

Workflow: Moving Out of a Notebook

Abigail Haddad

Full Stack Data Science DC

2/28/2024

Hi, I Am A Data Scientist

- I have been a data scientist for ~8 years
- I am trying to develop better development practices because, like everyone, I want my stuff to work better and break less
- But there are time costs to everything, and the answer to “what practice should I add and when?” is never “everything and all at once”
- I hope this will be helpful if your code currently lives in notebooks (Jupyter, Colab, Databricks)



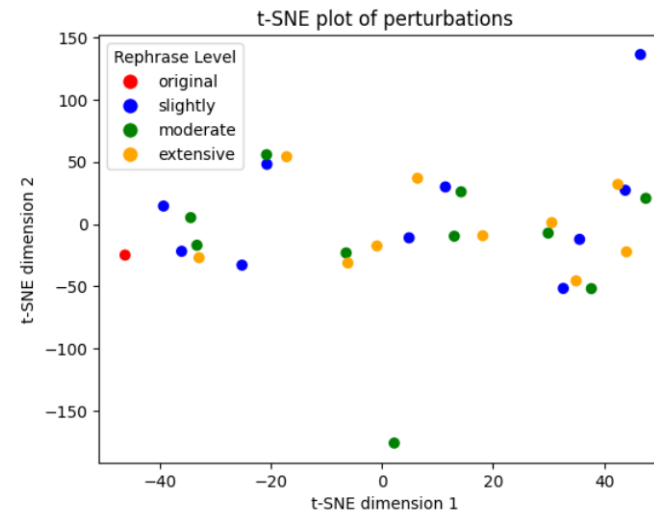
This was as good as I could get with GPT-4

Many Data Scientists Like Notebooks!

- Notebooks make seeing your data and outputs easy
- A lot of us grew up as data scientists using Jupyter
- You can easily intersperse markdown, code, and results

And then we plot to see how related the rephrase level is from the distance from the original prompt in this space

```
In [5]: plot_tsne_with_colors(result_df)
```



Markdown, code, and output all in one!

Wait, “Production”?

- An ML model getting used for something in the world
- A dashboard people are consuming
- A package people are using
- Code that is run automatically to create a report
- Something that is part of an ongoing process, rather than ad hoc

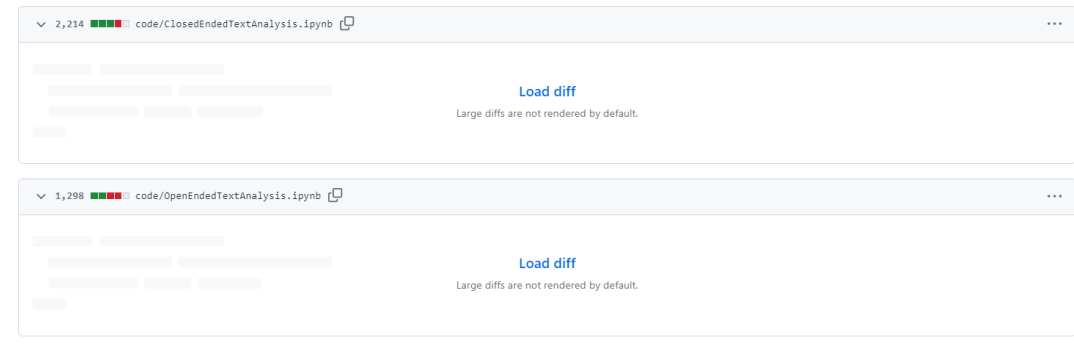
Wait, “Production”?

- An ML model getting used for something in the world
- A dashboard people are consuming
- A package people are using
- Code that is run automatically to create a report
- Something that is part of an ongoing process, rather than ad hoc

Most of your code might not get there!

Notebooks Are Not Ideal for Production

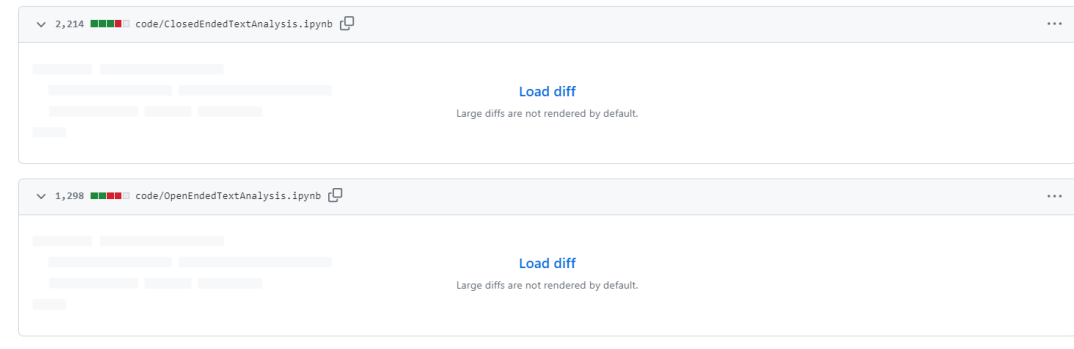
- Not well-suited for tracking, comparing changes
- Not well-suited for reuse
- Not well-suited for unit tests
- How data scientists tend to use notebooks (lack of functions and classes) is not transparent, maintainable



GitHub defaults to not showing my .ipynb changes

Notebooks Are Not Ideal for Production

- Not well-suited for tracking, comparing changes
- Not well-suited for reuse
- Not well-suited for unit tests
- How data scientists tend to use notebooks (lack of functions and classes) is not transparent, maintainable



GitHub defaults to not showing my .ipynb changes

Yes, there are tools to work around this.

Workflow: Notebooks -> .py files

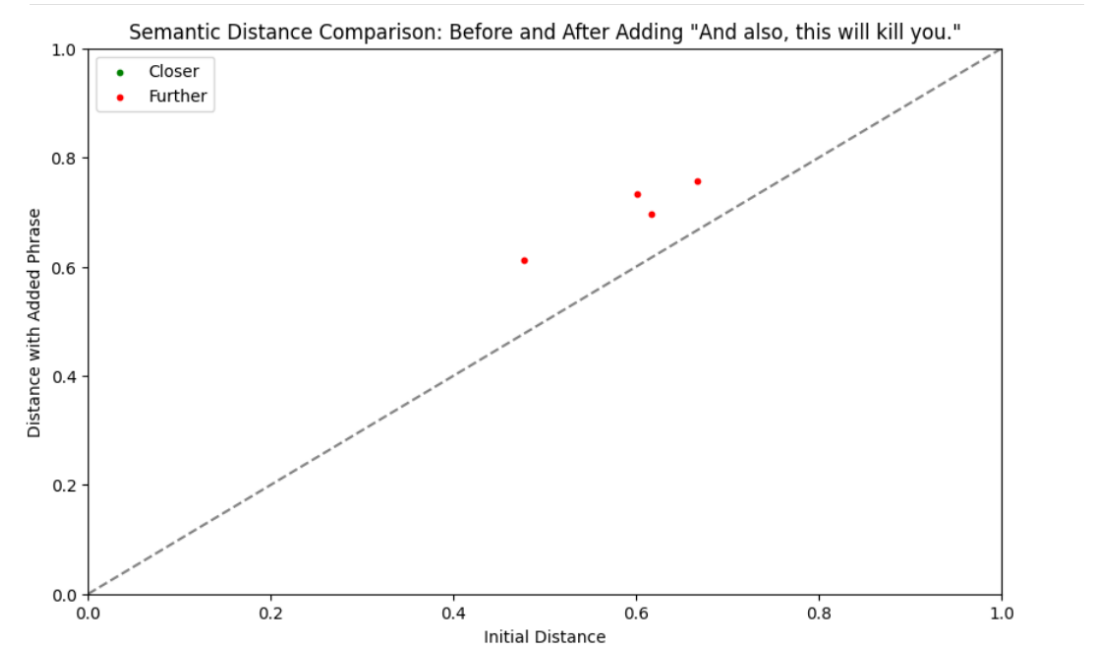
- Start developing in a notebook
- Temporary code stays in the notebook (or you delete)
- Long-term code or code that gets reused in multiple workflows goes in one or more .py files

Name
..
AuxiliaryOpenEndedAnalysis.ipynb
ClosedEndedTextAnalysis.ipynb
OpenEndedTextAnalysis.ipynb
PromptGenerationExamples.ipynb
Prompt_distance.ipynb
__init__.py
api_call_demo.py
functions.py
litellm_uuid.txt
population_max_simulation_demo.ipynb

This is in-between: still a bunch of notebooks and one .py file

Why Do It This Way?

- You like developing in a notebook
- A lot of the code you write is exploratory and temporary
- You eventually want something more production-ready



No one besides me ever needs to see or make this graph

My Steps For Moving Over

When I have a bunch of code that looks permanent or I need for multiple workflows:

- i. Refactor into classes and functions
- ii. Add docstrings (with GPT)
- iii. Maybe add tests
- iv. If I still have notebook code, import what I need into the notebook from the .py file

```
▼ class DataLoader:
    """
    A class for loading data into a DataFrame.

    Args:
        input_data (str or list): The input data to be loaded. If `is_file_path` is True, it sh
        is_file_path (bool, optional): Indicates whether `input_data` is a file path or a list

    Methods:
        load_data(): Loads the data into a DataFrame and performs some preprocessing.

    Returns:
        pandas.DataFrame: The loaded data.

    """

    def __init__(self, input_data, is_file_path=True):
        self.input_data = input_data
        self.is_file_path = is_file_path

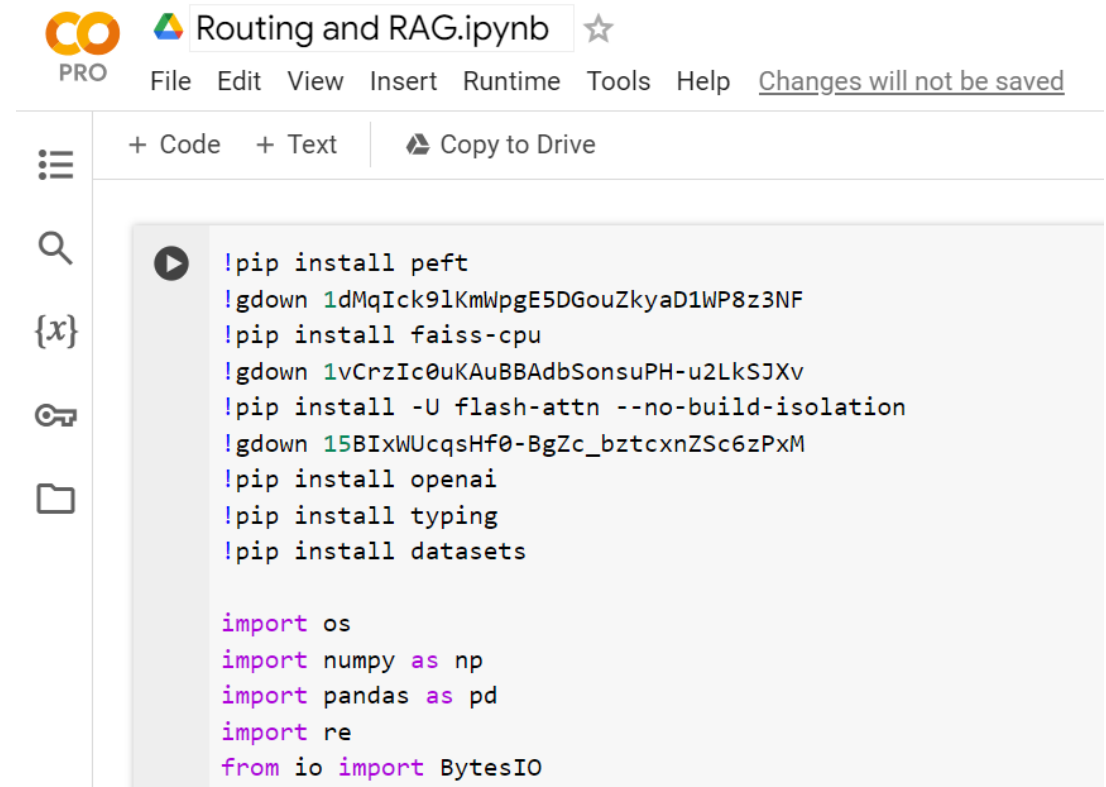
▼ def load_data(self):
    """
    Loads the data into a DataFrame and performs some preprocessing.

    Returns:
        pandas.DataFrame: The loaded data.
```

Hey, it's a class!

Keeping Demo Code In a Notebook

- Other people like interacting with notebooks, too!
- Combine your code, readme, and requirements.txt all in one file



The screenshot shows a Google Colab notebook interface. The title bar indicates the notebook is named "Routing and RAG.ipynb" and is in "PRO" mode. The menu bar includes "File", "Edit", "View", "Insert", "Runtime", "Tools", and "Help". A status message at the bottom of the menu bar says "Changes will not be saved". The notebook content is displayed in a code editor with a light gray background. The code is as follows:

```
!pip install peft
!gdown 1dMqIck9lKmWpgE5DGouZkyaD1WP8z3NF
!pip install faiss-cpu
!gdown 1vCrzIc0uKAuBBAdBonsuPH-u2LkSJXv
!pip install -U flash-attn --no-build-isolation
!gdown 15BIxWUcqsHf0-BgZc_bztcxnZSc6zPxM
!pip install openai
!pip install typing
!pip install datasets

import os
import numpy as np
import pandas as pd
import re
from io import BytesIO
```

This was for a Data Science DC demo and it was fine!

A Quick Note About Tests and GitHub Actions

- Dip a toe into CI/CD!
- A benefit of moving out of the notebook is automating testing
- Don't test code you're not keeping
- Ask GPT for help with the tests and the .yaml file

```
name: Python CI

on:
  push:
  pull_request:
  workflow_dispatch:

jobs:
  test:
    runs-on: ubuntu-latest

    strategy:
      matrix:
        python-version: ['3.9', '3.10']

    steps:
      - uses: actions/checkout@v2
      - name: Set up Python ${ matrix.python-version }}
        uses: actions/setup-python@v2
        with:
          python-version: ${ matrix.python-version }}
      - name: Install dependencies
        run: |
          python -m pip install --upgrade pip
          pip install -r requirements.txt
      - name: Run tests
        run: |
          python -m unittest discover -s tests
```

Hey, it's a .yaml file

An Even Quicker Note About Virtual Environments, Kernels, and Notebooks

I start my notebooks by typing 'jupyter notebook' into the command line from within my virtual environment

Thank you!

- I have a blog: <https://presentofcoding.substack.com/>
- Data Science DC is awesome! March 12, Deep Learning Generalization: <https://www.meetup.com/data-science-dc/>
- These slides are on GitHub:
<https://github.com/abigailhaddad/workflowSlides>