# Kickstarter ML Code Overview

## Summary:

All documentation regarding the code is within the code itself, along with the README.md file. The README file outlines how to run the application, and some troubleshooting advice on common issues. This document is simply briefly highlighting the steps undertaken during development, and some issues I encountered.

## Steps Undertaken:

* Refactoring:
  + Code Refactoring: Logically separated all the code into relevant files, classes, and functions.
  + Extraction: Migrated all relevant attributes and variables that would benefit from user customisability to outside of the python scripts, this allows for easy parameter tuning.
  + Comments: Added comments and relevant docstrings to all classes and functions.
  + Outputs: Decided to output both y\_hat and y\_test to an outputs.json file within /output\_data. This is to allow later analysis and visualisation of the model.
* Create Flask App:
  + While Flask is not inherently a production service on its own, it serves the purpose of demonstrating being able to access an ML model via HTTP requests, which provides a convenient and scalable way to deploy an ML model in Docker.
  + Due to the extraction of variables into configuration.json we can update the parameters within this file, and then pass them as data within the API call.
  + Before deploying to Docker, I tested the Flask API calls by deploying the Flask app locally and encountered a few port issues due to other system processes being mapped to the port I had intended to use, once I had exposed a port that was not in use the API call worked.
* Create Dockerfile:
  + I determined which packages and python version are necessary and specified these within the Dockerfile and requirements.txt.
    - Issues: Initially I specified the versions for each package within requirements.txt, this functioned fine on one laptop, however when running the script on another it caused issues during installation. To resolve this issue the versions were removed, and python set to the latest version.
  + Specified the port to expose within the container to be able to access Flask, this is the same as what is defined within the Flask app declaration in flask\_server.py. In addition, set the FLASK\_APP environment variable to define which python script to run when executing flask run.
* Create Bash Script:
  + Added the code to build and run the docker container, and then set a slight delay to allow for initialisation before calling the curl command to post the API call and write the responses to our outputs.json folder.