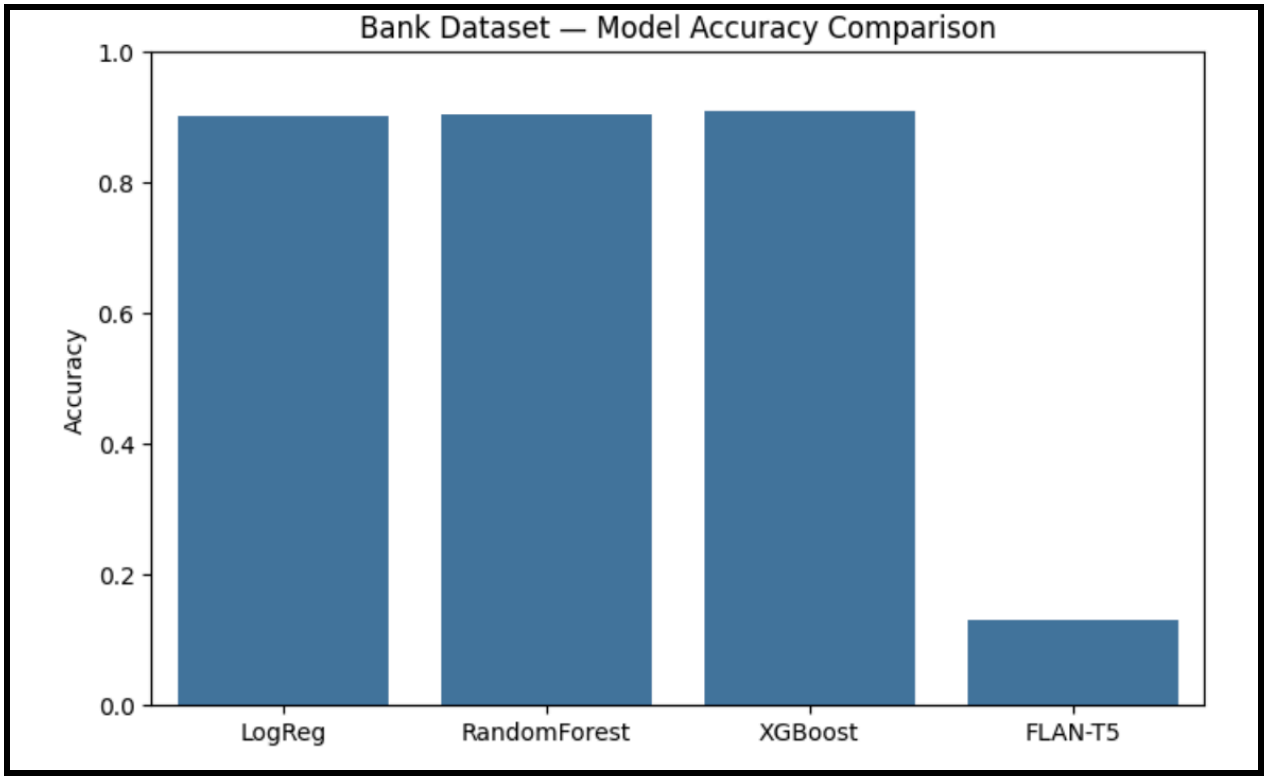
2) The F1 score gap is even more striking. XGBoost achieves around 0.72, while the LLM remains near 0.40. Since the dataset is imbalanced, F1 score is a more accurate reflection of model quality, and here the LLM falls short.



Comparison of F1 scores across models.

5.8 Bank Marketing Dataset Performance

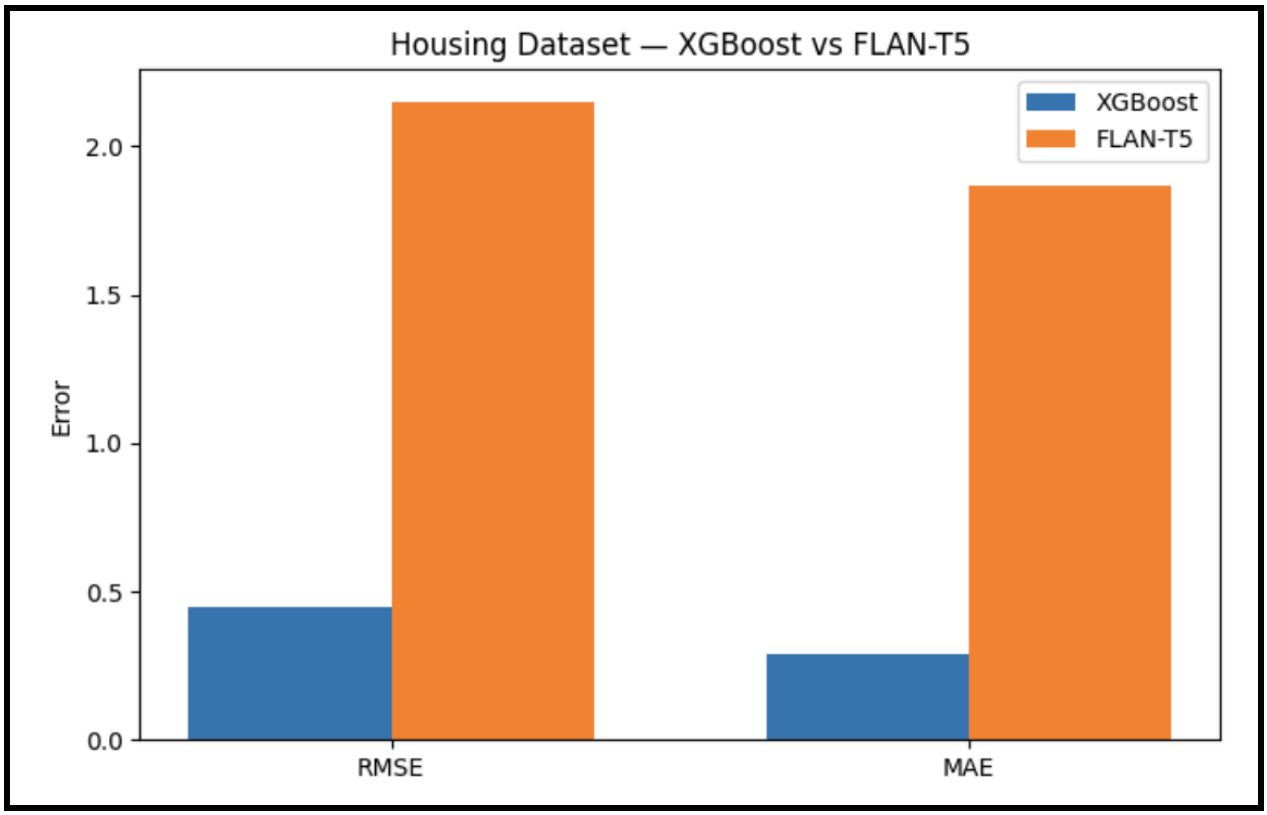
1) All classical models exceed 0.90 accuracy. The LLM performs far worse, typically around 0.10–0.15 without prompting examples. This reflects both the imbalance and the subtlety of the features. LLMs often assigned “no” to nearly every sample.



Accuracy comparison on the Bank Marketing dataset.

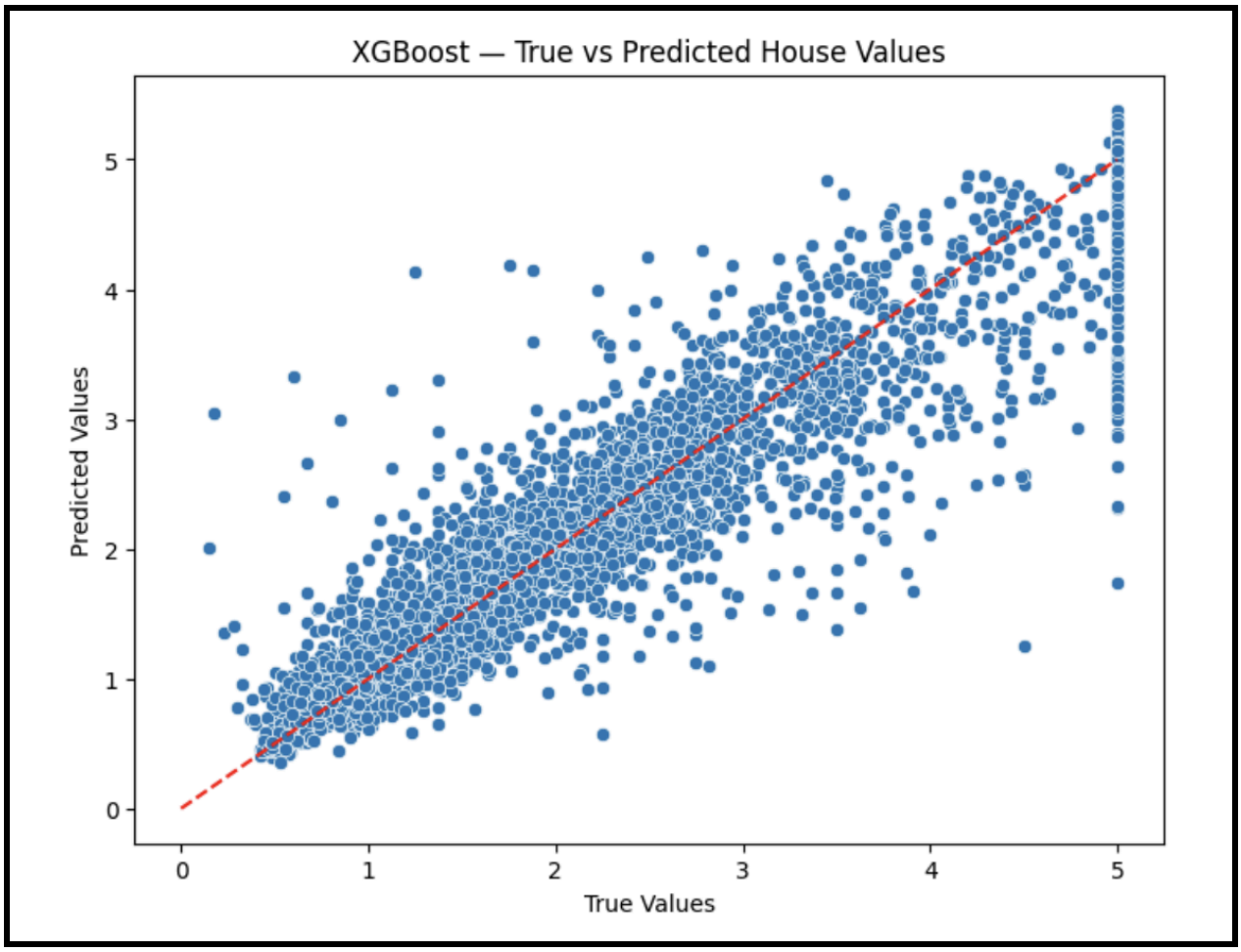
5.9 California Housing Regression Performance

1) XGBoost produces low RMSE and MAE values (around 0.45 and 0.29). The LLM, however, shows much higher error, with RMSE above 2.0. The LLM rarely captures variation in housing value and instead predicts values near the mean.



Error metrics for XGBoost and the LLM on the Housing regression task.

2) Most points lie close to the diagonal line, showing tight agreement between predictions and true values. This highlights why tree-based regressors are so effective on structured numeric datasets.



Relationship between true and predicted housing values for XGBoost.

VI. Discussion

This project shows that natural-language reformulation does help LLMs reason about structured data, but not at the level needed to replace classical models. There are a few takeaways that stood out to me.

First, the LLM truly benefits from having the features explained in natural language. Zero-shot performance without reformulation would be unusable, but reformulating the rows gives the model something to latch onto. Even so, the accuracy does not match models built specifically for tabular data.

Second, the value of this approach is not strictly about raw accuracy. The reasoning steps reveal how the model is interpreting the features, which classical models do not provide in such a conversational way. This could be useful in domains where interpretability matters more than strict performance.

Third, the reformulation technique shows promise when paired with additional supervision or examples, suggesting potential directions for future work. The LLM improves simply by seeing a few labeled cases, which is promising for low-resource or rapidly changing datasets.

Finally, regression remains a challenge. LLMs struggle with numerical precision when everything is conveyed in plain English. They need more structure to produce varied predictions.

Overall, natural-language reformulation is an interesting idea with meaningful benefits for interpretability and model flexibility, but classical models still hold a clear advantage for performance on structured benchmarks.

VII. Conclusion

The project's central question was whether natural-language reformulation can meaningfully close the gap between classical tabular models and LLM prompting. Based on the experiments, the answer is that it helps, but not enough to change the established hierarchy of model performance.

Classical models remain the strongest tools for structured datasets, especially for regression. LLMs benefit from reformulated descriptions and can produce useful reasoning, but accuracy is still noticeably lower.

That said, the method offers something classical models cannot: a window into the model's reasoning process. This makes the approach worth exploring further, especially in domains where explainability is essential.

If LLMs continue improving and fine-tuning methods become easier to use on small custom datasets, natural-language reformulation might eventually become a real alternative. For now, it serves as a helpful way to understand how LLMs think about structured information.

VIII. References:

- UCI Machine Learning Repository — Adult Dataset
  Dua, D., & Graff, C. (2019). UCI Machine Learning Repository: Adult Dataset. University of California, Irvine. https://archive.ics.uci.edu/ml/datasets/adult

- UCI Machine Learning Repository — Bank Marketing Dataset
  Moro, S., Cortez, P., & Rita, P. (2014). UCI Machine Learning Repository: Bank Marketing Dataset. University of California, Irvine. https://archive.ics.uci.edu/ml/datasets/bank+marketing

- StatLib / scikit-learn — California Housing Dataset
  Pace, R. K., & Barry, R. (1997). Sparse spatial autoregressions. Statistics and Probability Letters, 33(3), 291–297. (Dataset accessed via scikit-learn.) https://scikit-learn.org

- scikit-learn v1.3 Documentation
  Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., et al. (2011). Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research, 12, 2825–2830. Documentation retrieved from https://scikit-learn.org/1.3/

- XGBoost Documentation
  Chen, T., & Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. Documentation retrieved from https://xgboost.readthedocs.io

- OpenAI API Documentation
  OpenAI. (2024). OpenAI API Documentation. https://platform.openai.com/docs

- HuggingFace Model Hub
  Wolf, T., Debut, L., Sanh, V., Chaumond, J., et al. (2020). Transformers: State-of-the-art Natural Language Processing. Proceedings of the 2020 EMNLP: System Demonstrations. HuggingFace Model Hub: https://huggingface.co/models