Machine Learning(ML) Pipeline

Throughout the whole syllabus, you learned about many ML algorithms like supervised(regression, classification), unsupervised(clustering algorithms like KNN), and reinforcement learning techniques. Along with ML algorithms, you also learned about techniques to analyze the data quality, data transformation, and so on. The figure below contains some of the components of ML that you learned during this course.

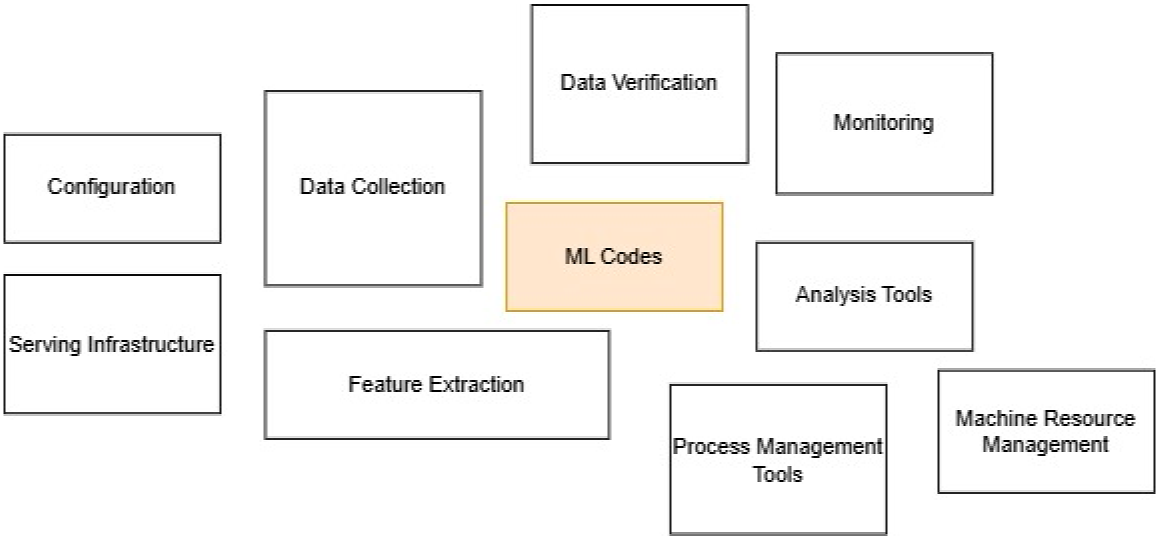

Fig. Machine learning components

Now, you are in the confusion that why we discuss all these things here and how these are related.

Here, in this chapter, later on, you learn how all of these relate to the chapter, and your's confusion will be clear out.

Introduction

Now, you might have got certain ideas where we are heading in this chapter. Till now, you learn how to solve certain problems using ML algorithms. But, in the real world, it not only related to solving them by implementing the code of some algorithms. There is more to it.

In real-world problems, you collect the data from different sources, analyze them, preprocess them, and then finally use different ML algorithms that you learned up to now. But, this not stops here. You need to do this step again and again as real-world data keep changing. This whole process consumes both time and resources.

Now, you got the idea of what you learned, and the diagram above shows is only the tip of the ML. ML is complicated, and there need to some methods or processes to organize the chaos present in the ML. This process that helps to organize the ML system by breaking into steps(by modularizing ML system) is known as ML pipeline. You must not confuse that ML pipeline solves all problems related to ML system building problems but, it helps to speed up the process, minimize the cost and time.

ML pipeline is the process that combines the different steps/methods of the ML system to produce the required output. ML pipeline helps to tackle the problems like data collection, deploying and reproducing models, model monitoring, handling model feedback.

ML Pipelines Architecture

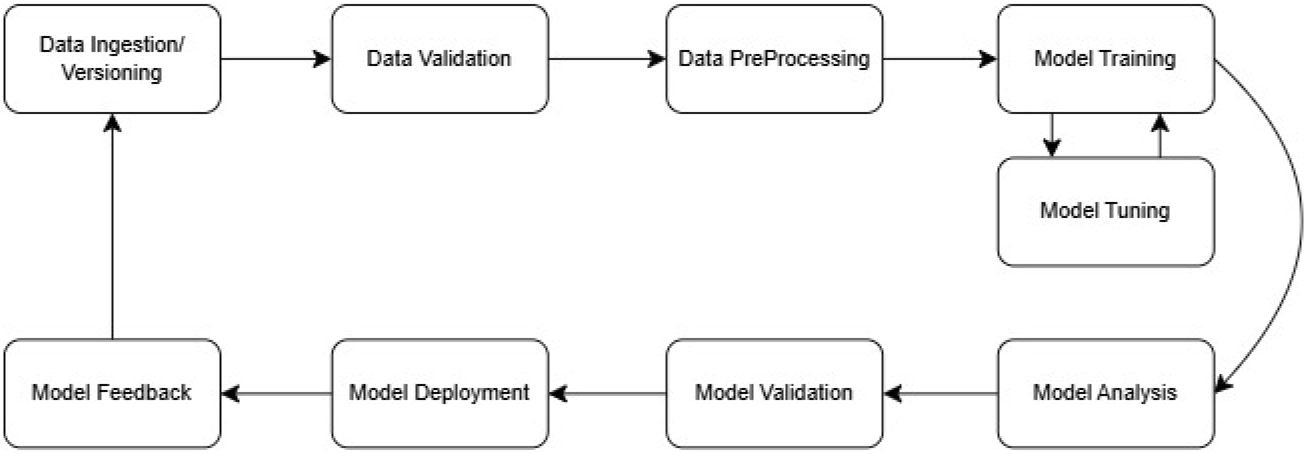

Fig. ML pipelines model life cycle

ML pipeline starts with new training data. It ends with receiving some feedback(production performance metric, users feedback using the product) on how the newly trained model is training. As shown in the above figure, the ML pipeline is a recurring cycle. The steps involved in working workflow of ML pipelines are as follows:

1. Data Ingestion/Data Versioning:

Here, we process the data into a format so that the following pipeline components can utilize or directly ingest the data if formatted. Besides this, we also split the following data into train and test sets.

We use technologies like Data Version Control(DVC) for data versioning in this step.

2. Data Validation

Here, we check the anomalies present in the data. We also check the statistics(frequency of the missing values; features are highly correlated or not) of the data. We also check if data schema for training and testing is changed or not. By observing results, we can address the data issue or stop the ML system workflow before implementing the ML system.

3. Data Preprocessing

This step is most likely the feature engineering step, where you do feature transformation things like mathematical transformation, categorical transformation, feature scaling. Here, you also do the things like normalization, feature extraction and selection. This is the step where you prepare the data for the ML model.

4. Model Training and Tuning

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, number of hidden layers) so that optimal model architecture or hyperparameters are found. Experiment tracking tool stores all the training parameters and evaluation metrics of every model training run in this step so that we can evaluate the model performance in a better way.

5. Model Analysis

In this step, we use the accuracy or the loss to determine whether the parameters of the model are optimal or not. In here, we carry out the in-depth analysis(metrics like precision, recall or calculating performance on larger datasets) of the model's performance in order to check that the model's performance is fair.

6. Model Versioning

In this step, we version the different models so that we can keep track of which model, set of hyperparameters and data sets have been used in the deployment.

7. Model Deployment

In this step, trained, tuned, and analyzed models are ready for deployment. We deploy these models in the server so that users can use it and provides us with feedback for its improvement.

8. Model Feedback/Feedback Loops

This is the last step of the ML pipeline and is also one of the crucial steps for the success of the ML projects. In this, we update the model and create a new version by observing the valuable information on the performance of the model, users feedback, and also introducing new training data in our data sets.

Benefits of ML pipeline

ML pipelines provide benefits to the data scientists/engineers, ML engineers, and organizations working in the ML field in various forms. Some of the benefits are as follows:

- ML pipeline provides the simplicity to the complex ML systems. ML system doesn't appear as complicated as it looks with pipelines. Pipelines also make code more cleaner and nice.
- The pipelines help to reduce the occurrence of bugs(for example, during data preparation, there is high chance of mistake if you do manually).
- As pipelines are modular in architecture, so ML system builds using the pipeline is scalable.
- Pipelines also help in tracking the different model's hyperparameters, datasets used, metrics like accuracy, loss.
- Pipelines also reduce both the time and resources used in building ML systems.

Challenges Associated with ML Pipelines

Although there are advantages of ML pipeline, there are also challenges in creating it. Some of them are listed below:

1. Data quality:

   The success of any ML models depends on the quality data used in building it. If data provided is not accurate, the ML model you build is of no use as it will produce incorrect output.
2. Data Reliabilty:

   If the sources form where data is obtained in not reliable, the ML model build by you is of use. So, you must ensure that data sources are trusted and reliable.
3. Data Accessibility:

   The ML pipelines provide the best results if the data used in it is accessible. In ML pipeline, you need to consolidate data obtained from different sources, clean, and curate it.

Now, you know what ML pipeline is. If you do such things manually, you will consume both your time and resources. There are many tools and technologies (like Sklearn pipelines, Cron, AzureML, MLFlow) to create the pipeline workflow. About these tools and techniques, you learn in the next chapter in this module.

But, still, till now, you don't know how to create an ML pipeline workflow. So for this, there is a subtopic in this chapter, where you will learn to create a simple ML pipeline workflow using the sklearn-pipeline method and also get familiar with the working of the sklearn workflow.