**Workflow and Functionalities Documentation**

Our project is focused on leveraging concepts learned during our MLOPs course. To do so, we developed machine learning models that would predict whether a review given by a user is positive (1) or negative (0). We created 4 simple models (Naïve Bayes, SVM, KNN and Logistic Regression), trained and evaluated them. The project includes an API and a Docker file, and can be ran directly through a Command-Line Interface.

**Functionalities**

1. **Model Training and Evaluation:** We developed 4 models that classify review as positive or negative based on a dataset that we stored in the *data* folder. There is a train and test dataset. In our *src* folder, we added the required functions to train and evaluate each model. Specifically, to ingest the data, clean the data, train the models and evaluate them. We save each model in the *models* folder.
2. **Command-Line Interface (CLI):** We developed a simple CLI to make the training and evaluating of the modelwithout having to run each function individually.The code is stored in the run\_cli.py file.
3. **API for Model Prediction:** To allow users to submit a review and have a model specific whether the review is positive or negative, we created an API. The API accepts as input the text (review) and which model to use. It output either a 0 (negative review) or a 1 (positive review).
4. **Docker file:** We build a Docker image using our Dockerfile. By running the Docker container, userscan leverage the API without having to worry about dependencies or setup.The Dockerfile leverages our pyproject.toml file that was generated with poetry and contains all of our dependencies.

This code runs locally. If we wanted to have it run for other users, as next steps, we would need to deploy our container onto either AWS, GCP or Azure. Then, users should be able to access on a website.