App.py

The API is configured in Pycharm and uses git addon to commit and push. Start by making basic 'hello world' type flask API, using GET endpoint as basic connectivity test for future debugging. Add POST endpoint. This is where the data processing will take place. Download Postman. Copy handful of rows from test csv and paste into a csv-to-json converter. Paste resulting JSON string into Postman for ongoing test of API performance. Pickle model, imputer, and scaler from python notebook and load into API. Also pickle lists [3] of float64 fields, object fields, and final features. These lists are imported in and then hardcoded into API. Apply checks for field type using object and float lists as well as check for input size. Take next 3 feature engineering steps from the notebook by [1] taking some symbol parsing code from notebook and apply it in API . [2] Run result through imputer and scaler. [3] One hot encode string fields. Apply the feature selection using the third pickled list and run those fields through the model. Generate predicted class from the resulting probability returned from the model. Combine predicted class, probability, and selected features into single dataframe and convert to JSON for return.

Docker Container

Download and install DockerDesktop. Create Dockerfile. In Dockerfile, setup container environment, ADD everything from local directory to container environment. Create requirements.txt and specify packages the API requires. Create shell file run_api.shell. In the script, require user input to choose container deployment or cluster. Copy Docker commands into shell to build image and launch container.

Kubernetes Cluster (Minikube)

Download and install Minikube as local container orchestration tool. Write up deployment.yaml file to blueprint Service and Deployment. Specify ports and Docker Image. Run commands to start minikube, dashboard for visual, push image to registry, deploy cluster. IP and port of endpoint returned as a result. Once everything works, place these commands in the second half of the shell file.

Docker Container on Amazon EC2

Create repository in AWS ECR. Docker push existing docker image from local to ECR repo. Run EC2 instance. Connect to EC2 and install docker. Run docker pull from EC2 to retrieve image from ECR. Docker Run image in EC2. POST to http://18.234.128.9/predict. GET from http://18.234.128.9/ .