1. LSTM
2. Cách hoạt động
3. Giải thích code mô hình

* Xây dựng mô hình LSTM:
  + model\_lstm = Sequential()
  + model\_lstm.add(LSTM(units=50, activation='relu',
  + input\_shape=(X\_train.shape[1], X\_train.shape[2])))
  + model\_lstm.add(Dense(units=3))
  + model\_lstm.compile(optimizer='adam', loss='mse')
* model\_lstm = Sequential(): Đây là cách tạo một mô hình tuần tự trong Keras. Mô hình tuần tự là một loại mô hình mạng nơ-ron, trong đó các layers được xếp chồng lên nhau theo một thứ tự cụ thể.
* model\_lstm.add(LSTM(units=50, activation='relu', input\_shape=(X\_train.shape[1], X\_train.shape[2]))):
* LSTM: Đây là layer của mô hình, thực hiện LSTM.
* units=50: Số lượng đơn vị LSTM (hoặc nơ-ron) trong layer. Điều này xác định kích thước của không gian trạng thái ẩn và cũng là kích thước đầu ra của layer này.
* activation='relu': Hàm kích hoạt được sử dụng trong LSTM layer. Ở đây, bạn đang sử dụng hàm kích hoạt Rectified Linear Unit (ReLU), một lựa chọn phổ biến.
* input\_shape=(X\_train.shape[1], X\_train.shape[2]): Đây là hình dạng của đầu vào cho layer LSTM. Nó sử dụng kích thước của dữ liệu đào tạo (X\_train), với X\_train.shape[1] là số lượng thời điểm (time steps), và X\_train.shape[2] là số lượng đặc trưng cho mỗi thời điểm.
* model\_lstm.add(Dense(units=3)):
* Dense: Đây là một layer kết nối đầy đủ (fully connected layer), trong đó mỗi nơ-ron trong layer trước đó được kết nối với tất cả các nơ-ron trong layer này.
* units=3: Số lượng nơ-ron trong layer này, và cũng là kích thước đầu ra của mô hình.
* model\_lstm.compile(optimizer='adam', loss='mse'):
* compile: Bước này được thực hiện để cấu hình quá trình đào tạo của mô hình.
* optimizer='adam': Thuật toán tối ưu hóa được sử dụng. Ở đây, bạn sử dụng Adam optimizer, một phương pháp tối ưu hóa thường được sử dụng trong học máy.
* loss='mse': Hàm mất mát được sử dụng trong quá trình đào tạo. Ở đây, bạn sử dụng Mean Squared Error (MSE), một hàm mất mát thường được sử dụng trong các bài toán dự đoán và hồi quy.

1. Decision tree
2. Random forest