Shape

Description automatically generated with medium confidence

**University of Information Technology**

**REPORT FINAL PROJECT**

****

**Instructor: Nguyen Minh Nhut**

**Students: Group 9**

January 5, 2023

**Menu:**

[TASK 1: 3](#_Toc123775209)

[Linear Regression Model 3](#_Toc123775210)

[A. What is linear regression? 3](#_Toc123775211)

[B. How to build the linear regression model 3](#_Toc123775212)

[C. Why 4](#_Toc123775213)

[D. Using Linear regression model to predict Stock HPG with the rate 10 – 0 4](#_Toc123775214)

[E. Using Linear regression model to predict Stock HAG with the rate 10 – 0 7](#_Toc123775215)

[TASK 2: 10](#_Toc123775216)

[Non-Linear Regression Model 10](#_Toc123775217)

[TASK 3: 10](#_Toc123775218)

[ARIMA Model 10](#_Toc123775219)

[A. Using ARIMA model to predict Stock HAG 10](#_Toc123775220)

[B. Using ARIMA model to predict Stock HPG 17](#_Toc123775221)

[TASK 4: 24](#_Toc123775222)

[LSTM Model 24](#_Toc123775223)

[A. What is LSTM model? 24](#_Toc123775224)

[B. How to build a LSTM model 25](#_Toc123775225)

[C. Why LSTMs are well-suited for time series tasks 25](#_Toc123775226)

[D. Using LSTM model to predict Stock HPG 25](#_Toc123775227)

[With the rate 7-3 25](#_Toc123775228)

[With the rate 8 – 2 32](#_Toc123775229)

[Predict Future 38](#_Toc123775230)

[E. Using LSTM model to predict Stock HAG 47](#_Toc123775231)

[With the rate 7 – 3 47](#_Toc123775232)

[With the rate 8 – 2 54](#_Toc123775233)

[Predict Future 61](#_Toc123775234)

[TASK 5: 71](#_Toc123775235)

[Prophet Model 71](#_Toc123775236)

[References 71](#_Toc123775237)

[Task and member assignment table 71](#_Toc123775238)

# TASK 1:

## Linear Regression Model

## A. What is linear regression?

Linear regression is a linear approach for modelling the relationship between *a scalar response and one or more explanatory variables* (also known as dependent variable and independent variables). The case of one explanatory variable is called simple linear regression; for more than one, the process is called multiple linear regression.[1]

## B. How to build the linear regression model

The basic model for multiple linear regression is:

- In there:

+ Y: dependent variable.

+ Xi: independent variable.

+ β: coefficient of freedom.

+ βi: partial slope.

+ ε: random error.

## C. Why

We are using multivariable linear regression to know that [2]

+ How strong the relationship is between two or more independent variables and one dependent variable.

+ The value of dependent variable at a certain value of the independent variables

## D. Using Linear regression model to predict Stock HPG with the rate 10 – 0

Step 1: Import library and data

Text

Description automatically generated

Step 2: Create a column ‘Index’ to point each line of the array

A screenshot of a computer

Description automatically generated

Step 3: Illustrated representation of Close Stock price

Chart

Description automatically generated

Step 4: In step 1, we imported the linear regression library. Dependent variable Y now we choose column 'price’, the influencing variable is column 'Index’.

A screenshot of a computer

Description automatically generated with medium confidence

🡪From the result, we receive the Intercept and the partial slope

Step 5: Prediction with the result from step 4 with the formula

With n run from 0 to len(HPG) .After calculating ,we stor the result into the new column called ‘Predicted’





Step 6: Evaluate the model through MAE, MAPE, MSE, RSME, R2

Text

Description automatically generated

## E. Using Linear regression model to predict Stock HAG with the rate 10 – 0

Step 1: Import library and data

A screenshot of a computer

Description automatically generated

Step 2: Create a column ‘Index’ to point each line of the array

Text

Description automatically generated

Step 3: Illustrated representation of Close Stock price

Chart

Description automatically generated

Step 4: In step 1, we imported the linear regression library. Dependent variable Y now we choose column 'price’, the influencing variable is column 'Index' .

A screenshot of a computer

Description automatically generated with medium confidence



🡪From the result, we receive the Intercept and the partial slope

Step 5: Prediction with the result from step 4 with the formula

With n run from 0 to len(HAG) .After calculating ,we stor the result into the new column called ‘Predicted’



Graphical user interface, text

Description automatically generated



Step 6: Evaluate the model through MAE, MAPE, MSE, RSME, R2

Text

Description automatically generated

# TASK 2:

## Non-Linear Regression Model

# TASK 3:

## ARIMA Model

### A. Using ARIMA model to predict Stock HAG

Step 1: Import library and data

A screenshot of a computer

Description automatically generated with medium confidence

Step 2: Stationary or non-stationary test Text

Description automatically generated

🡪 The output above shows that the p-value is less than the significance level of 0.05, so we reject the null hypothesis. The series is stationary

Step 3: Selecting p , d ,q order in ARIMA(p , d , q )

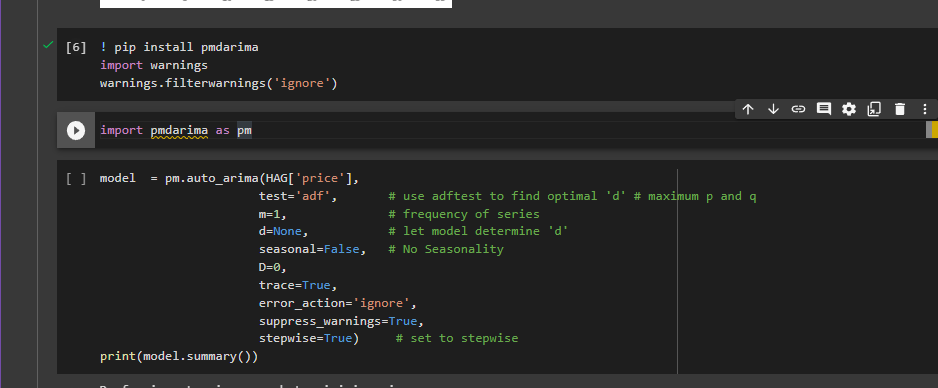
Plot the ACF - AutoCorrelation Function and the PACF - Partitial AutoCorrelation Function though 2 below line code

Graphical user interface, application

Description automatically generated

But before that we must be import plot\_acf,plot\_pacf from the statsmodels.graphics.tsaplots libraries .

Or we can use the auto\_arima in order to find the best model automatic.



Where:  
+ HAG[‘price’]: the one-dimensional array

+ test = ‘adf”:Type of unit root test to use in order to dectect stationarity .

+ m = 1: The period for seasonal differencing, m =1 here mean it’s for annual (non-seasonal) data.

+ d = None: The order of first-differencing,in here we want the it to be selected automatically so we’ll let it be None .

+ seasonal = False: because we had already set m = 1, the seasonal will be set back to False no matter it’s setting True of False

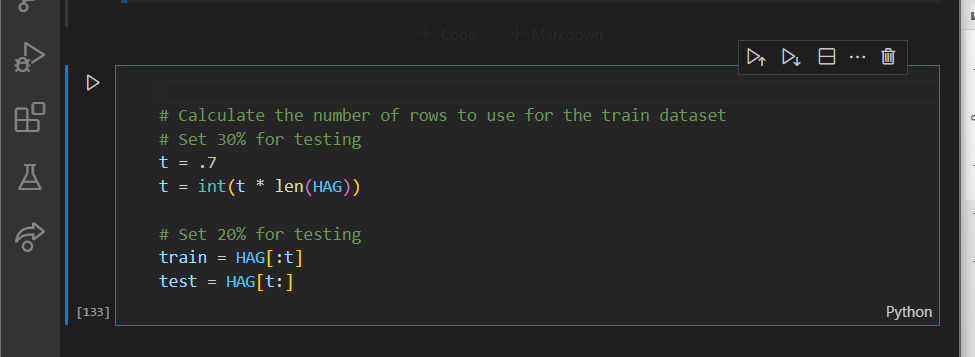
+ trace = True:

+ stepwise = True:

After the processing, the p,d,q order we receive are ARIMA(5 . 1. 3) is best model for this dataset

Step 4: We divided the dataset into 2 small following as train and test .

With the rate of 7 - 3



With the rate of 8 - 2

A screenshot of a computer

Description automatically generated with medium confidenceStep 5: Start to create the model though statmodels.api and train dataset



Text

Description automatically generated

A screenshot of a computer

Description automatically generated with medium confidence

Step 6: Obtain predicted values by using predict() function

**With the rate of 7 - 3**

Graphical user interface, text, application

Description automatically generated

🡪As shown in the image above, the prediction data will be generated and respectively from the index line 1268, which is the end of the train set, and 1811, the length of the whole HAG data set.

**With the rate of 8 - 2**

Graphical user interface, text

Description automatically generated

🡪As shown in the image above, the prediction data will be generated and respectively from the index line 1449, which is the end of the train set, and 1811, the length of the whole HAG data set.

Step 7: Plot the predicted and test dataset

But first, we will copy the test and replace the column ‘price’ with the df in step 6

**With the rate of 7 - 3**

Graphical user interface, chart

Description automatically generated

**With the rate of 8 - 2**

Graphical user interface, chart, line chart

Description automatically generated

Step 8: Evaluate the model through MAE, MAPE, MSE, RSME, R2

**With the rate of 7 - 3**

Text

Description automatically generated

**With the rate of 8 - 2**

Text

Description automatically generated

### B. Using ARIMA model to predict Stock HPG

Step 1: Import library and data

A screenshot of a computer

Description automatically generated with medium confidence

Step 2: Stationary or non-stationary test

Text

Description automatically generated

🡪 The output above shows that the p-value is greater than the significance level of 0.05, so we fail to reject the null hypothesis. The series is not stationary and requires differencing.

Text

Description automatically generated

Step 3: Selecting p , d ,q order in ARIMA(p , d , q )

Plot the ACF - AutoCorrelation Function and the PACF - Partitial AutoCorrelation Function though 2 below line code and import plot\_acf,plot\_pacf from the statsmodels.graphics.tsaplots libraries .

Graphical user interface, application

Description automatically generated

Graphical user interface, chart

Description automatically generated

Or we can use the auto\_arima in order to find the best model automatic.

Graphical user interface, text

Description automatically generated

Where:  
+ HPG[‘price’]: the one-dimensional array

+ test = ‘adf”:Type of unit root test to use in order to dectect stationarity .

+ m = 1: The period for seasonal differencing, m =1 here mean it’s for annual (non-seasonal) data.

+ d = None: The order of first-differencing,in here we want the it to be selected automatically so we’ll let it be None .

+ seasonal = False: because we had already set m = 1, the seasonal will be set back to False no matter it’s setting True of False

+ trace = True:

+ stepwise = True:

After the processing, the p,d,q order we receive are ARIMA(5 . 1. 2) is best model for this HPG dataset

Step 4: We divided the dataset into 2 small following as train and test.

**With the rate of 7 - 3**

**Graphical user interface, text

Description automatically generated**

**With the rate of 8 - 2**

Graphical user interface, text, application

Description automatically generated



Step 5: Start to create the model though statmodels.api and train dataset

Text

Description automatically generated

Text

Description automatically generated

Step 6: Obtain predicted values by using predict() function

**With the rate of 7 - 3**

Graphical user interface, text, application

Description automatically generated

🡪As shown in the image above, the prediction data will be generated and respectively from the index line 1268, which is the end of the train set, and 1811, the length of the whole HAG data set.

**With the rate of 8 - 2**

Graphical user interface, text, application

Description automatically generated

🡪As shown in the image above, the prediction data will be generated and respectively from the index line 1449, which is the end of the train set, and 1811, the length of the whole HPG data set.

Step 7: Plot the predicted and test dataset

But first, we will copy the test and replace the column ‘price’ with the df in step 6

Text

Description automatically generated

**With the rate of 7 - 3**

Chart

Description automatically generated

**With the rate of 8 - 2**

Graphical user interface

Description automatically generated

Step 8: Evaluate the model through MAE, MAPE, MSE, RSME, R2

**With the rate of 7 - 3**

Text

Description automatically generated

**With the rate of 8 - 2**

Text

Description automatically generated

# TASK 4:

## LSTM Model

## A. What is LSTM model?

1. Recurent Neural Networks

* It is a class of neural networks tailored to deal with temporal data. The neurons of RNN have a cell state/memory, and input is processed according to this internal state, which is achieved with the help of loops with in the neural network. There are recurring module of ‘tanh’ layers in RNNs that allow them to retain information. However, not for a long time, which is why we need LSTM models.[3]

2. LSTM model

* It is special kind of recurrent neural network that is capable of learning long term dependencies in data. This achieved becaused the recurring module of the model has a combination of four layers interacting with each other.[4]
* The LSTM cell works by using a combination of three types of gates: forget gate, input gates and output gates.
* The LSTM cell has three internal states, the cell state, the input state, output state. The cell state is able to remember information over a long period of time and is controlled by forget and input gates. The input state is responsible storing new information and is controlled by the input gate. The output state is responsible for passing on information to the next time step and is controlled by the output gate.

Graphical user interface, application

Description automatically generated

## B. How to build a LSTM model

* Preprocessing of time series data: before training an LSTM model on a time series dataset, it is usually necessary to preprocess the data.
* Splitting the data into training, and test sets. After preprocessing the data, it is typically split into training, and test sets. The training sets is used to train the LSTM model, and test set is used to evaluate the model’s performance.
* Training the LSTM model: the LSTM model is trained on the training set using an optimization algorithm.
* Evaluating the model’s performance on the test set: after training the LSTM model, its performance is evaluated on the test set.

## C. Why LSTMs are well-suited for time series tasks

* LSTM are well-suited for time series tasks because they are able to process and learn from long-term dependencies in data. Time series data often exhibits patterns and trends over time, and LSTMs are able to capture and learn from these dependencies to make accurate predictions.

## D. Using LSTM model to predict Stock HPG

### With the rate 7-3

* Import the necessary libraries.

Ảnh có chứa văn bản

Mô tả được tạo tự động

* Import the dataset

Ảnh có chứa văn bản

Mô tả được tạo tự động

* Converts the ‘date’ column in the df Dataframe to a datetime data type using the to\_datetime function from pandas. The format parameter specifies the format of the dates in the ‘date’ column. In this case, the dates are in the format ‘%m/%d/%Y’

Ảnh có chứa văn bản

Mô tả được tạo tự động

* Sets the ‘date’ column as the index of the df DateFrame. An index is a lebel for each row in a Dataframe, and it is used to identify to locate the rows.
* Using DatetimeIndex function of pandas libraries to set the index of date column. This function takes a Datetiem array-like object and return an Index of datetime objects.
* The df[‘date’].values is an array-like object containing the datetime values in the date column.

Ảnh có chứa văn bản

Mô tả được tạo tự động

* Generate descriptive statistics of DataFrame. It returns a new DataFrame containing the count, mean, standard deviation, minimum, maximum and quartiles of the data in each column of the original DataFrame

Ảnh có chứa bàn

Mô tả được tạo tự động

* Data visualization of the stock HPG
* Chart, histogram

  Description automatically generated
* Create a new DataFrame called ‘df1’ that contains only the date and price columns from the original ‘df’ DateFrame.
* Sets the index of df1 to the date column using the index attribute.
* Drops the date column from df1 using drop function.

Ảnh có chứa văn bản

Mô tả được tạo tự động

* Visualize data of the stock HPG

Graphical user interface, chart, histogram

Description automatically generated

* Splits the df1 dataframe into train and test sets.
* Convers the df1 dataframe to a number array using the values attribute.
* Calculates the number of rows to use for the train set using the t variable. T is set to 0.7 which means that 70% of the rows will be used for the train dataset and the remaining 30% for the test set

Ảnh có chứa văn bản

Mô tả được tạo tự động

* Use a MinMaxScaler from sklearn to scale the data in the data array and scale the data by transforming the data to a specific range, in this case the feature\_range parameter is set to (0,1), so the data will be scaled to the range 0 to 1.
* To fit the scaler to the data and transform the data, use the fit\_transform methods of the MinMaxScaler object and passes the data arrays as a parameter.

Ảnh có chứa văn bản

Mô tả được tạo tự động

* Create 2 variables. X\_train is used as input and y\_train is used as target variable
* Loop from the 5 value of the train\_data array and uses the last 5 values as the input features for each iteration. The current value is then used as the target variable.

Text

Description automatically generated

* Converts the x\_train and y\_train arrays to numpy arrays.
* Reshapes the x\_train and y\_train arrays to be one-dimensional.

Ảnh có chứa văn bản

Mô tả được tạo tự động

* Builds a neural network using the Sequential model from the tensorflow.keras library. We can add one or more layers to the model by calling the ‘add’ method of the model object.
* We add three LSTM layers to the models using the LSTM layer from tenserflow.keras. LSTM layers can learn long-term dependencies in the data and are able to remember information from previous time steps.
* We also add two dropout layers using the Dropout layer from tensorflow.keras. dropout is a regularization technique that helps prevent overfitting by randomly droppping a certain percentage of the units in the layer during training. The dropout layers are set to 0.5 , which mean that 50% of the units in the layer will be randomly droppped during training.
* We adds a dense layer with 1 unit using the Dense layer from tensorflow.keras. A dense layer is a fully-connected layer, where all units in the layer are connected to all units in the previous layer. The dense layer is the output layer of the model and has a single unit, which mean that the model will output a single value.
* Compile the model ưith the ‘mean\_absolute\_error’ loss function and the ‘adam’ optimizer.
* Save the model as the name save\_model\_7-3.dhf5
* The model will be saved using the ModelCheckpoint object. The save\_best\_only parameter is set to True, which mean that only the best model will be saved
* Trains the model using the fit method of the model object. The fit method takes the input and output data (x\_train and y\_train), the number of epochs to train for (epochs), and the batch size (batch\_size). The verbose parameter is et to 2, which means that the training progress will be printed to the console. The callback parameter is set to a list containing the ModelCheckpoint object, which will cause the model to be saved during training.

Ảnh có chứa văn bản

Mô tả được tạo tự động

* Load the trained model from the file save\_model\_7-3.hdf5
* Transform the y\_train array back to the original scale
* Use the model to predictions on the x\_train data
* Transform the y\_train\_predict array back to the original scale

Ảnh có chứa văn bản

Mô tả được tạo tự động

* Preprocesses the test data to prepare it for use with the trained model.
* Create the x\_test array for the test data. Loops throuhg the test data, starting from the 5thh value, and appends the last 5 values to the x\_test array as the input features for each interation. Then converts the x\_test to a numpy array and reshapes it to be one-dimensional
* Extracts the true values for the test data from the data array and stores them in the y\_test array.
* The trained model to make predictions on the x\_test data using the predict method of the model.
* Transform the y\_test\_predict array back to the original scale

Ảnh có chứa văn bản

Mô tả được tạo tự động

* data visualization to compare predicted and true values on the df1

Ảnh có chứa văn bản

Mô tả được tạo tự động

Chart, line chart

Description automatically generated

* Evaluate the model through the evaluation metrics on the test set.

Ảnh có chứa văn bản

Mô tả được tạo tự động

* Data visualization to compare predicted and true values on the test set

### With the rate 8 – 2

* Import the necessary libraries.

Ảnh có chứa văn bản

Mô tả được tạo tự động

* Import the dataset

Ảnh có chứa văn bản

Mô tả được tạo tự động

* Converts the ‘date’ column in the df Dataframe to a datetime data type using the to\_datetime function from pandas. The format parameter specifies the format of the dates in the ‘date’ column. In this case, the dates are in the format ‘%m/%d/%Y’

Ảnh có chứa văn bản

Mô tả được tạo tự động

* Sets the ‘date’ column as the index of the df DateFrame. An index is a lebel for each row in a Dataframe, and it is used to identify to locate the rows.
* Using DatetimeIndex function of pandas libraries to set the index of date column. This function takes a Datetiem array-like object and return an Index of datetime objects.
* The df[‘date’].values is an array-like object containing the datetime values in the date column.

Graphical user interface, application

Description automatically generated

* Generate descriptive statistics of DataFrame. It returns a new DataFrame containing the count, mean, standard deviation, minimum, maximum and quartiles of the data in each column of the original DataFrame

Ảnh có chứa bàn

Mô tả được tạo tự động

* Data visualization of the stock HPG
* Chart, histogram

  Description automatically generated
* Create a new DataFrame called ‘df1’ that contains only the date and price columns from the original ‘df’ DateFrame.
* Sets the index of df1 to the date column using the index attribute.
* Drops the date column from df1 using drop function.

Ảnh có chứa văn bản

Mô tả được tạo tự động

* Splits the df1 dataframe into train and test sets.
* Convers the df1 dataframe to a number array using the values attribute.
* Calculates the number of rows to use for the train set using the t variable. T is set to 0.8 which means that 80% of the rows will be used for the train dataset and the remaining 20% for the test set

Ảnh có chứa văn bản

Mô tả được tạo tự động

* Use a MinMaxScaler from sklearn to scale the data in the data array and scale the data by transforming the data to a specific range, in this case the feature\_range parameter is set to (0,1), so the data will be scaled to the range 0 to 1.
* To fit the scaler to the data and transform the data, use the fit\_transform methods of the MinMaxScaler object and passes the data arrays as a parameter.

Ảnh có chứa văn bản

Mô tả được tạo tự động

* Create 2 variables. X\_train is used as input and y\_train is used as target variable
* Loop from the 5 value of the train\_data array and uses the last 5 values as the input features for each iteration. The current value is then used as the target variable.

Ảnh có chứa văn bản

Mô tả được tạo tự động

* Converts the x\_train and y\_train arrays to numpy arrays.
* Reshapes the x\_train and y\_train arrays to be one-dimensional.

Ảnh có chứa văn bản

Mô tả được tạo tự động

* Builds a neural network using the Sequential model from the tensorflow.keras library. We can add one or more layers to the model by calling the ‘add’ method of the model object.
* We add three LSTM layers to the models using the LSTM layer from tenserflow.keras. LSTM layers can learn long-term dependencies in the data and are able to remember information from previous time steps.
* We also add two dropout layers using the Dropout layer from tensorflow.keras. dropout is a regularization technique that helps prevent overfitting by randomly droppping a certain percentage of the units in the layer during training. The dropout layers are set to 0.5 , which mean that 50% of the units in the layer will be randomly droppped during training.
* We adds a dense layer with 1 unit using the Dense layer from tensorflow.keras. A dense layer is a fully-connected layer, where all units in the layer are connected to all units in the previous layer. The dense layer is the output layer of the model and has a single unit, which mean that the model will output a single value.
* Compile the model ưith the ‘mean\_absolute\_error’ loss function and the ‘adam’ optimizer.
* Save the model as the name save\_model\_7-3.dhf5
* The model will be saved using the ModelCheckpoint object. The save\_best\_only parameter is set to True, which mean that only the best model will be saved
* Trains the model using the fit method of the model object. The fit method takes the input and output data (x\_train and y\_train), the number of epochs to train for (epochs), and the batch size (batch\_size). The verbose parameter is set to 2, which means that the training progress will be printed to the console. The callback parameter is set to a list containing the ModelCheckpoint object, which will cause the model to be saved during training.

Ảnh có chứa văn bản

Mô tả được tạo tự động

* Load the trained model from the file save\_model\_7-3.hdf5
* Transform the y\_train array back to the original scale
* Use the model to predictions on the x\_train data
* Transform the y\_train\_predict array back to the original scale

Ảnh có chứa văn bản

Mô tả được tạo tự động

* Preprocesses the test data to prepare it for use with the trained model.
* Create the x\_test array for the test data. Loops throuhg the test data, starting from the 5thh value, and appends the last 5 values to the x\_test array as the input features for each interation. Then converts the x\_test to a numpy array and reshapes it to be one-dimensional
* Extracts the true values for the test data from the data array and stores them in the y\_test array.
* The trained model to make predictions on the x\_test data using the predict method of the model.
* Transform the y\_test\_predict array back to the original scale

Ảnh có chứa văn bản

Mô tả được tạo tự động

* data visualization to compare predicted and true values on the df1

Ảnh có chứa văn bản

Mô tả được tạo tự động

Chart, line chart, histogram

Description automatically generated

* Evaluate the model through the evaluation metrics on the test set.

Ảnh có chứa văn bản

Mô tả được tạo tự động

* Data visualization to compare predicted and true values on the test set

Chart, histogram

Description automatically generated

### Predict Future

* Import the necessary libraries.

Ảnh có chứa văn bản

Mô tả được tạo tự động

* Import the dataset

Ảnh có chứa văn bản

Mô tả được tạo tự động

* Converts the ‘date’ column in the df Dataframe to a datetime data type using the to\_datetime function from pandas. The format parameter specifies the format of the dates in the ‘date’ column. In this case, the dates are in the format ‘%m/%d/%Y’

Ảnh có chứa văn bản

Mô tả được tạo tự động

* Sets the ‘date’ column as the index of the df DateFrame. An index is a lebel for each row in a Dataframe, and it is used to identify to locate the rows.
* Using DatetimeIndex function of pandas libraries to set the index of date column. This function takes a Datetiem array-like object and return an Index of datetime objects.
* The df[‘date’].values is an array-like object containing the datetime values in the date column.

Graphical user interface, application

Description automatically generated

* Generate descriptive statistics of DataFrame. It returns a new DataFrame containing the count, mean, standard deviation, minimum, maximum and quartiles of the data in each column of the original DataFrame

Ảnh có chứa bàn

Mô tả được tạo tự động

* Data visualization of the stock HPG
* Chart, histogram

  Description automatically generated
* Create a new DataFrame called ‘df1’ that contains only the date and price columns from the original ‘df’ DateFrame.
* Sets the index of df1 to the date column using the index attribute.
* Drops the date column from df1 using drop function.

Ảnh có chứa văn bản

Mô tả được tạo tự động

* Converts the df1 to a numpy array and stores it in the data variable.
* Create a MinMaxScaler object with a range of 0 to 1, which will be used to scaled the data.
* The scaler is then fit to the data and used to transform the data.

Ảnh có chứa văn bản

Mô tả được tạo tự động

* Create training datasets for the model
* Loop through the train data, starting from the 5th value
* These 5 values represent the input data for the model. Then, it appends the current value of the ‘sc\_train’ array to the ‘y’ array. This current value represents the target or label for the model.

Ảnh có chứa văn bản

Mô tả được tạo tự động

* Converts the x and y arrays to numpy arrays.
* Reshapes the x and y arrays to be one-dimensional.

Ảnh có chứa văn bản

Mô tả được tạo tự động

* Builds a neural network using the Sequential model from the tensorflow.keras library. We can add one or more layers to the model by calling the ‘add’ method of the model object.
* We add three LSTM layers to the models using the LSTM layer from tenserflow.keras. LSTM layers can learn long-term dependencies in the data and are able to remember information from previous time steps.
* We also add two dropout layers using the Dropout layer from tensorflow.keras. dropout is a regularization technique that helps prevent overfitting by randomly droppping a certain percentage of the units in the layer during training. The dropout layers are set to 0.5, which mean that 50% of the units in the layer will be randomly droppped during training.
* We adds a dense layer with 1 unit using the Dense layer from tensorflow.keras. A dense layer is a fully connected layer, where all units in the layer are connected to all units in the previous layer. The dense layer is the output layer of the model and has a single unit, which mean that the model will output a single value.
* Compile the model ưith the ‘mean\_absolute\_error’ loss function and the ‘adam’ optimizer.
* Save the model as the name save\_model. dhf5
* The model will be saved using the ModelCheckpoint object. The save\_best\_only parameter is set to True, which mean that only the best model will be saved
* Trains the model using the fit method of the model object. The fit method takes the input and output data (x and y), the number of epochs to train for (epochs), and the batch size (batch\_size). The verbose parameter is set to 2, which means that the training progress will be printed to the console. The callback parameter is set to a list containing the Model Checkpoint object, which will cause the model to be saved during training.

Ảnh có chứa văn bản

Mô tả được tạo tự động

* The shape of sc\_train.
* Creates an input array using the last 5 days of the sc\_train, which is scaled version of the df1. Then reshaped into a 2d array with a single row.
* Print the shape of the input array

Ảnh có chứa văn bản

Mô tả được tạo tự động

* Convert the x\_input to a list ans stores it in the temp\_input variable. Then, it extracts the first element of the temp\_input list using indexing ans stores it in the same variable.
* Converts this first element, which is an array, back to list and stores it in the temp\_input variable

Ảnh có chứa văn bản

Mô tả được tạo tự động

* Make predictions for the next 30 days using the LSTM model. It does this by looping through the next 30 days and using the model to make a prediction for each day.
* If the length of temp\_input is greater than 31, use the first 31 elements to make a prediction

+ Extract the first 3 elements of temp\_input and reshape then for input to the model

+ Make a prediction using the model

+ Add the prediction to the temp\_input list and remove the first element

* If the length of temp\_input is less than 31, use the entire list to make a prediction

+ rehspae temp\_input for input the model

+ make a prediction using the model

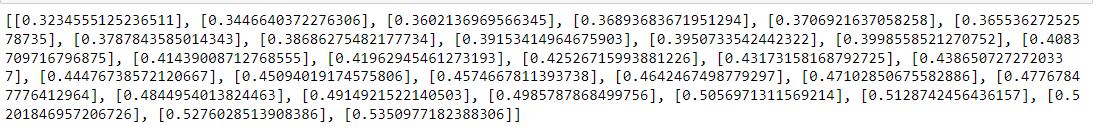
+add the prediction to the temp\_input list

+ add the priction to the lst\_output list

Ảnh có chứa văn bản

Mô tả được tạo tự động

* The result of that after prediction for the next 30 days of the HPG stock



* Converts the date column in the df to a list of strings
* Parse the date strings using strptime(), using the correct format string
* Generate a list of future dates, starting from the last date in the training data and going for 30 days

Ảnh có chứa văn bản

Mô tả được tạo tự động

* Inverse transform the lst\_\_output list using the StandardScaler object

Ảnh có chứa văn bản

Mô tả được tạo tự động

* Creates a p[lot to visualize the actual and predicted prices of a HPG stock.

Ảnh có chứa văn bản

Mô tả được tạo tự động

Chart, line chart

Description automatically generated

* Clearer visualization for the next 30 days of HPG stock

Chart, line chart

Description automatically generated

Ảnh có chứa văn bản

Mô tả được tạo tự động

## E. Using LSTM model to predict Stock HAG

### With the rate 7 – 3

* Import the necessary libraries.

Ảnh có chứa văn bản

Mô tả được tạo tự động

* Import the dataset



* Converts the ‘date’ column in the df Dataframe to a datetime data type using the to\_datetime function from pandas. The format parameter specifies the format of the dates in the ‘date’ column. In this case, the dates are in the format ‘%m/%d/%Y’

Ảnh có chứa văn bản

Mô tả được tạo tự động

* Sets the ‘date’ column as the index of the df DateFrame. An index is a lebel for each row in a Dataframe, and it is used to identify to locate the rows.
* Using DatetimeIndex function of panda’s libraries to set the index of date column. This function takes a Datetiem array-like object and return an Index of datetime objects.
* The df[‘date’]. values is an array-like object containing the datetime values in the date column.

Graphical user interface, application

Description automatically generated

* Generate descriptive statistics of DataFrame. It returns a new DataFrame containing the count, mean, standard deviation, minimum, maximum and quartiles of the data in each column of the original DataFrame

Ảnh có chứa bàn

Mô tả được tạo tự động

* Data visualization of stock HAG.

Chart

Description automatically generated

* Create a new DataFrame called ‘df1’ that contains only the date and price columns from the original ‘df’ DateFrame.
* Sets the index of df1 to the date column using the index attribute.
* Drops the date column from df1 using drop function.

Ảnh có chứa văn bản

Mô tả được tạo tự động

* Splits the df1 dataframe into train and test sets.
* Convers the df1 dataframe to a number array using the values attribute.
* Calculates the number of rows to use for the train set using the t variable. T is set to 0.7 which means that 70% of the rows will be used for the train dataset and the remaining 30% for the test set

Ảnh có chứa văn bản

Mô tả được tạo tự động

* Use a MinMaxScaler from sklearn to scale the data in the data array and scale the data by transforming the data to a specific range, in this case the feature\_range parameter is set to (0,1), so the data will be scaled to the range 0 to 1.
* To fit the scaler to the data and transform the data, use the fit\_transform methods of the MinMaxScaler object and passes the data arrays as a parameter.

Ảnh có chứa văn bản

Mô tả được tạo tự động

* Create 2 variables. X\_train is used as input and y\_train is used as target variable
* Loop from the 5 values of the train\_data array and uses the last 5 values as the input features for each iteration. The current value is then used as the target variable.

Ảnh có chứa văn bản

Mô tả được tạo tự động

* Converts the x\_train and y\_train arrays to numpy arrays.
* Reshapes the x\_train and y\_train arrays to be one-dimensional.

Ảnh có chứa văn bản

Mô tả được tạo tự động

* Builds a neural network using the Sequential model from the tensorflow.keras library. We can add one or more layers to the model by calling the ‘add’ method of the model object.
* We add three LSTM layers to the models using the LSTM layer from tenserflow.keras. LSTM layers can learn long-term dependencies in the data and are able to remember information from previous time steps.
* We also add two dropout layers using the Dropout layer from tensorflow.keras. dropout is a regularization technique that helps prevent overfitting by randomly droppping a certain percentage of the units in the layer during training. The dropout layers are set to 0.5, which mean that 50% of the units in the layer will be randomly droppped during training.
* We adds a dense layer with 1 unit using the Dense layer from tensorflow.keras. A dense layer is a fully connected layer, where all units in the layer are connected to all units in the previous layer. The dense layer is the output layer of the model and has a single unit, which mean that the model will output a single value.
* Compile the model ưith the ‘mean\_absolute\_error’ loss function and the ‘adam’ optimizer.
* Save the model as the name save\_model\_7-3. dhf5
* The model will be saved using the ModelCheckpoint object. The save\_best\_only parameter is set to True, which mean that only the best model will be saved
* Trains the model using the fit method of the model object. The fit method takes the input and output data (x\_train and y\_train), the number of epochs to train for (epochs), and the batch size (batch\_size). The verbose parameter is et to 2, which means that the training progress will be printed to the console. The callback parameter is set to a list containing the ModelCheckpoint object, which will cause the model to be saved during training.

Ảnh có chứa văn bản

Mô tả được tạo tự động

* Load the trained model from the file save\_model\_7-3. hdf5
* Transform the y\_train array back to the original scale
* Use the model to predictions on the x\_train data
* Transform the y\_train\_predict array back to the original scale

Ảnh có chứa văn bản

Mô tả được tạo tự động

* Preprocesses the test data to prepare it for use with the trained model.
* Create the x\_test array for the test data. Loops throuhg the test data, starting from the 5thh value, and appends the last 5 values to the x\_test array as the input features for each interation. Then converts the x\_test to a numpy array and reshapes it to be one-dimensional
* Extracts the true values for the test data from the data array and stores them in the y\_test array.
* The trained model to make predictions on the x\_test data using the predict method of the model.
* Transform the y\_test\_predict array back to the original scale

Ảnh có chứa văn bản

Mô tả được tạo tự động

* data visualization to compare predicted and true values on the df1

Ảnh có chứa văn bản

Mô tả được tạo tự động

Chart, line chart, histogram

Description automatically generated

* Evaluate the model through the evaluation metrics on the test set.

Ảnh có chứa văn bản

Mô tả được tạo tự động

* Data visualization to compare predicted and true values on the test set

Chart

Description automatically generated

### With the rate 8 – 2

* Import the necessary libraries.

Ảnh có chứa văn bản

Mô tả được tạo tự động

* Import the dataset



* Converts the ‘date’ column in the df Dataframe to a datetime data type using the to\_datetime function from pandas. The format parameter specifies the format of the dates in the ‘date’ column. In this case, the dates are in the format ‘%m/%d/%Y’

Ảnh có chứa văn bản

Mô tả được tạo tự động

* Sets the ‘date’ column as the index of the df DateFrame. An index is a lebel for each row in a Dataframe, and it is used to identify to locate the rows.
* Using DatetimeIndex function of panda’s libraries to set the index of date column. This function takes a Datetiem array-like object and return an Index of datetime objects.
* The df[‘date’]. values is an array-like object containing the datetime values in the date column.

Graphical user interface, application

Description automatically generated

* Generate descriptive statistics of DataFrame. It returns a new DataFrame containing the count, mean, standard deviation, minimum, maximum and quartiles of the data in each column of the original DataFrame

Ảnh có chứa bàn

Mô tả được tạo tự động

* Data visualization of stock HAG.

Chart

Description automatically generated

* Create a new DataFrame called ‘df1’ that contains only the date and price columns from the original ‘df’ DateFrame.
* Sets the index of df1 to the date column using the index attribute.
* Drops the date column from df1 using drop function.

Ảnh có chứa văn bản

Mô tả được tạo tự động

* Splits the df1 dataframe into train and test sets.
* Convers the df1 dataframe to a number array using the values attribute.
* Calculates the number of rows to use for the train set using the t variable. T is set to 0.8 which means that 80% of the rows will be used for the train dataset and the remaining 20% for the test set

Ảnh có chứa văn bản

Mô tả được tạo tự động

* Use a MinMaxScaler from sklearn to scale the data in the data array and scale the data by transforming the data to a specific range, in this case the feature\_range parameter is set to (0,1), so the data will be scaled to the range 0 to 1.
* To fit the scaler to the data and transform the data, use the fit\_transform methods of the MinMaxScaler object and passes the data arrays as a parameter.

Ảnh có chứa văn bản

Mô tả được tạo tự động

* Create 2 variables. X\_train is used as input and y\_train is used as target variable
* Loop from the 5 values of the train\_data array and uses the last 5 values as the input features for each iteration. The current value is then used as the target variable.

Ảnh có chứa văn bản

Mô tả được tạo tự động

* Converts the x\_train and y\_train arrays to numpy arrays.
* Reshapes the x\_train and y\_train arrays to be one-dimensional.

Ảnh có chứa văn bản

Mô tả được tạo tự động

* Builds a neural network using the Sequential model from the tensorflow.keras library. We can add one or more layers to the model by calling the ‘add’ method of the model object.
* We add three LSTM layers to the models using the LSTM layer from tenserflow.keras. LSTM layers can learn long-term dependencies in the data and are able to remember information from previous time steps.
* We also add two dropout layers using the Dropout layer from tensorflow.keras. dropout is a regularization technique that helps prevent overfitting by randomly droppping a certain percentage of the units in the layer during training. The dropout layers are set to 0.5, which mean that 50% of the units in the layer will be randomly droppped during training.
* We adds a dense layer with 1 unit using the Dense layer from tensorflow.keras. A dense layer is a fully connected layer, where all units in the layer are connected to all units in the previous layer. The dense layer is the output layer of the model and has a single unit, which mean that the model will output a single value.
* Compile the model ưith the ‘mean\_absolute\_error’ loss function and the ‘adam’ optimizer.
* Save the model as the name save\_model\_7-3. dhf5
* The model will be saved using the ModelCheckpoint object. The save\_best\_only parameter is set to True, which mean that only the best model will be saved
* Trains the model using the fit method of the model object. The fit method takes the input and output data (x\_train and y\_train), the number of epochs to train for (epochs), and the batch size (batch\_size). The verbose parameter is et to 2, which means that the training progress will be printed to the console. The callback parameter is set to a list containing the ModelCheckpoint object, which will cause the model to be saved during training.

Ảnh có chứa văn bản

Mô tả được tạo tự động

* Load the trained model from the file save\_model\_8-2. hdf5
* Transform the y\_train array back to the original scale
* Use the model to predictions on the x\_train data
* Transform the y\_train\_predict array back to the original scale

Ảnh có chứa văn bản

Mô tả được tạo tự động

* Preprocesses the test data to prepare it for use with the trained model.
* Create the x\_test array for the test data. Loops throuhg the test data, starting from the 5thh value, and appends the last 5 values to the x\_test array as the input features for each interation. Then converts the x\_test to a numpy array and reshapes it to be one-dimensional
* Extracts the true values for the test data from the data array and stores them in the y\_test array.
* The trained model to make predictions on the x\_test data using the predict method of the model.
* Transform the y\_test\_predict array back to the original scale

Ảnh có chứa văn bản

Mô tả được tạo tự động

* data visualization to compare predicted and true values on the df1

Ảnh có chứa văn bản

Mô tả được tạo tự động

Chart, line chart

Description automatically generated

* Evaluate the model through the evaluation metrics on the test set.

Ảnh có chứa văn bản

Mô tả được tạo tự động

* Data visualization to compare predicted and true values on the test set

Chart, histogram

Description automatically generated

### Predict Future

* Import the necessary libraries.

Ảnh có chứa văn bản

Mô tả được tạo tự động

* Import the dataset

Ảnh có chứa văn bản

Mô tả được tạo tự động

* Converts the ‘date’ column in the df Dataframe to a datetime data type using the to\_datetime function from pandas. The format parameter specifies the format of the dates in the ‘date’ column. In this case, the dates are in the format ‘%m/%d/%Y’

Ảnh có chứa văn bản

Mô tả được tạo tự động

* Sets the ‘date’ column as the index of the df DateFrame. An index is a lebel for each row in a Dataframe, and it is used to identify to locate the rows.
* Using DatetimeIndex function of pandas libraries to set the index of date column. This function takes a Datetiem array-like object and return an Index of datetime objects.
* The df[‘date’].values is an array-like object containing the datetime values in the date column.

Graphical user interface, application

Description automatically generated

* Generate descriptive statistics of DataFrame. It returns a new DataFrame containing the count, mean, standard deviation, minimum, maximum and quartiles of the data in each column of the original DataFrame

Ảnh có chứa bàn

Mô tả được tạo tự động

* Data visualization of the stock HPG
* Chart, line chart

  Description automatically generated
* Create a new DataFrame called ‘df1’ that contains only the date and price columns from the original ‘df’ DateFrame.
* Sets the index of df1 to the date column using the index attribute.
* Drops the date column from df1 using drop function.

Ảnh có chứa văn bản

Mô tả được tạo tự động

* Converts the df1 to a numpy array and stores it in the data variable.
* Create a MinMaxScaler object with a range of 0 to 1, which will be used to scaled the data.
* The scaler is then fit to the data and used to transform the data.

Ảnh có chứa văn bản

Mô tả được tạo tự động

* Create training datasets for the model
* Loop through the train data, starting from the 5th value
* These 5 values represent the input data for the model. Then, it appends the current value of the ‘sc\_train’ array to the ‘y’ array. This current value represent the target or label for the model.

Ảnh có chứa văn bản

Mô tả được tạo tự động

* Converts the x and y arrays to numpy arrays.
* Reshapes the x and y arrays to be one-dimensional.

Ảnh có chứa văn bản

Mô tả được tạo tự động

* Builds a neural network using the Sequential model from the tensorflow.keras library. We can add one or more layers to the model by calling the ‘add’ method of the model object.
* We add three LSTM layers to the models using the LSTM layer from tenserflow.keras. LSTM layers can learn long-term dependencies in the data and are able to remember information from previous time steps.
* We also add two dropout layers using the Dropout layer from tensorflow.keras. dropout is a regularization technique that helps prevent overfitting by randomly droppping a certain percentage of the units in the layer during training. The dropout layers are set to 0.5, which mean that 50% of the units in the layer will be randomly droppped during training.
* We adds a dense layer with 1 unit using the Dense layer from tensorflow.keras. A dense layer is a fully connected layer, where all units in the layer are connected to all units in the previous layer. The dense layer is the output layer of the model and has a single unit, which mean that the model will output a single value.
* Compile the model ưith the ‘mean\_absolute\_error’ loss function and the ‘adam’ optimizer.
* Save the model as the name save\_model. dhf5
* The model will be saved using the ModelCheckpoint object. The save\_best\_only parameter is set to True, which mean that only the best model will be saved
* Trains the model using the fit method of the model object. The fit method takes the input and output data (x and y), the number of epochs to train for (epochs), and the batch size (batch\_size). The verbose parameter is set to 2, which means that the training progress will be printed to the console. The callback parameter is set to a list containing the ModelCheckpoint object, which will cause the model to be saved during training.

Ảnh có chứa văn bản

Mô tả được tạo tự động

* The shape of sc\_train.
* Creates an input array using the last 5 days of the sc\_train, which is scaled version of the df1. Then reshaped into a 2d array with a single row.
* Print the shape of the input array

Ảnh có chứa văn bản

Mô tả được tạo tự động

* Convert the x\_input to a list ans stores it in the temp\_input variable. Then, it extracts the first element of the temp\_input list using indexing ans stores it in the same variable.
* Converts this first element, which is an array, back to list and stores it in the temp\_input variable

Ảnh có chứa văn bản

Mô tả được tạo tự động

* Make predictions for the next 30 days using the LSTM model. It does this by looping through the next 30 days and using the model to make a prediction for each day.
* If the length of temp\_input is greater than 31, use the first 31 elements to make a prediction

+ Extract the first 3 elements of temp\_input and reshape then for input to the model

+ Make a prediction using the model

+ Add the prediction to the temp\_input list and remove the first element

* If the length of temp\_input is less than 31, use the entire list to make a prediction

+ rehspae temp\_input for input the model

+ make a prediction using the model

+add the prediction to the temp\_input list

+ add the priction to the lst\_output list

Ảnh có chứa văn bản

Mô tả được tạo tự động

* The result of that after prediction for the next 30 days of the HPG stock

Chart, scatter chart

Description automatically generated

* Converts the date column in the df to a list of strings
* Parse the date strings using strptime(), using the correct format string
* Generate a list of future dates, starting from the last date in the training data and going for 30 days

Ảnh có chứa văn bản

Mô tả được tạo tự động

* Inverse transform the lst\_\_output list using the StandardScaler object

Ảnh có chứa văn bản

Mô tả được tạo tự động

* Creates a p[lot to visualize the actual and predicted prices of a HPG stock.

Chart

Description automatically generated

* Clearer visualization for the next 30 days of HPG stock

Chart, line chart

Description automatically generated

Ảnh có chứa văn bản

Mô tả được tạo tự động

# TASK 5:

## Prophet Model

# References

[1]Resource:<https://en.wikipedia.org/wiki/Regression_analysis><https://en.wikipedia.org/wiki/Linear_regression>

[2] Retrieved from multiple-linear-regression (n.d.). [https://www.scribbr.com/statistics/multiple-linear-regression/](http://www.scribbr.com/statistics/multiple-linear-regression/)

[3] Lstm model (n.d.). *https://www.tutorialspoint.com/time\_series/time\_series\_lstm\_model.htm*.

[4] Lstm neural network model (n.d.). *https://towardsdatascience.com/exploring-the-lstm-neural-network-model-for-time-series-8b7685aa8cf*

# Task and member assignment table

Group Member: Group 9

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | ID | Task | Status | Leader |
| Nguyễn Minh Thành | 20521920 | Task 1 – task 3 | Done | X |
| Nguyễn Văn Tân | 20521880 | Task 4 | None |  |
| Nguyễn Phan Hiếu Thuận | 20521994 | Task 2 – task 5 | Done |  |