Automated Role Classification System

Replacing Manual Title Classification for Databricks

One-Page Project Summary

Problem Statement

Current Workflow:

- **Audience Personas**: Regex-based SQL CASE statements classify titles into roles (Data Engineer, Data Scientist, Data Analyst, Data Architect, Business Execs, CIO, CDO and Unprioritized)
- **Chief-Level Classification**: Titles flagged as "false" by default; manual review updates to "true" for CXO roles (CIO, CDAO, CTO, CAO, CDO) and categorises to the respective role otherwise classification remains null

Challenges:

- Manual classification is error-prone and time-consuming
- Regex patterns struggle with title variations and abbreviations

Business Objective

Automating title classification to:

- Replace regex-based persona classification
- Automate Chief-level role detection
- Use ML based automation instead of manual/SQL approach in the two classifications

Data Sources

- ***Primary***: CSV file with 2M+ classified titles into two different classification categories respectively for Model I & Model II
- ***Labels***: 8 categories (Audience Persona: 7 roles + ”Unprioritized”) for Model I & 6 categories (Chief-level: 5 roles + ”null”) for Model II

Proposed Approach

Model Architecture:

- ***LSTM***: Efficient architecture with SMOTE oversampling
- ***BERT***: Pre-trained transformer for contextual understanding
- ***Gen AI***: LLM-based classification for ambiguous titles
- ***SQL AI***: Integration with Databricks SQL AI functions

Evaluation Strategy

- ***Validation***: Confusion matrix, misclassification analysis
- ***Testing***: Held-out test set with SMOTE sampling
- ***Improvement***: Error case inspection for model refinement

Key Metrics

- ***Primary***: Recall (prevent false negatives for classification roles),

  Switch to accuracy if recall calculation not feasible
- ***Secondary***: F1-score, Accuracy, Confusion Matrix Analysis

Timeline

Approach/Model	**Timeline**	**Est. Runtime**	**Status**
Approach 1: LSTM Training Full Notebook: Link	7th March	Model I: 40 min. Model II: 80 min.	Model I ***[Acc.~97%, Rec.~93%]*** Model II: ***[Acc.~93%, Rec.~80%]***
Approach 2: BERT Training	–	–	Planned
Approach 3: Gen AI Testing	–	–	Planned
Approach 4: SQL AI	6th March	Not Known	Planned

Potential Risks & Mitigation

- ***Risk 1***: Class imbalance

  Mitigation: SVM-SMOTE sampling, Undersampling
- ***Risk 2***: Overfitting

  Mitigation: Early stopping, dropout layers

Expected Outcomes

- ***Accuracy***: >90% for both the classification systems
- ***Recall***: >80% for both the classification systems (if feasible)