Elena Markoska

Exercise 1: Prioritisation

I'm grateful for my team's suggestions and encourage a continuous free-flowing debate. Given the complexity of the problem and the short amount of time available to develop a prototype, we must be efficient in how we split our time across different tasks.

1. The very first step that should be undertaken once the data becomes available is some initial Exploratory Data Analysis (EDA). I would first ask my team to produce summary statistics of the existing data to gain an understanding of the variety of the data across products, factories, types of time-series, and possible correlations between them. Various visualisation techniques should also be employed to understand the distributions, patterns, and trends that likely exist, which is what **Suggestion 2** partially points to. This is a relatively low-effort task that yields very high benefit and informs the future steps meaningfully.
2. Once the results from the EDA are available, seniors within the team should approach domain experts (**Suggestions 6 and 8**) to gain a better understanding of the problem at hand that the team is attempting to automate. Experts can provide ample advice on their current processes, rule-based approaches that likely already exist, statistical tests that have proven useful. As the data is varied, the domain expert can further advise on differences between factories themselves (some factories can be much larger than others and therefore data from them can be much different; some factories can produce products with differing properties than the products produced in other factories; etc), as well as the differences between the individual time-series data (similar time-series may be grouped and receive different treatment, e.g. labour costs are likely different than quantity of various materials). Experts are likely to have witnessed anomalous data points, or even anomalous series entirely, and will likely be able to advise on what anomalies are likely to look like, which in turn could help produce target variables of anomalous/correct data. This is a medium-effort, but high benefit task.
3. With the knowledge obtained from our EDA and the domain experts, **Suggestion 9** is a logical next step. By examining previously received inaccurate data, we can consider various possibilities for features that may be relevant in the future steps when we attempt to build models to detect anomalies. Possible features may be measures of the magnitude of deviation from the data to be classified from the previous (or mean) data point, measures of seasonality, volatility, dispersion, fitting linear regression models to estimate a slope as a proxy for the trend of the time-series, moving averages, z-scores, first and second differences, measurements of entropy to quantify the complexity/irregularity of the data, etc. This is a low-effort task that will yield a number of potentially highly valuable features.

4. Depending on how many people are in my team, **Suggestions 3, 4, and 5** could all be done in parallel, which may offer further understanding of the problem through collaboration. These are all medium-effort / high-benefit tasks, but given the nature of iterative, experimental model development, they could become time-consuming.
    a. I would recommend starting with **Suggestion 5** as it is relatively simple to implement some ML models (random forest, xgboost, logistic regressions, SVMs, etc) given an existing target variable. These models should be evaluated for their performance in specificity (high specificity indicates the model is performant at minimising false positives and effectively identifies true negatives) or negative predictive value (how well the model predicts true negatives among all predicted negatives), as opposed to relying on simple accuracy. Care should be taken to ensure training on a balanced dataset as there will inevitably be more non-anomalous data points than anomalous ones.
    b. **Suggestion 3** indicates a clustering approach on a factory level, however lower granularity would be better. Further, grouping similar time-series across different factories could yield improvements as models would be trained on groups individually (instead of all data, e.g. modeling labour costs separate from transport costs). I'd recommend K-Means clustering or DBSCAN (using dynamic time warping as a distance metric between time series) — K-Means demands configuration of the number of clusters, while DBSCAN does not, which is likely to yield differing and informative outcomes. Given the differences in factories/products, the data would have to be standardised or normalised for a meaningful peer comparison.
    c. **Suggestion 4** could be used to flag possibly inaccurate data, however considering that it is an unsupervised approach, it could also be used to generate new features that may benefit the models in a) and b), too. Algorithms I'd ask my team to attempt under this suggestion are an Isolation forest (unsupervised ensemble decision tree designed to isolate anomalies), or a One-Class SVM (train an SVM on non-anomalous data to find a boundary around the non-anomalous ones, that can then be used to detect the anomalous ones.).
5. The necessity of **Suggestion 7** should be examined in relation to the success of the approaches from Suggestions 3, 4, and 5. While there may be a need to impute data in an attempt to improve accuracy and make the methods more robust, this is not a technique I would resort to as a first step as it creates artificial data which may not be representative of reality. Had the dataset been too sparse, it would have been a suggestion with higher priority, and even then, depending on the properties of the data, I may suggest polynomial interpolation (captures non-linear relationships) or even an expectation-minimisation imputation (computes maximum probability estimates based on observed data and the assumed data distribution).

Regarding **Suggestion 1**, I recognise the value of this suggestion as blockchain technology can have a high impact on ensuring data integrity, however this is a very complex and very high effort task. Given the abundance of other techniques that are lower effort and potentially high

benefit, I would not instruct my team to undertake Suggestion 1 unless other techniques have been shown to fail.

Machine learning development is an iterative and highly experimental process, therefore regular liaising with the domain experts and continued EDA is incredibly important, especially given the meaningful feedback loop between these two. Finally, appropriate documentation of tasks and experiments is necessary, especially when working across a team of data scientists tasked with different parts of the experimentation process.

Exercise 2: Interpretation

1. At a first glance, I'd be inclined to recognise the two outliers as depicted in the graph and I'd offer praise to my colleague for their initial work in this direction. However I see two potential issues with this approach: (1) using dimensionality reduction, and (2) detecting anomalies on a factory level (as opposed to more granular). On (1), while using a dimensionality reduction approach helps prevent overfitting and allows for computational efficiency, it also often comes with a variety of problems such as: **information loss** (which features were removed, why those, and are there other existing features that compensate for this information loss?), **sensitivity to outliers** (many dimensionality reduction techniques are sensitive to outliers and may distort the reduction process), **assumption in linearity** (there's no reason to assume linear relationships between our variables — common in many standard dimensionality reduction techniques, like PCA). I would also be concerned about potential **difficulties in interpretability**. On (2), it is less meaningful to identify a *factory* being anomalous, or a time-series as a whole being anomalous, as opposed to parts of the time-series or even the last data point being anomalous in comparison to the previous ones (also see Exercise 1 4.b). This approach should be applied at a lower granularity level; and given that data may vary significantly across factories and products, it must be standardised or normalised. Great care should be taken in selecting features to be kept or removed, using appropriate techniques for numerical data (PCA, Autoencoders for complex, non-linear data, etc).

2. Although accuracy of 95.3% may seem favourable, and I applaud my colleague's efforts to employ an ML model on past data, I'd encourage them to keep the objective of the overall task in mind, *especially* as it pertains to outlier/anomaly detection. Seeing as the purpose here is to build a model that can accurately detect anomalies (true negatives (TN)), as mentioned in Exercise 1 point 4.a, we need other metrics that measure the model's performance on detecting these, namely the specificity measure:

$$Specificity = \frac{True\ Negatives\ (TN)}{True\ Negatives\ (TN) + False\ Positives\ (FP)} = \frac{5}{5+43} = 10.42\%$$

which suggests that the model is highly inaccurate for TN detection and is not ready for deployment. While tinkering with the model itself (different choice of algorithm, different choice

of features) may be helpful, it's likely that the main cause of the low specificity is the imbalanced dataset (fewer anomalies than non-anomalies). To balance the dataset, we could **undersample the majority class** (randomly remove non-anomalies to balance the class distribution; this can lead to information loss), **oversample the minority class** (replicate anomalies so they're more representative; may work better for this case), **synthetic minority oversampling techniques** (synthetically generating examples of anomalous observations), etc. However, it may also be helpful to use algorithms that are better able to deal with imbalanced datasets (e.g. balanced bagging or balanced random forest) or perhaps assign class weights to assign higher weights to the anomalous class.

3. I would initially congratulate my colleague on pursuing a worthwhile avenue such as a tree-based approach, however given that that the accuracy on the validation data is less than a random approach (which would yield a 50% accuracy), I would encourage my colleague to explore further - especially on data points in the paler areas where the model has been less confident (perhaps even liaising with the domain expert). I'd advise that it's very likely that the model is suffering from overfitting, which is a known weakness of tree-based models, especially if the tree(s) has/have been allowed to grow too deep, the model is too complex, which is more likely to happen when the data is numerical. Overfitting tree-based models can benefit from some form of regularisation: **pruning** (limiting the depth of the tree to prevent the model from becoming too complex and fitting noise rather than patterns), **splitting** (set a limit to the number of samples required to be at a leaf node, which encourages larger and more generalised nodes), etc. It would also be helpful to improve the feature engineering endeavor and select features that are meaningful, perhaps using expert knowledge or feature importance techniques. Using techniques like k-fold cross-validation can help assess the model's performance across different subsets of the data, which will provide a more reliable estimate of the model's true performance throughout the process. Finally, the choice of algorithm is important, too: ensemble models (like random forest or gradient boosting trees) could improve the situation, but if those also fail, we could also try models outside the tree-based family.