**Machine Learning Pipeline: -**

Surround is a lightweight framework for serving machine learning pipelines in Python.

As the name suggests, pipeline class allows sticking multiple processes into a single scikit-learn estimator. pipeline class has fit, predict and score method just like any other estimator.

To implement pipeline, we separate features and labels from the data-set at first.

X=winedf.drop(['quality'],axis=1)  
Y=winedf['quality']

from sklearn.pipeline import Pipeline  
pipeline = Pipeline(steps) # define the pipeline object.

* We can use make\_pipeline instead of Pipeline to avoidnaming the estimator or transformer.
* Each pipeline component is separated from the others, and takes in a defined input, and returns a defined output.
* We store the raw log data to a database. This ensures that if we ever want to run a different analysis, we have access to all of the raw data.
* We remove duplicate records. It’s very easy to introduce duplicate data into your analysis process, so deduplicating before passing data through the pipeline is critical.
* Each pipeline component feeds data into another component. We want to keep each component as small as possible, so that we can individually scale pipeline components up, or use the outputs for a different type of analysis.

Advantages of pipelines

1. Use of pipelines gives you a kind of meta-language to describe your model and abstract from some implementation details.
2. With pipelines, you don't need to carry test dataset transformation along with your train features - this is taken care of automatically.
3. Hyperparameter tuning made easy - set new parameters on any estimator in the pipeline, and refit - in 1 line. Or use GridSearchCV on the pipeline.