# README: Supervised Data Mining Project

## Introduction

This project demonstrates the implementation of three supervised classification algorithms on a binary classification dataset, the Pima Indians Diabetes Dataset. The objective is to evaluate and compare the performance of the algorithms using a variety of classification metrics calculated manually.

## Features

1. Implementation of three algorithms:  
 - Random Forest  
 - Decision Tree  
 - LSTM (Long Short-Term Memory)  
2. Use of 10-fold cross-validation to ensure reliable model evaluation.  
3. Calculation of classification metrics:  
 - Precision  
 - Recall  
 - False Positive Rate (FPR)  
 - False Negative Rate (FNR)  
 - F1-Score  
 - True Skill Statistic (TSS)  
 - Heidke Skill Score (HSS)  
4. ROC curve and AUC score computation and visualization for each algorithm.  
5. Performance comparison and visualization using bar plots.

## Dataset

The dataset used is the Pima Indians Diabetes Dataset, sourced from the UCI Machine Learning Repository. It contains 8 medical diagnostic features and a binary target indicating whether the patient has diabetes.

## Prerequisites

Ensure you have the following libraries installed before running the code:  
- numpy  
- pandas  
- scikit-learn  
- tensorflow  
- matplotlib  
You can install these libraries using pip:  
```  
pip install numpy pandas scikit-learn tensorflow matplotlib  
```

## Files

1. `data\_mining\_project.ipynb` - The Jupyter Notebook containing the implementation.  
2. `pima-indians-diabetes.data.csv` - The dataset file.  
3. `README.docx` - This README file providing an overview of the project.

## Instructions to Run

1. Clone the repository to your local machine:  
 ```  
 git clone: https://github.com/Rajtharun-nagi/DataMiningfinalProject  
 ```  
2. Navigate to the project directory:  
 ```  
 https://github.com/Rajtharun-nagi/DataMiningfinalProject/tree/main/Nagipogu\_Tharun\_kumar\_Final\_project

4. Run the notebook cells sequentially to execute the code.  
5. View the results and visualizations generated in the notebook.

## Results

The project evaluates the performance of the three algorithms using 10-fold cross-validation. The metrics are calculated manually for each fold, and the results are averaged. Additionally, ROC curves and AUC scores are computed to compare model performance visually.

## Discussion

From the evaluation, LSTM showed the best performance in terms of Precision, Recall, and AUC score. Random Forest also performed well due to its ensemble learning capability. Decision Tree, while interpretable, had lower performance due to its tendency to overfit.

## Conclusion

This project highlights the application of supervised learning techniques to a binary classification problem. It demonstrates the importance of performance evaluation metrics and visualizations in comparing different algorithms.