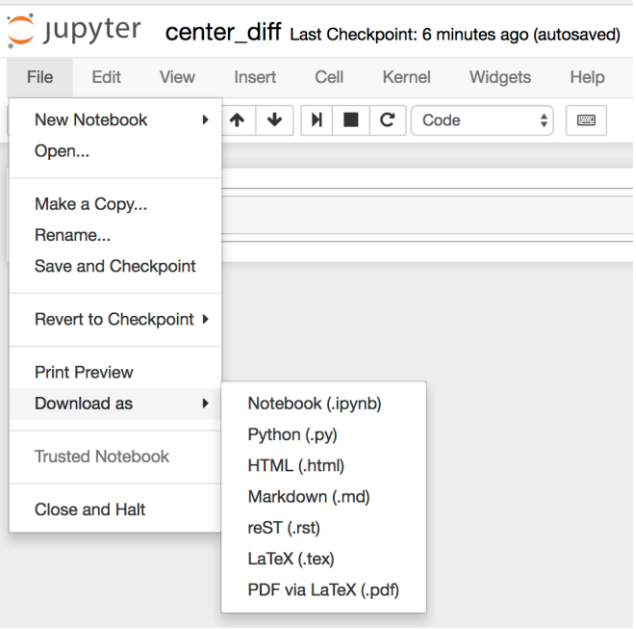
Homework 5
Programming, Due 9:00, Tuesday, December 6, 2022

Late submission within 24 hours: score*0.9;
Late submission before post of solution: score*0.8 (the solution will usually be posted within a week); no late submission after the post of solution)

HW Submission Procedure (請仔細閱讀)

1. For concept and derivation, please write them in a professional format and submit a pdf file. Name your pdf file `YourID_HW5.pdf`, for example, `n96081494_HW5.pdf`

2. You should submit your Jupyter notebook and Python script (`*.py`, in Jupyter, click File, Download as, Python (`*.py`)).



3. Name a folder using your student id and HW number (e.g., n96081494_HW5), put the pdf and all the Jupyter notebooks and python scripts into the folder and zip the folder (e.g., n96081494_HW5.zip).

4. Submit your HW directly through the course website.

Total 100%

1. (**50%**) **Please download the zip file HW5.zip from Moodle.** Name your Jupyter notebook `RNN_SPAM` and Python script `RNN_SPAM.py`. Please create an RNN model to classify spam. The dataset SPAM.csv can be downloaded from Moodle.

   Data preprocessing is to convert the text in Message to a vector. You can use the function `Tokenizer` to complete.
   (`from tensorflow.keras.preprocessing.text import Tokenizer`)

   Message

   Text: `'Received, understood n acted upon!'`

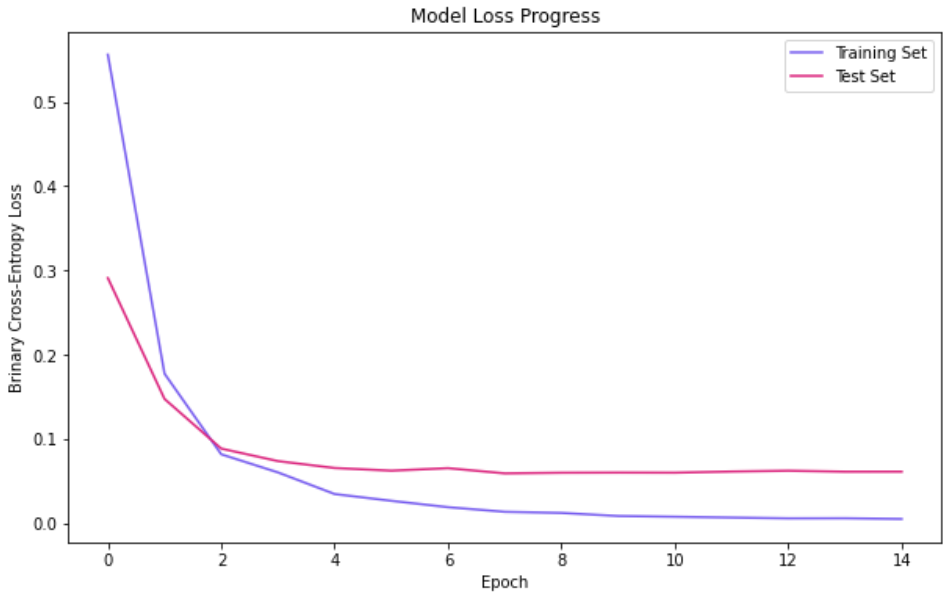
   ↓

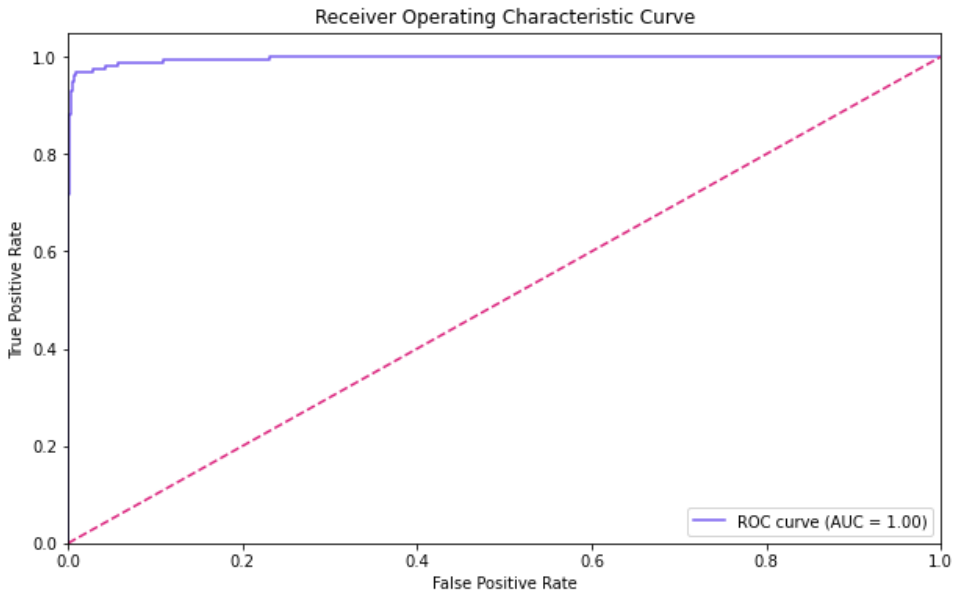
   Vector: `[1903. 3620. 98. 2462. 3621.]`

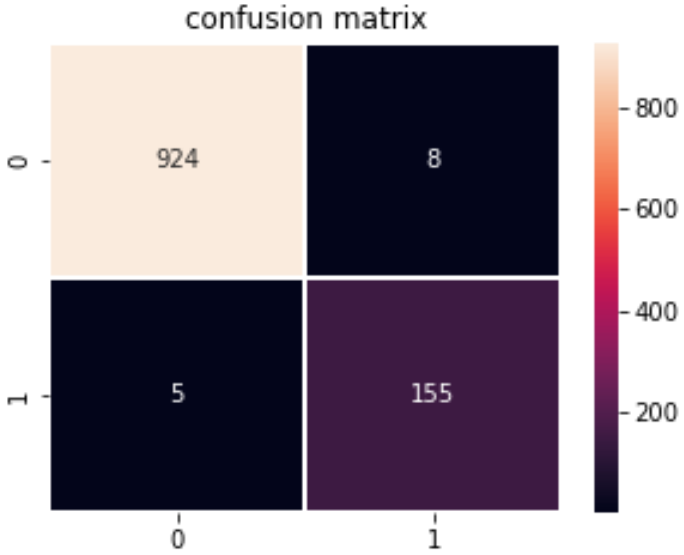
   After converting the text to a vector, use `sequence.pad_sequences` (`from tensorflow.keras.preprocessing import sequence`) to fill the length of all sequences to the same length. The following below is the shape of data after pre-processing.

   ```
   x_train shape: (4367, 200)
   x_test shape: (1092, 200)
   ```

   Use the pre-processed data to train the model, please report training history, roc curve, confusion matrix and balanced_accuracy_score.



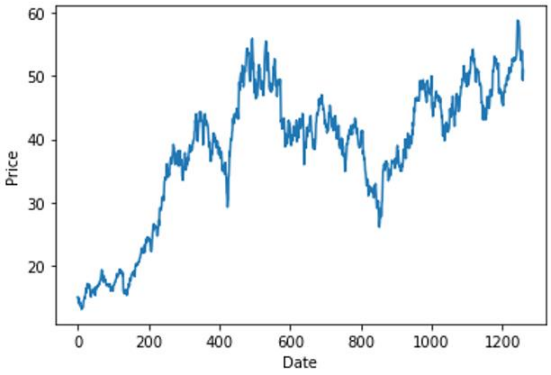




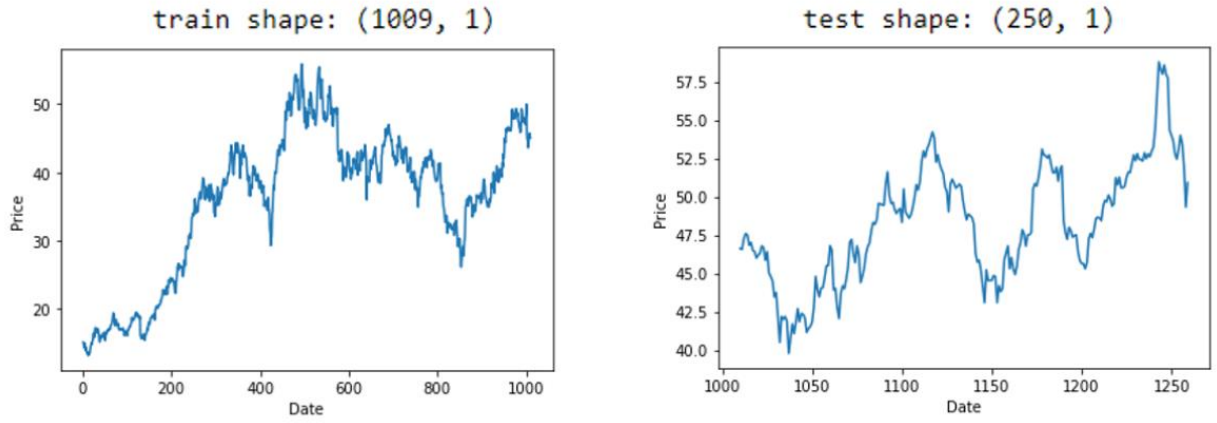

```
balanced_accuracy_score: 0.98008
```

3. (50%) **Please download the zip file HW5.zip from Moodle.** Name your Jupyter notebook `lstm_stock` and Python script `lstm_stock.py`. Please create a lstm model to predict stock price. The dataset `stock`.csv can be downloaded from Moodle. The dataset contains dates and prices, the following is the data presented in tables and graphs.

	date	open
0	2013/2/8	15.07
1	2013/2/11	14.89
2	2013/2/12	14.45
3	2013/2/13	14.30
4	2013/2/14	14.94
...	...	...
1254	2018/2/1	54.00
1255	2018/2/2	53.49
1256	2018/2/5	51.99
1257	2018/2/6	49.32
1258	2018/2/7	50.91



Please divide the data into training set and testing set. The first 1009 data are used for model training, and the last 250 data are the test set. The figure below shows data shape and graphs.



The data pre-processing is to take the data of previous dates to predict the data of the next date. The figure below is an example. The figure uses the 3 data as inputs to predict the next data. (Here we use 3 pieces of data as an example, you can decide by yourself how many data you want to use as inputs.)

Features

	date	open
0	2013/2/8	15.07
1	2013/2/11	14.89
2	2013/2/12	14.45

Label →

	date	open
3	2013/2/13	14.30

	date	open
1	2013/2/11	14.89
2	2013/2/12	14.45
3	2013/2/13	14.30

	date	open
4	2013/2/14	14.94

.....

.....

	date	open
1245	2018/1/19	58.59
1246	2018/1/22	57.99
1247	2018/1/23	57.74

	date	open
1248	2018/1/24	54.35

X_train shape: (1006, 3, 1)

	date	open
0	2013/2/8	15.07
1	2013/2/11	14.89
2	2013/2/12	14.45
3	2013/2/13	14.30
4	2013/2/14	14.94
...	...	...
1004	2017/2/3	44.31
1005	2017/2/6	44.80
1006	2017/2/7	45.75
1007	2017/2/8	45.26
1008	2017/2/9	45.07

1006

X_test shape: (250, 3, 1)

	date	open
1006	2017/2/7	45.75
1007	2017/2/8	45.26
1008	2017/2/9	45.07
1009	2017/2/10	46.62
1010	2017/2/13	46.56
...	...	...
1254	2018/2/1	54.00
1255	2018/2/2	53.49
1256	2018/2/5	51.99
1257	2018/2/6	49.32
1258	2018/2/7	50.91

250

Please show the model and plot training history and the results of testing dataset (the last 250 data) predictions.

```
Model: "sequential_3"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
lstm_3 (LSTM)                (None, 3)                 60
_________________________________________________________________
dense_3 (Dense)              (None, 1)                 4
=================================================================
Total params: 64
Trainable params: 64
Non-trainable params: 0
```