Introduction

This report summarizes the evaluation of several machine learning models and a fine-tuned Large Language Model (LLM) for classifying developer roles from commit messages and metadata. The goal was to identify the most effective approach for this multi-class task.

Results

Traditional Model Performance (Validation Set) Confusion Matrix - Logistic Regression (Validation Set)

	macro_f1	accuracy	precision	recall
Fine-tuned LLM	0.984599	0.986667	0.989329	0.980769
XGBoost	0.812510	0.813333	0.818597	0.808337
Logistic Regression	0.780700	0.777778	0.778339	0.784588
Random Forest	0.761118	0.751111	0.765595	0.758689

Final Evaluation on Test Set (Fine-tuned LLM)

```
FINE-TUNED LLM PERFORMANCE (Test Set)
============================================================
Test loss 0.0797 accuracy 0.9822 Macro F1 0.9791

Classification Report:
              precision    recall  f1-score   support

     backend      1.000     1.000     1.000        73
    frontend      0.943     1.000     0.971        66
   fullstack      1.000     0.897     0.946        39
          qa      1.000     1.000     1.000        47

    accuracy                          0.982       225
   macro avg      0.986     0.974     0.979       225
weighted avg      0.983     0.982     0.982       225

<Figure size 1000x800 with 0 Axes>
```

Major Failure Modes and Lessons Learned

Failure Modes

- **Traditional models** struggled mainly with predicting ‘frontend’ and ‘fullstack’ developer roles, often confusing them with ‘backend’, indicating overlap in responsibilities and ambiguous commit messages.

- **Fine-tuned LLM** almost completely eliminated misclassification, correctly identifying all roles except rare single misclassifications, indicating robust contextual learning from commit messages.

Lessons Learned

- **Fine-tuned LLM** showed significantly higher performance for text-heavy classification tasks, capturing nuance traditional models could not.

- **Class imbalance** and ambiguity in commit messages were constant challenges, but handled effectively by the LLM.

- All import and padding issues in LLM setup were resolved, enabling scalable deployment and repeatability.

Conclusion

The **fine-tuned LLM** achieved the highest Macro F1 score and overall accuracy, demonstrating superior capability for developer role classification from commits. Advanced text understanding in transformer models leads to more reliable multi-class predictions than traditional ML approaches. Future improvements could involve further LLM tuning and exploring richer text features.

Data Splitting: The dataset was split into stratified training, validation, and test sets (70/15/15) for balanced representation.

Model Training:

- Traditional models: Logistic Regression, Random Forest, and XGBoost were trained using pipelines for both numerical and categorical features.

- Fine-tuned LLM: Pretrained BERT was fine-tuned for classification based on commit messages.

Evaluation:

- Models were validated and selected using metrics: **Macro F1 Score**, **Accuracy**, **Precision**, and **Recall**.

Data quality and preprocessing (feature engineering, stratified splits) were vital for balanced, reliable evaluation.