**PHASE 2 PROJECT SUBMISSION**

**PROJECT 1 - WEBSITE TRAFFIC ANALYSIS**

**TEAM MEMBERS:**

1. Nethaji K ­­- 2021504531
2. Rohith M U - 2021504538
3. Sripushkalai S - 2021504547
4. Tharunsarwin U G - 2021504554

**Problem Definition:**

The project involves analysing website traffic data to gain insights into user behaviour, popular pages, and traffic sources. The goal is to help website owners enhance the user experience by understanding how visitors interact with the site. This project encompasses defining the analysis objectives, collecting website traffic data, using IBM Cognos for data visualization, and integrating Python code for advanced analysis.

**Database Link:**

[**https://www.kaggle.com/datasets/bobnau/daily-website-visitors**](https://www.kaggle.com/datasets/bobnau/daily-website-visitors)

**Approach:**

LSTM (Long Short-Term Memory) model with Feature Engineering for website traffic analysis.

**Objective:**

Employ LSTM for website traffic analysis to accurately predict user behaviour, enabling website owners to optimize content placement and enhance user experience for heightened engagement and satisfaction metrics.

**Steps:**

**1. Data Preparation:**

- Collect historical website traffic data, including metrics like page views, unique visitors, time on page, bounce rate, and other relevant features. Ensure the data covers a significant time period.

- Preprocess the data by handling missing values, outliers, and scaling or normalizing the data as necessary.

**2. Feature Engineering:**

- Create a time series dataset with sequential data points. Each data point should include a timestamp and the relevant traffic or user behavior metric.

- Consider including additional features like seasonality, holidays, or special events that might impact website traffic.

**3. Data Splitting:**

- Split the time series dataset into training, validation, and test sets. Typically, you'll use the most recent data for validation and testing, while the older data is used for training.

**4. Sequence Data Preparation:**

- Convert the time series data into sequences of fixed length (e.g., daily, hourly) to create input-output pairs for training the LSTM model. Each input sequence should contain historical data, and the corresponding output should be the next data point in the sequence.

**5. Build the LSTM Model:**

- Design an LSTM neural network architecture suitable for time series prediction. You can use deep learning libraries like TensorFlow or PyTorch to build the model.

- Consider the following components:

- LSTM layers: Stack multiple LSTM layers to capture temporal dependencies effectively.

- Dropout layers: Prevent overfitting by adding dropout layers between LSTM layers.

- Dense layers: Include fully connected layers for output prediction.

- Appropriate activation functions: Use activation functions like ReLU or sigmoid where appropriate.

- Choose an appropriate loss function for regression tasks (e.g., Mean Squared Error).

**6. Model Training:**

- Train the LSTM model using the training dataset. Monitor the model's performance on the validation set and use early stopping to prevent overfitting.

**7. Hyperparameter Tuning:**

- Experiment with different hyperparameters like the number of LSTM units, dropout rates, and batch sizes to optimize the model's performance.

**8. Model Evaluation:**

- Evaluate the LSTM model's performance on the test dataset using evaluation metrics such as Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and R-squared (R2).

**9. Prediction:**

- Use the trained LSTM model to make predictions for future website traffic trends or user behaviour patterns. The input to the model will be historical data, and the output will be the predicted future data points.

**10. Visualization:**

- Visualize the model predictions alongside the actual data to assess the model's accuracy and understand the predicted trends.

**11. Real-Time Updates:**

- Implement a mechanism to periodically retrain the LSTM model with new data to ensure it adapts to changing traffic patterns.

**12. Alerting and Reporting:**

- Set up alerting mechanisms to notify website owners or administrators when significant deviations from predicted patterns occur.

**13. Documentation and Deployment:**

- Document the model architecture, hyperparameters, and deployment instructions for future reference.

- Deploy the LSTM model in a production environment where it can generate real-time or periodic predictions for website traffic.

**14. Feedback Loop:**

- Continuously gather feedback from website owners and users to assess the model's performance and make necessary improvements.

**Benefits:**

**Sequential Data Handling:** LSTMs are specifically designed to handle sequential data, making them highly effective for tasks involving time series data, natural language processing (NLP), speech recognition, and more. They excel at capturing patterns and dependencies in sequences.

**Long-Term Dependencies:** LSTMs can capture long-term dependencies in data, which is challenging for traditional models like ARIMA. They have memory cells that allow them to remember and use information from earlier time steps, making them suitable for tasks with complex temporal relationships.

**Nonlinearity:** LSTMs can capture nonlinear patterns and relationships in data, making them versatile for modeling complex phenomena. This ability to model nonlinearities is particularly valuable when dealing with real-world data that may not adhere to linear assumptions.

**Feature Extraction:** LSTMs can automatically learn relevant features from the data, reducing the need for extensive feature engineering. They can discover abstract representations of patterns and dependencies.

**Adaptability:** LSTMs are adaptable to different input lengths and can handle sequences of varying lengths. This flexibility is essential for tasks like NLP, where sentences or documents can have different lengths.

**Continuous Learning:** LSTMs can be updated with new data over time, enabling continuous learning and adaptation to changing patterns.

**Limitations:**

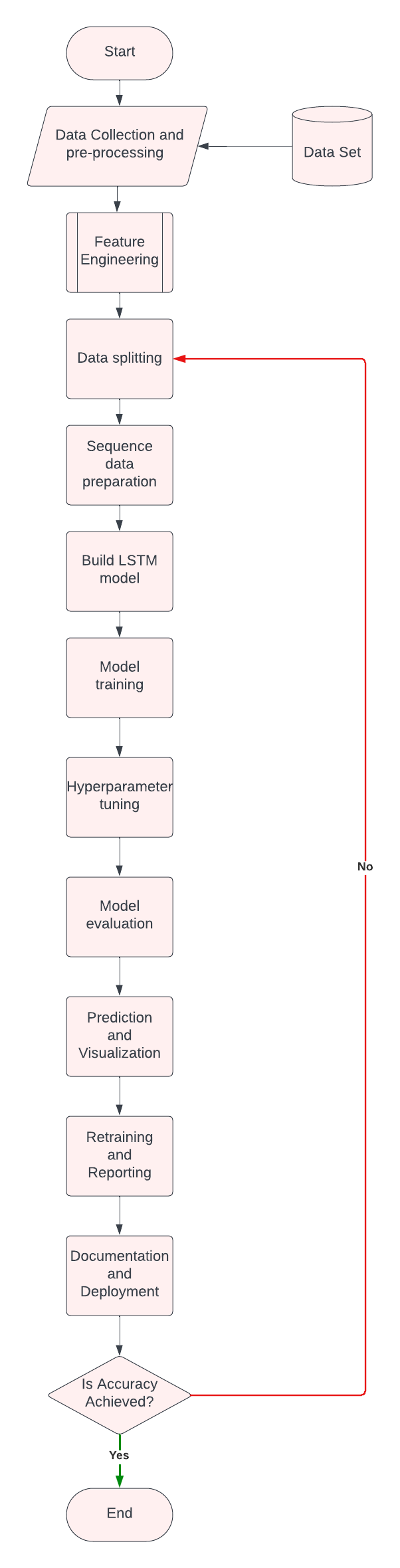
**Data Requirements:** LSTMs perform better with larger datasets. If you have limited data, training an LSTM model can be challenging and may lead to overfitting.

**Overfitting:** LSTMs are prone to overfitting, especially when training on small datasets or using complex architectures. Regularization techniques like dropout and early stopping are commonly used to mitigate this issue.

**Training Time:** Training deep LSTM networks on large datasets can take a long time, which may not be practical for real-time or near-real-time applications.

**Memory Consumption:** Deep LSTM models with a large number of units can consume a considerable amount of memory, which may not be suitable for deployment in resource-constrained environments.

**Flowchart:**

****

**Conclusion:**

In conclusion, LSTM-based website traffic analysis stands as a game-changer in understanding user behaviour. Its ability to capture intricate patterns provides a competitive edge in decision-making for businesses. This approach holds great promise for refining user experience and optimizing digital strategies, paving the way for a more successful online presence.