KAGGLE: DATA SCIENCE MANAGEMENT FRAMEWORK

FERNANDO RAMIREZ

1. Mission: Less Code, More Information

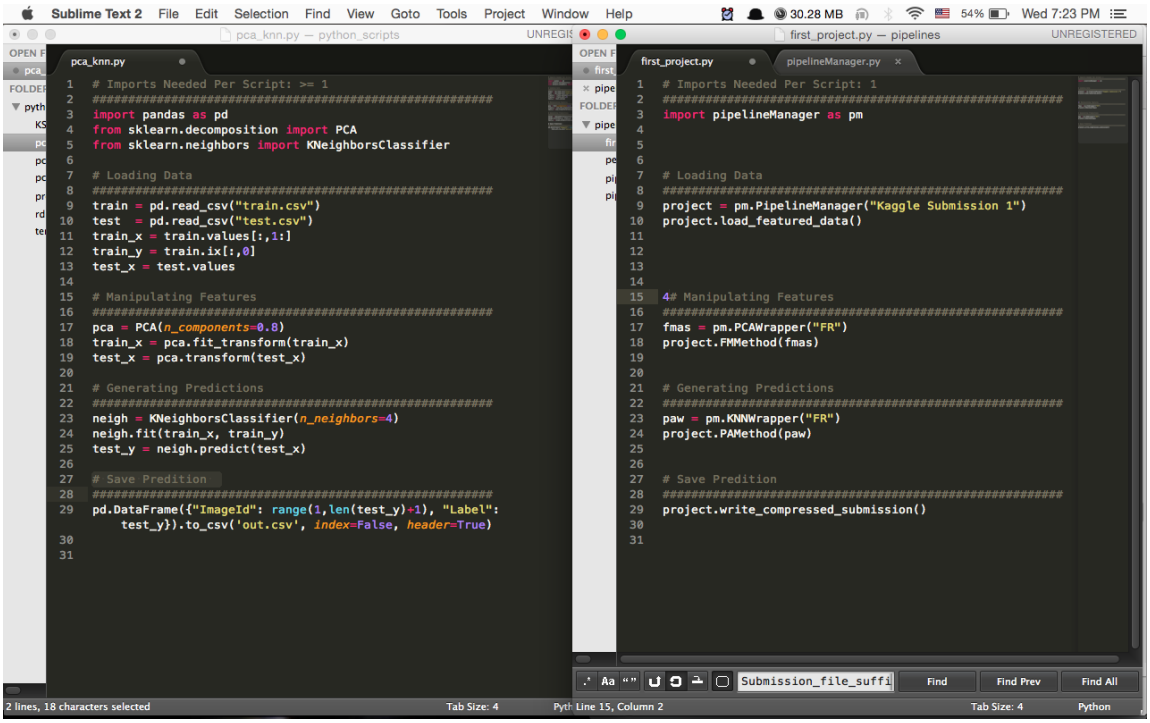

FIGURE 1. Left: Simple AI Script in python; Right: Enhanced and Simplified AI Script with our framework

2. Intellectual Basis

2.1. **Tidy Data and Kaggle.** My interest in machine learning came from the same place every body else does, a desire to create a better smarter world. However, when I first attempted to enter the space, I found that my troubles came mostly from semantic code compilation problems.

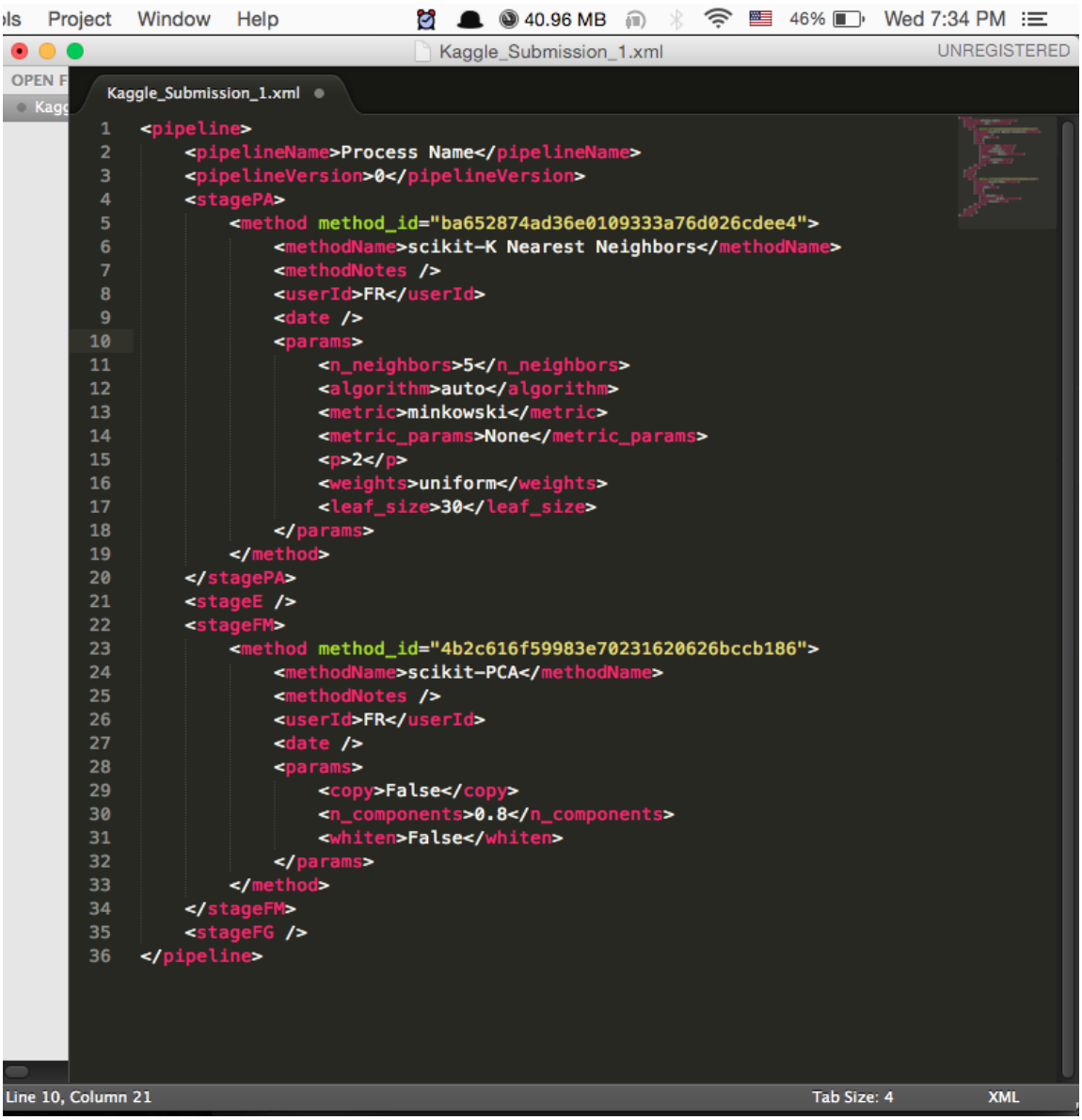

FIGURE 2. Pipeline XML: Generated when script is compiled!

I left for a while, but was dragged back in when I discovered Kaggle! I started competitions but the problems still persisted, so I set out to make entering data science competitions about the data and the math.

Which brings us here!!! This framework is loosely based on the Tidy Data paper by Hadley Wickham which stresses the importance of documenting how data science scripts are documented, and git hashing for optimization.

Below I describe the fundamentals of this framework which is based around modularizing, and reusing what we refer to as "Pipelines", which are scripts that take a problem from input data to prediction!

2.2. **Framework.** Work flow Each Pipeline Manager(PM) object should be a unique attempt to create a prediction The following are functions(maybe classes, but for now should I have them as functions) that have functions that take a PM object. They take PM object because they manipulate the data or make prediction and need to write what they do to the log of the object, which in theory should help us generate a complete report of how the output was developed from raw data to prediction.

Here are the basic function/class contracts:

- PM class:
  - abstract away the input and output of the data to the correct folder
  - Separate training data into train and cross validation set
  - track data manipulation
  - print report for Pipeline
  - Different logs for different types of transformations described below that should just be compiled together visualize data(later feature)
- Feature generator: Responsible for turning raw data into features
- Feature Selector: Responsible for selecting a number of features dimensionality reduction falls into this
- Prediction Algorithm: Takes in feature data and returns a prediction
- Ensemble: Takes in data and a set of learning algorithms and returns a prediction.

2.3. **Key Benefits.** Improved Script Tracking: Using the framework allows the learning process to be modularized and analyzed at a higher level which improves the script development process.
Improved Script Reusability: Scripts are compositions, which means that they are independent of the data that is used. This feature allows for improved milage on different components, and a quicker conceptualization or development of intuition of what different components should be used and when.
Improved Readability: Tracking legacy code is made simpler and teams can focus on developing components and back testing them, which is important for data science problems, where small tweaks could make a big difference. This feature also allows for new people to the field to not need to code as much and can focus on the math!

2.4. **If this was a Google Project.** Aka: Unlimited resources!

I would want this to be a web hosted service, where people could share their components and talk about their performance. You could license the use of scripts, and sell them for different applications. Make big data projects more collaborative and more lucrative. I dont necessarily like capitalism, but I think that reducing the barrier to entry and making this type of profession fun, could in the long run make the functionality more affordable.