# COS30018 - Option B - Task 2: Data processing 1

As stated in the Task B.2 document, the current model (v0.1) has a few fatal flaws with the way it’s designed. Specifically, with the data processing, as it requires users:

* Must manually choose the start date and end date for the training data and then start date and end date for the test data.
* The current version v0.1 just ignores the dataset features that can be obtained (e.g., Open, High, Low, Volume, AdjClose), instead opting to use the feature Close only.

The new version of the file is titled v0.2, these issues were rectified by altering the v0.1 file and learning from P1 and doing additional research:

**This function will allow you to specify the start date and the end date for the whole  
dataset as inputs.**

Data was originally loaded directly into the file using a hard-coded start and end date. The user also had to manually input the dates for training and testing periods, whilst only the “close” price feature was able to be utilised.

In v0.2, the new function load\_and\_process\_data was implemented, allowing the date range for the complete dataset to be specified, and the ability to use multiple features ("Open," "High," "Low," "Volume," and "Adj Close"). The data can also be saved locally if the user wants to use it in the future, meaning that the user doesn’t have to download it repeatedly.

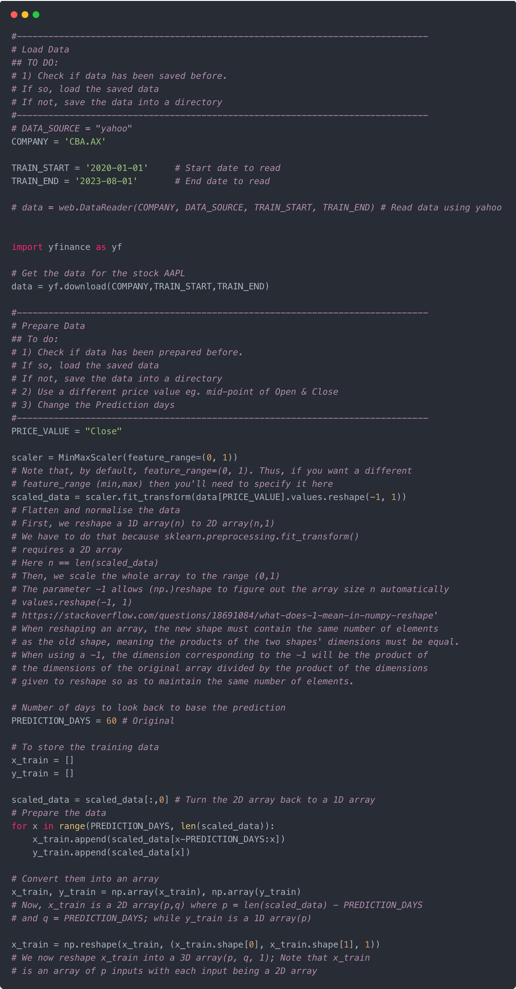


Figure 2: v0.2 Data Loading

Figure 1: v0.1 Data Loading

**This function will allow you to deal with the NaN issue in the data.**

In v0.1 there was no handling of NaN values, whereas in v0.2 the load\_and\_process\_data function has an if statement to handle NaN values. This allows for better data integrity before it is fed into the model.

A screen shot of a computer code

Description automatically generated

Figure 3: NaN values

**This function will also allow you to use different methods to split the data into  
train/test data; e.g. you can split it according to some specified ratio of train/test and  
you can specify to split it by date or randomly.**

In v0.1, data was split manually into train and test sets by defining separate start and end dates. This needs to be altered in v0.2 to allow the data to be split based on a ratio specified by the user (e.g., 50% training, 50% testing) or by specific dates. This change will allow for experimentation with different approaches for splitting data.

**This function will have the option to allow you to store the downloaded data on your  
local machine for future uses and to load the data locally to save time.**

No method to store / download data locally existed in v0.1 of the program. This resulted in the user needing to download the dataset whenever they ran the program. However, in v0.2 the processed data can be saved locally, reducing redundancy and saving time.

A computer screen shot of a code

Description automatically generated

Figure 4: Locally stored

**This function will also allow you to have an option to scale your feature columns and  
store the scalers in a data structure to allow future access to these scalers.**

Originally in v0.1, only the Close price was scaled using MinMaxScaler, with the scaling range hardcoded to (0, 1). In v0.2, the newly implemented function allows the scaling of multiple features and flexibility when defining the scaling range. Also, the scaler objects can be stored which allows for consistent scaling across different runs and new incoming data.

A screenshot of a computer program

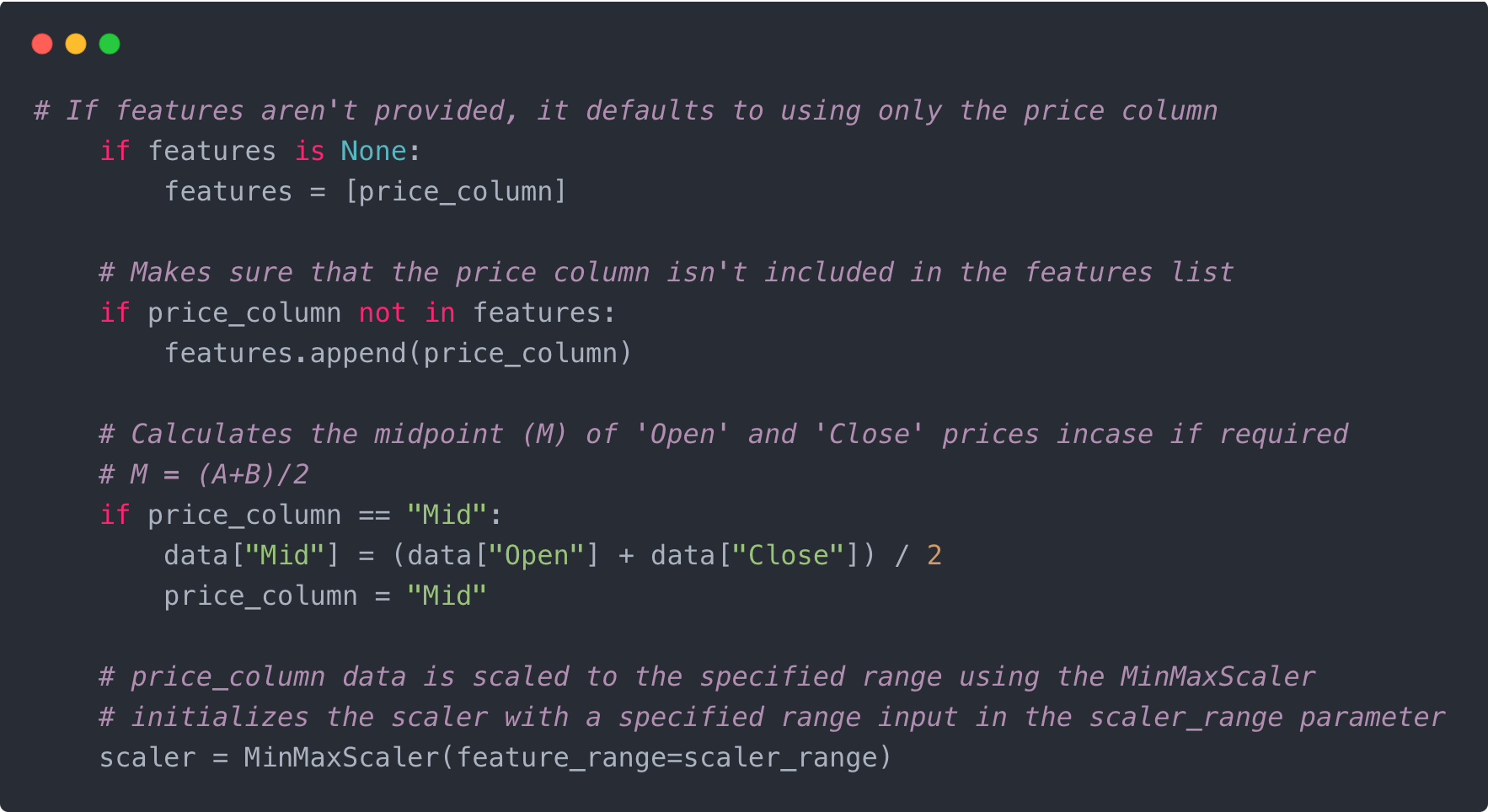
Description automatically generated

Figure 6: List of Features and data processing

Figure 5: Features