

Name: Pranav Ingale

Batch code: LISUM09

Submission date: 28-MAY-2022

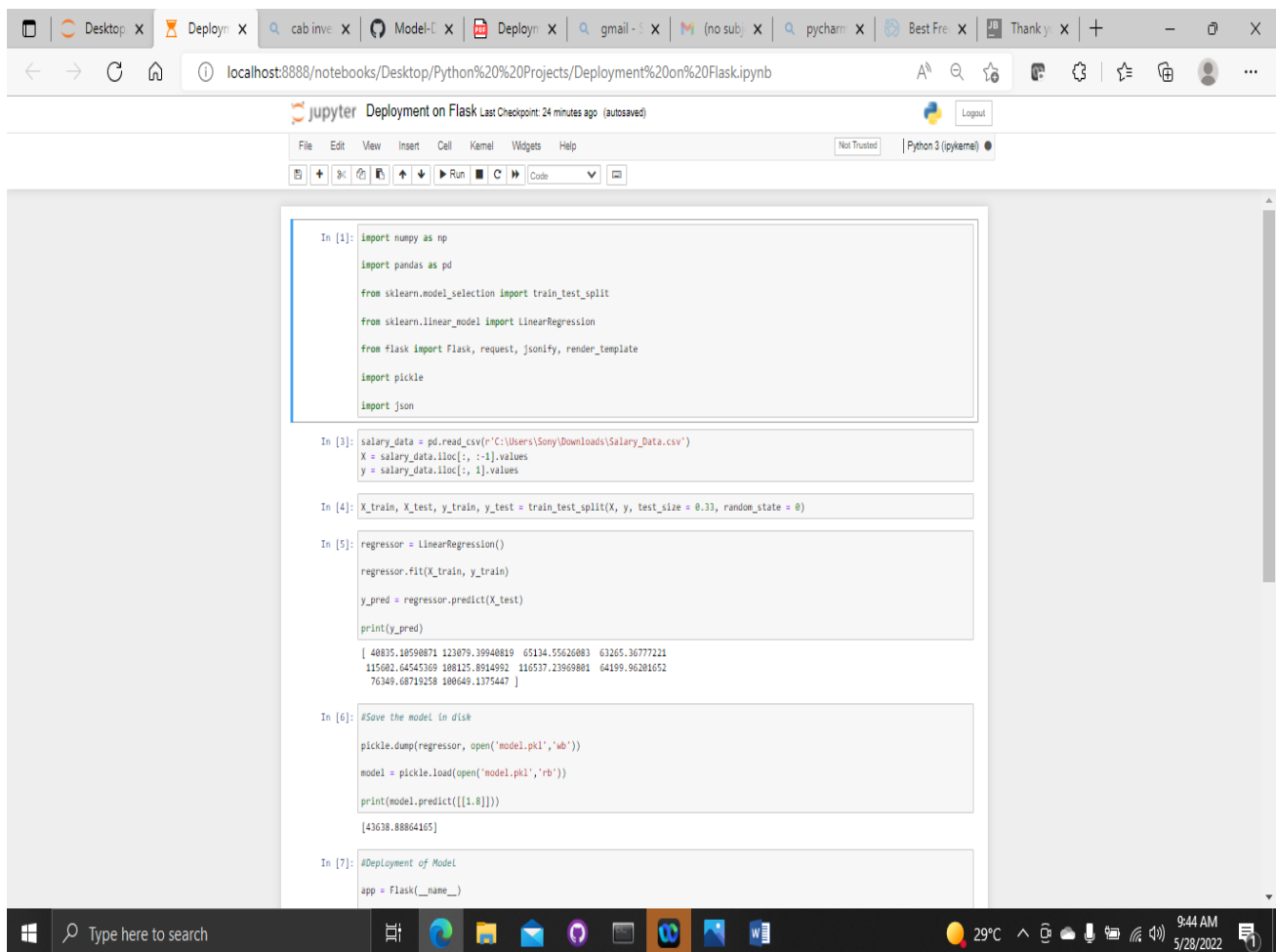
Submitted to: Data Glacier

Deployment on Flask

Step 1:

Develop ML model:

**Predict the salary of an employee based on experience using
Linear Regression Model.**



The screenshot displays a Jupyter Notebook titled "Deployment on Flask" running on a local host. The notebook contains several code cells for data loading, model training, and deployment. The output of the training cell shows a list of predicted salaries for a set of input features. The deployment cell initializes a Flask application.

```
In [1]: import numpy as np
import pandas as pd

from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from flask import Flask, request, jsonify, render_template

import pickle
import json

In [3]: salary_data = pd.read_csv(r"C:\Users\Sony\Downloads\Salary_Data.csv")
X = salary_data.iloc[:, :-1].values
y = salary_data.iloc[:, 1].values

In [4]: X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.33, random_state = 0)

In [5]: regressor = LinearRegression()
regressor.fit(X_train, y_train)
y_pred = regressor.predict(X_test)
print(y_pred)

[ 40835.10590871 123079.39940819  65134.55626083  63265.36777221
 115082.64545369 188125.8914992  116537.23969881  64199.96281652
 76349.68719258 180649.1375447 ]

In [6]: #Save the model in disk
pickle.dump(regressor, open('model.pkl', 'wb'))
model = pickle.load(open('model.pkl', 'rb'))
print(model.predict([[1.8]]))

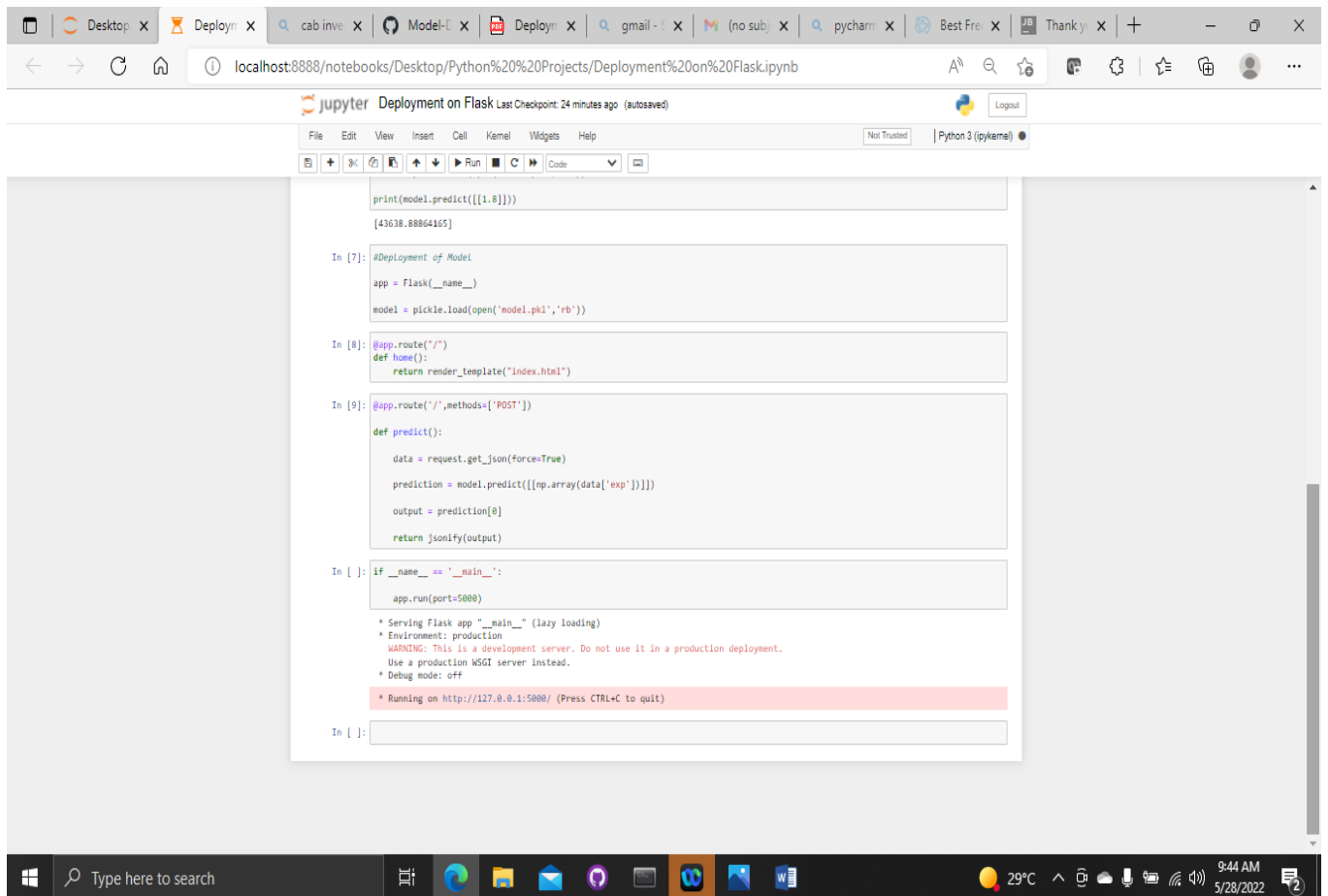
[43638.88864165]

In [7]: #Deployment of Model
app = Flask(__name__)
```

Step 2: Saving the trained model to the disk using the pickle library

Step 3: Deployment of Model

Step 4: Creating the Web App by typing the URL in the browser.



The screenshot shows a Jupyter Notebook titled "Deployment on Flask" running in a browser at localhost:8888. The notebook contains several code cells. The first cell shows a prediction result: `print(model.predict([[1.8]]))` resulting in `[43638.88864165]`. The second cell, labeled "In [7]:", shows the deployment setup: `app = Flask(__name__)`, `model = pickle.load(open('model.pkl', 'rb'))`. The third cell, labeled "In [8]:", shows a route for the home page: `@app.route('/') def home(): return render_template("index.html")`. The fourth cell, labeled "In [9]:", shows a route for predictions: `@app.route('/', methods=['POST']) def predict(): data = request.get_json(force=True) prediction = model.predict([np.array(data['exp'])]) output = prediction[0] return jsonify(output)`. The fifth cell, labeled "In []:", shows the main execution block: `if __name__ == '__main__': app.run(port=5000)`. Below the code, there are status messages: `* Serving Flask app "__main__" (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: off`, and `* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)`. The browser's address bar shows the URL `localhost:8888/notebooks/Desktop/Python%20%20Projects/Deployment%20on%20Flask.ipynb`. The Windows taskbar at the bottom shows the time as 9:44 AM on 5/28/2022.

```
print(model.predict([[1.8]]))
[43638.88864165]

In [7]: #Deployment of Model
app = Flask(__name__)
model = pickle.load(open('model.pkl', 'rb'))

In [8]: @app.route('/')
def home():
    return render_template("index.html")

In [9]: @app.route('/', methods=['POST'])
def predict():
    data = request.get_json(force=True)
    prediction = model.predict([np.array(data['exp'])])
    output = prediction[0]
    return jsonify(output)

In [ ]: if __name__ == '__main__':
    app.run(port=5000)

* Serving Flask app "__main__" (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: off
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)

In [ ]:
```