Problem Framing

1: Identify your problem space
Broadly speaking, what is your goal?
(e.g. Build a system that routes contacts better)

_____
_____
_____

2: Identify the success metric for your business problem
What defines success in this problem space? What is the metric you are most interested in?
(e.g. Success for Intelligent Routing (IR) was defined by reducing transfers. The metric of interest is the transfer rate)

_____
_____
_____

3: Conduct opportunity analysis
Is there currently anything in production performing this operation? How is it performing? How many errors is it making? What kind of errors is it making?
(e.g. We had rules in place to determine transfers, it was making a lot of errors, big opportunity to improve)

_____
_____
_____

4: Understanding the problem space and identifying key people.
Who are the people involved in this problem space? Who are the domain experts? Who owns the data?
(e.g. Additional people involved in Intelligent Routing were the CS agents, their managers and capacity planners, SDEs that own the routing system, and data engineers)

_____
_____
_____

5: Framing your problem as an ML problem

5a: Are you trying to predict a discrete value *(e.g. Kindle vs Non-Kindle)* or a continuous value *(e.g. a price)*? That is, do you want to do classification or regression?

_____

5b: If discrete, how many possible outcomes are there? *(e.g. [Kindle, Non-kindle] = 2, [Kindle, AWS, Returns, Where's My Stuff] = 4)*

_____

5c: Do you care about the top 1 or top X?

_____

5d: Is it possible to have multiple outcomes for the same data point? (e.g. someone could be calling about a kindle problem AND returning a package)

_____

5e: What is the relationship between these predictions and the problem space? How can we use these prediction to make a data-driven business decision?
(e.g. We can try routing customers based on the most likely issues predicted by the model)

_____
_____
_____

6: Given all of the above, can you convert your general problem statement to an ML problem statement?

(e.g. Can we reduce transfers by building a model that predicts the most likely 3 customer issue for any given customer and presenting predictions as personalized options in the phone IVR?)

_____
_____
_____

Data

1: How do you get the data? Fetch a small sample to test the data fetching method.

_____
_____
_____

2: What format is the data? The most common format is a .csv or a JSON.

_____
_____
_____

3: Time to explore the data. Does your data already have labels? This determines whether you can use supervised machine learning techniques.

_____
_____
_____

4: If there is no obvious label, what could you use as a label? (If you do not have any labels, you will need to explore semi-supervised or unsupervised techniques, detailed in future courses)

_____
_____
_____

4a: If so, how many labels are there? Which labels are common and which labels are rare? Are there any labels you might want to exclude from the model for a business reason?

_____
_____
_____

5: What other data sources might existing for your problem space? Can you augment your current data in any way to make it bigger and better?

_____
_____
_____

6: After performing a manual analysis of a random sample (e.g. reading 50 phone call records), does it seem like there is generally enough information for a human to guess the label?

_____
_____
_____

7: What kinds of data points (features) are there? Can you imagine any other kinds of data points that might be useful for this classification task?

_____
_____
_____

8: Is there anything odd-seeming about your data? Missing data points? Anything that does not match your expectations? (e.g. Why don't I see any phone calls related to XXXX?)

_____
_____
_____

9: Visualize the distribution of your labels. Are any labels very rare? Are any labels more important from a business perspective? Do you want to exclude any labels from your model? (e.g. We always want certain phone calls to go to a certain skill group)

_____
_____
_____

10: Given the quantity of data, what is the best experimental set up (how do you want to split your data across training, development and test sets?) If you do not have a lot of data (hundreds or low thousands of observations), you may want to try 80%-10%-10%, otherwise, try 70%-15%-15%.

_____
_____
_____

Modeling

Evaluation

In order to calculate precision and recall, you need to decide which class if your "positive class". This is somewhat arbitrary, but you can think of it as which label is your model designed to detect. (e.g. Our positive was Kindle and our negative was "Not-Kindle")

1: What is your positive class?

_____
_____
_____

2: Is accuracy a meaningful metric for your problem? If you have an unbalanced problem, where the positive class is much rarer than the negative class, accuracy may not be meaningful. Check your answer to question 4a from the data section.

_____
_____
_____

3: Fill our your 2x2 error matrix. How many false positive and false negatives did your model produce?

4: What is the precision for this model? What is the recall for this model? How does this compare to your baseline system's performance?

_____
_____
_____

5: Can you describe in words what is a false positive for your business problem? (e.g. A false positive for IR was when we predicted a contact was Kindle but the True label was Not-Kindle)

_____
_____
_____

6: For your business problem, do you want to prioritize precision or recall? Are false positives more expensive than false negatives? Do you want to prioritize them equally?

_____
_____
_____

Feature Engineering

The best place to start for feature engineering is error analysis. Look at a sample of 20 of the errors that are the most expensive to your business problem (false positive or false negatives). For each error, examine the data point and look at the prediction.

1: Can you take a guess at what features may have led the model to make this mistake? Do you see any patterns in why these mistakes are happening?

_____
_____
_____

2: Pick the most convincing pattern you identified in the previous step. In consultation with your domain experts and your own intuition, can you imagine some kind of information that would help the model learn to correct this kind of mistake?

_____
_____
_____

3: Given that you would like to make this kind of information available to your model during training, how can you actually construct a feature that represents this kind of information?

_____
_____
_____

4: Most features fall into one of three categories: numerical (a real number), categorical (discrete value) and binary (categorical with only two options). Which of these feature types makes the most sense for the feature you want to build?

_____
_____
_____

5: How will you build this feature? Is this a feature that you can create from your existing data? It may be the case that this information is not present in your original data set.

_____
_____
_____

You can repeat this entire process for other error patterns identified during error analysis, and you can also repeat the process for the other type of error (FP/FN)

Application

Before you launch your model, you will want to revisit your business metric. Talk to your domain experts and work to decide how to relate model performance to your business metric. This part is a bit tricky, but try looking at individual cases.

1: In shadow mode, if you act on the model prediction instead of the action taken by the baseline, would you have the same outcome? What might change as a result of acting on the model's suggestion?

_____
_____
_____
_____
_____
_____

2: This is a good time to think about what will happen when you model makes a mistake. What kinds of errors are you seeing in your offline experiments or in shadow mode? How do these errors change the customer experience?

_____
_____
_____

3: If you model breaks and the system goes offline, what back up will you have to ensure that the customer experience does not degrade too much?

_____
_____
_____

4: **In your estimation, is the model likely to improve performance in terms of your business metric?**

_____
_____
_____

If you answered yes to the above question, you are ready to perform A/B testing. Start by randomly applying your treatment to a small group of customers, maybe 1% of the volume.

Take this opportunity to ensure that the data from your launch is being logged properly. This is also another opportunity to do error analysis.

5: Are there errors that from the 1% treatment group similar to what you saw in offline experimentation?

_____
_____
_____

6: Based on what you see in 1% treatment, do you still think this model will meet your business needs?

_____
_____
_____

If so, you can ramp up to 5%, repeat the process with the eventual goal of running a 50% A/B test. The amount of time needed to run your experiment will depends on your use case, but you will want to make sure that you run for a long enough time period to a) gather sufficient data for significance testing and b) gather data from a wide enough time frame to be representative. As a suggestion, two full weeks of A/B testing is what we used for the Intelligent Routing system.