

# **ML Model Deployment on Azure for Deep learning Time-series Project**

## **Business Objective**

Machine Learning Operations (MLOps) is based on DevOps principles and practices that increase the efficiency of workflows. For example, continuous integration, delivery, and deployment. MLOps enables the automated management of the end-to-end machine learning lifecycle. The main goal of MLOps is the faster deployment of models in production.

We will be using Azure as our cloud platform. Microsoft Azure provides many robust services in its ecosystem to create an end-to-end MLOps pipeline. In this project, we will be deploying our time-series deep learning model on the Azure cloud platform in a multi-part format.

## **Aim**

To create an MLOps project using the Microsoft Azure platform to deploy a deep learning model in production.

## **Tech stack**

- Language - Python
- Libraries - Flask, pickle, pandas
- Services - Flask, Azure

## **Prerequisites**

It is advisable to have a basic knowledge of the following services to understand the project.

- Docker
- Flask
- [Deep Learning Project for Time Series Forecasting in Python](#)

## **Approach**

1. Sign-in to the Microsoft Azure account
2. Create a virtual machine
  - Select the tab to create a new VM
  - Add the basic configuration details to create a VM instance.
3. Connect to the Virtual machine
  - Download and install the putty application

- Add the configuration details
  - Refer to the 'setup-new-vm.sh' file and add the commands.
- 4. Import codebase in a virtual machine
  - Download the Filezilla and create a new client
- 5. Flask app deployment
  - Run the app.py file on the putty terminal
  - Download the postman application and call the Flask API on it.
- 6. Docker installation
  - Follow the steps from the 'install-docker.sh' to install docker
- 7. Docker-based deployment

## **Files description**

1. Input: CallCenterData.xlsx
2. MLPipeline: this folder contains all the functions put into different python files
3. Notebook: cnn-rnn model ipynb file
4. Output: 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. readme. md:
9. requirements.txt: essential libraries with their versions
10. setup-new-vm.sh – file that contains the steps to connect to the virtual machine.

## **Project Takeaways**

1. Introduction to ML model deployment
2. What is Microsoft Azure?
3. How to connect to the Microsoft Azure portal?
4. What is an Azure console?
5. What is a cloud shell?
6. How to create a Virtual machine (VM) instance
7. How to connect to the VM?
8. Download and install the putty application
9. What is a File Transfer Protocol (FTP)?
10. Download and install the Filezilla application
11. What is the flask app?
12. How to perform flask app deployment?
13. What is a postman application?
14. Introduction and installation of docker
15. How to perform a docker-based deployment?