Chapter

1

Framing ML Problems

Artificial intelligence and machine learning have dramatically changed our lives, and we are still in the early stages of adoption. Google adopted machine learning two decades ago and continues to inspire us all with its innovations. Google's innovations are part of its brand, and millions of customers eagerly look forward to the annual conference Google I/O, which is described as "innovation in the open" for new path-breaking creations. Many innovations, although considered disruptive initially, are adopted rapidly, sometimes by billions of users; they can even achieve common household usage in just a few years.

For example, a universal translator is a device that can translate text or voice from one language to another in near real time; this was a common theme in old sci-fi movies that is now a reality today. Another use case is the ability to search for objects in images by showing an image instead of using a keyword, called "search by image." You might have used the "suggested replies" to emails, which saves you the time and effort to type out an email reply on Android phones. While that use case might save a few minutes of time, other innovations like the AlphaFold AI system saves years, accelerating research by predicting a protein's 3D structure from its amino acid sequence instead of using means based on a physical lab.

Businesses across the globe have started embracing AI and machine learning. These businesses are not "rare innovators" but cut across all domains, in a range of industries including agriculture, healthcare, transportation, retail, and manufacturing. One can safely say that any field can benefit from machine learning by identifying the right use cases.

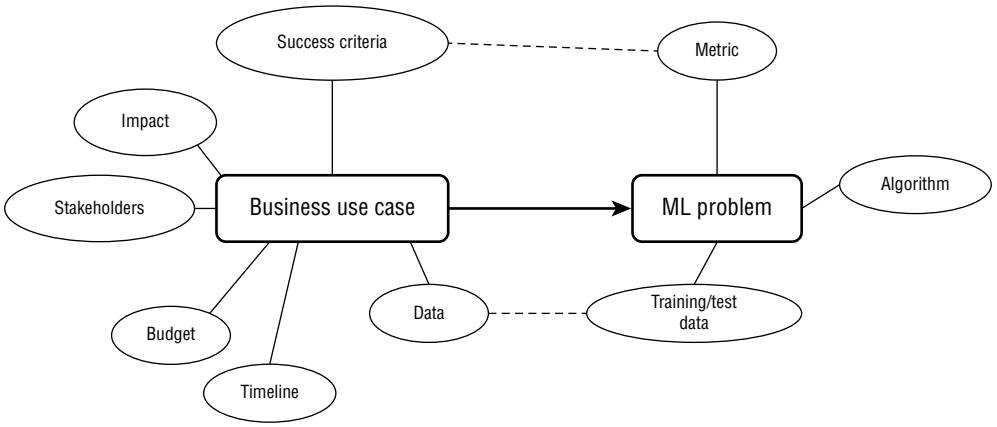
You see use cases everywhere around you, and the motivations for innovation might be varied. Some industries might be trying to solve critical problems like trying to understand the protein structure of the coronavirus; other industries might be looking for ways to increase efficiency. Sometimes machine learning can solve a problem that was previously impossible to solve, while in other cases it might be providing an incremental improvement to an existing situation. Sometimes these innovations might be competitive, where one business is able to grow rapidly due to innovation that puts pressure on all others, while in other cases all businesses might be trying to come together to solve a single problem. Having a historical understanding of an industry helps you appreciate the previous efforts in solving problems and exploring new approaches.

Innovations all start out as simple use cases that are translated into machine problems, which can then be tackled by machine learning engineers. The ability to identify these use cases and assess the impact of machine learning is the first step in your journey toward certification.

Translating Business Use Cases

The goal of this chapter is to help you to first identify the impact, success criteria, and data available for a use case. Then, match this with a machine learning approach (an algorithm and a metric) as shown in Figure 1.1. We will look at how to fit the ML project into the budget and timeline.

FIGURE 1.1 Business case to ML problem



Now, imagine you are being tasked with using ML to solve a problem. You first need to identify the use case and fit it to a machine learning problem.

For example, say you are trying to predict house prices, you can use a regression model. The performance requirements of the model will be determined by the business.

Say, you have had a discussion with the key people and understood the use case; you now need to identify the key *stakeholders*, the people related to the use case. The stakeholders can be executives, the CFO, data engineers, tech support staff who may have to approve the project to proceed. Each of these stakeholders might have very different expectations of this ML project, and your ability to communicate the value could make the difference between approval or rejection. Executives are looking for impact to business, CFOs are typically interested in the budget of the solution, managers might be keen on timelines, and data managers might be interested in data privacy and security. If you are able to understand these five aspects, your pathway to approvals will be smooth.

The stakeholders will help you measure the *impact* of this use case for your company and the end user. The impact could be increasing profit, reducing fraud, improving the quality of life, or even saving lives. The impact is probably the most important element of the use case.

For example, say your company has a learning management system (LMS), a platform where students subscribe to courses. You have data of students' activities and using this you want to improve the experience using machine learning. You could do several things:

- Create a recommendation engine to show new courses for students.
- Churn prediction to see if a student is going to quit the course.
- Churn prediction to see if a teacher is not going to come back.
- Identify what makes a course interesting for students (sample questions, more images, more tables, short videos, etc.).
- Identify what kind of learning a student prefers (auditory, visual, or kinesthetic).

Which of these would be most impactful is a question that can be answered only by the business owner.

Once you have identified the use case with the highest impact on your business, you need to identify the outcome of your machine learning solution. In short, what would happen if you implemented your solution? Sometimes, your model would make accurate predictions, but the environment might react in a counterproductive way to these predictions. This is because the environment is seldom static; the users could adapt or the users could get confused with the behavior of the predictions.

For example, say your company has a video sharing website, and you have millions of videos. You are trying to build an ML model to recommend videos to your users. You could choose from among the following:

- An ML model to recommend unseen videos from popular video creators. The problem is that this is not personalized. What if the user does not like some of the creators?
- An ML model to recommend videos that get a lot of clicks. But what if these are just clickbait, where people click and regret wasting time?
- An ML model to recommend videos that have been watched fully by similar users. This would lead to improving the user experience.

In this example, you need to have a good understanding of the use case, the overall goal, and the end user to be able to find the right fit.

Next, find out if the problem is even solvable using machine learning. Business leaders hear inspiring stories in the media about how a business solves a problem with ML and it sounds magical and the business leaders would love to use it to solve their business problems. They need an expert like you to figure out if it is even feasible to solve their problem using ML. This is not as easy as it sounds; it depends on several things, like existing technology, available data, and budget. For example, natural language processing has advanced leaps and bounds and has made it possible to do things that were impossible just a few years ago, such as using ML to answer a question from a piece of text. Familiarity with the latest advancements in natural language processing would help you identify easier, faster, and better ML methods to solve your business problems.

As the next step, you will need to identify an ML learning approach that fits your use case.

Machine Learning Approaches

Many machine learning problems have been well researched and have elegant solutions, but some algorithms are not perfect and some problems can be solved in multiple ways. Sometimes, a use case will fit perfectly with an ML framework and other times not so well. You need to be aware of the landscape of ML problems. There are several approaches to machine learning methods. Some of these approaches have been studied for decades, and others are fairly new. There are hundreds if not thousands of ways to apply machine learning techniques. To help us get a grasp of the breadth of these methods, we organize them into categories (also called methods, or types or approaches or problems). Each of these approaches solves a specific class of problems, distinguished by the type of data, the type of prediction, and so on.

On the exam, you will be given the details of a use case and will be expected to understand the nature of the problem and find the appropriate machine learning approach to solve it. To accomplish that, you need to have wide knowledge of the landscape of these machine learning approaches.

We will look at the different ways to classify the approaches in the following sections.

Supervised, Unsupervised, and Semi-supervised Learning

A common method of classifying machine learning approaches is based on the type of learning. When you have a labeled dataset that you can use to train your model, it is called *supervised learning*. For example, supervised learning would be trying to build a model to classify images of dogs or cats and having the ability to use a dataset of images that have been labeled accordingly.

There are some cases where you have only unlabeled data, such as a set of images (without any labels or tags), and you will be asked to classify or group them. This would be an *unsupervised* ML model. Clustering algorithms are a suite of algorithms that belong to this type and are used to group and/or classify data. Autoencoders are also a family of algorithms that belong to this type. Autoencoders are used to reduce the dimensionality of input data, a preprocessing step in many machine learning models.

Another popular unsupervised ML use case is *topic modeling*, a type of document clustering problem. The algorithm takes documents and classifies them into N number of classes based on the commonality of words and sentences in the texts. Comparing this to how a human being would classify books, say, in a library, you may classify them into fiction, nonfiction, science, history, and so on. In other times, you may classify the books based on languages (for example, English, Chinese, Hindi). Similarly, an unsupervised algorithm may or may not classify in the way you expected. The output of unsupervised learning methods

cannot be fully controlled, and it is almost never perfect and so requires careful tuning to get required results. Table 1.1 provides the details of some of the popular ML model types that are readily available in Google Cloud.

TABLE 1.1 ML problem types

Name	Data Type	Supervised/Unsupervised
Regression – Tables	Tabular	Supervised
Classification – Tables	Tabular	Supervised
Forecasting	Series	Supervised
Image classification	Image	Supervised
Image segmentation	Image	Supervised
Object detection	Image	Supervised
Video classification	Video	Supervised
Video object tracking	Video	Supervised
Video action recognition	Video	Supervised
Sentiment analysis	Text	Supervised
Entity extraction	Text	Supervised
Translation	Text	Supervised
K-means clustering	Tabular	Unsupervised
Principal component analysis	Tabular	Unsupervised
Topic modeling	Text	Unsupervised
Collaborative filtering/ recommendations	Mixed	Supervised/Unsupervised

Source: Adapted from Google cloud/ `https://cloud.google.com/vertex-ai/docs/training-overview` last accessed December 16, 2022.

To solve the problem of uncertainty in unsupervised learning, there is a hybrid solution called *semi-supervised learning*, where some data is labeled and other data is not. This is like guiding the algorithm toward the clusters that you want to see. While semi-supervised models are interesting research topics and have some utility, in a majority of use cases, supervised models are used.

There are many other kinds of machine learning models beyond these, including reinforcement learning (where the algorithm is not given data but is given an environment that the agent explores and learns) and active learning algorithm, but they are beyond the scope of the certificate exam.

Another way to classify the machine learning algorithms is based on the type of prediction. The type of data the model will predict determines several aspects of the machine learning algorithm and the method used. We will explore that next.

Classification, Regression, Forecasting, and Clustering

Classification is the process of predicting the "labels" or "classes" or "categories." Given a picture of a pet, classifying dogs versus cats is a classification problem. If there are just two labels, it is called *binary classification*, and if there are more labels, it is called *multiclass classification*. You could have a classification with thousands of labels; for example, the Cloud Vision API can classify millions of different objects in a picture, which is a more difficult problem to solve. You cannot apply the same model for binary classification, multiclass classification, and classification with thousands of classes.

In *regression*, the ML model predicts a number—for example, prediction of house price (given the number of bedrooms, square footage, zip code), prediction of the amount of rainfall (given temperature, humidity, location). Here the predicted value's range depends on the use case. The ML algorithms used for regression are usually different from classification. Typically, you would find structured data (data in rows and columns), as shown in Table 1.2, being used for regression problems.

TABLE 1.2 Structured data

Student ID	Age	Exam Scores (Out of 100)
1	34	75
2	23	59
3	36	92
4	31	67

Forecasting is another type where the input is time-series data and the model predicts the future values. In a time-series dataset (Table 1.3), you get a series of input values that are indexed in time order. For example, you have a series of temperature measurements taken every hour for 10 hours from a sensor. In this case, one temperature reading is related to the previous and next reading because they are from the same sensor, in subsequent hours, and usually only vary to a small extent by the hour, so they are not considered to be "independent" (an important distinction from other types of structured data).

TABLE 1.3 Time-Series Data

	Temperature
Series 1	29, 30, 40, 39, 23, 20
Series 2	10, 11, 13, 23, 43, 34
Series 2	19, 18, 19, 20, 38, 20
Series 4	14, 17, 34, 34, 12, 43

Some forecasting problems can be converted to regression problems by modifying the time-series data into independent and identically distributed (IID) values. This is done either for convenience or availability of data or for preference for a certain type of ML model. In other cases, regression problems can be converted into classification problems by bucketizing the values. We will look into details in the following chapters. There is an art to fitting an ML model to a use case.

Clustering is another type of problem, where the algorithm creates groups in the data based on inherent similarities and differences among the different data points. For example, if we are given the latitude and longitude of every house on Earth, the algorithm might group each of these data points into clusters of cities based on the distances between groups of houses. K-means is a popular algorithm in this type.

ML Success Metrics

A business problem can be solved using many different machine learning algorithms, so which one to choose? An *ML metric* (or a suite of metrics) is used to determine if the trained model is accurate enough. After you train the model (supervised learning), you will predict the values (y) for, say, N data points for which you know the actual value (y). We will use a formula to calculate the metric from these N predictions.

There are several metrics with different properties. If so, what is our metric? What is the formula for calculating the metric? Does the metric align with the business success criteria? To answer these questions, let us look at each class of problems, starting with classification.

Say you are trying to detect a rare fatal disease from an X-ray. This is a binary classification problem with two possible outcomes: positive/negative. You are given a set of a million labeled X-ray images with only 1 percent of the cases with the disease, a positive data point. In this case, a wrong negative (false negative), where we predict that the patient does not have the disease when they actually do have it, might cause the patient to not take timely action and cause harm due to inaction. But a wrong positive prediction (false positive), where we predict that the patient has the disease when in fact they do not, might cause undue concern for the patient. This will result in further medical tests to confirm the prediction. In this case, accuracy (the percentage of correct prediction) is not the correct metric.

Let us now consider an example with prediction numbers for a binary classification for an unbalanced dataset, shown in Table 1.4.

TABLE 1.4 Confusion matrix for a binary classification example

		Predicted	
		Positive Prediction	**Negative Prediction**
Actual	**Positive Class**	5	2
	Negative Class	3	990

There are two possible prediction classes, positive and negative. Usually the smaller class (almost always the more important class) is represented as the positive class. In Table 1.4, we have a total of 1,000 data points and have predictions for each. We have tabulated the predictions against the actual values. Out of 1,000 data points, there are 7 belonging to the positive class and 993 belonging to the negative class. The model has predicted 8 to be in the positive class and 992 in the negative class. The bottom right represents true negatives (990 correctly predicted negatives) and the top left represents true positives (5 correctly predicted positives). The bottom left represents false positives (3 incorrectly predicted as positive) and the top right represents false negatives (2 incorrectly predicted as negative). Now, using the numbers in this confusion matrix, we can calculate various metrics based on our needs.

If this model is to detect cancer, we do not want to miss detecting the disease; in other words, we want a low false negative rate. In this case, *recall* is a good metric.

$$\text{Recall} = \frac{True\ Positive}{True\ Positive + False\ Negative}$$

In our case, recall = 5/(5 + 2) = 0.714. If false positives are higher, the recall metric will be lower because false negative is in the denominator. Recall can range from 0 to 1, and a higher score is better. Intuitively, recall is the measure of what percentage of the positive data points the model was able to predict correctly.

On the other hand, if this is a different use case and you are trying to reduce false positives, then you can use the precision metric.

$$\text{Precision} = \frac{True\ Positive}{True\ Positive + False\ Positive}$$

In our case, we have 3 false positives, so our precision score is 5/(5 + 3) = 0.625. Intuitively, precision quantifies the percentage of positive predictions that were actually correct.

Sometimes, your use case might be interested in reducing both false positives and false negatives simultaneously. In that case, we use a harmonic mean of both precision and recall, and it is called the F1 score. (There is a more general F_β score depending on how you wish to weight precision and recall and F1 is just one case.)

$$\text{F1} = 2 \times \frac{Precision * Recall}{Precision + Recall}$$

In our example, we get 2 x (0.625 x 0.714)/(0.625 + 0.714) = 0.666. Here again, F1 ranges from 0 to 1, and a higher score indicates a higher-quality model. The three metrics are summarized in Table 1.5.

TABLE 1.5 Summary of metrics

	Scenario	Formula
Precision	Lower false positive	$\text{Precision} = \frac{True\ Positive}{True\ Positive + False\ Positive}$
Recall	Lower false negative	$\text{Recall} = \frac{True\ Positive}{True\ Positive + False\ Negative}$
F1	Lower false positive and false negative together	$\text{F1} = 2 \times \frac{Precision * Recall}{Precision + Recall}$

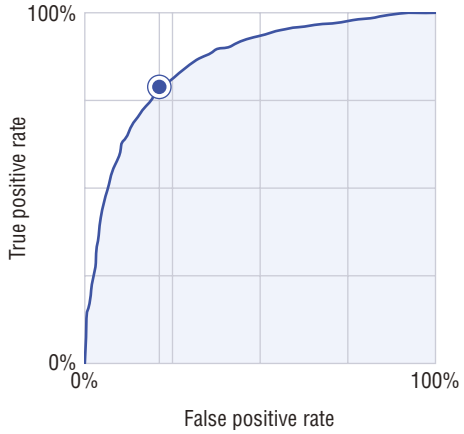
Area Under the Curve Receiver Operating Characteristic (AUC ROC)

ROC stands for receiver operating characteristic curve (it comes from the field of signal processing) and is a graphical plot that summarizes the performance of a binary classification model (Figure 1.2). The x-axis is the false positive rate, and the y-axis is the true positive rate, and the plot is generated at different classification thresholds. The ideal point for this plot is the top-left corner, which has 100 percent true positive and 0 percent false positive, but in practice you will never see this. You can also calculate the precision, recall, and F1 at each point on the curve. When you visually inspect the curve, a diagonal line is the worst case, and we want the curve to stretch as far from the diagonal as possible.

When you have two models, you get two ROC curves, and the way to compare them is to calculate the area under the curve (AUC).

Once you have chosen the model based on AUC, you can find the threshold point that maximizes your F1 (as indicated in Figure 1.2).

FIGURE 1.2 AUC



This method has the following advantages:

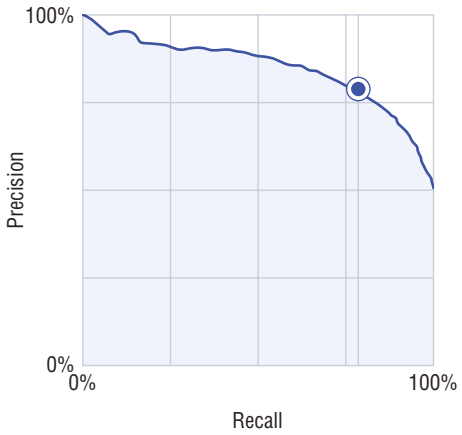
- **Scale-invariant:** It measures how well the predictions are ranked and not their absolute values.
- **Classification threshold-invariant:** It helps you measure the model irrespective of what threshold is chosen.

Classification threshold invariance is not always desirable because sometimes there are huge disparities between false positives and false negatives. Therefore, AUC is not usually the best metric for picking a model when there is class imbalance.

The Area Under the Precision-Recall (AUC PR) Curve

The area under the precision-recall curve is a graphical plot that illustrates the relationship between a precision-recall pair (Figure 1.3). The x-axis is the recall and the y-axis is the precision. The best AUC PR curve is a horizontal line across the top. In this curve, the optimal point is the top-right corner, which has 100 percent precision and 100 percent recall, which is never seen in practice but always aimed at.

FIGURE 1.3 AUC PR



If the dataset is highly imbalanced, the AUC PR is preferred because a high number of true negatives can cause the AUC curve to be skewed.

Regression

Regression predicts a numerical value. The metric should try to show the quantitative difference between the actual value and the predicted value.

MAE The mean absolute error (MAE) is the average absolute difference between the actual values and the predicted values.

RMSE The root-mean-squared error (RMSE) is the square root of the average squared difference between the target and predicted values. If you are worried that your model might incorrectly predict a very large value and want to penalize the model, you can use this. Ranges from 0 to infinity.

RMSLE The root-mean-squared logarithmic error (RMSLE) metric is similar to RMSE, except that it uses the natural logarithm of the predicted and actual values +1. This is an asymmetric metric, which penalizes under prediction (value predicted is lower than actual) rather than over prediction.

MAPE Mean absolute percentage error (MAPE) is the average absolute percentage difference between the labels and the predicted values. You would choose MAPE when you care about proportional difference between actual and predicted value.

R^2 R-squared (R^2) is the square of the Pearson correlation coefficient (r) between the labels and predicted values. This metric ranges from zero to one; and generally a higher value indicates a better fit for the model.

Responsible AI Practices

AI and machine learning are powerful new tools, and with power comes responsibility. You should consider fairness, interpretability, privacy, and security in your ML solution. You can borrow from best practices in software engineering in tandem with considerations unique to machine learning.

General Best Practices Always have the end user in mind as well as their user experience. How does your solution change someone's life? Solicit feedback early in the design process. Engage and test with a diverse set of users you would expect to use your solution. This will build a rich variety of perspectives and will allow you to adjust early in the design phase.

Fairness Fairness is very important because machine learning models can reflect and reinforce unfair biases. Fairness is also difficult in practice because there are several definitions of fairness from different perspectives (academic, legal, cultural, etc.). Also, it is not possible to apply the same "fairness" to all situations as it is very contextual. To start with, you can use statistical methods to measure bias in datasets and to test ML models for bias in the evaluation phase.

Interpretability Some popular state-of-the-art machine learning models like neural networks are too complex for human beings to comprehend, so they are treated as black boxes. The lack of visibility creates doubt and could have hidden biases. *Interpretability* is the science of gaining insights into models and predictions. Some models are inherently more interpretable (like linear regression, decision trees) and others are less interpretable (deep learning models). One way to improve interpretability is to use *model explanations*. Model explanations quantify the contributions of each input feature toward making a prediction. However, not all algorithms support model explanations. In some domains, model explanations are mandated, so your choice of algorithms is restricted.

Privacy The only connection between the training data and prediction is the ML model. While the model only provides predictions from input values, there are some cases where it can reveal some details about the training data. This becomes a serious issue if you trained with sensitive data like medical history, for example. Although the science of detecting and preventing data leakage is still an area of active research, fortunately there are now techniques to minimize leakage in a precise and principled fashion.

Security The threat of cybersecurity is very much applicable to machine learning. In addition to the usual threats to any digital application, there are some unique security challenges to machine learning applications. These threats are ever present, from the data collection phase (poison data), training phase (leakage of training data), and deployment phase (stealing of models). It is important to identify potential threats to the system, keep learning to stay ahead of the curve, and develop approaches to combat these threats.

You can read more at `https://ai.google/responsibilities`.

Summary

In this chapter, you learned how to take a business use case and understand the different dimensions to an ask and to frame a machine learning problem statement as a first step.

Exam Essentials

Translate business challenges to machine learning. Understand the business use case that wants to solve a problem using machine learning. Understand the type of problem, the data availability, expected outcomes, stakeholders, budget, and timelines.

Understand the problem types. Understand regression, classification, and forecasting. Be able to tell the difference in data types and popular algorithms for each problem type.

Know how to use ML metrics. Understand what a metric is, and match the metric with the use case. Know the different metrics for each problem type, like precision, recall, F1, AUC ROC, RMSE, and MAPE.

Understand Google's Responsible AI principles. Understand the recommended practices for AI in the context of fairness, interpretability, privacy, and security.