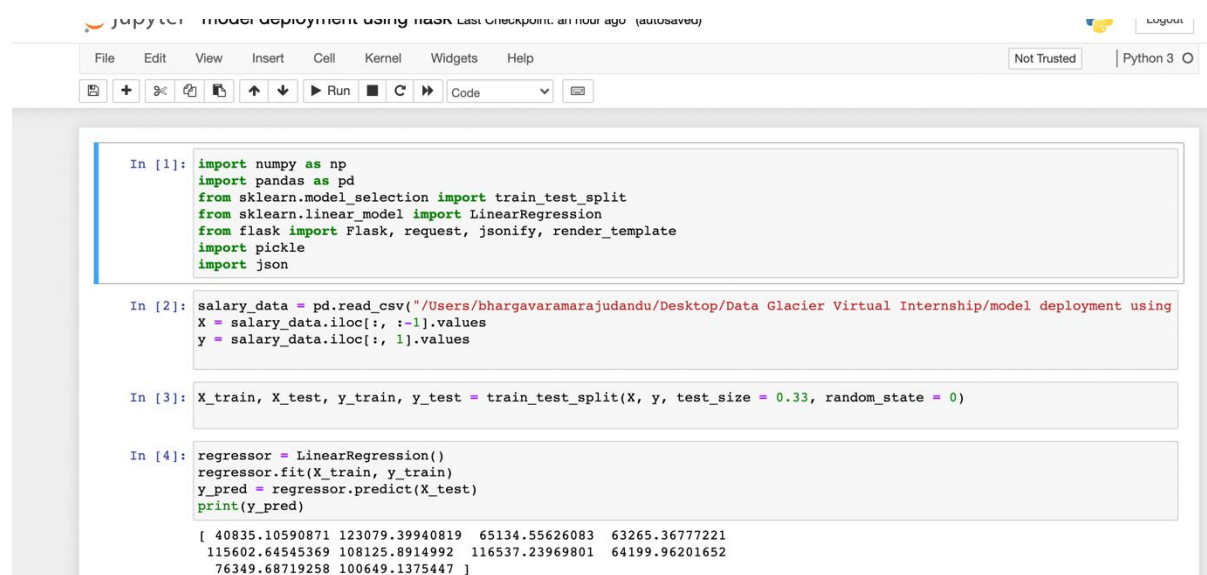


Name: Bhargava Rama Raju Dandu.
Batch code: LISUM01.
Submission Date: 10-June-2021.
Submitted to: Data Glacier.

Deployment on Flask

Step 1: Develop a Machine learning model.

Predict the salary of an employee using Linear Regression Model.



The image shows a Jupyter Notebook interface with the title "model deployment using flask". The notebook contains four code cells. The first cell imports necessary libraries: numpy, pandas, sklearn's train_test_split and LinearRegression, flask's request, jsonify, and render_template, and pickle and json. The second cell reads a CSV file from the desktop, loads it into a pandas DataFrame, and separates the features (X) and target variable (y). The third cell splits the data into training and testing sets using train_test_split with a test_size of 0.33. The fourth cell creates a LinearRegression model, fits it to the training data, and prints the predictions for the test data. The output of the fourth cell is a 5x2 array of predicted salaries.

```
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 [2]: salary_data = pd.read_csv("/Users/bhargavaramarajudandu/Desktop/Data Glacier Virtual Internship/model deployment using
X = salary_data.iloc[:, :-1].values
y = salary_data.iloc[:, 1].values

In [3]: X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.33, random_state = 0)

In [4]: 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
 115602.64545369 108125.8914992  116537.23969801  64199.96201652
 76349.68719258 100649.1375447 ]
```

Step 2: Save the trained model by using pickle library.



The image shows a Jupyter Notebook interface with the title "model deployment using flask". The notebook contains three code cells. The first cell is a comment: "#Save the model in disk". The second cell uses pickle.dump to save the trained LinearRegression model to a file named "model.pkl" in binary mode. The third cell loads the model back from "model.pkl" using pickle.load and prints the prediction for a new input value of 1.8. The output of the third cell is the predicted salary: 43638.88864165.

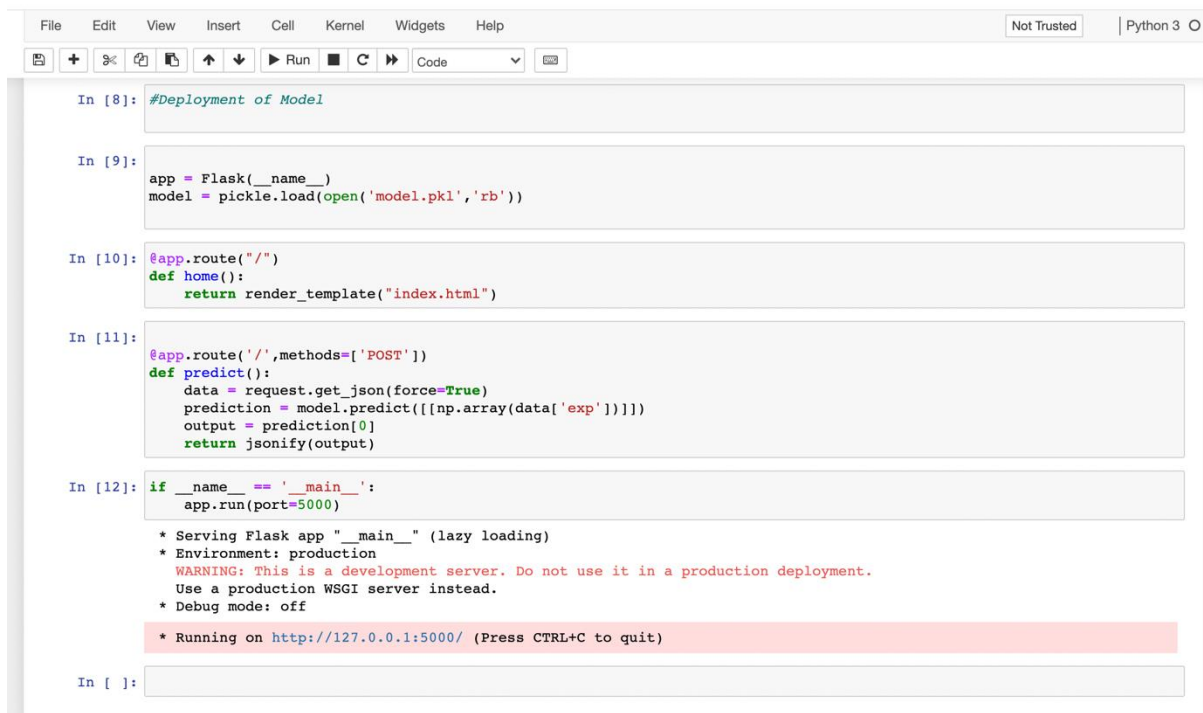
```
In [5]: #Save the model in disk

In [6]: pickle.dump(regressor, open('model.pkl', 'wb'))

In [7]: model = pickle.load(open('model.pkl', 'rb'))
print(model.predict([[1.8]]))

[43638.88864165]
```

Step 3: Deployment of the model



```
In [8]: #Deployment of Model

In [9]: app = Flask(__name__)
        model = pickle.load(open('model.pkl', 'rb'))

In [10]: @app.route("/")
        def home():
            return render_template("index.html")

In [11]: @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 [12]: 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

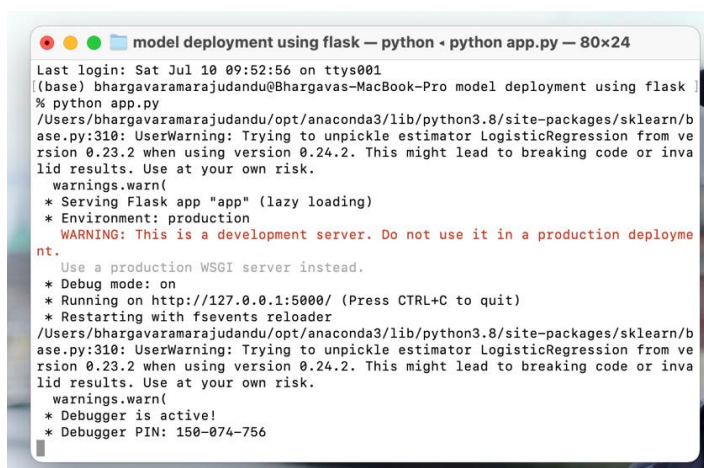
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)

In [ ]:
```

- Created the instance of the *Flask()* and loaded the model.
- Bounded “/” with the method *predict()* in which predict method gets the data from the json passed by the requestor.
- *model.predict()* method takes input from the json and converts it into 2D *numpy array* the results are stored into the variable named *output*.
- Return this variable after converting it into the json object using flask's *jsonify()* method.
- Run our server by following above code section and using port 5000.

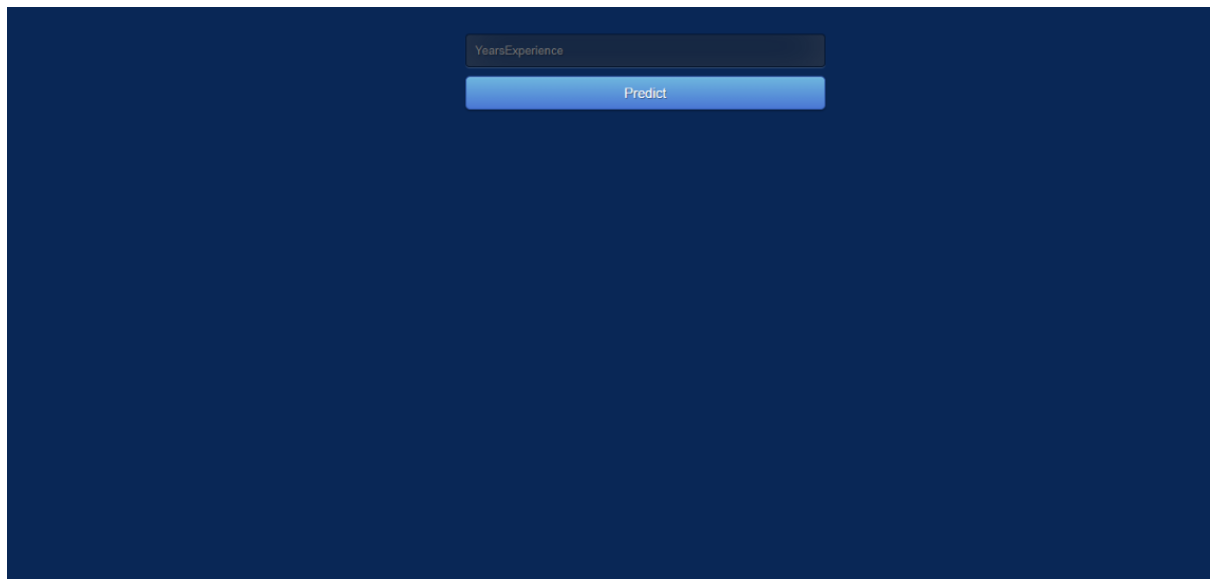
Step 4:

Checking python app.py file in terminal



```
model deployment using flask — python • python app.py — 80x24
Last login: Sat Jul 10 09:52:56 on ttys001
[(base) bhargavaramarajudandu@Bhargavas-MacBook-Pro model deployment using flask]
% python app.py
/Users/bhargavaramarajudandu/opt/anaconda3/lib/python3.8/site-packages/sklearn/b
ase.py:310: UserWarning: Trying to unpickle estimator LogisticRegression from ve
rsion 0.23.2 when using version 0.24.2. This might lead to breaking code or inva
lid results. Use at your own risk.
  warnings.warn(
* Serving Flask app "app" (lazy loading)
* Environment: production
  WARNING: This is a development server. Do not use it in a production deployme
nt.
  Use a production WSGI server instead.
* Debug mode: on
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
* Restarting with fsevents reloader
/Users/bhargavaramarajudandu/opt/anaconda3/lib/python3.8/site-packages/sklearn/b
ase.py:310: UserWarning: Trying to unpickle estimator LogisticRegression from ve
rsion 0.23.2 when using version 0.24.2. This might lead to breaking code or inva
lid results. Use at your own risk.
  warnings.warn(
* Debugger is active!
* Debugger PIN: 150-074-756
```

Step 5: Creating the web app using the using in browser



The image shows a web application interface. It has a dark blue background. At the top center, there is a text input field with the placeholder text "YearsExperience". Below the input field is a blue button with the text "Predict".

THANK YOU