DMP03 FAQs

1. How can we use technical indicators like RSI in machine learning for predictions?

   We can use RSI or any other technical indicators as an input feature. Adding new features would involve adding new columns (that we assume have predictive power) in the dataset that we use to train the ML model. The following resources discuss similar examples:
   - https://blog.quantinsti.com/machine-learning-classification-strategy-python/
   - https://blog.quantinsti.com/gold-price-prediction-using-machine-learning-python/
   - https://blog.quantinsti.com/machine-learning-logistic-regression-python/
   - https://blog.quantinsti.com/support-vector-machines-introduction/
   - https://blog.quantinsti.com/covered-call-strategy-machine-learning/
   - https://blog.quantinsti.com/trading-using-machine-learning-python/

2. How do we know how the model arrives at the prediction?

   Machine learning models mostly work as black boxes. We can figure out the weights that the model has calculated for each feature. However, it is difficult to specifically identify the non-linearity that the model captured and how it gave us the predictions.

3. Is it possible that a completely different pattern may evolve when analyzing data using machine learning outside of a textbook?

   Yes. We, as practitioners, can recognize only a few of the known patterns by looking at the data or charts. But, machines have the power to analyze and recognize many hidden and unknown patterns which are impossible for humans to identify.

4. Is the machine learning function purely based on training 'numbers' as input? Or can we feed in price movement 'patterns' and let the machine identify which stocks have a high degree of 'fit' against those patterns?

   Machine learning models usually take numeric values as inputs. Now, the numeric values that we feed can come from any source. It might be technical analysis metrics, chart patterns, volatility regimes, or anything we think carries predictive power. No matter what kind of data we choose to work with, we need to convert it to numbers before feeding it to ML models.

5. What is the **linalg** that we used in the class?

   It is a sub-module in **numpy** for linear algebra. We use the **lstsq** function from it to find the least-squares estimators for multiple linear regression. We perform the linear regression using first principles with the help of this sub-module. Packages like **statsmodels** and **sklearn** provide alternatives to perform linear regression where we can directly provide the input and get the regression coefficients.

6. What do we mean by a time series?

   A time series is a sequence of observations indexed over time. For example, stock price, records of quarterly results, stock returns observed over time. The sequence mentioned is mostly equally spaced observations in time. It can also be irregular observations but they are harder to analyze and work with.