Commands Used to deploy flask application in Ec2:

Step 1: Launch a Ec2 machine. Also, update the security group to have custom TCP rule on port 5000 for all ip's.

Step 2: ssh into ec2 machine

Step 3: Install Docker using following commands:
sudo amazon-linux-extras install docker

Step 4: Start the docker service using following command:
sudo service docker start

Step 5: update user permissions:
sudo usermod -a -G docker ec2-user
sudo chmod 666 /var/run/docker.sock

Step 6: Make templates and static directories as below:
mkdir templates
mkdir static

Step 7: copy all files from application code in local to the ec2 machine using scp:

scp -i akshay-personal.pem requirements.txt ec2-user@ec2-54-227-175-19.compute-1.amazonaws.com:/home/ec2-user
scp -i akshay-personal.pem Dockerfile ec2-user@ec2-54-227-175-19.compute-1.amazonaws.com:/home/ec2-user
scp -i akshay-personal.pem app.py ec2-user@ec2-54-227-175-19.compute-1.amazonaws.com:/home/ec2-user
scp -i akshay-personal.pem logic.py ec2-user@ec2-54-227-175-19.compute-1.amazonaws.com:/home/ec2-user
scp -i akshay-personal.pem data_preparation.py ec2-user@ec2-54-227-175-19.compute-1.amazonaws.com:/home/ec2-user
scp -i akshay-personal.pem age_prediction_model.pkl ec2-user@ec2-54-227-175-19.compute-1.amazonaws.com:/home/ec2-user
scp -i akshay-personal.pem gender_prediction_model.pkl ec2-user@ec2-54-227-175-19.compute-1.amazonaws.com:/home/ec2-user
scp -i akshay-personal.pem scenario1_test.csv ec2-user@ec2-54-227-175-19.compute-1.amazonaws.com:/home/ec2-user
scp -i akshay-personal.pem templates/index.html ec2-user@ec2-54-227-175-19.compute-1.amazonaws.com:/home/ec2-user/templates
scp -i akshay-personal.pem static/styles.css ec2-user@ec2-54-227-175-19.compute-1.amazonaws.com:/home/ec2-user/static

Your directory in EC2 should look like below:

```
[ec2-user@ip-172-31-18-185 ~]$ sudo service docker start
Redirecting to /bin/systemctl start docker.service
[ec2-user@ip-172-31-18-185 ~]$ sudo usermod -a -G docker ec2-user
[ec2-user@ip-172-31-18-185 ~]$ sudo chmod 666 /var/run/docker.sock
[ec2-user@ip-172-31-18-185 ~]$ mkdir templates
[ec2-user@ip-172-31-18-185 ~]$ mkdir static
[ec2-user@ip-172-31-18-185 ~]$ ls
static  templates
[ec2-user@ip-172-31-18-185 ~]$ ls
Dockerfile  age_prediction_model.pkl  app.py  data_preparation.py  gender_prediction_model.pkl  logic.py  requirements.txt  scenario1_test.csv  static  templates
[ec2-user@ip-172-31-18-185 ~]$ ls /templates
ls: cannot access /templates: No such file or directory
[ec2-user@ip-172-31-18-185 ~]$ ls templates/
index.html
[ec2-user@ip-172-31-18-185 ~]$ ls static/
styles.css
[ec2-user@ip-172-31-18-185 ~]$
```

Step 8: Build docker image using below command:
docker build -t prediction .

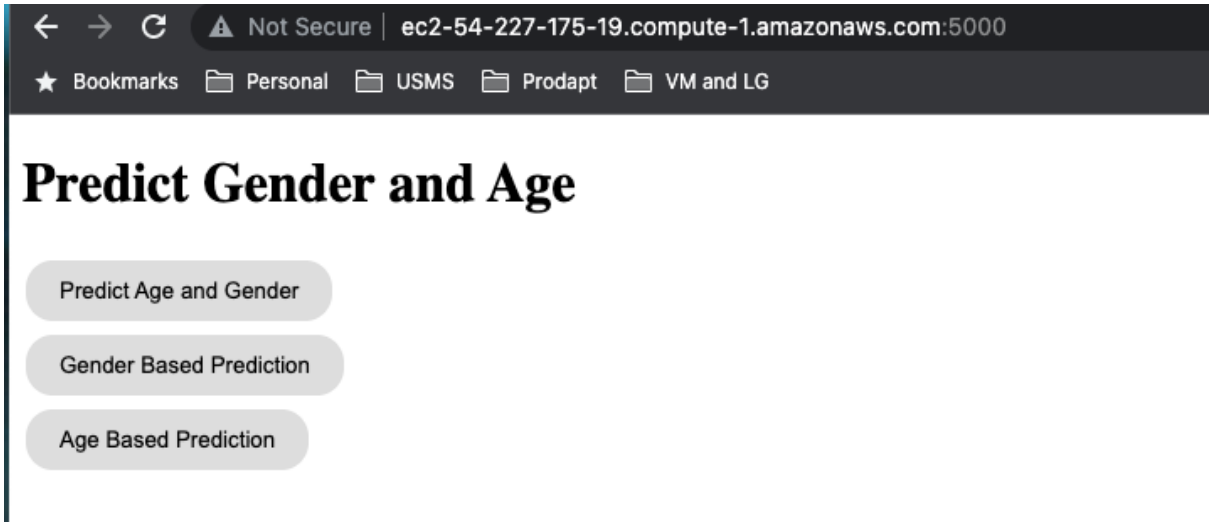
Step 9: Run docker image using below command:
docker run -p 5000:5000 prediction

It should look like this once started:

```
[ec2-user@ip-172-31-18-185 ~]$ docker run -p 5000:5000 prediction
 * Serving Flask app "app" (lazy loading)
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployment.
   Use a production WSGI server instead.
 * Debug mode: on
 * Running on http://0.0.0.0:5000/ (Press CTRL+C to quit)
 * Restarting with stat
 * Debugger is active!
 * Debugger PIN: 188-218-095
```

Note: You see that screenshot says application is running on http://0.0.0.0:5000/ which is local host. However, to run from our machine we need to provide ec2 public dns to run our application in this case it would be:
ec2-54-227-175-19.compute-1.amazonaws.com:5000

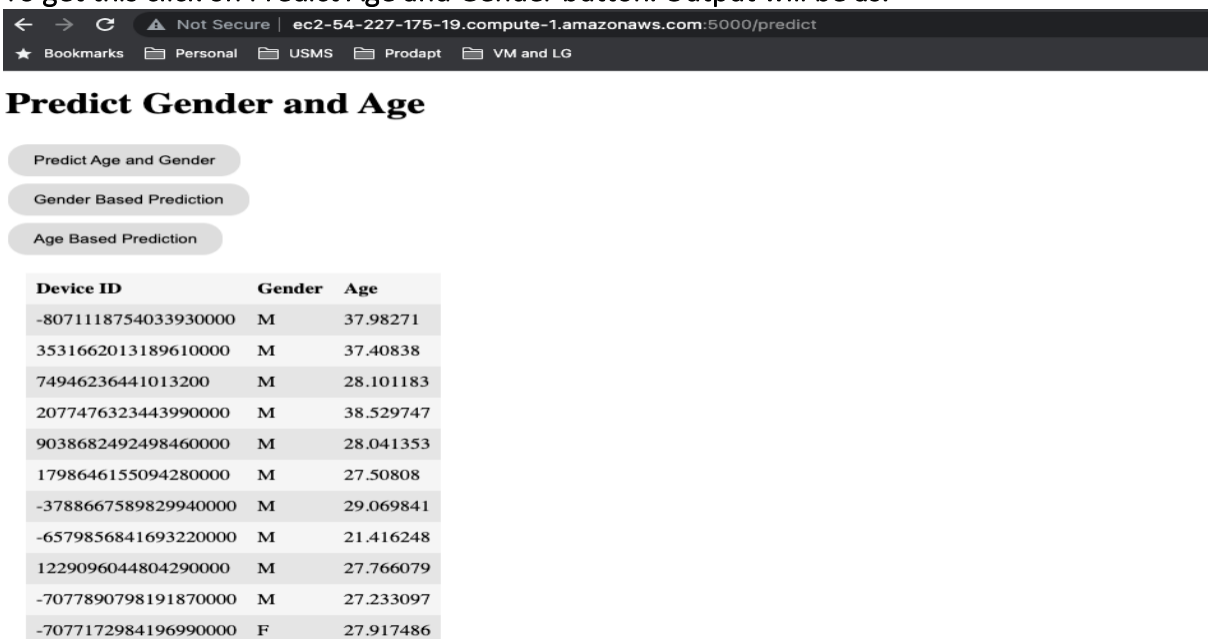
Step 10: Open the above dns and host in browser, API should be ready to execute as shown below:



Final Output for predicting the campaigns

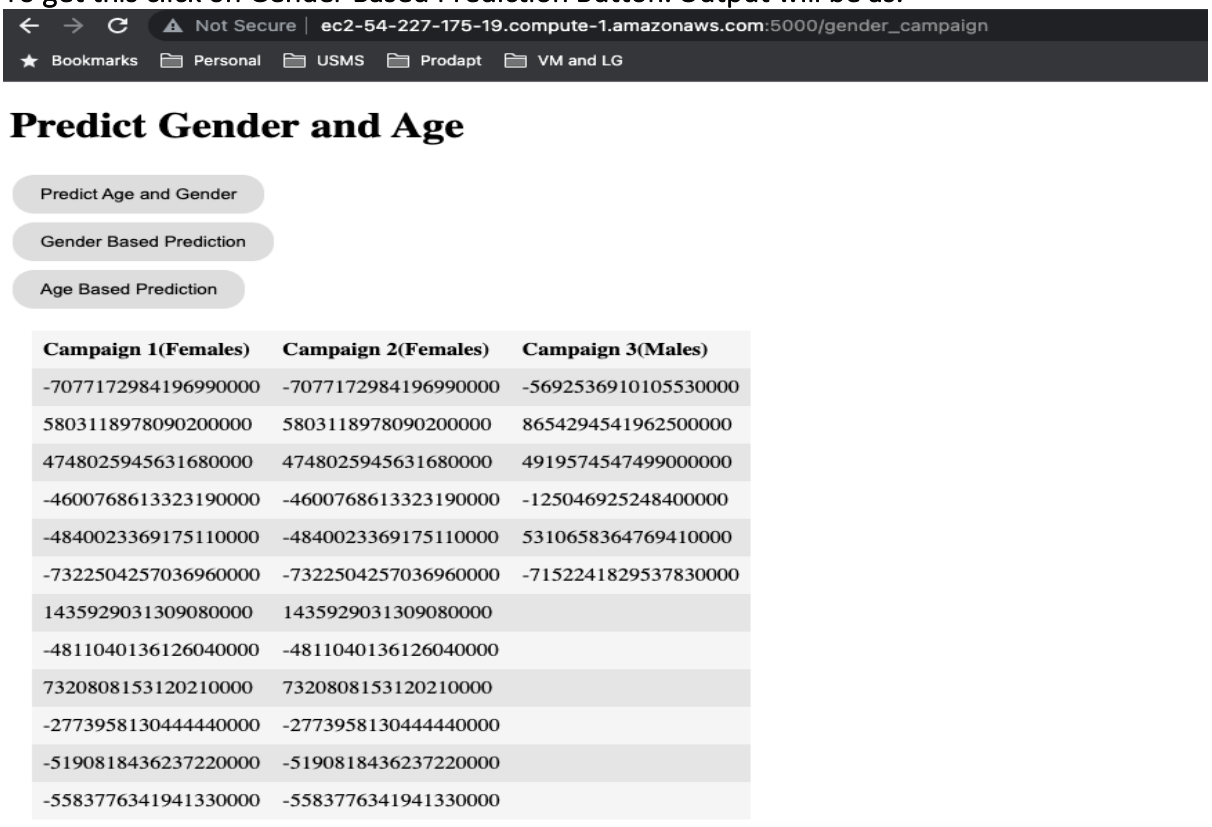
1. Age and Gender prediction for Sample of 50 test data

To get this click on Predict Age and Gender button. Output will be as:



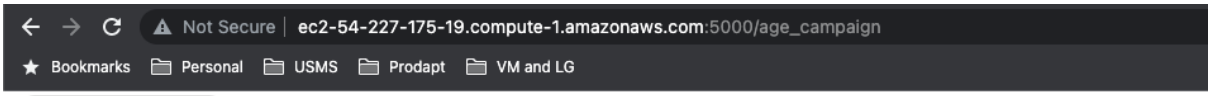
Device ID	Gender	Age
-8071118754033930000	M	37.98271
3531662013189610000	M	37.40838
74946236441013200	M	28.101183
2077476323443990000	M	38.529747
9038682492498460000	M	28.041353
1798646155094280000	M	27.50808
-3788667589829940000	M	29.069841
-6579856841693220000	M	21.416248
1229096044804290000	M	27.766079
-7077890798191870000	M	27.233097
-7077172984196990000	F	27.917486

2. Gender based campaign Prediction:
   To get this click on Gender Based Prediction Button. Output will be as:



Campaign 1(Females)	Campaign 2(Females)	Campaign 3(Males)
-7077172984196990000	-7077172984196990000	-5692536910105530000
5803118978090200000	5803118978090200000	8654294541962500000
4748025945631680000	4748025945631680000	4919574547499000000
-4600768613323190000	-4600768613323190000	-125046925248400000
-4840023369175110000	-4840023369175110000	5310658364769410000
-7322504257036960000	-7322504257036960000	-7152241829537830000
1435929031309080000	1435929031309080000	
-4811040136126040000	-4811040136126040000	
7320808153120210000	7320808153120210000	
-2773958130444440000	-2773958130444440000	
-5190818436237220000	-5190818436237220000	
-5583776341941330000	-5583776341941330000	

3. Age based campaign prediction:
   To get this click on Age Based Prediction button in UI. Output will be as:



Age Based Prediction

Campaign 4:Age Bucket(0-24)	Campaign 5:Age Bucket(24-32)	Campaign 6:Age Bucket(32+)
-6579856841693220000	74946236441013200	-8071118754033930000
2864448110369580000	9038682492498460000	3531662013189610000
5134205187608460000	1798646155094280000	2077476323443990000
7700833807167330000	-3788667589829940000	4748025945631680000
-6099309864413210000	1229096044804290000	-6096615357726460000
-4840023369175110000	-7077890798191870000	-7322504257036960000
-125046925248400000	-7077172984196990000	928040807049483000
-4811040136126040000	8470056818325450000	5850850356474440000
9085370879774260000	-5692536910105530000	-3374882087563500000
6087658991573520000	5803118978090200000	-1407669367877110000
	8654294541962500000	-8060821165622630000
	-4600768613323190000	-7152241829537830000
	6428543206524510000	5481079504430610000
	4110551735503610000	
	4919574547499000000	
	1435929031309080000	
	-7381976878520170000	
	7320808153120210000	