Question 1 of 6

Q1/6: The accuracy of the predictions about the live data are not as good as the results that you showed during the training of the model. Why do you think this is?

- ○ You must have implemented the code wrong
- ○ The live data must include different columns than the ones I trained it on
- ◉ The model is underfit
- ○ The model is overfit

Great Work!

Well done! This is most likely the case, it seems as though the model is not generalizing to new data well, so the trained model was not trained on a representative sample.

Question 2 of 6

Q2/6: Can you suggest a way to improve the predictions on the live data?

- ○ Train it again on the same data
- ○ Reduce the dataset that it was trained on
- ○ Use a larger percentage when splitting the data for training
- ◉ Collect more rows of data to train the model with

Great Work!

Well done! It appears as though a classic case of "underfitting" is occurring, where the trained model has not been trained on enough examples to generalize to unseen data. To alleviate this issue, we should train the model on a larger sample.

Question 3 of 6

Q3/6: The client has offered some additional datasets that may be useful. Can you let us know which two datasets you think would be best to include in the model?

- [x] Weather, deliveries
- [ ] Weather, HR
- [ ] Weather, customer satisfaction
- [ ] Weather, pollution

Great Work!

Well done! It is hard to know without seeing any of the data which combination would be best, but considering that we are predicting stock levels, stock levels will be dependent on how much stock the store has available. So, using data about stock being delivered could be useful. Weather will also play a big part since people buy specific products when the weather changes!

Question 4 of 6

Q4/6: We would like to explore the possibility of using a more complex machine learning algorithm to see how it compares to the current Random Forest. Can you suggest one to try?

- ○ Support Vector Machine Regressor
- ◉ Neural Network
- ○ Linear Regression
- ○ Naive Bayes Regressor

Great Work!

Well done! Neural networks are very complex and powerful algorithms, due to the inherent structure of how the algorithm learns from data.

Question 5 of 6

Q5/6: What would be a disadvantage of using the more complex model from the previous question, against the current Random Forest?

- ○ Worse results
- ○ Having to re-write all the code
- ◉ More difficult to explain the algorithms results
- ○ Having to use different packages

Great Work!

Well done! With great power and complexity comes a great challenge when interpreting results. With the random forest regressor, you can output the splitting points that the algorithm used to make predictions as well as feature importance. Due to the nature of how a neural network is built, it is very difficult to explain how the trained model makes its decisions.

Question 6 of 6

Q6/6: Can you suggest a way that we can optimize the performance of the current Random Forest algorithm? In particular, we want to know how we can improve the MAE of the current algorithm.

- [x] Tune hyperparameters
- [ ] Add more folds to cross-validation
- [ ] Duplicate rows within the data for training
- [ ] Train it on just 1 fold

Great Work!

Well done! Every algorithm has a number of parameters that can be adjusted and has an impact on how the algorithm learns. The random forest regressor has a number of parameters that can be tweaked. We can run an optimization job to try and find the best combination of these parameters for our dataset!