**Production Code Types**

**Procedural Programming**

**Custom Pipeline Code**

**Third Party Pipeline Code**

**Production Code Overview**

* **Towards deployment code**
  + You need to write production code for the entire Machine Learning Pipeline
  + We do not deploy models using Jupyter Notebook, we deploy models using scripts. So you have to transition the code from a Jupyter Notebook to a Python script
* **How to write deployment code for ML**
  + Procedural Programming, which the same as we do in Jupyter Notebook
  + Custom Pipeline Code is object oriented and calls the procedures in order
  + Third Party Pipeline Code is object oriented and uses the scikit-learn pipeline

**Procedural Programming Pipeline**

* We create three scripts:
  + Functions script
  + Training script:
    - Notice that only the *train* and data is used
  + Scoring script:
    - Notice that the *scaler* and *model* are imported
* **Procedural Programming: yaml file (or py file)**
  + Features: the list of feature that are used to train our models
* **Procedural Programming: Overview**
  + Advantages:
    - Easy to manually check if it reproduces the original model, as it is quite similar to the original code
  + Disadvantages:
    - Can get buggy, it’s difficult to pick those error by the design of the code
    - Difficult to test, as it relies mostly on manual testing, which is time consuming
    - Difficult to build software on top of it, as it generally carries on a poor design
  + Overall, it’s not recommended
    - If this is the only model that your company wants to deploy and you are the only data scientist, then this paradigm would be fine

**Designing a Custom Pipeline**

**Custom ML Pipeline: Summary**

* We call it custom because we design the pipeline. But different organizations have different pipelines

**Custom Pipeline: Overview**

* Advantages:
  + Can be tested, versioned, tracked and controlled as it is well design from the start
  + As it is well design, we can build future models on top of it or embed it in other pre-existing pipelines
  + Built to satisfy business needs as it is a custom design
* Disadvantages:
  + Data Scientists may be familiar with specific business software, making it difficult for him to add additional steps or integrate new methods to preexisting pipelines
  + As they are custom pipelines, they are often not reusable. This means that the pipeline that we build to train and deploy one model may not be suitable to train and deploy another model
  + In a nutshell, custom pipeline may lack the versatility to be used across multiple projects

**Shallow Dive into Scikit-learn API**

* We will use BaseEstimator and TransformerMixin.
  + BaseEstimator:
    - It *gets* the parameters that we give to the class when we initialize it, and returns it as Scikit-learn returns all its parameters
  + TransformerMixin:
    - Fit\_transform has the functionality to implement the fit method followed immediately by the transform method
* We will inherit these two base estimators and rewrite the fit and transform method.