Topics in Deep Learning

“DeepAlgae: Predicting Optimal Time Frame for Biofuel Extraction from Algae”

Team Number: 5
Team Members: Spoorthi Shivaprasad (PES2UG21CS536)
Sumukh Shetty (PES2UG21CS554)
Sumukh Suresh (PES2UG21CS555)
Surabhi A. Chilkunda (PES2UG21CS557)

Goal of the project:

This project aims to develop a **deep learning model** that **predicts the optimal time frame for extracting biofuel from algae cultures**.

What is the problem we are trying to solve?

The current process of **identifying the ideal time for biofuel extraction** from algae **relies on manual monitoring and analysis**, which can be time-consuming and imprecise. This can lead to missed opportunities for harvesting at peak efficiency and potential yield losses.

Proposed Solution:

This project proposes an **LSTM-based deep learning model** to analyze trends and dependencies within algae growth data. The model will consider:

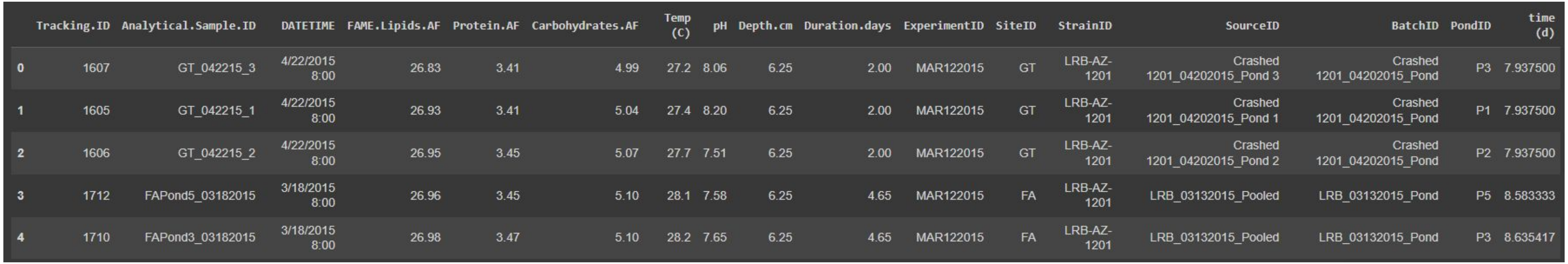
- **Essential Biomolecule Concentrations: - Lipid content - Carbohydrate content - Protein content**
- **Environmental Growth Conditions: - Algal Culture Temperature - Algal Culture pH - Algal Culture Depth**

Benefits of this DL approach:

- Automated Prediction
- Improved Efficiency
- Increased Yield
- Data-Driven Optimization

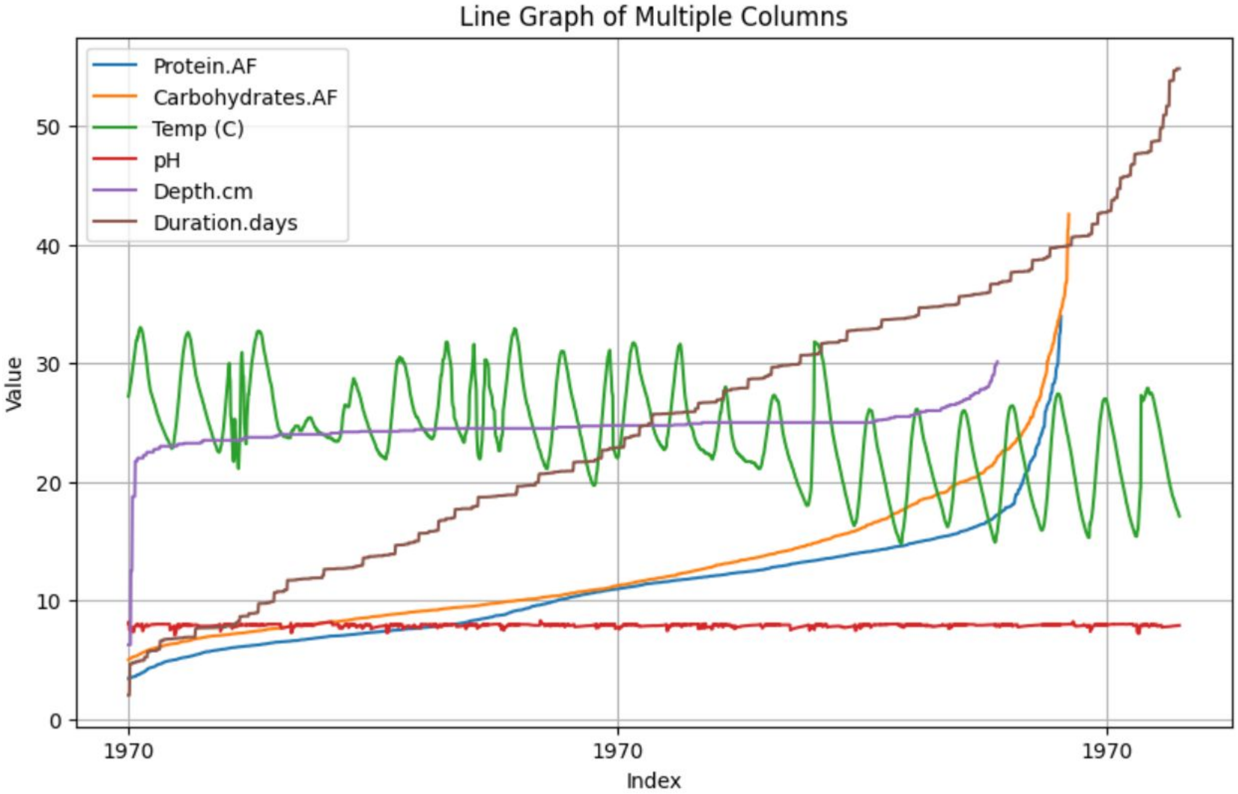
Overview of the Dataset:

The ATP3 Unified Field Study Data (ATP3 UFS) is a publicly available dataset that provides insights into large-scale, open pond cultivation of algae. Compiled by the Algae Testbed Public-Private Partnership (ATP3), this data encompasses information on algal growth and biomass composition across various geographical locations (continental US and Hawaii) and seasonal conditions throughout the year.

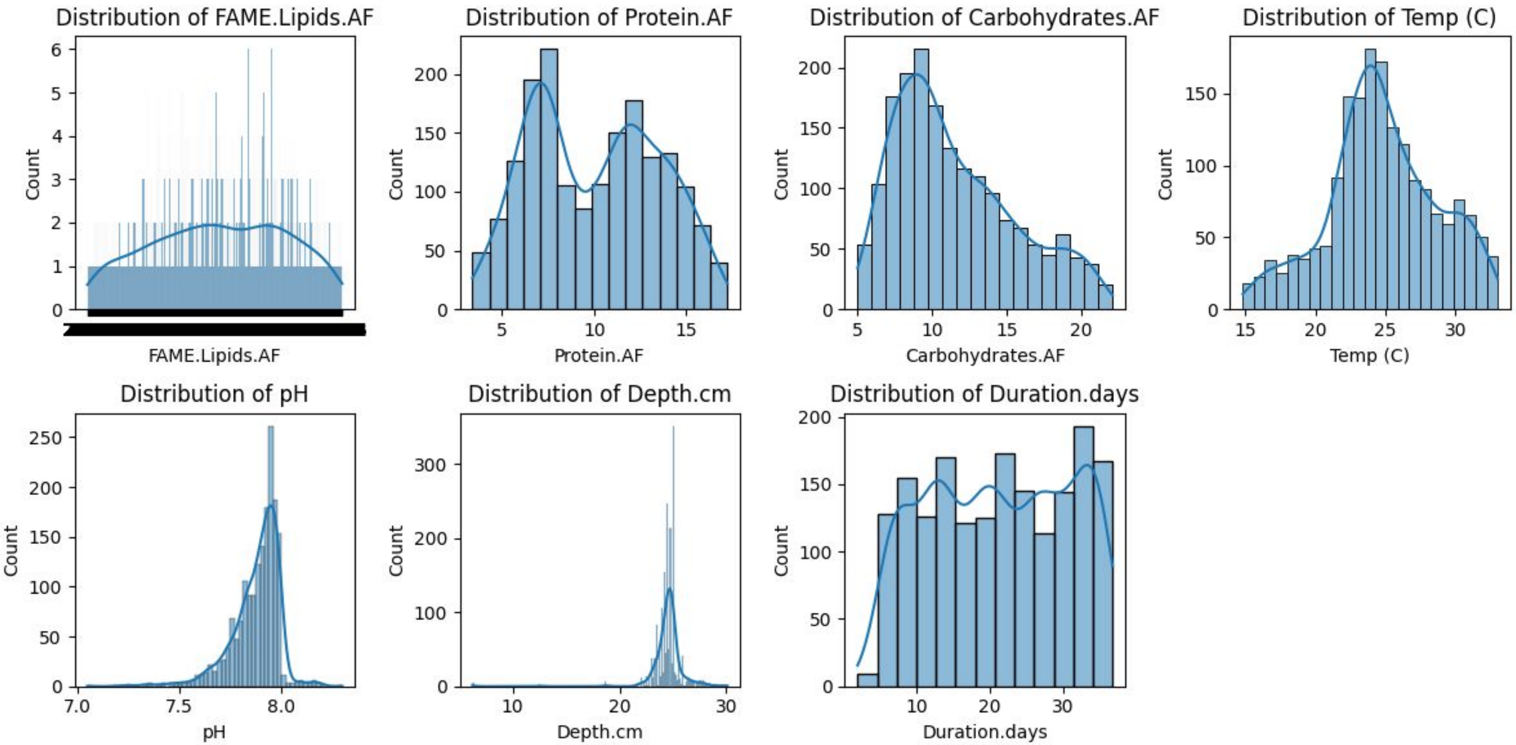


	Tracking.ID	Analytical.Sample.ID	DATETIME	FAME.Lipids.AF	Protein.AF	Carbohydrates.AF	Temp (C)	pH	Depth.cm	Duration.days	ExperimentID	SiteID	StrainID	SourceID	BatchID	PondID	time (d)
0	1607	GT_042215_3	4/22/2015 8:00	26.83	3.41	4.99	27.2	8.06	6.25	2.00	MAR122015	GT	LRB-AZ-1201	Crashed 1201_04202015_Pond 3	Crashed 1201_04202015_Pond	P3	7.937500
1	1605	GT_042215_1	4/22/2015 8:00	26.93	3.41	5.04	27.4	8.20	6.25	2.00	MAR122015	GT	LRB-AZ-1201	Crashed 1201_04202015_Pond 1	Crashed 1201_04202015_Pond	P1	7.937500
2	1606	GT_042215_2	4/22/2015 8:00	26.95	3.45	5.07	27.7	7.51	6.25	2.00	MAR122015	GT	LRB-AZ-1201	Crashed 1201_04202015_Pond 2	Crashed 1201_04202015_Pond	P2	7.937500
3	1712	FAPond5_03182015	3/18/2015 8:00	26.96	3.45	5.10	28.1	7.58	6.25	4.65	MAR122015	FA	LRB-AZ-1201	LRB_03132015_Pooled	LRB_03132015_Pond	P5	8.583333
4	1710	FAPond3_03182015	3/18/2015 8:00	26.98	3.47	5.10	28.2	7.65	6.25	4.65	MAR122015	FA	LRB-AZ-1201	LRB_03132015_Pooled	LRB_03132015_Pond	P3	8.635417

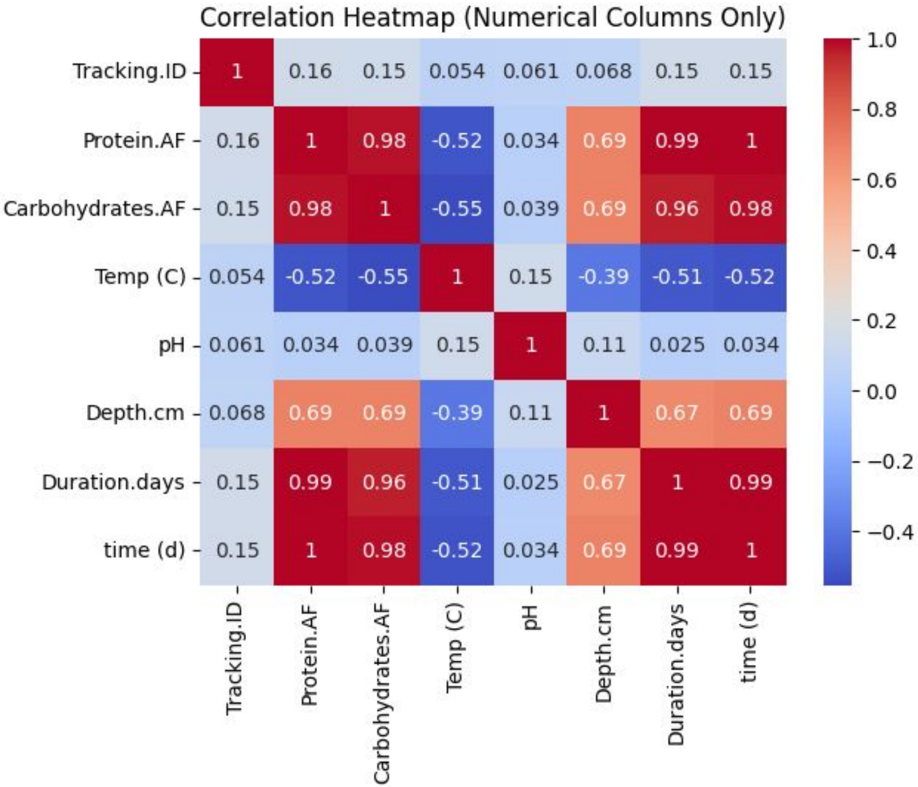
Overview of the Dataset:



Overview of the Dataset:



Overview of the Dataset:

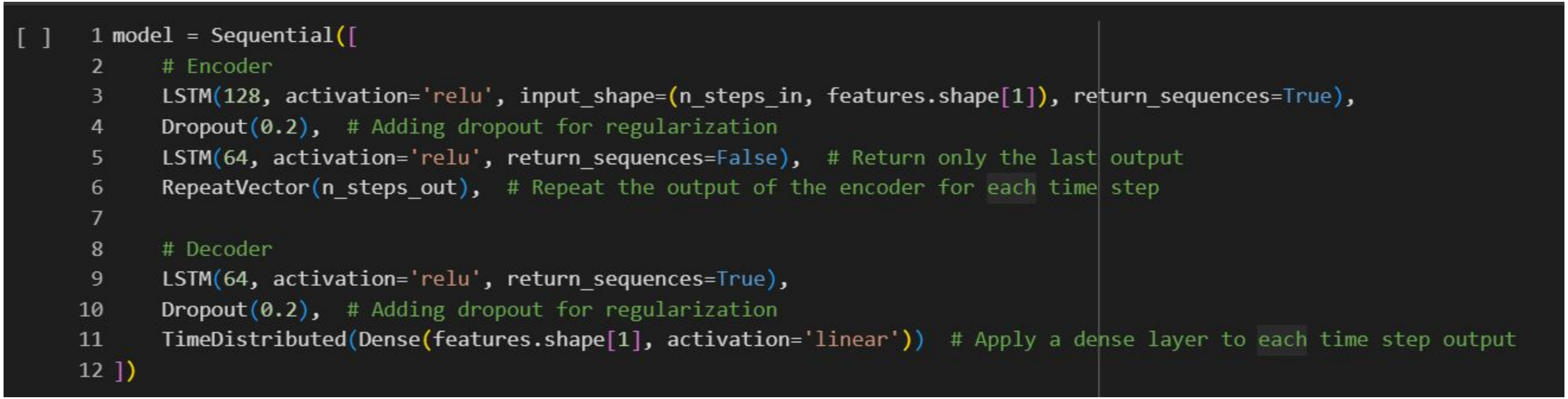


Correlation Heatmap (Numerical Columns Only)

	Tracking.ID	Protein.AF	Carbohydrates.AF	Temp (C)	pH	Depth.cm	Duration.days	time (d)
Tracking.ID	1	0.16	0.15	0.054	0.061	0.068	0.15	0.15
Protein.AF	0.16	1	0.98	-0.52	0.034	0.69	0.99	1
Carbohydrates.AF	0.15	0.98	1	-0.55	0.039	0.69	0.96	0.98
Temp (C)	0.054	-0.52	-0.55	1	0.15	-0.39	-0.51	-0.52
pH	0.061	0.034	0.039	0.15	1	0.11	0.025	0.034
Depth.cm	0.068	0.69	0.69	-0.39	0.11	1	0.67	0.69
Duration.days	0.15	0.99	0.96	-0.51	0.025	0.67	1	0.99
time (d)	0.15	1	0.98	-0.52	0.034	0.69	0.99	1

Model Selected

“An LSTM based Encoder-Decoder Neural Network”



```
model = Sequential([
    # Encoder
    LSTM(128, activation='relu', input_shape=(n_steps_in, features.shape[1]), return_sequences=True),
    Dropout(0.2),  # Adding dropout for regularization
    LSTM(64, activation='relu', return_sequences=False),  # Return only the last output
    RepeatVector(n_steps_out),  # Repeat the output of the encoder for each time step

    # Decoder
    LSTM(64, activation='relu', return_sequences=True),
    Dropout(0.2),  # Adding dropout for regularization
    TimeDistributed(Dense(features.shape[1], activation='linear'))  # Apply a dense layer to each time step output
])
```

“An LSTM based Encoder-Decoder Neural Network”

PES UNIVERSITY

Assumptions for suitable conditions to extract biofuel from algae

- **Lipids above 20%** of dry mass
- **Protein 10% - 30%** of dry mass
- **Carbohydrate 10% - 30%** of dry mass
- Algal Culture **Temperature 20°C to 25°C**
- Algal Culture **pH 7 to 9**
- Algal Culture **Depth less than 30 cm**

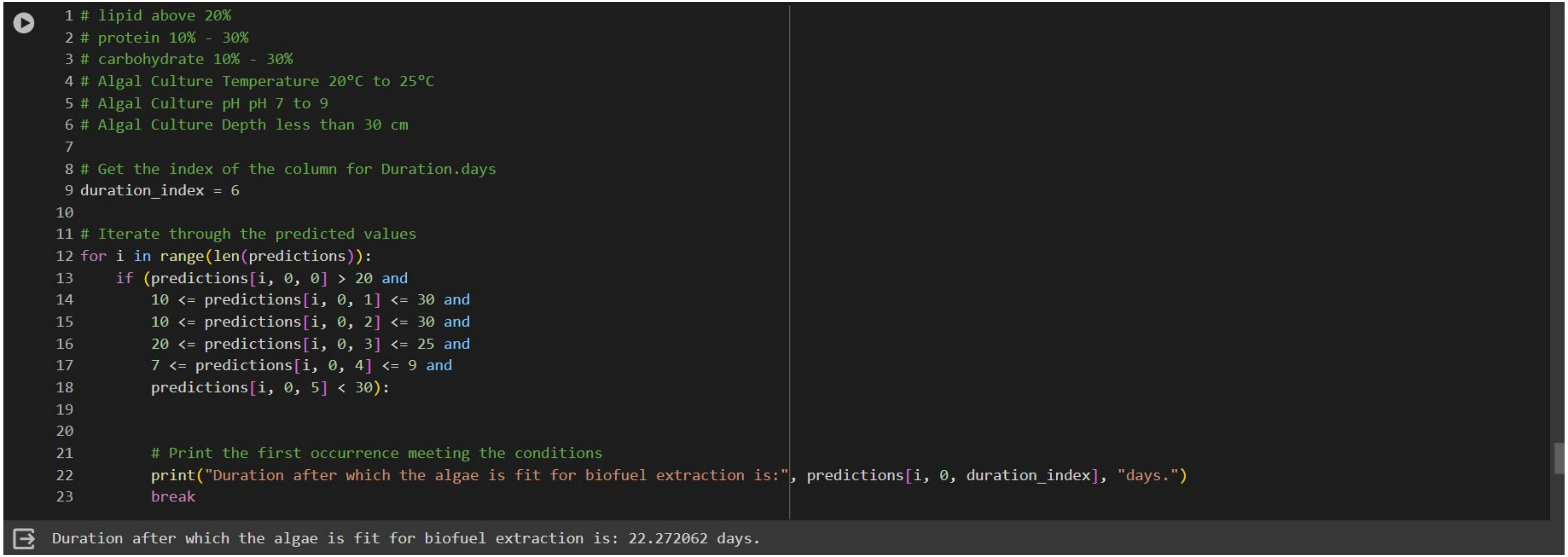
Why this particular model?

- **Sequential Data Handling:**
  - LSTMs are apt for **processing sequential data**, making them suitable for **time series forecasting tasks**.
- **Long-Term Dependencies:**
  - LSTMs can **capture long-term dependencies in data**, crucial for **understanding patterns** in algae growth duration.
- **Encoder-Decoder Architecture:**
  - The encoder-decoder architecture is good at **understanding complex sequential relationships**, making it **appropriate for time series prediction.**

Customisations:

- **Stacked LSTMs**:
  - The model uses **two LSTM layers** with **128 and 64 units** respectively.
- **Dropout:**
  - **Dropout layers** (with **0.2 probability**) are **included after each LSTM layer**.
- **Return Sequences vs. Return only last output:**
  - **The first LSTM layer** uses **return_sequences=True**, which means it **returns the entire output sequence for each input**.
  - **The second LSTM layer** uses **return_sequences=False**, **returning only the last output from the sequence**. This final output is then repeated for each time step in the decoder using **RepeatVector**.
- **Decoder with LSTM and Dense Layer:**
  - A final **TimeDistributed layer with a dense layer having the same number of units as the number of features is used**. **This dense layer predicts the value of each feature** for each time step in the output sequence.

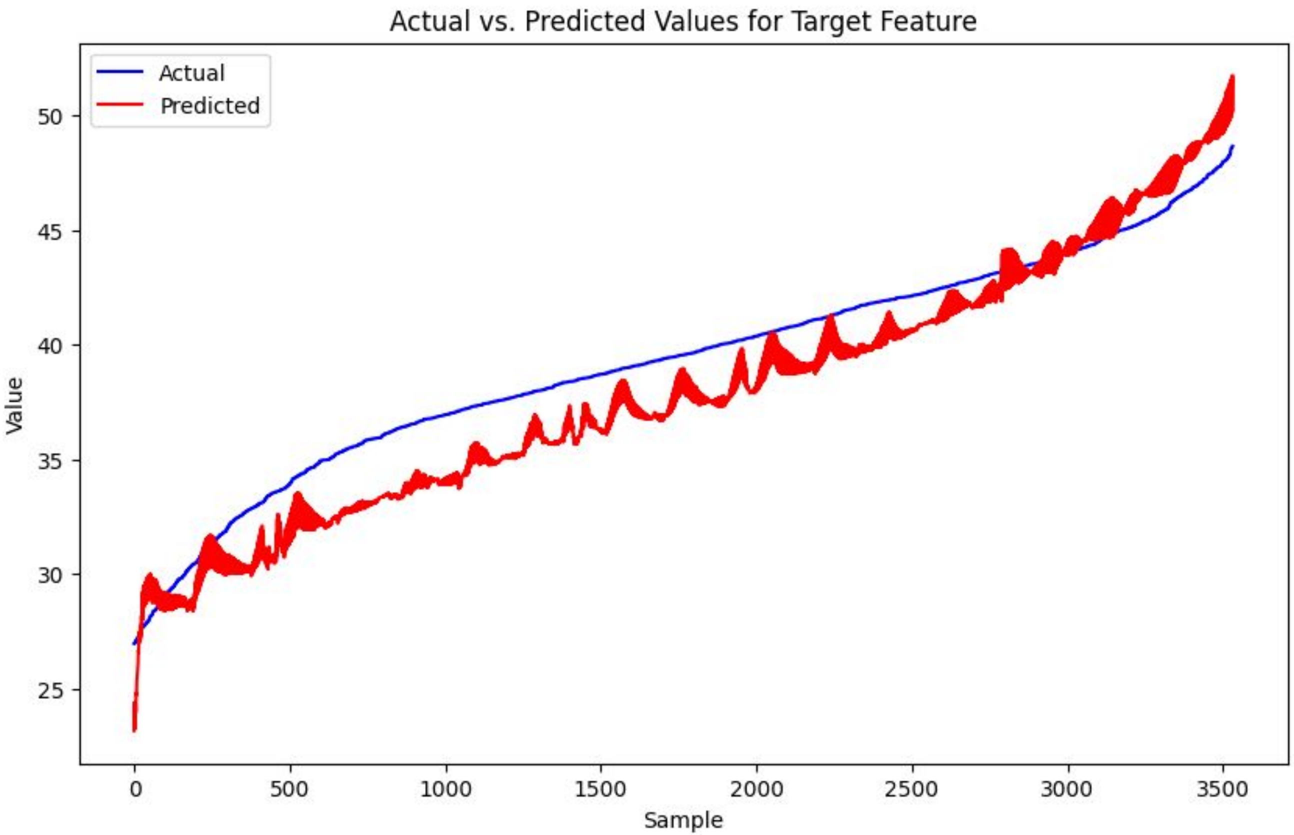
Results



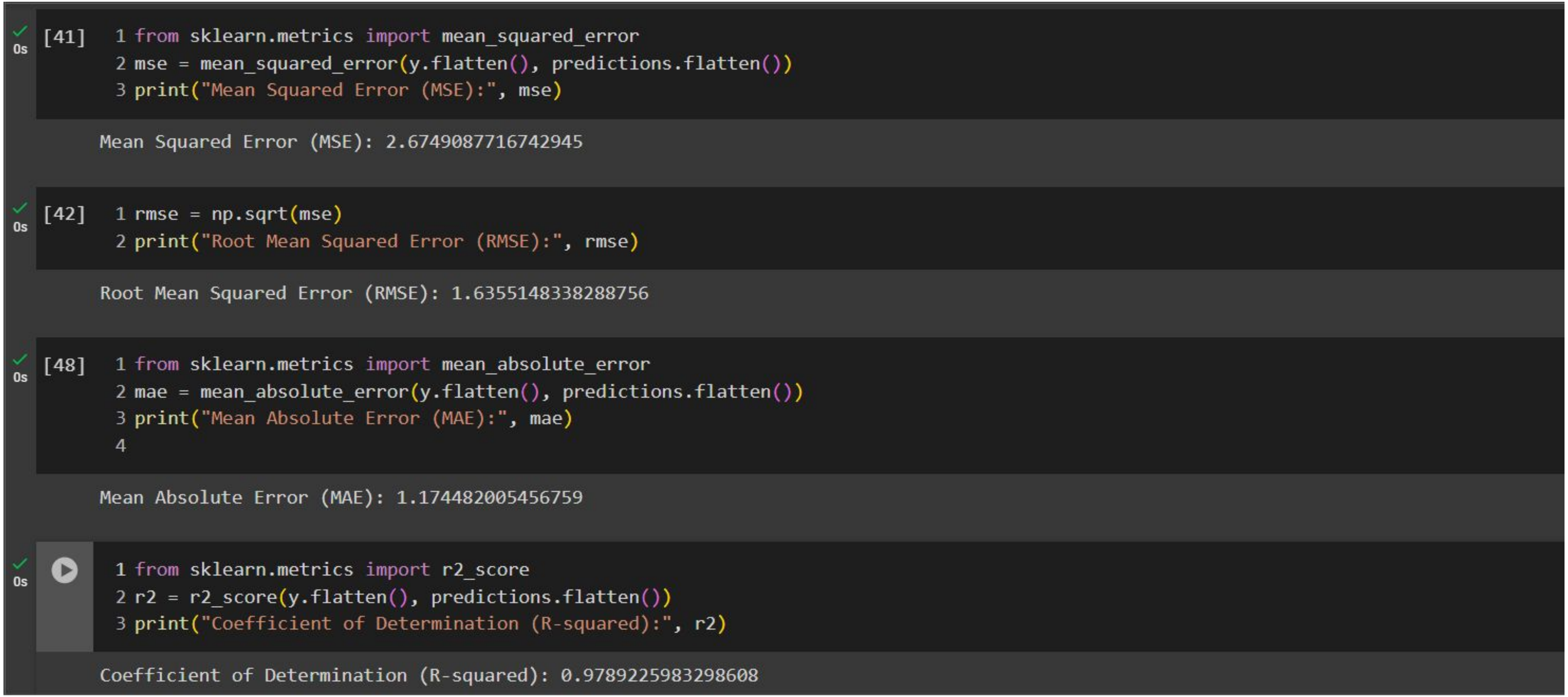
```python
# lipid above 20%
# protein 10% - 30%
# carbohydrate 10% - 30%
# Algal Culture Temperature 20°C to 25°C
# Algal Culture pH pH 7 to 9
# Algal Culture Depth less than 30 cm

# Get the index of the column for Duration.days
duration_index = 6

# Iterate through the predicted values
for i in range(len(predictions)):
    if (predictions[i, 0, 0] > 20 and
        10 <= predictions[i, 0, 1] <= 30 and
        10 <= predictions[i, 0, 2] <= 30 and
        20 <= predictions[i, 0, 3] <= 25 and
        7 <= predictions[i, 0, 4] <= 9 and
        predictions[i, 0, 5] < 30):


        # Print the first occurrence meeting the conditions
        print("Duration after which the algae is fit for biofuel extraction is:", predictions[i, 0, duration_index], "days.")
        break
```

```
Duration after which the algae is fit for biofuel extraction is: 22.272062 days.
```

Results


Actual vs. Predicted Values for Target Feature
Actual
Predicted
Value
Sample
25
30
35
40
45
50
0
500
1000
1500
2000
2500
3000
3500

Results



```python
from sklearn.metrics import mean_squared_error
mse = mean_squared_error(y.flatten(), predictions.flatten())
print("Mean Squared Error (MSE):", mse)
```

```
Mean Squared Error (MSE): 2.6749087716742945
```

```python
rmse = np.sqrt(mse)
print("Root Mean Squared Error (RMSE):", rmse)
```

```
Root Mean Squared Error (RMSE): 1.6355148338288756
```

```python
from sklearn.metrics import mean_absolute_error
mae = mean_absolute_error(y.flatten(), predictions.flatten())
print("Mean Absolute Error (MAE):", mae)

```

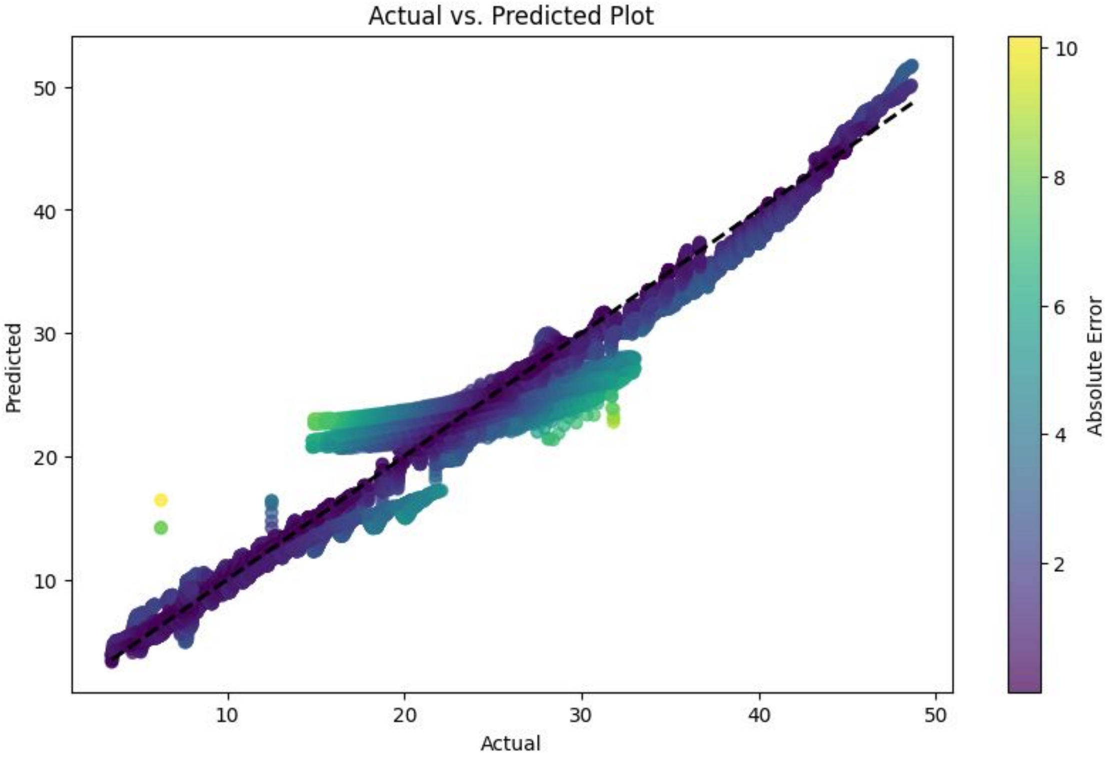
```
Mean Absolute Error (MAE): 1.174482005456759
```

```python
from sklearn.metrics import r2_score
r2 = r2_score(y.flatten(), predictions.flatten())
print("Coefficient of Determination (R-squared):", r2)
```

```
Coefficient of Determination (R-squared): 0.9789225983298608
```

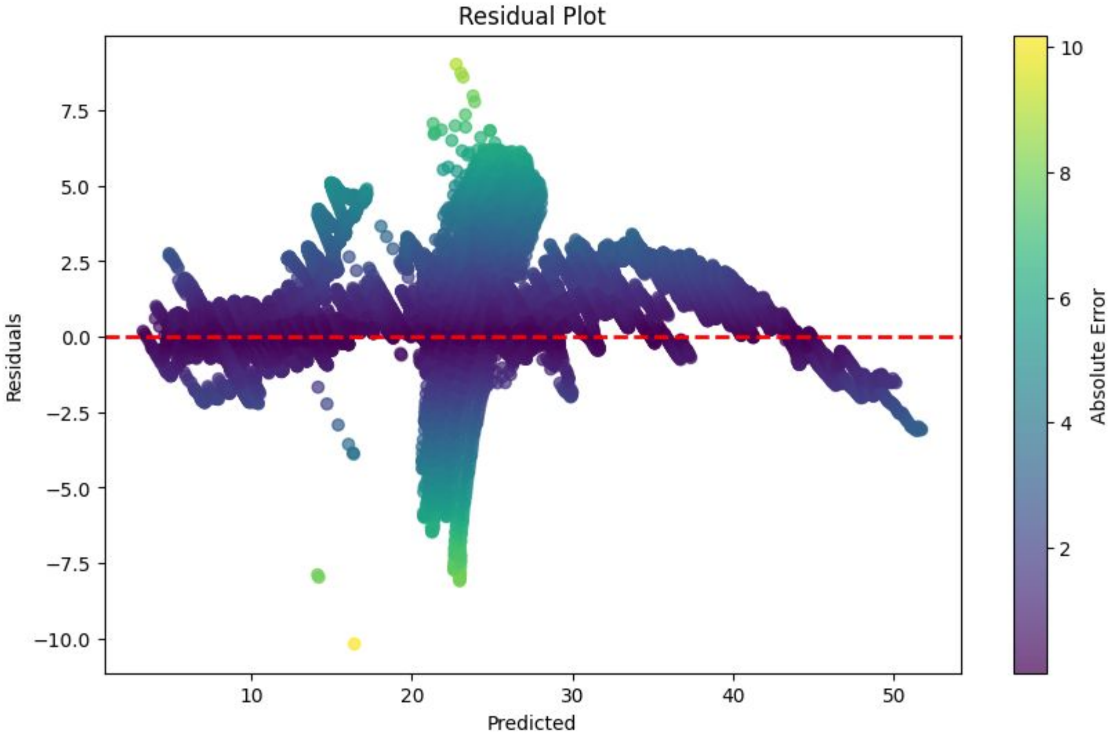
Results

A scatter plot with a color gradient based on the absolute errors



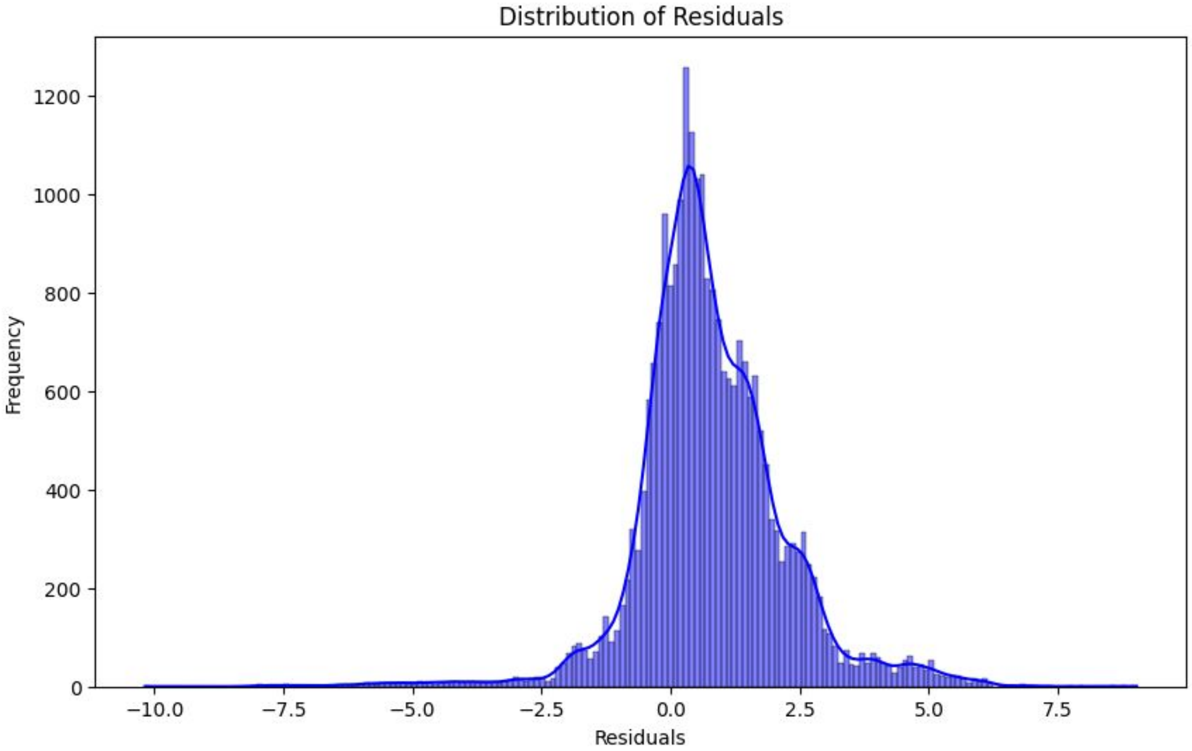
Results

Residual Plot with Color Gradient



Results

Distribution Of Residuals



Analysis and Interpretation of Results:

Note: The target variable ranges from a minimum of 2 days to a maximum of 54 days.

- **Mean Squared Error (MSE)**: An MSE of 2.674 can be considered relatively good.

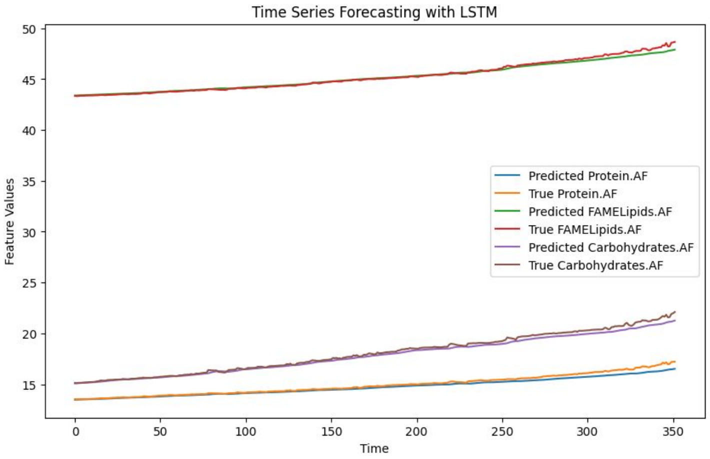
- **Root Mean Squared Error (RMSE):** Similar to MSE, an RMSE of 1.635 represents an average error of about 3% of the target variable range (approximately 1.6 days), indicating a good level of error.

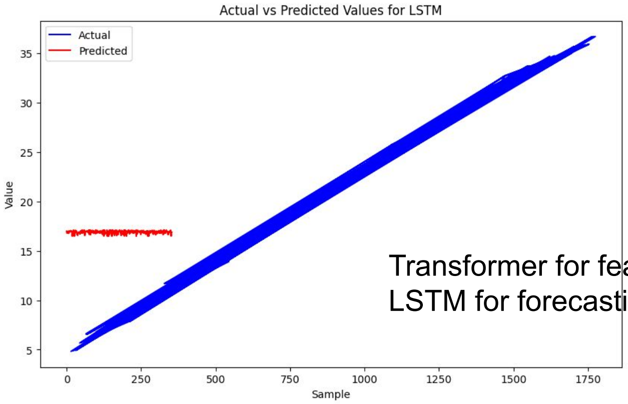
- **Mean Absolute Error (MAE):** An MAE of 1.17 translates to an average absolute difference of approximately 2.2% of the target variable range (approximately 1.2 days). This is a good value, suggesting most predictions are fairly close to the actual values in terms of duration (days).

- **Coefficient of Determination (R-squared):** This remains an exceptionally good value, indicating that 97.8% of the variance in the actual data is explained by the predictions. This reinforces the model's ability to capture the underlying relationships in the data.

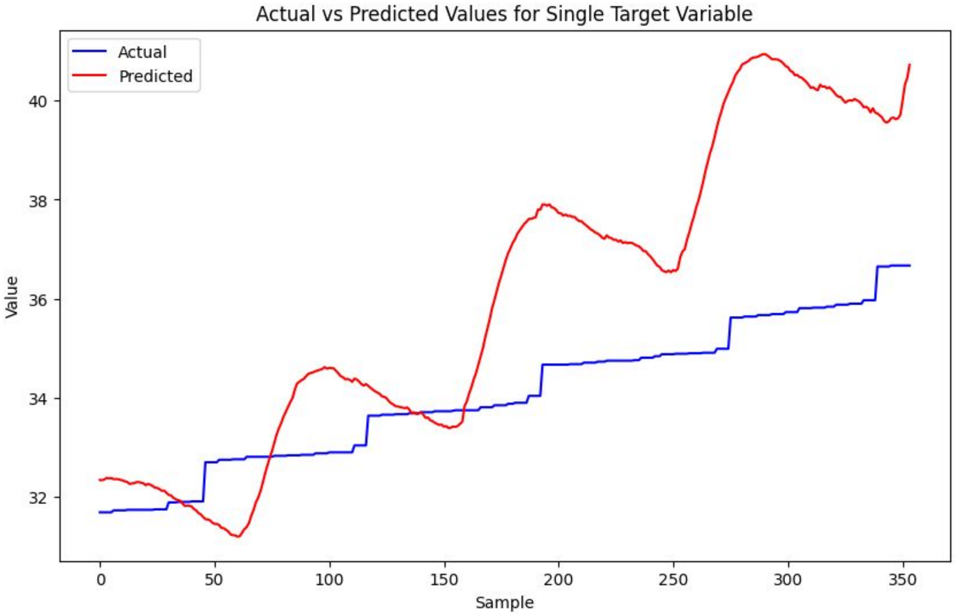
Comparison with other approaches

Vanilla LSTM





Transformer for feature extraction and LSTM for forecasting



Multivariate LSTM

Comparison with related work

Model Type	Features Considered	Accuracy/Precision/R-squared
2-level Random Forest	Nitrogen content	89.6%
3-level Random Forest	Nitrogen content	77.0%
4-level Random Forest	Nitrogen content	66.7%
Regression Model	Total nitrogen in the past year, Average maximum temperature in the past 7 days	R-squared: 0.69
LSTM for Temporal Trends	Nitrogen content	Accuracy for Combined LSTM with 2-level classification: 86.0% for predicting HAB
SVM (Support Vector Machine)	Total nitrogen over the past year, Average maximum temperature over the past 7 days	Microcystis spp: Accuracy 92.3%, Precision 86.4% Dolichospermum spp: Accuracy 71.4%, Precision 77.5%
Gradient Boosting Regression (GBR)	Nitrogen content	MAE: 4.22 mg m−3, RMSE: 6.27 mg m−3
LSTM	Nitrogen content	MAE: 3.87 mg m−3, RMSE: 6.00 mg m−3

Limitations

- **Limited Context Window:**
  - LSTMs can only look back a few steps, missing long-term dependencies.
  - Important past events might not influence predictions if they're too far back.
- **Data Dependency:**
  - LSTMs need lots of good data to learn effectively.
  - Insufficient or noisy data can lead to poor model performance.
- **Black-Box Nature:**
  - Understanding why LSTMs make specific predictions is challenging.
  - It's like a magic box; hard to peek inside and debug or explain easily.

Future Enhancements:

- **Advanced LSTM Architectures:**
  - **Bidirectional LSTMs:** Incorporate information from both past and future time steps for enhanced context understanding.
- **Ensemble Methods:**
  - **Combine Predictions:** Aggregate predictions from diverse LSTM configurations or other time series models to improve prediction accuracy and robustness.
- **Attention Mechanisms:**
  - **Attention Layers:** Integrate attention mechanisms within the LSTM model to focus on crucial parts of the input sequence, enhancing prediction accuracy and interpretability.

Thank you

1. Spoorthi Shivaprasad - PES2UG21CS536
2. Sumukh Shetty - PES2UG21CS554
3. Sumukh Suresh - PES2UG21CS555
4. Surabhi A. Chilkunda - PES2UG21CS557