Integration Issue	Source of Problem	Solution
Model Prediction between research and production model don't match	• Different python library versions between research and production environments. • Different python version • No seeds set during research and / or productionising the code • Different data source used in different environments to train the models	• Check python package versions, ensure they are the same • Check python versions, ensure same • Set seed all throughout research and deployment code • Check that the sources are identical
Outcome of feature engineering steps between research and production code do not match	• Different python library versions between research and production environments. • Different python version • No seeds set during research and / or productionising the code • Different data source used in different environments to train the models	• Check python package versions, ensure they are the same • Check python versions, ensure same • Set seed all throughout research and deployment code • Check that the sources are identical
Different data obtained in research and production environment	• Environments are reading from different sources	• Check that the sources are identical
Different train and test split in research and production environment	• No random_state or seed set while splitting dataset • Data rows uploaded in different order if using SQL	• Set seed all throughout research and deployment code • Order by date or foreign key to avoid random upload of rows
Model doesn't score	• New data contains null values, not present in training data • New data contains categories not present in categorical variables of training set	• Check with domain expert if variable can take null values. If it can, you need to include a feature engineering step during training. If it can't write an error handler not to score observations with null values and send an alert • Include a feature engineering step during training to account for unseen labels. Alternatively, write an error handler not to score observations with unseen values and send an alert.

	• New data contains unusual compared to training data	• Check with domain expert if variable can take such values. On occasions they are placeholders for code. Include a feature engineering step during training to account for unseen labels. Alternatively, write an error handler not to score observations with unseen values and send an alert.
Model prediction matches for most observations but not for a few	• Potential randomness when calculating the prediction	• Check if you are using random sampling to fill null values within variables and control the seed for sampling
Some variables my model need are not available in the live systems	• Common issue found when deploying models for the first time	• Try and replace that variable for the most similar one. Or remove the variable and re-train and re-evaluate the model both in the research and production environments. Ideally, you want to spend a significant amount of time making sure data will be available at the point the live system calls the model, ahead of modelling
The scores in the research and production environment match, but the performance of the model is below the expected	• Another common issue	• Check that the distribution of the variables is the same for the training data and the data you are scoring live. Often, there were filters applied when gathering the data that are not applied during live scoring and vice-versa. If this is the case, performance might be lower than seen during training. Ideally, when deciding which data you will use for training, you spend a significant amount of time, understanding the population that will be entering into your model • Corroborate that the target you used to train your model is a true representation of the real outcome. Often, we use targets with information that is not reliably updated / saved by the company. If target is not reliable, model performance might be lower than anticipated.

		• Check that the model implementation satisfies your requirements. Are there any exceptions made to the population of data that is sent to your API?