**DATA SCIENCE INTERN at DATA GLACIER**

## **Week 5: Cloud and API deployment**

**Name: Shubh Goyal**

**Batch Code: LISUM20**

**Date: 5th May 2023**

**Submitted to: Data Glacier**

**Table Of Content**

1. **Introduction**
2. **Building Machine Learning Model**
3. **Turning Model into Flask Framework**

* **App.py**
* **Index.html**
* **Running Procedure**

# **INTRODUCTION**

In this project, we are going to deploying machine learning model using the Flask Framework. Diagram

Description automatically generated

we will focus on both: building a machine learning model, then create an API for the model, using Flask, the Python micro-framework for building web applications. This API allows us to utilize predictive capabilities through HTTP requests.

# **Building Machine Learning Model**

A screenshot of a computer program

Description automatically generated with medium confidence

# **Turning** **Model into Flask Framework**

First, we create a folder for this project called YouTube Spam Filtering, this is the directory tree inside the folder. We will explain each file “ML MODEL DEPLOYMENT”

**APPLICATION FOLDER FILE DIRECTORY**

* App.py
* Templates/
  + Index.html
* Model.pkl
* hiring.csv

## **App.py**

The app.py file contains the main code that will be executed by the Python interpreter to run the Flask web application, it included the ML code for classifying SD.

A screen shot of a computer

Description automatically generated with medium confidence

## **INDEX.HTML**

A screenshot of a computer screen

Description automatically generated with medium confidence

## **RUNNING PROCEDURE**

Once we have done all of the above, we can start running the API by either double click app.py.

When you navigate to “ <http://127.0.0.1:5000>” You will see the below page

A screenshot of a computer

Description automatically generated with medium confidence

Enter the information:

A screenshot of a computer

Description automatically generated with medium confidence

After you enter the data, click on “Predict” Button

Result:  
A picture containing text, screenshot, font, line

Description automatically generated

## **Model deployment using Heroku**

We’re ready to start our Heroku deployment now that our model has been trained, the machine

learning pipeline has been set up, and the application has been tested locally. There are a few

ways to upload the application source code onto Heroku. The easiest way is to link a GitHub

repository to your Heroku account.

**Requirement.txt**

It is a text file containing the python packages required to execute the application.

* **Steps for Model Deployment Using Heroku**

Once we uploaded files to the GitHub repository, we are now ready to start deployment on

Heroku. Follow the steps below:

1. **After signing up on heroku.com then click on Create new app**

**A screenshot of a computer

Description automatically generated with medium confidence**

1. **Enter App name and region.**

**A screenshot of a computer

Description automatically generated with medium confidence**

1. **Connect to GitHub repository where code is I uploaded.**

**A screenshot of a computer

Description automatically generated with medium confidence**

1. **Deploy Branch**

**A screenshot of a computer

Description automatically generated with medium confidence**

1. **After waiting 5 to 15 minutes our application is Ready**

**A screenshot of a computer

Description automatically generated with medium confidence**

**The app is published at:**

https://ml-salary-predictions.herokuapp.com/