

---

# PLACEMENT INTERVIEW TASK

---

## SECTION 1 - INFORMATION

### SECTION 1.1 - OVERVIEW

This interview task will allow you to show off your ability to use python and demonstrate your understanding of machine learning. You be able apply your critical thinking and problem-solving skills with respects to an industry scenario. You will be asked to complete a small project that has been splits into 4 primary tasks and a completely optional secondary task.

You will be provided with a python Jupyter notebook file that has some starting code in which will help you with some of the tasks. The notebook and code provided is optional to use but highly recommended. Also, you will be provided with a Json file with all the data needed for the tasks.

We would like the finish code to be completed in a structured and well documented python Jupyter notebook (.ipynb file). Additionally, we would like any other files to be clear and easy understand with meaningful names where applicable.

Once you have completed all the tasks you were able to do, upload the code and other files to a private GitHub repository, have your name & student ID somewhere in the repository, and add the user 7n14 (<https://github.com/7n14>) (project member called Harry) to the repository. More information about the submission can be found in section 2.5.

Initially, the document might seem extensive or overwhelming, but there's no need for concern. Below, you'll find several questions. We encourage you to tackle as many as you can. It's perfectly fine if some questions seem challenging or if you can't solve them. Please, do not feel pressured. We are interested in observing your approach and progress, as it will offer insightful information to determine if this position is a good fit for you.

*Please make notes within the repository of software used, plugins, programming languages versions and system details so we can make sure we can reproduce your results. More details on how to make your code reproduce the same results can be found in the python Jupyter notebook provided.*

---

## AI USAGE

Usage of AI tools like ChatGPT is allowed but please clearly document in your code and any other documents created where you have used AI tools to generate content. Otherwise, we won't be able to get clear idea what areas your strengths and weaknesses are.

Example of documentation in code:

```
# AI Tool Used  
Insert Code Here  
# End of AI Tool Used
```

## SECTION 1.2 - CONCEPT

For this interview task you will be taking the role as a data analyst/programmer working for Pizza Hut and have been tasked to finish a project that an employee did not complete before needing to be moved to a more important task. The employee has started work on some of the code but has left out key components. Your job is to complete the code, produce results to meet your managers requirements, then share the code & results with your manager via GitHub.

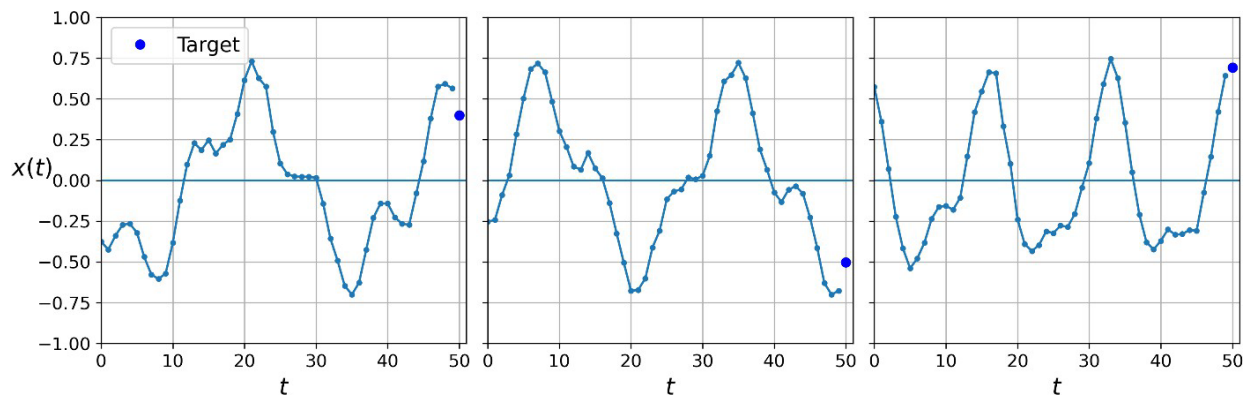
## SECTION 1.3 - QUESTIONS

If you have any questions or have any problems, please contact either [t.chen@hud.ac.uk](mailto:t.chen@hud.ac.uk), [h.gouge@hud.ac.uk](mailto:h.gouge@hud.ac.uk), [ai4sw@hud.ac.uk](mailto:ai4sw@hud.ac.uk) or [u2150435@unimail.hud.ac.uk](mailto:u2150435@unimail.hud.ac.uk) and we will aim to back to you within 24 hours.

## SECTION 2 - THE BRIEF

### SECTION 2.1 - OVERVIEW

Your manager, who overseeing this project, wants to see if there is a way to predict the sales across all of Pizza Huts physical locations. You have been provided a Json file with all 15,000 physical stores sales across 62 months compared to average sales of the month, where positive being above average and negative being below average. See below for example graphs with first 50 months sales,  $x(t)$ , across time,  $t$ , and then 51st as a target at  $t_{50}$  :



*Disclaimer: Not actual data from Pizza Hut, its randomly generated but uses the fact in 2019 Pizza Hut had 15,00 physical stores*

### SECTION 2.2 - PRIMARY TASKS

*Try and attempt as many tasks as possible, if you get stuck/unable to complete any of the tasks move onto the next task. If you struggle with either task 2 or 3, you can move onto task 4 and use the data that's already provided or extra data generated from task 2.*

#### Task 1

First, you need to load the provided Json file into your code and format it. Your manager wants to first see if you can predict 1 month in advance. They want you to store the first 50 months store sales in an array as an input/X and then the 51<sup>st</sup> store sale in a parallel array as the output/Y. Use the example below to help understand.

*To confirm that you formatted correctly for Parallel Array when you convert them into a NumPy array their shape (X.shape) value should (15000, 50, 1) and (15000, 1)*

## Parallel Array Example

---

```
X = [[0.12], [-0.24], [-0.38], [-0.02], [0.49], ...],  
[[0.56], [-0.74], [-0.91], [-0.63], [0.23], ...], ...]  
Y = [[0.87], [0.63], ...]
```

---

## Task 2

---

### Part 1

Now, in the code provide you need to change the existing time series model as it's a place holder, and it's considered to be sub-optimal. So, you need to change, remove, and add new layers to the model to best suit the problem. Make brief notes on what you tried and how successful it was with regards to part 2 of the task. Structure the notes in a way that you state all the layers of the attempt, what changed from previous, why, testing parameters and results.

*See below for example TensorFlow Keras layers you can use:*

- *Keras.layers.Flatten*
- *Keras.layers.Dense*
- *Keras.layers.SimpleRNN*
- *Keras.layers.GRU*
- *Keras.layers.LSTM*
- *Keras.layers.TimeDistributed*

*You can use any other layers that are available as long as you explain why you choose to use them. Additionally, you can change the learning rate of the optimizers but changing loss or the optimizer type is not needed to complete the task. See documentation for more information: [TensorFlow python docs](#)*

*Tip: Try to balance the width and depth of your model to help with training time & keeping it simple*

---

### Part 2

Using the data from the Json that you formatted, convert it into a NumPy array and then train the model using a split 70% for training, 20% for validation and 10% for testing. To train the model, use the provided `model.compile` and `model.fit` methods.

After finishing training evaluate the model and see if it meets the requirements of your manager. To evaluate the model use `model.evaluate(X_valid, y_valid)` and your manager wants the output to be lower than 0.004168. If you struggle to get lower than 0.004168 but achieve an output lower than 0.01 that would be considered acceptable but not ideal.

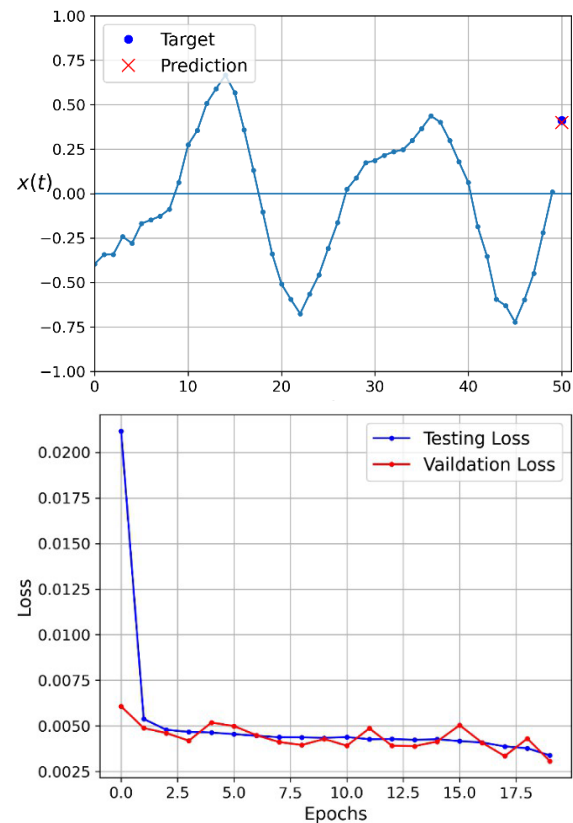
*Hint: To meet the requested  $<0.004168$ , the model will need to be a deep recurrent neural network (Have multiple layers by using `return_sequences=True` in the layer parameters)*

---

## PART 3

Your manager wishes to see visually see the prediction of the model. Produce a graph similar to one on the right either by creating your own graphs or by using some code that the previous developer left you:

Also, your manger would like to see a training & validation loss graph diagrams for each model you try. Utilize the data returned by the model fit method which contains the testing and validation loss data. They want you store them in a folder called “images” and the image names need to be clearly labeled. See example on the right:



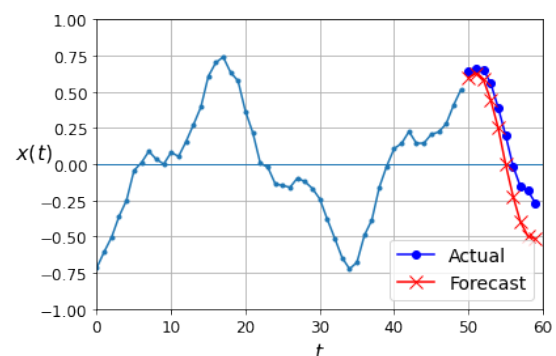
---

## Task 3

Once you meet the managers' requirements, they want to explore the idea of predicting multiple months in advance. Refactor your code so it's able to forecast 12 months instead of 1 month. Manager leaves it up to you how you wish to complete this task. 12 months is the main goal but if your able to get forecast at least 6 months your manager will consider this acceptable.

*Tip: The shape of the input and output arrays should have a shape of (15000, 50, 1) and (15000, 12, 1)*

Your manager wants you to display the results as a graph. Some code has been provided but potentially needs a bit of refactoring to work with your solution or you can code your own graphs. Example graph to the right.



---

## Task 4

After showing off the graph to your manager, they want you to create a csv file that includes the predicted sales for the next 12 months of the stores (month 63 and onwards). They want the csv file to include the store's unique ID (found in the Json file), all of the previous sales, the 12 months of predicted sales and any other relevant information you can think could be useful for analysis of each store (e.g., average, lower quartile, upper quartile, ...).

## SECTION 2.3 - SECONDARY TASK

This is a completely optional task and is only here if you want to demonstrate your existing experience with JavaScript and the frameworks surrounding it. Do not worry about this task if you don't have time to complete or don't know how to complete the task. Please prioritize other schoolwork/assignments over this task. If you do take on the task don't feel stressed to complete every feature requested, complete as many as you can.

Once you have completed the 4 primary tasks, your manager suggests it would be nice to have an app to easily view all of the different stores' predicted sales. They say to use Expo ([Expo Docs](#)) as they have used it before. They have given you a list of features in order of importance that they would like to see if you do create the app:

- A graph to view the sales data from the csv file or any other file type you created for task 4
  - Clear labels what is previous and predicted sales average
  - (Optional) 2 separate lines for previous and predicted sales average
- Way to search for a store ID and get the graph for said store
  - Either on the same page and the graph updates, or 2 pages one for searching and the other where the graphs are
- Additional graphs for any other data generated in task 4
- The ability to add new sale record to a store, replacing the predicted value
- Ability to view 2 store sales averages at the same time
  - Either on the same graph, or separate graphs
- Have a list of the top 10 best performing store
  - One for the previous 12 months
  - One for the predicted best performing stores across the next 12 months

Make sure to document clearly how you ran and tested the app, so it's easier for us to test the app ourselves.

## SECTION 2.4 - OTHER PROJECTS

If you have any other previous projects that is relevant with regards to the job descriptions person specification, and you wish to show them off to us to prove you are capable of said skills and programming languages. Either have link to the public repository or add the project files to the repository in a sperate clearly labeled folder. Make sure to document what the purpose of the project is, what skills/languages used and if it was a solo project or group project.

*Note: Make sure you have permission to show the project code and files from anyone who might have worked on the project, and remove any sensitive data that might be stored with in the project*

## SECTION 2.5 - SUBMISSION

You have until the interview date/time to work on these questions. Before then, you need to have created and shared a private GitHub repository with team member Harry by adding the user 7n14 (<https://github.com/7n14>). We will continue to monitor the progress that may be updated by interview time, but you need to share the repository **by March 06**. Please divide the repository with folders if you do attempt the optional task & include any other projects. The repository should include:

- All of your code in a .ipynb file
- Any other files & folders used for optional task
- Folders for any other project you wish to show to us
- Folder holding all images of graph plots
- Notes of versions of software & languages used
- File that has your name and student ID (recommend writing these in a README.md file)
- Include any other files that you created or used whether this being a file for recording your different model attempts