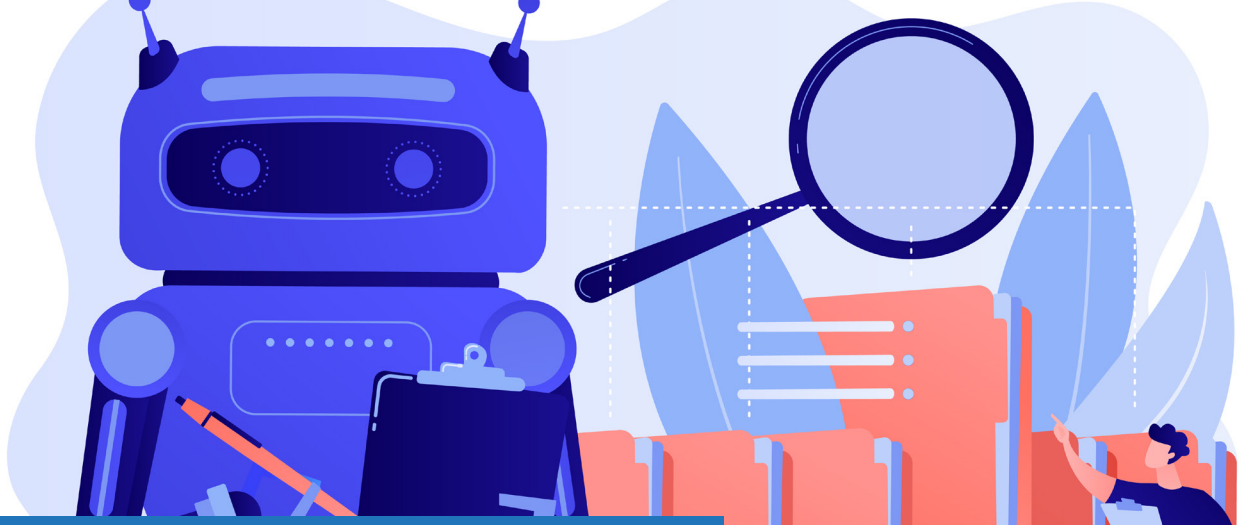

1. Building A ML Model

Learning Objectives

By the end of this chapter, readers will be able to

1. Frame Business problem to ML problem
2. Understand the ML model building Pipeline
3. Describe each phase of CRISP-DM in his/her own words
4. Learn to build, evaluate ML models, and select the best one.

Prerequisites

Readers should have knowledge of the following concept before starting this read.

1. Basic concepts on AI.
2. Basic Mathematics
3. Basic Python Programming.

1. Framing ML Problem

Machine learning problem framing starts with business problems. Every business wants AI to automate their workflow, save cost, expand profit margin, and eventually increase customer satisfaction. Most of the time, we get into this mode of thinking, should I build an AI system to do x, y, and z. But, without really thinking, this is going to save me more money? Or is it going to increase the revenue for the business and so forth? Does it increase sales or saves the cost or ROI that you were trying, tracking, and being very thoughtful about what the potential impact is going to be crucial.

All problems don't need machine learning. Some are traditionally better. Most of the business today using AI has made a significant impact on business growth. So formulating the Business problems into the ML solution is crucial.

Let's look at a case where the company's marketing department wants to collect customer feedback, whether it is a new product, campaign, or a new logo. Just put it to a concept test and analyze the sentiments attached to it. Predicting customer sentiment becomes the problem to solve, and using ML is one approach that can be used to solve it.

2. The End-to-End ML Model Building Process

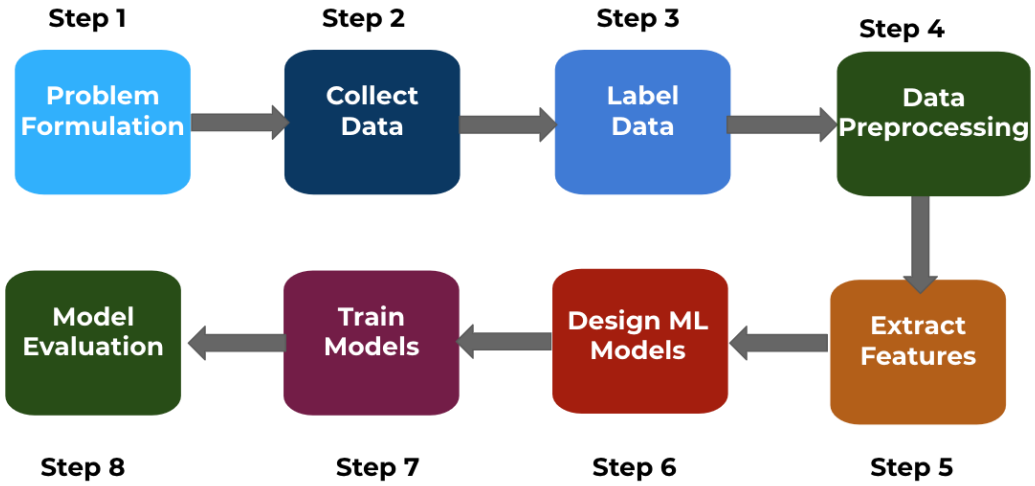

Figure 1: End-to-End ML model building pipeline.

Here is an eight-step recipe or a process to build a machine learning model. And what we've done here is distilled the steps from the perspective of building a machine learning model for a given business problem into eight different steps. But there might be other professors or scientists who might divide into nine steps, you know, some divide only five steps, and so forth. So, what are the eight steps?

1. The first step is for your given problem, for your given business problem, what are you trying to achieve? Is ML an excellent approach to solve the problem or is the traditional approach better? Will ML automate workflow, save cost, expand profit margin, and eventually increase customer satisfaction.
2. The second step is collecting data. Once you make sure that building an ML model is themost efficient way to solve your problem you will need data. So, one of the things you need to make sure is before you even start building the model, you have to make sure that you have the right kind, the right amount of data.
3. The third step is the labeling of the data. It provides the target values that are going to be used by the machine learning model.

4. Step number four is Data preprocessing, which involves several steps like data transfor mation, data cleaning, normalization, dimensionality reduction, etc.
5. Step five, which is extracting features. Many machine learning algorithms require a feature representation that extracts the usable information of data in numeric form and helps the model work as it is designed for.
6. Step number six is figuring out what kind of machine learning algorithm we can use or train to build a system to solve our business problem.
7. Step number seven is all about training the model, squeezing the data with some useful algorithm to find the data pattern.
8. Step eight is figuring out whether the prediction of the model is close to the actual output.

The right setup is more important than the right choice of algorithm. Now, the natural query is, do we have some standard framework for model building that blends all these discussed steps into a single framework? Yes!! CRISP-DM is the one.

3. CRoss Industry Standard Process for Data Mining (CRISP-DM)

Figure 2: Process diagram showing the relationship between the different phases of CRISP- DM

The Cross-Industry Standard Process for Data Mining (CRISP-DM) is an open standard process model with six phases that naturally describes the data science life cycle. It's like a set of guard-rails to help you plan, organize, and implement your data science project.
Some of the steps involved in CRISP-DM are below:

A. Business understanding:

The first step is business understanding, where we decide what we want to accomplish from a business perspective and define problems and objectives. Basically, we assess the situation, available resources, requirements and formulate success criteria. Business Understanding involves several steps:

- Requirement gathering is the initial step where we basically perform the following task:
  - Don't think around the technical solution.
  - Prioritize the most imperative features.
  - Requirements should come from Stakeholders before Developers.
  - Interviews, workshops, conceptualizing, and written communications are all compelling ways of gathering data from key stakeholders.
- Problem formulation is to identify the key business problem, start simple and straight forward. What would you like your ML model to do? Maybe you want the ML model to predict how many likes and subscribers a video just uploaded will gain in the future.
- Define success and failure metrics. Are those metrics measurable? A measurable metric helps to track the success of your project. For example, If you are building a sentiment analysis classifier, Accuracy-recall helps evaluate the classifier you built.
- Feasibility Analysis identifies the success of the project. It may answer some key questions like:
  1. What type of data do we need? Is it text, image, video, speech?
  2. How tough is data acquisition?
  3. How costly is data labeling?
  4. The volume of data required?
  5. How risky is the cost of false predictions?
  6. How frequently does the system need to be right to be useful?
  7. Is there sufficient literature on the problem?

B. Data Understanding:

Data collection is a crucial step because machine learning requires data; that's just how it works. We are building a statistical algorithm on lots and lots of data, figuring out patterns on it. So, we need a lot of data, Especially, new models like neural networks, deep learning systems require a ton of data. Before we even start building the model, one of the things we need to make sure of is if we have the right kind, the right amount of data, and so forth. For example, if we want to build sentiment classifiers, we need to collect a huge number of emails, tweets,

reviews, or text pieces from customers.

After the collection of data, we need to label them. This is particularly relevant only for supervised learning systems. In supervised learning, you need human label data. So, let's look at an example of sentiment detection. In sentiment detection, if you are building a machine learning system to predict sentiments, how happy or unhappy somebody is on a piece of text. Labeling here means taking such emails or pieces of text from customers, and having a group of humans go through them, and then label by reading them. What it does is, it provides the target values that are going to be used by the machine learning model, to build a model, to be able to classify sentiments by feeding pieces of text.

In order to understand the data, we have to explore them. Data exploration is the starting step in data analysis, where we explore huge information set in an unstructured way to reveal initial patterns, characteristics, and points of interest. This process isn't implied to reveal every bit of details a dataset holds, but rather assist in creating a wide picture of critical trends and major points to study in more prominent detail.

Box plots, histograms, scatter plots, heat maps, etc. are different data exploration tools used in machine learning.

C. Data Preparation:

In any Machine Learning process, Data Preprocessing is the step that transforms or encodes the data to such a state that the Algorithm can now easily interpret it. In other words, the data features can now be easily interpreted by an algorithm. In this step, common tasks are:

- Deleting or updating records with incorrect or invalid values from raw data and deleting records lacking a significant number of columns.
- Transforming data so that it can be feed in an ML model is called data transformation. It also contains categorical feature encoding. Data normalization involves converting all data variables into a given range. Discretization is a process of transforming continuous data into sets of small intervals/discrete data.
- Reducing the number of features by creating lower-dimensional, more efficient data representations using PCA techniques, embedding extraction is called dimensionality reduction. Using such created lower-dimensional data in machine learning algorithms is called feature extraction. If you choose a subset of the input features to train the model and ignore the obsolete or redundant feature, then it is called attribute subset selection or feature selection.

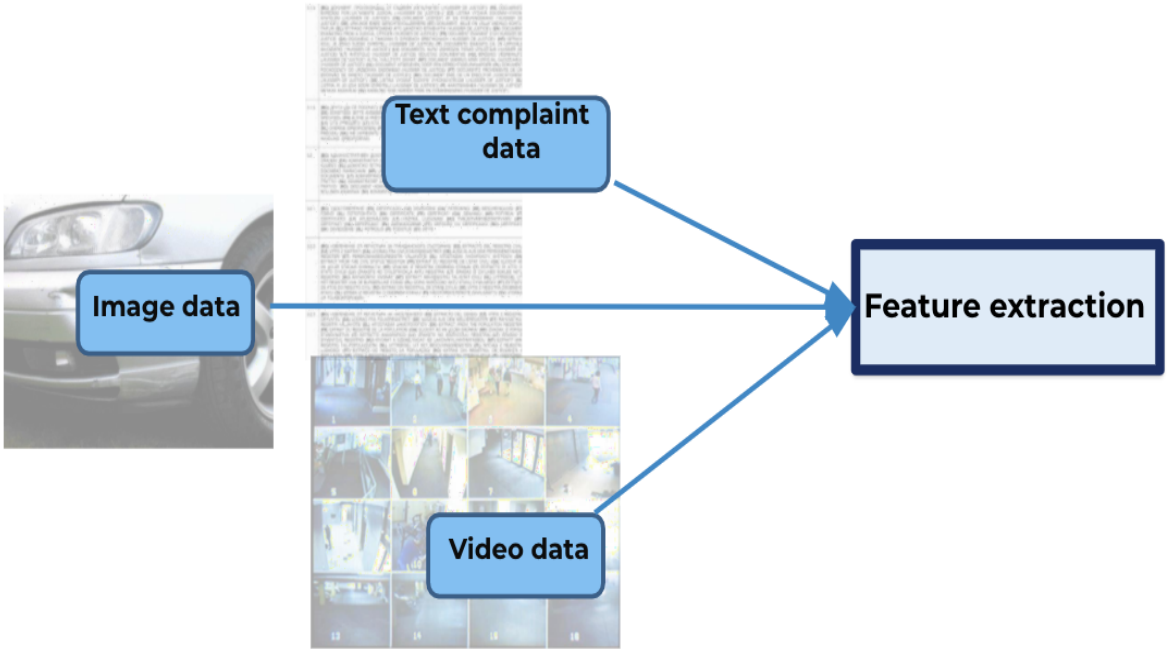

Figure 3: Extracting features from an image, video, and text data.

Machine learning algorithms require a feature representation that could be used in ML models. Particularly, suppose you have unstructured data, such as text data, image data, video data, and so forth. In that case, it does require a feature extraction system that would convert them into some sort of numerical form: real numbers or categorical numerical numbers or classes. Extracted features are intended to be informative and non-redundant, facilitating the subsequent learning and generalization steps, and in some cases leading to better human interpretations.

D. Modelling:

The next step in the CRISP-DM framework is modeling, where we try to fit different machine learning models or their combination on the prepared data. We prepare the final model with the best performing algorithm. Different models require different data preparation steps. So there's a back and forth between modeling and data preparation.

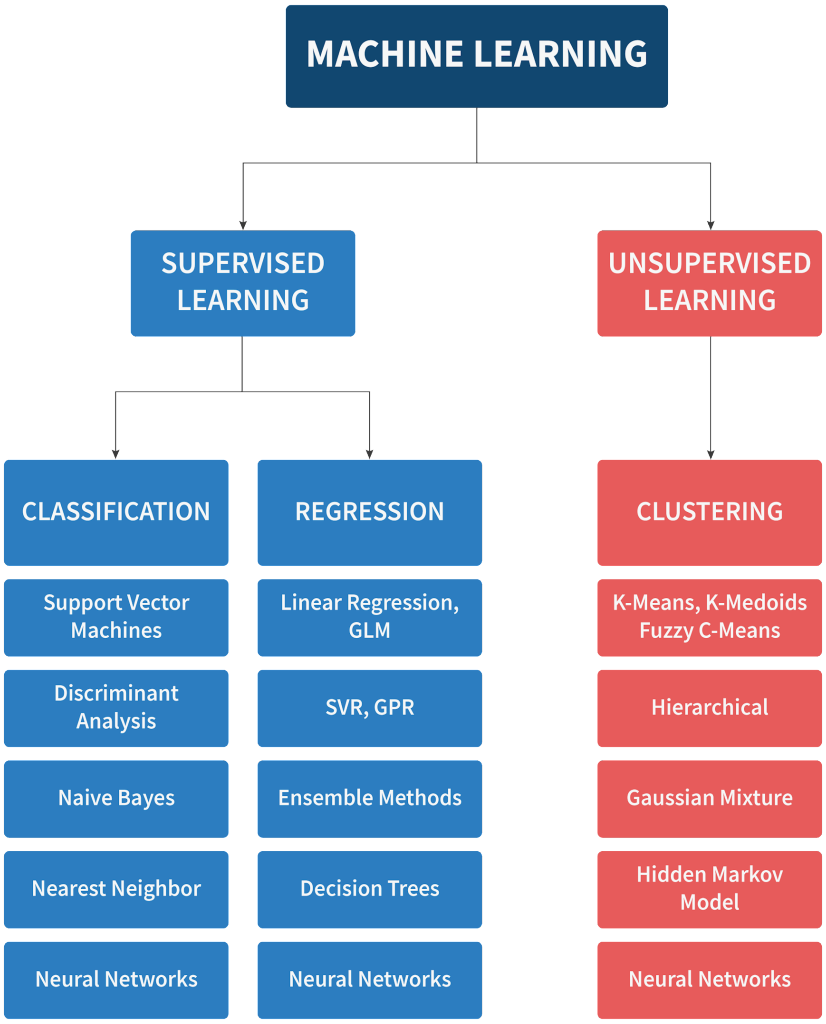

Figure 4: Machine learning Algorithms chart

Machine learning has a pool of methods and algorithms based on the supervised and Unsupervised learning paradigm. Supervised learning can be divided into classification and regression problems. Regression algorithms are used to predict continuous values, such as price, salary, age, etc. Classification algorithms are used to predict/Classify the discrete values such as Male or Female, True or False, Spam or Not Spam, etc. We select the appropriate algorithm based on our use case. The selection of the appropriate method depends on your understanding of the data.

It is the core step of the ML pipeline. In this step, we train the model to take inputs and predict an output with the lowest error possible/ better accuracy. In this step, we tune the ML model hyperparameters(like learning rate and number of hidden layers) to find optimal model architecture or hyperparameters.

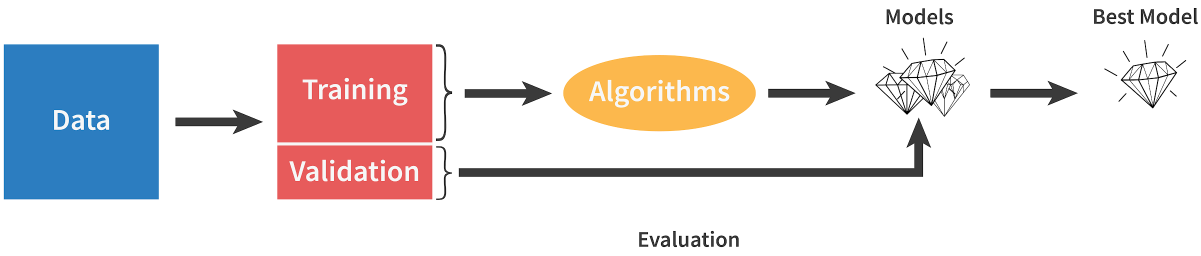

Figure 5: ML Model Development Ecosystem

The training set is further divided into the train set and the validation set. 80/20 split is indeed a good starting point. The validation set helps to evaluate different models and select the best one. We train the model using a training set on specific hyperparameter settings, evaluate the trained model using a validation set, and check different evaluation metrics. After repeating this process several times, we get a different model with different hyperparameters. This process continues until we are not satisfied with our model performance on the validation set. Finally, we pick the model with a high evaluation metric score.

E. Evaluation:

As mentioned above, after fitting the model, we evaluate its performance in the evaluation step. Here, we assess the model and evaluate the model concerning success criteria defined in the business understanding phase. It helps find the best model representing our data and how well the chosen model will work in the future. Understanding the business better almost always improves the evaluation score of the model. So here, too, there is a back and forth between evaluation and business understanding.

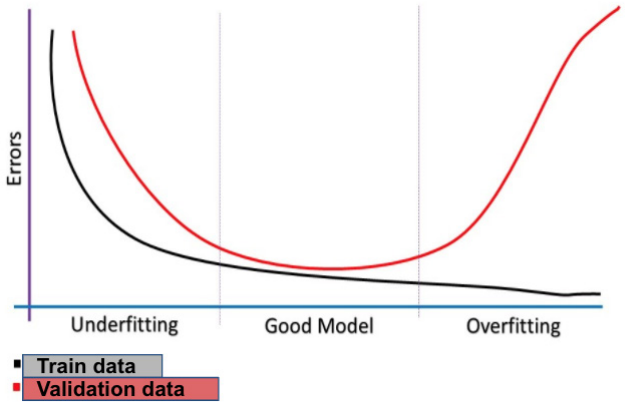

Figure 6: Learning Curve

To determine whether the model is predicting well in unseen data, we plot the error plot for training data and validation data. If the error is high for both training and validation data, then it is a case for underfitting. Underfitting is the case where the model has " not learned enough" from the training data, resulting in low generalization and unreliable predictions.

If an error is low for training data and high for validation data, which means there is an increased gap between the error curve of train and validation data) then the model is overfitting. Overfitting is the case where the overall cost is minimal for a training set, but the model's generalization is unreliable. This occurs because the model learns "too much" from the training data set. It performs well in the training set but poor at the validation set or unseen data. This problem can be understood as too much memorization of training set data patterns, so it performs well at train set and worst at validation set.

The solution for underfit is to increase the model complexity, and add more features , add polynomial features. The solution for overfitting is to reduce model complexity, Add training data, introduce regularization and dropout.

Machine learning has several evaluation metrics under different categories.

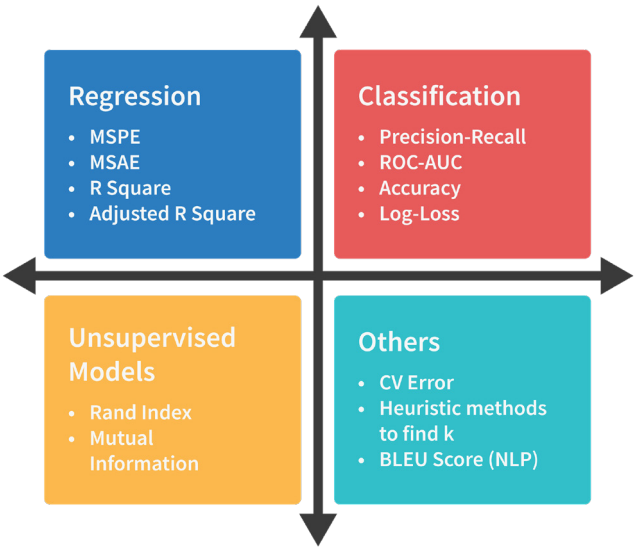

Figure 7: Evaluation Metrics

MSPE, MSAE, MSE, R-SQUARE, etc., are some of the evaluation metrics for regression, which tells how much is the prediction deviated from original values

Accuracy, Precision-recall, F1 score, Roc-AUC, etc., are the performance evaluation metrics for classification problems. Similarly, rand index, Mutual information, etc. are metrics for evaluating unsupervised models, which is out of scope for this Reading.

To calculate the MSE, we take the difference between our model's predictions (y hat) and the actual value (y), square it, and average it out across the whole dataset.

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

The MSE is excellent for ensuring that our trained model has no outlier predictions with huge errors since the MSE puts larger weight on these errors due to the squaring part of the error.
The Mean Absolute Error (MAE) is only slightly different from the MSE, but interestingly provides almost exactly opposite properties! To calculate the MAE, we take the difference between our model's predictions and the actual value, apply the absolute value to that difference, and then average it out across the whole dataset.

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$$

Like the MSE, the MAE will never be negative since we are always taking the absolute value of the errors. The beauty of the MAE is that its advantage directly covers the MSE disadvantage. We are taking the absolute value, all of the errors will be weighted on the same linear scale. Thus, unlike the MSE, we won't be putting too much weight on our outliers, and our loss function provides a generic and even measure of how well our model is performing.
Accuracy is the simplest performance metric for classification models. It is the fraction of predictions that the classifier predicted correctly.

$$\text{Accuracy} = \frac{\text{No. of correct prediction}}{\text{Total No. of prediction}}$$

For example, if the test accuracy for the sentiment detection model is equal to 0.75, it correctly classifies 75% of the test data documents.

A confusion matrix is a 2d array that looks like the figure below,

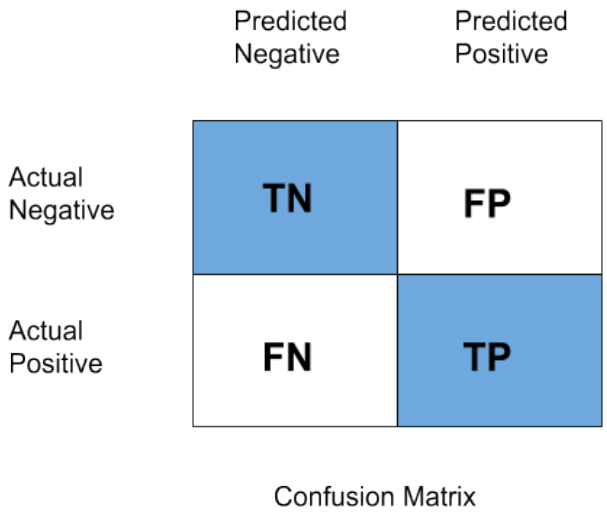

Figure 8: Confusion Matrix

The rows of the matrix represent the true classes' instances, while the columns represent the instances of predicted classes (or vice versa). It summarizes correct and incorrect predictions made by the classifier. Each box of the confusion matrix represent

Classifier prediction, which can be summarized as:

- TN is True Negatives. It is the number of examples that our classifier predicted as nega tive.
- FP is False Positives which is the number of examples that our classifier predicted as positive but are actually negative.
- FN is False Negatives. It is the number of examples that our classifier predicted as nega tive but are actually positive
- TP is True Positives. It is the number of examples that our classifier predicted as positive and are actually positive.

F. Deployment:

The last step is a deployment where we deploy the model. This can be done by packaging the system into a Docker container and exposing a REST API.

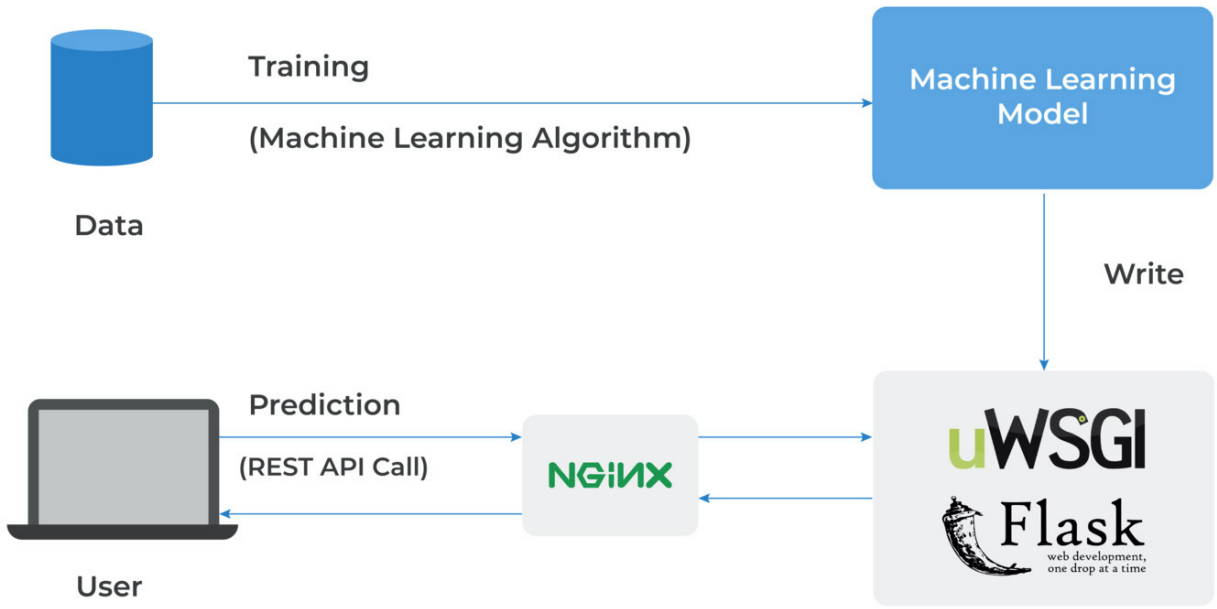

Figure 9: Deployment Lifecycle

Basically, when we are deploying ML models, we serve them through a server. When we deploy the model to the server, we have to create an endpoint so that our application can access it (say REST service or REST API). REST API increases our ML model's reach to a broader audience and can be called from any Application – mobile app, Java application, web applica-tion, .Net application, PHP/javascript, Python, etc.

The most common Python framework, which is used to create a REST API in Python is Flask. Flask is a lightweight micro web framework written in Python, which is used to create

small web applications in Python.

Django is also the popular web framework for Python, widely used for web app development of medium to large scale. Due to the lightweight and simplicity of flask, the flask is preferred by many machine learning engineers to create API to serve ML models.

Frameworks and tools like TensorFlow, tensorflow.js can help to achieve the model deployment.

We have reached the end of this read on Building A ML Model, do tackle the quiz, and look at programming material to get an overview on how to build, evaluate and select the best model for a simple binary sentiment classifier.