

This code snippet is designed to perform stock price prediction using a pre-trained model and input data. The prediction process involves several steps that make use of the **numpy**, **pandas** and **sklearn.preprocessing** libraries, as well as the **pickle** module for model loading.

Here's a breakdown of the steps involved in the code:

1. Data Preparation

- The input data is assumed to be in the form of a pandas DataFrame.
- The code extracts the "Close" column from the DataFrame, which represents the stock closing prices.
- Missing values in the "Close" column are interpolated to handle any gaps in the data.

2. Data Scaling

- The `MinMaxScaler` from `sklearn.preprocessing` is employed to scale the data between the range of 0 and 1.
- The `MinMaxScaler` ensures that all the values fall within the specified range, making them suitable for processing by the machine learning model.

3. Model Loading

- The pre-trained model is loaded using the `pickle` module's `load` function.
- The model should be stored in a file named "model.sav" and is loaded into the `loaded_model` variable.

4. Prediction

- The scaled data is utilized to predict the next stock price.
- The predicted value is then inverse scaled back to the original range using the `inverse_transform` method of the `MinMaxScaler`.

5. Data Update

- The predicted value is added to the original DataFrame by appending it to the "Close" column.
- The first row of the DataFrame is removed to maintain its length, discarding the oldest stock price.

6. Re-Scaling and Second Prediction

- The updated DataFrame is re-scaled using the `MinMaxScaler`.
- The re-scaled data is fed into the loaded model to predict the subsequent stock price.
- Similar to before, the predicted value is inverse scaled.

7. Result and Return

- The two predicted stock prices are stored in the `ans` list.
- The `ans` list, containing the predicted values, is converted to a numpy array and returned as the output of the function.

