CS634 Final Term Project Report (Sample Template)

Student Name: [Your Full Name]

Course: CS634 – Data Mining

Instructor: Dr. Yasser Abduallah

Project Title: [e.g., Binary Classification of Diabetes Data Using Random Forest, LSTM, and SVM]

Date: [Submission Date]

1. Introduction

- Briefly describe the purpose of this project.
- Explain what binary classification is and why it's important.
- State the dataset and algorithms used: Random Forest (mandatory), one Deep Learning model (LSTM, Bi-LSTM, GRU, or Conv1D), and one classic ML algorithm (SVM, Decision Tree, KNN, Naïve Bayes, or Bayesian Networks).

2. Dataset

- Name & Source (e.g., Pima Indians Diabetes – UCI Repository).
- Link to dataset.
- Description: number of rows, features, target variable (binary).
- Preprocessing steps: missing values, normalization/standardization, label encoding, class balance handling (if applicable).

3. Algorithms Overview

- Random Forest: brief rationale and how it works at a high level.
- Chosen Deep Learning model: why it suits the data/task.
- Chosen classic ML algorithm: expected strengths/limitations.

4. Implementation

- Programming language: Python.
- Development environment: Jupyter Notebook and .py scripts.

- Required Python packages and installation instructions (example shown below).

```
pip install numpy pandas scikit-learn tensorflow keras matplotlib seaborn
```

- Screenshots of the running program are mandatory. Include at minimum:
- – Dataset loading/preview
- – Training each algorithm
- – Per-fold evaluation outputs/results

5. Evaluation Setup

- Method: 10-fold cross-validation.
- Metrics to report for each fold and overall averages: TP, TN, FP, FN, Accuracy, Precision, Recall, F1, FPR, FNR, Specificity, Balanced Accuracy, TSS, HSS, ROC, AUC, BS, BSS.

6. Results

- For each fold, list results for all three algorithms (Random Forest, your Deep Learning model, and your classic ML model). Include averages at the end.

Example Results Table (abbreviated headers to save space):

Fold	Alg	TP	TN	FP	FN	Acc	Prec	Rec	F1	TSS	HSS	AUC

- Also include ROC curves (one per algorithm) and a brief comparative summary across algorithms.

7. Discussion

- Which algorithm performed best overall and why? Consider bias/variance, dataset characteristics, and hyperparameters.
- Challenges encountered during preprocessing/training/evaluation and how you addressed them.
- Potential improvements or future work (feature engineering, class rebalancing, tuning, architecture changes).

8. Conclusion

- Summarize key findings and takeaways.
- Reflect on the impact of algorithm choice and evaluation methodology.

9. References

- Dataset source link.
- Tutorials, documentation, and research articles consulted.

10. GitHub Link

- Provide the link to your repository (must follow naming convention: lastname_firstname_finaltermproj).
- Ensure instructor/TA access is granted.

11. Appendix (Mandatory)

- Screenshots of Jupyter Notebook and .py execution (training and evaluation).
- Additional plots (e.g., confusion matrices per algorithm, ROC curves).
- Supporting notes or clarifications (environment details, versions, hardware).