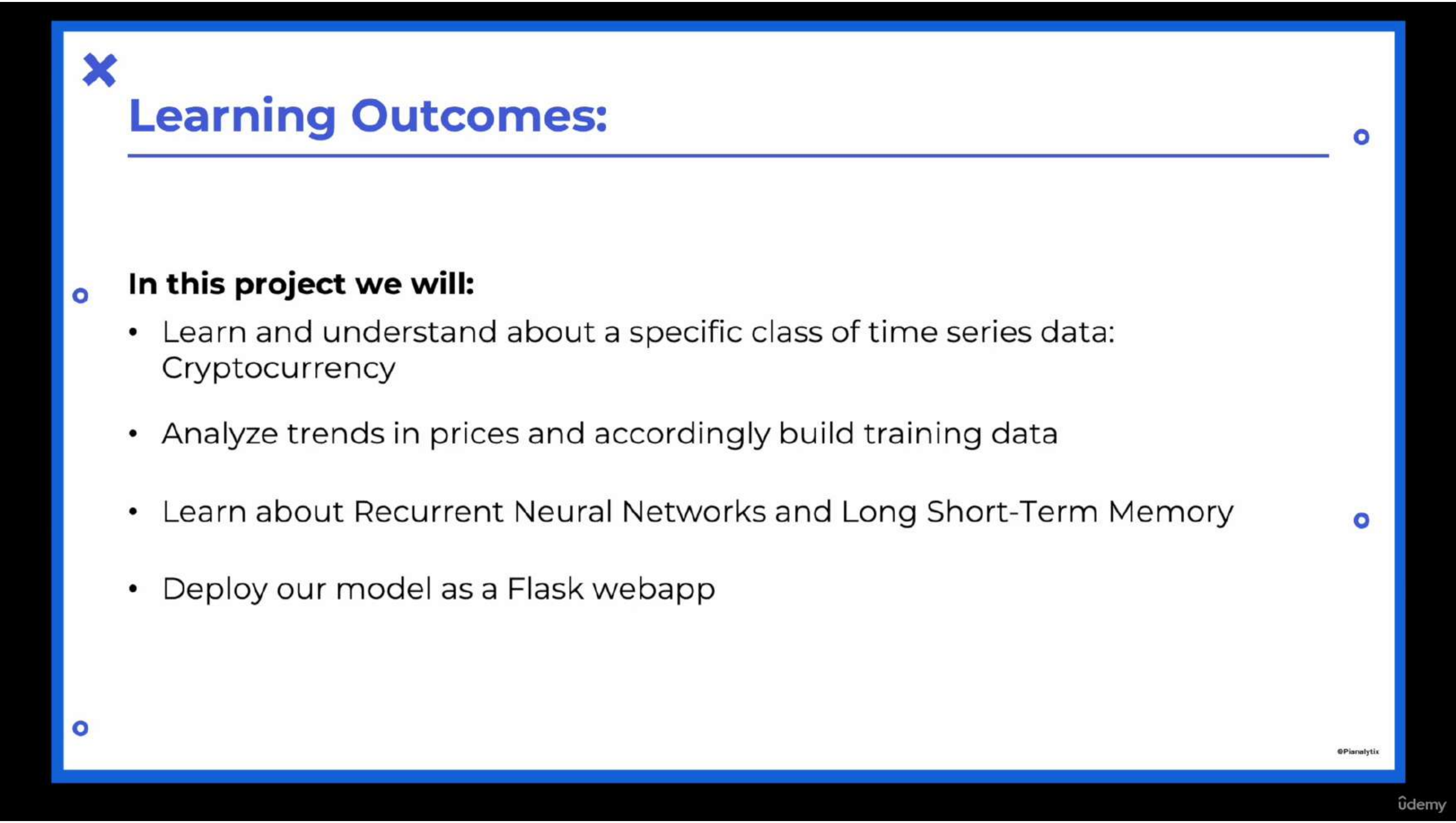
Data Science
Project

# Predicting Ethereum Prices with LSTMs

©Pianalytix



# Learning Outcomes:

**In this project we will:**

- Learn and understand about a specific class of time series data: Cryptocurrency
- Analyze trends in prices and accordingly build training data
- Learn about Recurrent Neural Networks and Long Short-Term Memory
- Deploy our model as a Flask webapp

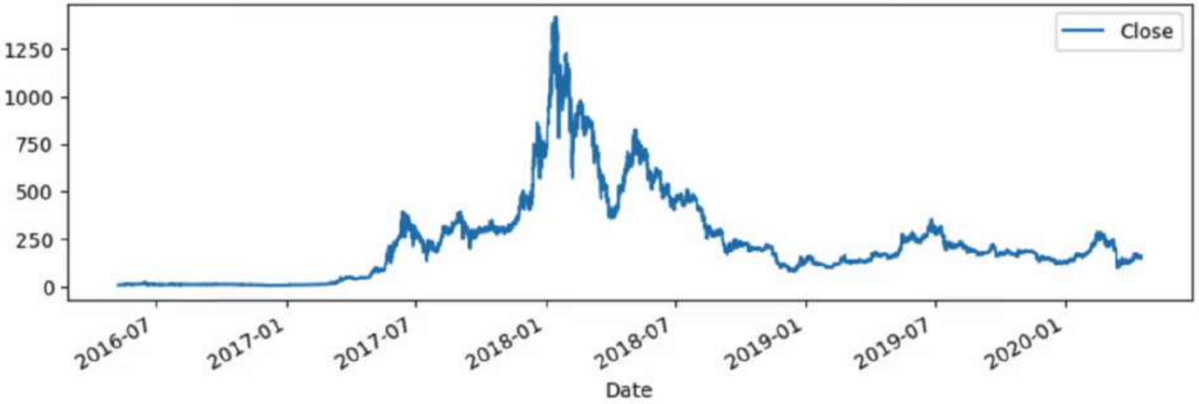
# Part I: Cryptocurrency

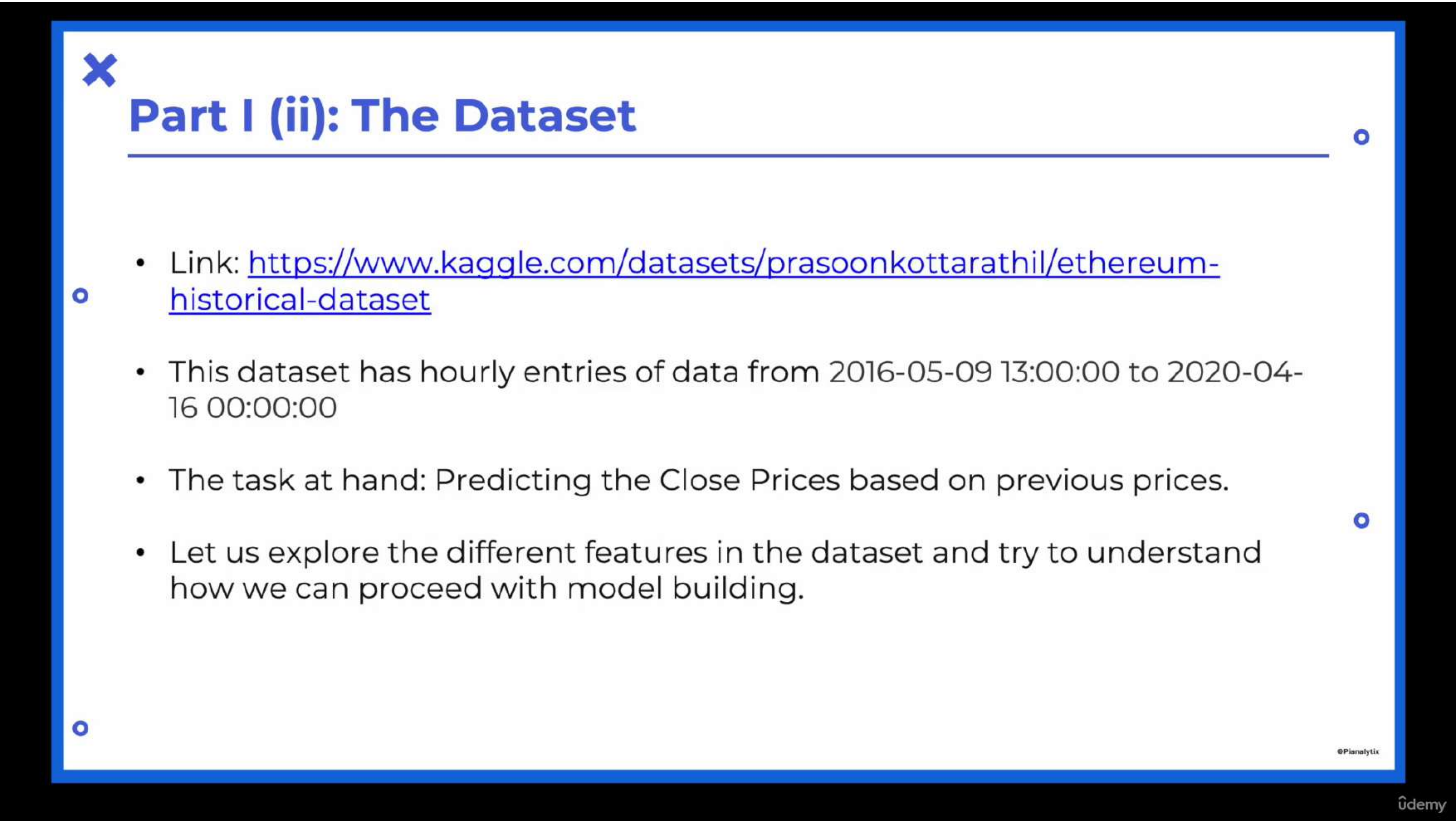
- With the recent boom in blockchain technology and digital currency, cryptocurrencies have gained a lot of traction, and investors!
- It has essentially become a variant of stock market trading, making it a variable data scientist can analyse and predict.
- As crypto is more unreliable and unstable, there has been an increasing demand in services and tools to help in crypto investing, compared to traditional stock market analysis.
- In this project, we are going to employ LSTMs, a form of RNNs, to atleast help us understand trends in crypto , *if not make us billionaires!*

©Pianalytix

# Part I (i): Ethereum

- Ethereum is a decentralized blockchain with smart contract functionality. Ether (is the native cryptocurrency of the platform. Among cryptocurrencies, ether is second only to bitcoin in market capitalization. It is open-source software.
- Below we can see the chart for Ethereum close prices (taken from the dataset itself)
- Lets head on to the notebook for further analysis.



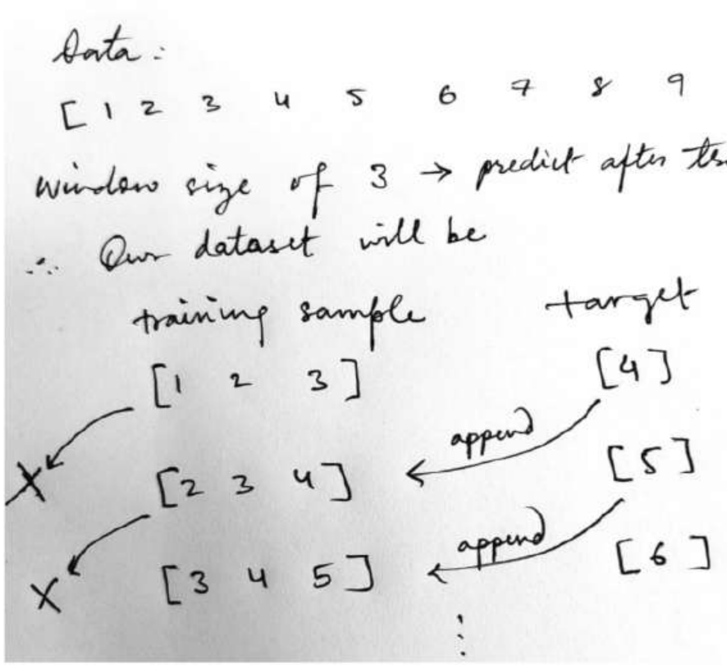


# Part I (ii): The Dataset

- Link: https://www.kaggle.com/datasets/prasoonkottarathil/ethereum-historical-dataset
- This dataset has hourly entries of data from 2016-05-09 13:00:00 to 2020-04-16 00:00:00
- The task at hand: Predicting the Close Prices based on previous prices.
- Let us explore the different features in the dataset and try to understand how we can proceed with model building.

# Part II (iii): Predicting Future Unseen Values

## Rolling Window data generation:



Data:

[1 2 3 4 5 6 7 8 9 10]

window size of 3 → predict after training on 3 samples.

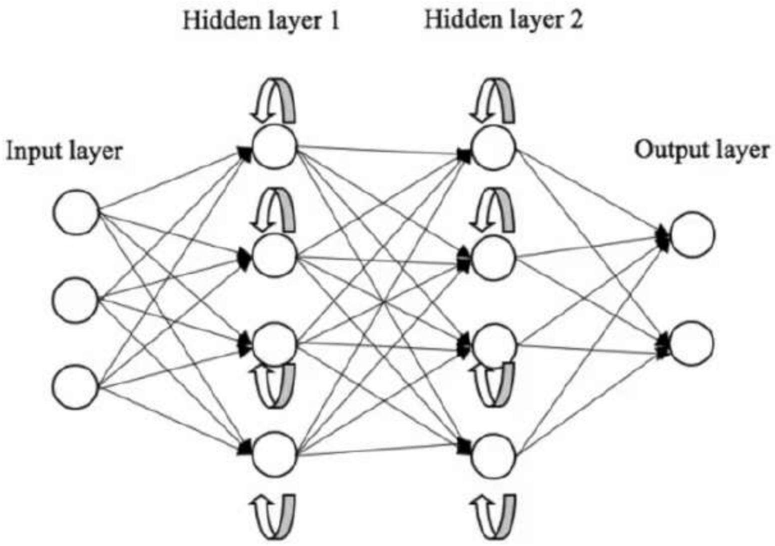
∴ Our dataset will be

| training sample | target |
|---|---|
| [1 2 3] | [4] |
| [2 3 4] | [5] |
| [3 4 5] | [6] |
| ⋮ | |

append
append
X
X

# Part II (i): Recurrent Neural Networks

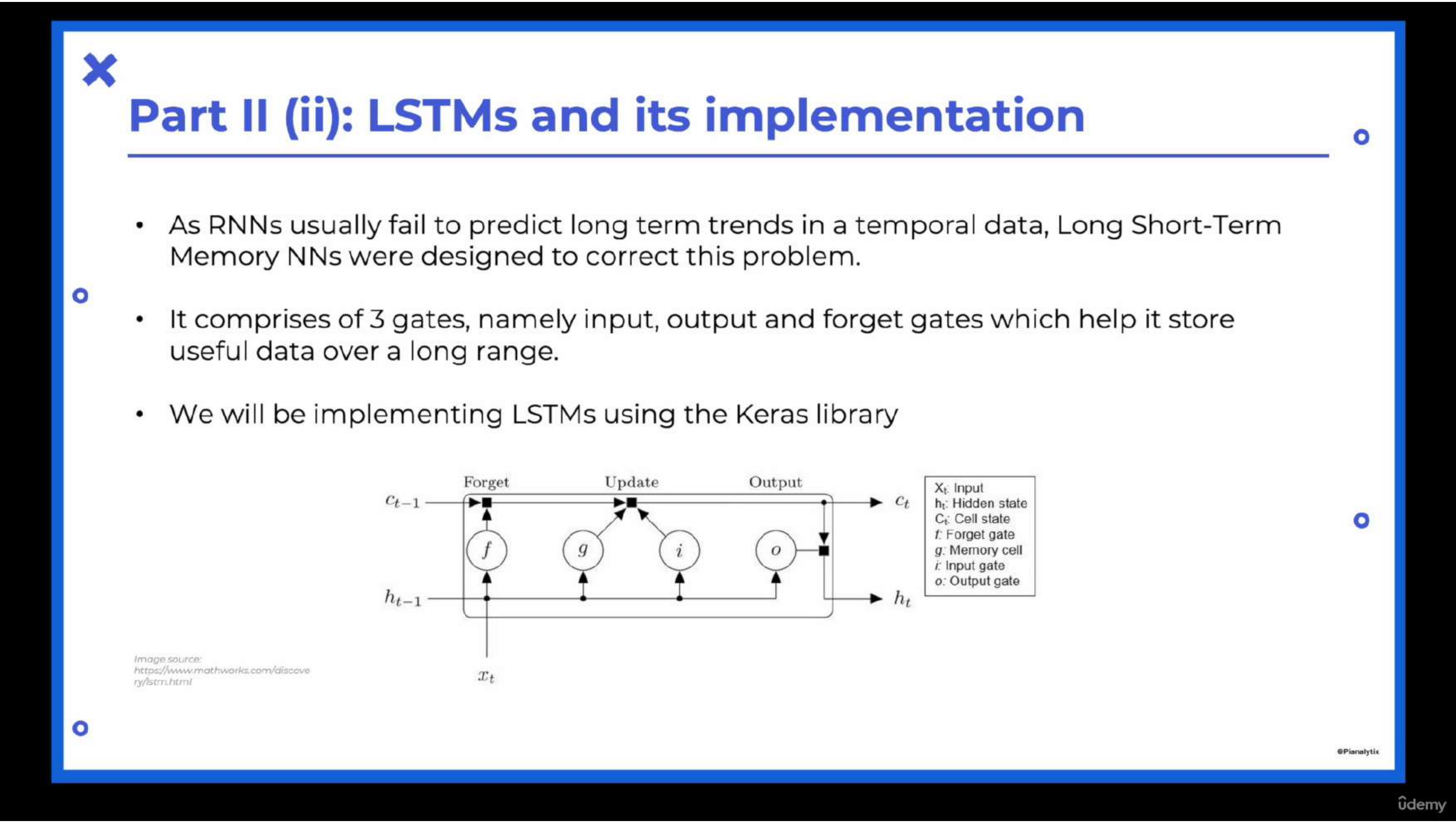
- The basic idea behind an RNN is memory. Regular Neural Networks like Feed Forward or CNNs consider each sample data independent of other data in the dataset.
- This is why RNNs are very useful for NLP and temporal problems as it considers past data to influence its predictions.
- Variants of RNN include:
  Bi-directional RNNs
  ***LSTMs***
  Gated Recurrent Units



General Form of RNNs

*Image source: https://opendatascience.com/understanding-the-mechanism-and-types-of-recurring-neural-networks/*

©Pianalytix



# Part II (ii): LSTMs and its implementation

- As RNNs usually fail to predict long term trends in a temporal data, Long Short-Term Memory NNs were designed to correct this problem.
- It comprises of 3 gates, namely input, output and forget gates which help it store useful data over a long range.
- We will be implementing LSTMs using the Keras library

*Image source: https://www.mathworks.com/discovery/lstm.html*

©Pianalytix