

Time Series ARCH Model Deployment on AWS

Business Objective

Deployment is the method by which you integrate a machine learning model into an existing production environment to make practical business decisions based on data. MLOps is a means of continuous delivery and deployment of a machine learning model. Practicing MLOps means that you advocate for automation and monitoring at all steps of ML system construction, including integration, testing, releasing, deployment, and infrastructure management.

In this project, we aim to create the MLOps project for the time series arch model [Build ARCH and GARCH Models in Time Series using Python](#) on the AWS cloud platform (Amazon Web Services) that is cost-optimized. Cost-optimized model deployment means that we will be using minimal services.

Aim

To create an MLOps project using the Amazon Web Services (AWS) platform to deploy time series arch model in production.

Tech stack

- Language - Python
- Libraries - Flask, pickle
- Services - Flask, AWS, Docker, Lightsail, EC2

Approach

1. Model creation
 - Save the model in a pickle format(.pkl)
2. Flask app
 - Create a flask file
3. EC2 machine creation
 - Create an instance on the AWS management console
 - Launch the instance
 - Install the 'putty' tool on your local for login
4. EC2 and Docker setup
 - Refer to the steps present in the 'install-docker.sh' file
5. AWS CLI installation
 - Refer to the steps from 'install-aws-cli.sh' file
6. Lightsail installation
 - Refer to the steps from 'install-lightsail-cli.sh'
7. Upload files into the EC2 machine
 - Method1**
 - Upload the code file in zip format on AWS console (cloud shell)
 - Method 2**
 - Create a S3 storage bucket
 - Copy the object URL, get the URL on ec2 machine
 - Unzip the bitbucket folder
8. Deployment
 - Follow the order of installation, from 'lightsail-deployment.md' file

Files description

1. Input: CallCenterData.xlsx
2. MLPipeline: this folder contains all the functions put into different python files
3. Notebook: time series arch model ipynb file
4. Output: arch model saved in a pickle format
5. App.py: flask app configuration
6. Dockerfile: docker image
7. Engine.py: File where the MLPipeline files are called
8. install-aws-cli.sh: steps for aws cli installation
9. install-docker-.sh: steps for docker installation
10. install-lightsail-cli.sh: steps for lightsail installation
11. lightsail-deployment.md: Lightsail deployment readme file
12. requirements.txt: essential libraries with their versions

Project Takeaways

1. Introduction to ML model deployment
2. What is Docker?
3. Creation of docker file
4. What is flask?
5. EC2 machine setup
6. EC2 and docker setup
7. Perform AWS CLI installation
8. What is Lightsail?
9. Install Lightsail CLI
10. Upload code files using cloud shell
11. Create a bucket and upload code files on S3
12. Docker deployment
13. Lightsail Setup
14. Lightsail Deployment