Attachment C – Modelling the impact

This section provides a guide to developing a predictive model for how the AI system's actions will impact the community and identifying any unintended consequences that may require the AI system's actions to be reviewed.

Predicting the outcomes of the AI system's actions

This step is about predicting how the AI systems actions will impact the community.

Agencies need to model how each action (or combination of actions) will impact the performance metrics chosen for this project/AI-enabled system. These models can be a collection of simple hand-made rules, a re-trainable machine learning algorithm, or a sophisticated causal model.

Agencies may need to engage specialist advice on how to design and build appropriate models for their AI system's objectives. Some general considerations are:

- Models must be able to directly observe or reliably infer the numerical values of the performance metrics.
- Models must be able to predict how these performance metrics will change for the different possible operating parameters identified. This will allow the AI system and the agency to plan future actions.

In some cases, agencies may have to conduct experiments, run trials, simulations or use observational studies to ascertain the effect of their system's actions.

Consult with experts to augment the set of performance metrics

A redesign of the AI system's performance metrics will be required to eliminate any unintended consequence identified previously. During this process agencies should consult again with stakeholders to identify any further issues that may not have been picked up in the initial consultation process. The following should also be considered:

- Is more data is needed to improve the AI system's performance?
- Is there missing data? What was the process that produced missing entries in the data? How will the distribution of missing data and an agency's approach to "imputation" affect the performance of the model?
- Ensure that missing data imputation and feature engineering are both inside the model selection pipeline: they are part of the model under consideration, and they can greatly affect prediction. These techniques should be able to be tested and validated in addition to the models themselves.
- Start with a simple model as a baseline and increase complexity only if it improves performance. Simpler models tend to be easier to train, review and debug.

- Consider if adding known costs or benefits to a prediction algorithm’s objective function improve its performance
- Use held-out data (that was never observed during training) for model selection, using techniques such as cross-validation. Depending on the agency’s hyperparameter or model selection procedure, it may be necessary to have an additional validation set or use nested cross-validation
- Model selection should not be based (solely) on standard accuracy measurements, but all the system metrics. The most accurate model overall may not be the one whose properties are most suitable for the problem at hand.
- Use K-fold cross-validation rather than a single “leave out” set unless the team has more data than required or prohibitively long training times. Remember there is a trade-off between accuracy of the validation estimate (higher values of K) and computation time
- Is the AI model is using the features like it would expect to make its predictions? Or has it found some spurious correlation that has no generalisation power? Use tools such as feature importance, partial dependence, local surrogates (LIME) to look for these issues.
- Think about also using a “naive” or “random” prediction strategy as an additional comparison. This will test if the modelling covariates (independent variables) are giving the model any information, or if the model is overfitting
- How sensitive is the system to the introduction of noise into inputs, and is this sensitivity reflected in the predicted uncertainty?
- If it is important for the model to output a level of confidence, or an associated probability, ensure that this uncertainty is itself validated.
- Be wary of covariate shift, which may lead to a drop-in performance once the system is deployed but will not appear in cross-validation results on the training data. It may be necessary to account for it through re-weighting or re-sampling.
- Beware of asking predictive models causal questions, unless the agency has explicitly taken into account all potential confounding variables, and are explicitly applying covariate adjustment techniques. For example, historical hospital admissions data is only relevant to the admissions policy in place when it was recorded, not for training a model that influences admissions.
- Can the causal effect of potential actions be estimated from historical data or is it necessary to run experiments? Do the relationships vary across the population or overtime?
- How long does it take for the consequences of an action to be observed? Could this create problems for measuring the performance of the system once it is deployed?
- Concept shift can arise when the relationships learned from the training data are no longer valid. Real-world phenomena often change over time, and not all such change is gradual: sudden changes in laws or policies may make all historical data useless overnight.