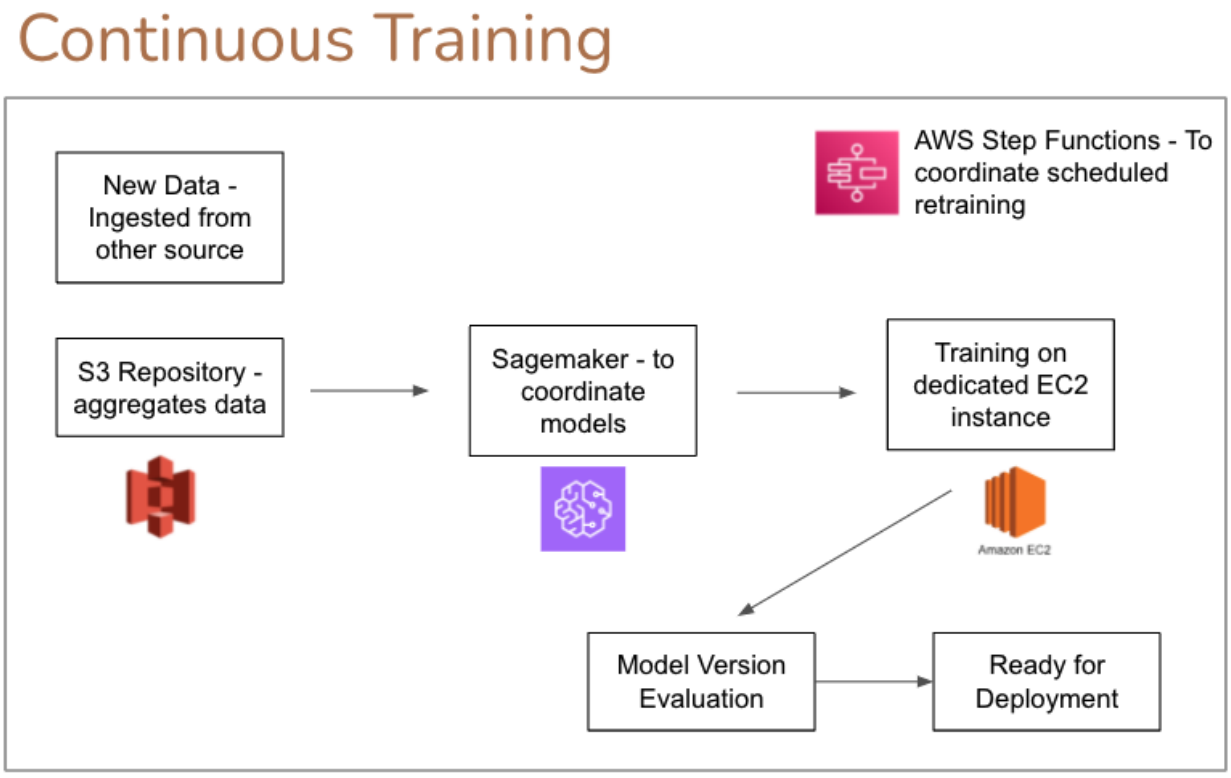
Background:

For my data pipeline project, I built an end to end deployment and training system around a house price prediction model. I made the data pipeline of this project continuous, from data ingestion to model training and re-deployment. In this way, our model can quickly add new data to the system and provide more updated results continuously. I will also build a Flask app that can demonstrate the model in action as well.

Training:

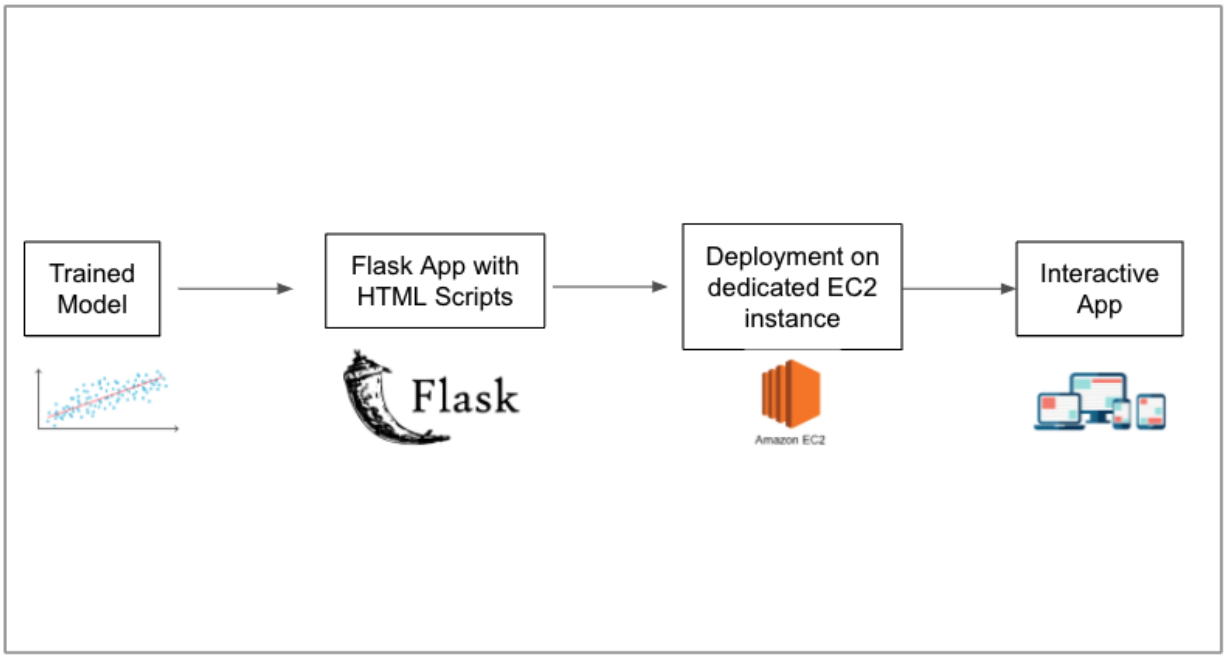


The training was done primarily using AWS Sagemaker with interaction with other AWS services, such as EC2, S3 and Step Functions. The general outline of work was:

1. Set up AWS S3 Bucket with Data
2. Use Sagemaker to pull data from S3 model and prepare for training
3. Training on Sagemaker can be performed on the notebook's instance, or can specify a separate instance for training
4. Schedule Retrainings with Step Functions using new data
5. Compare retrained models with old models

Deployment:

Deployment

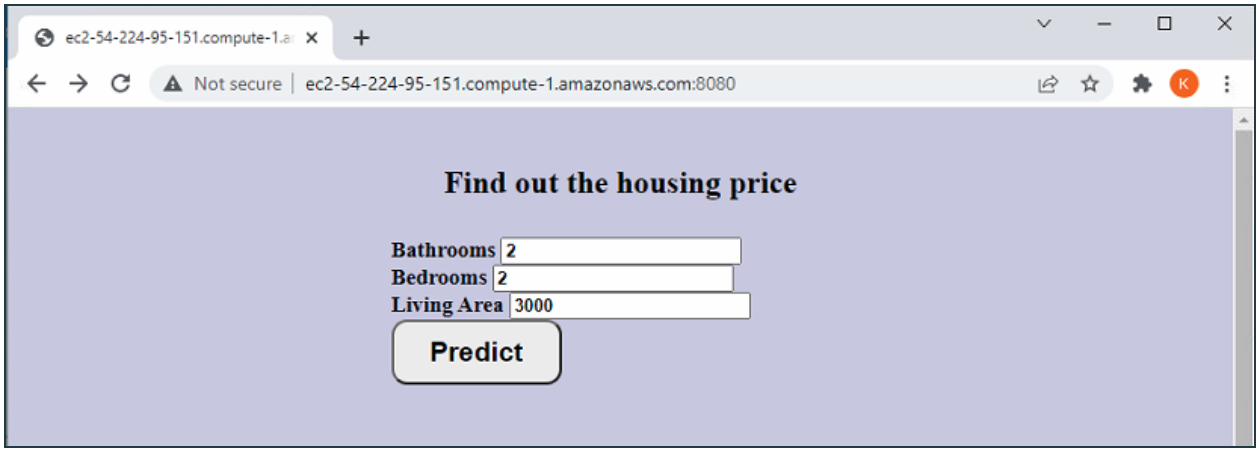


The deployment was build using Flask API and HTML pages, and was deployed to a dedicated EC2 instance. The general outline of this work was:

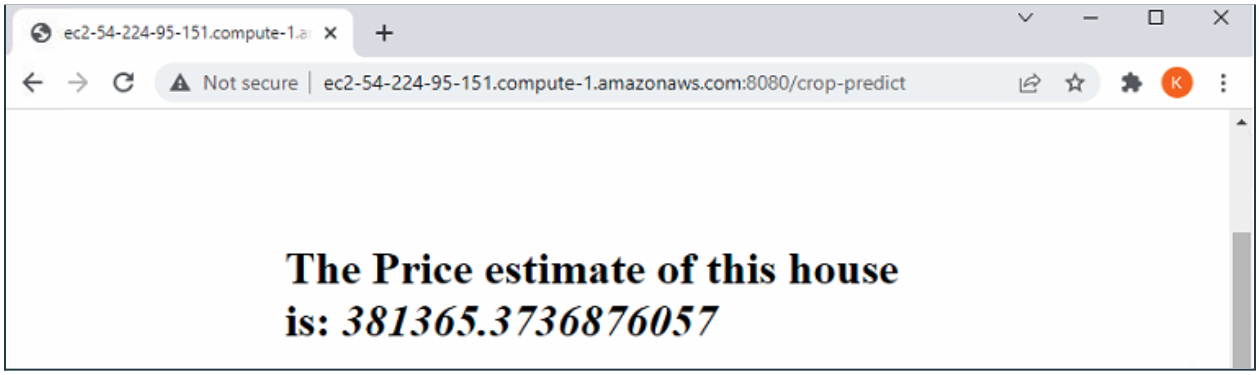
1. Load trained model
2. Create Flask API to run model/predictions
3. Create HTML page
4. Create EC2 instance
5. Create PuttyKey to connect to EC2 instance
6. Load files to EC2 instance
7. Run application file
8. Interact with web app

Final Product:

Flask App Input Page:



Predicted outcome:



Next Steps:

- Expand to larger dataset
- Deploy deep learning networks
- Integrate training and deployment steps more continuously
- Redesign Flask API for more aesthetic appearance

Tools:

AWS: For ML Workflow, S3 storage and instances for training
AWS Sagemaker
AWS S3
AWS EC2
AWS Step Functions

Flask: For building interactive app