Date: 12.09.2025



Copyright @ 2024 PibyThree.com All Rights Reserved

Containerized Linear Regression Model for Shoe Size Prediction

Trained in Jupyter, Stored in S3, and Deployed via ECS Fargate with Streamlit UI

Contents

[Trained in Jupyter, Stored in S3, and Deployed via ECS Fargate with Streamlit UI 1](#_Toc208591696)

[1. Overview 2](#_Toc208591697)

[2. Dataset 2](#_Toc208591698)

[3. Model Training 2](#_Toc208591699)

[4. Model Storage – AWS S3 2](#_Toc208591700)

[5. Web Application – Streamlit 2](#_Toc208591701)

[6. Docker Image 2](#_Toc208591702)

[7. Deployment on AWS ECS Fargate 3](#_Toc208591703)

[8. Accessing the App 3](#_Toc208591704)

## 1. Overview

Built a web application that predicts shoe size from a person’s height and gender using a trained linear regression model, fully deployed on AWS.

Here is the github repo: [aryan-madhavi/S3-Model-and-Containerized-App](https://github.com/aryan-madhavi/S3-Model-and-Containerized-App)

## 2. Dataset

**Sample Dataset:**

|  |  |  |
| --- | --- | --- |
| Height | Gender | Shoe Size |
| 180 | M | 12 |
| 165 | F | 6 |

* + - Gender encoded as: M → 1, F → 0
* Final model trained on only Height and Gender
* Actual Dataset: [Dataset](https://figshare.com/articles/dataset/Study_of_the_Relationship_between_Height_Shoe_Size_and_Gender/1541176)

## 3. Model Training

Performed in a Google Collab Research notebook: [model.ipynb](https://colab.research.google.com/drive/1L5Rd4colihBGibHEU9hLHv99L8-cpODz?usp=sharing)

Used it to download the model.pkl file as well after the model was trained

## 4. Model Storage – AWS S3

* Created a Bucket: `am-regression-model`
* Uploaded the `model.pkl` file in the bucket

## 5. Web Application – Streamlit

**Features:**

* User inputs height and gender.
* Button triggers prediction.
* Displays predicted shoe size.

**Caching:**

* Uses @st.cache\_resource(ttl=14400) to cache model for 4 hours.

## 6. Docker Image

The dockerfile has been used to build, tag and push the image to DockerHub: docker.io/aryanfafo/shoe-size-predictor:v1

## 7. Deployment on AWS ECS Fargate

**Steps:**

1. **Created IAM Policy:**
   * Created a `ecsTaskS3ReadPolicy` and ` ecsTaskS3ReadRole` for `ECS` Service to allow S3 read

ecsTaskS3ReadPolicy

{

"Version": "2012-10-17",

"Statement": [

{

"Effect": "Allow",

"Action": [

"s3:GetObject"

],

"Resource": [

"arn:aws:s3:::am-regression-model/\*"

]

}

]

}

1. **Created ECS Cluster:**
   * Created a cluster name `shoe-size-predictor-cluster` with AWS Fargate as infra
2. **Created Task Definition:**
   * Created a cluster name `shoe-size-predictor-task` with AWS Fargate as infra
   * Set CPU to 0.5 vCPU and 1GB Memory
   * Assigned the `ecsTaskS3ReadRole` as Task Role
   * Map port 8501 container to 8501 host
   * Other default settings
3. **Start the Task:**
   * Navigate to the created cluster
   * Go to the Task subtab and run new task
   * Select `shoe-size-predictor-task`
   * Ensure the security group allows port 8051
   * Rest default configuration

## 8. Accessing the App

* Task gets a public IP: http://<public-ip>:8501
* You may also use the `nslookup public-ip` to get the FQDN