**WEEK 10&11 ASSIGNMENT**

1. **Host a Ubuntu Virtual Machine using Oracle VM Virtual Box. (5 marks)**

My personal laptop is not supporting virtualization and hence I’m unable to install VirtualBox or Docker or other relevant software that use or require virtualization. So, for this exercise, I’m using AWS to create a Ubuntu EC2 instance. But since I can’t use this instance for installing Visual Studio code and other stuff, I’m cloning the initial template code into my personal machine, modify some things into the newly created repo, clone the code into the VM of AWS and run further steps there.

1. **Set up Visual Studio code on Ubuntu VM. (5 marks)**

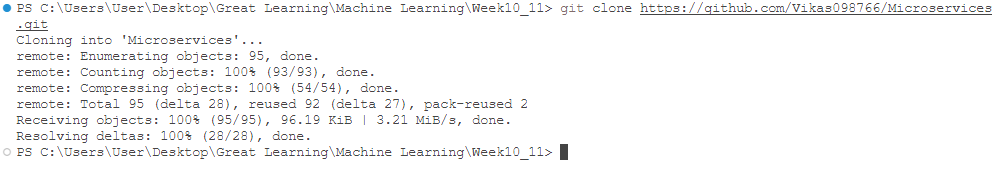
As explained previously, I can’t setup Visual Studio Code in a VM that is hosted in AWS and I am accessing via SSH using Putty.

1. **Setup Python. (5 marks)**

I’ve already installed python in my personal machine  


1. **Clone this Github repositoryhttps://github.com/Vikas098766/Microservices.git(1 mark)**

I’m cloning the repo mentioned here using the command  
git clone <https url>



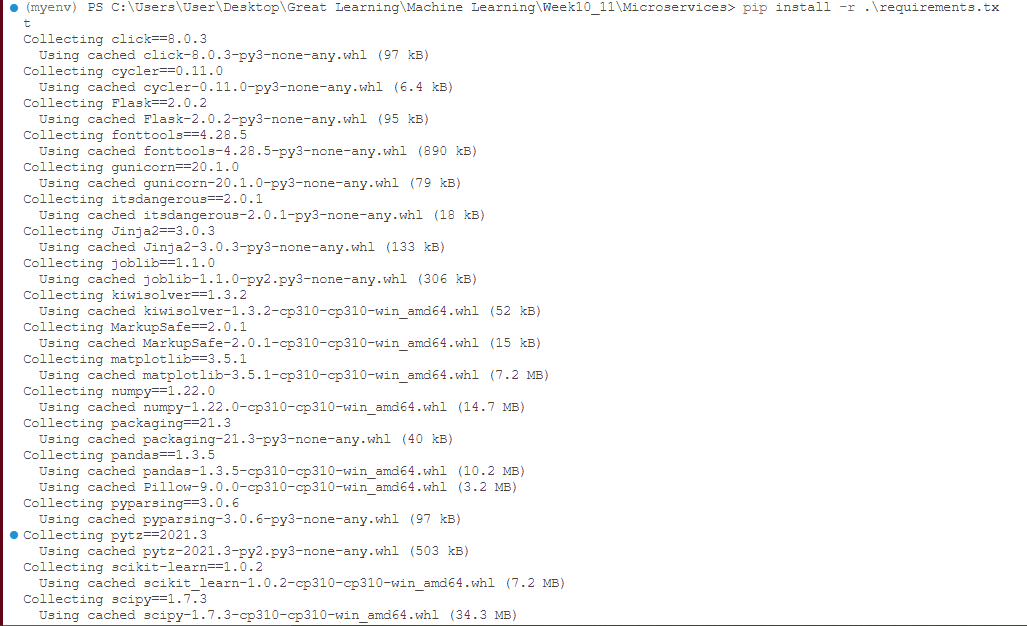
1. **Create a virtual environment (1 Mark)**

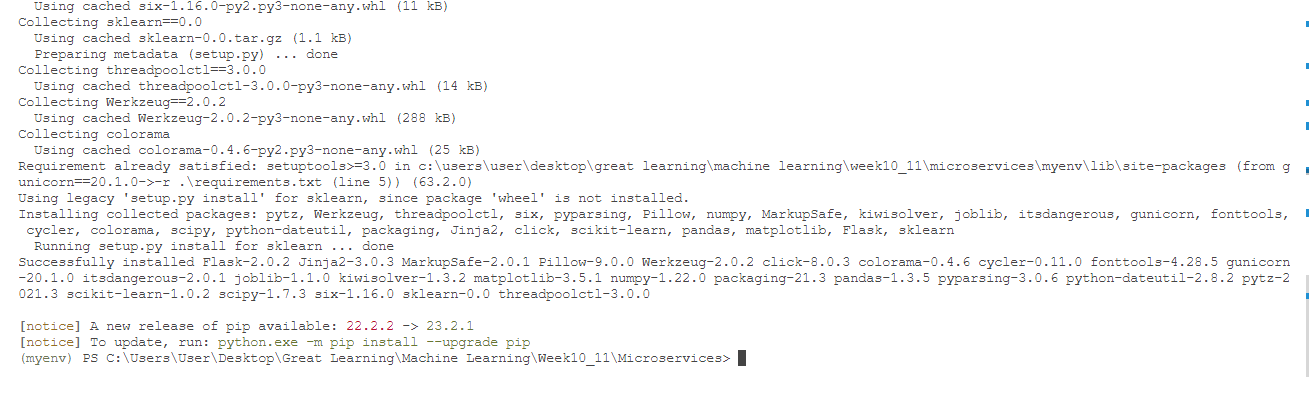
In order to create a virtual environment, I’m using the command below which will create a new virtual env called myenv. In order to activate it, use the other command  
python -m venv myenv  
myenv\Scripts\activate = Activates venv  
deactivate = Deactivates venv



1. **Install the requirements from requirements.txt file (1 Mark)**

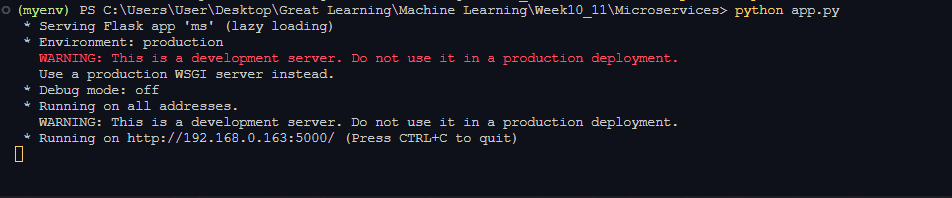
In order to install necessary requirements, use the below commands  
pip install -r requirements.txt

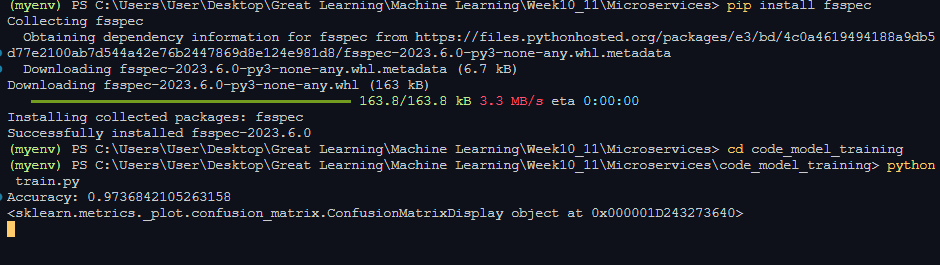


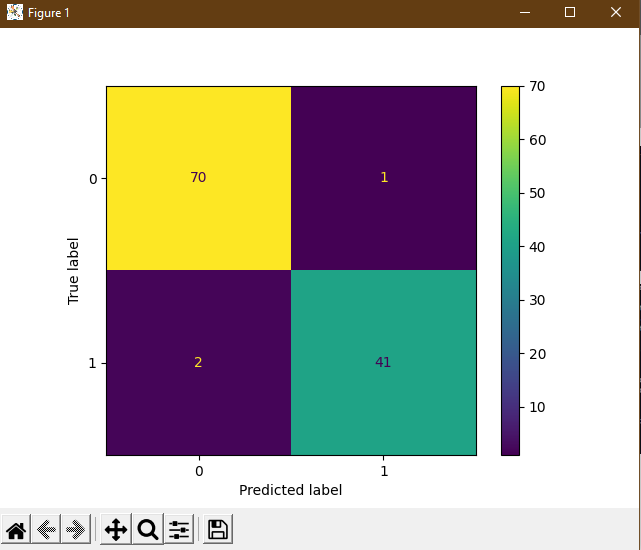


1. **Train and save the model (2 marks)**

In order to train and save the mode, first we have to run main.py and then go to code\_model\_training in the editor and run the model.py file preferably in a separate terminal.

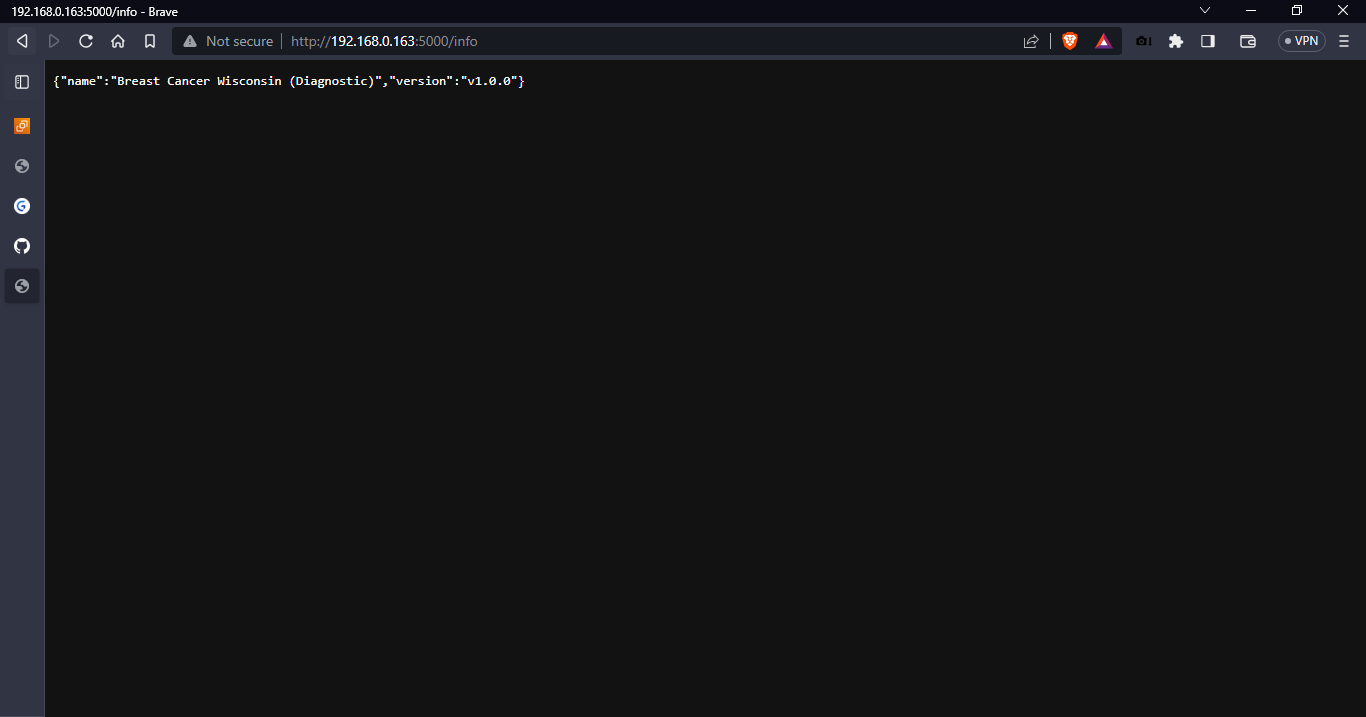




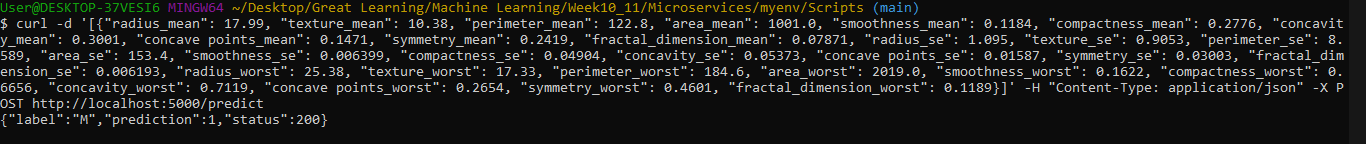


1. **Test the flask web application (5 marks)**

In order to test the application, go to any browser, type the url and check the response.



1. **Test the application and make predictions using the example calls available in the folder/tests.(5 marks)**

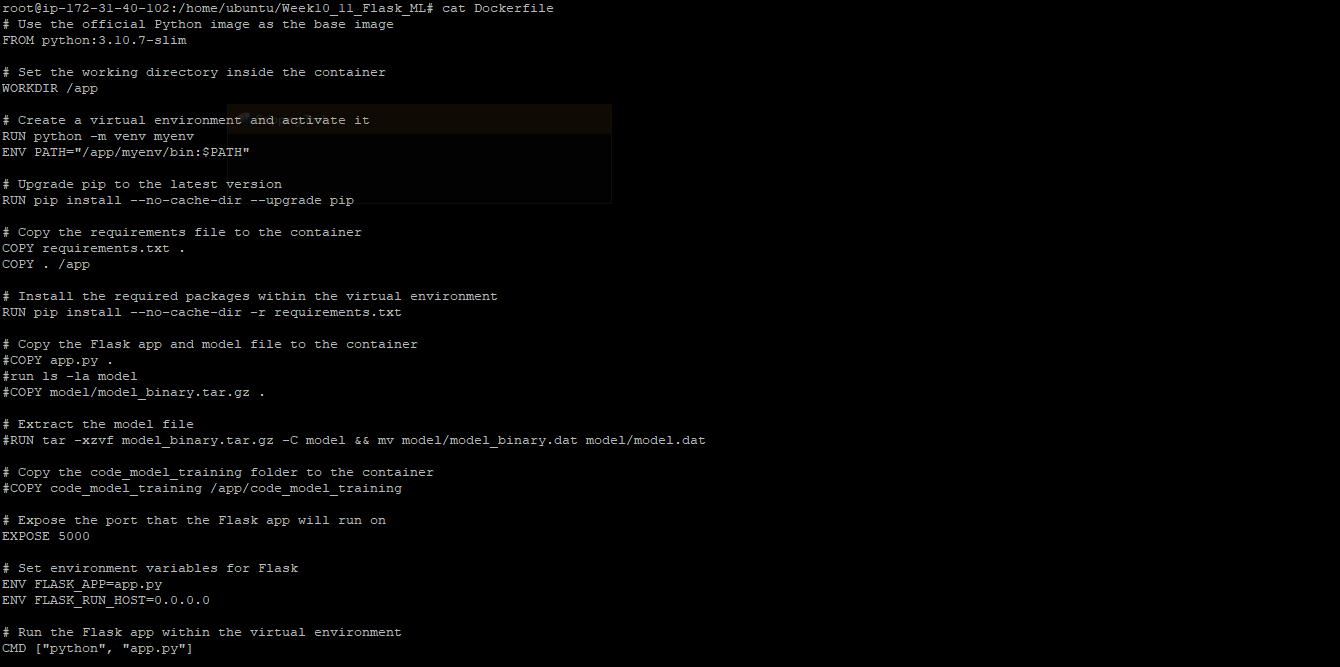






1. **Create a docker image containing everything needed to run the application.(10 marks)**

I created a base dockerfile and then went on modifying by doing trail and error by creating an container and checking whether the app is working or not.



1. **Run the containerized application as a prediction service and test it locally by passing some example calls and get the prediction. (10 marks)**

In order to test the application, we will have to build the image by using the below command and run the image using the next command  
docker build -t flask\_ml .

docker run -d -p 5000:5000 –name flask\_ml flask\_ml

Since I’m using an EC2 instance to host the application, instead of localhost or 0.0.0.0, I’ll use the machine IP in the url and test the commands for checking in browser

