

Name: [Wafa Faisal](#)

Batch Name: [LISUM01](#)

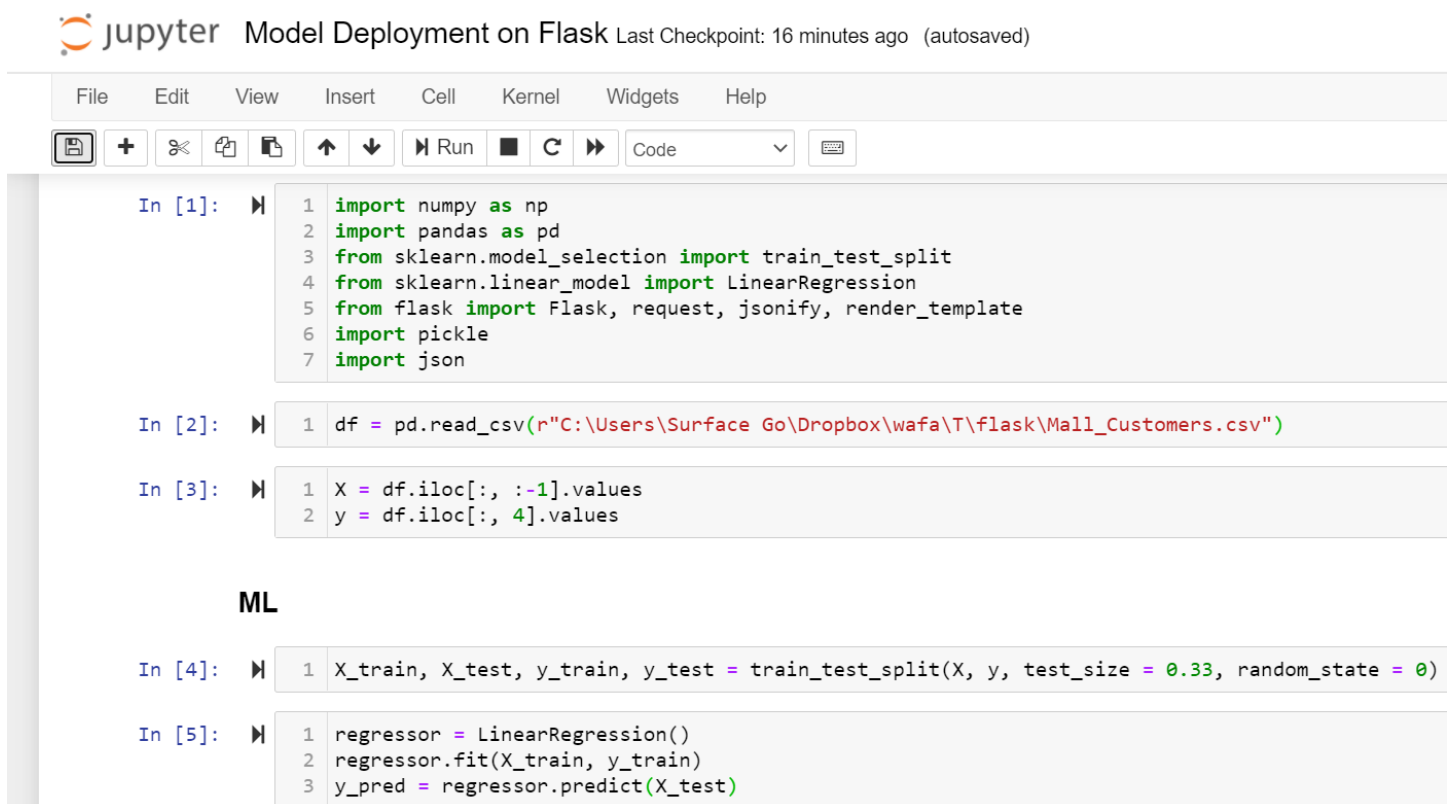
Submission date: [5-July-2021](#)

Submission to: [Data Glacier](#)

## Deployment Model on Flask

Step 1:

Develop ML model for mall customers datasets using Linear Regression Model



The image shows a Jupyter Notebook interface with the title "Model Deployment on Flask" and a status bar indicating "Last Checkpoint: 16 minutes ago (autosaved)". The interface includes a menu bar (File, Edit, View, Insert, Cell, Kernel, Widgets, Help) and a toolbar with icons for saving, adding cells, undo, redo, and running code. The notebook contains five code cells:

```
In [1]: 1 import numpy as np
        2 import pandas as pd
        3 from sklearn.model_selection import train_test_split
        4 from sklearn.linear_model import LinearRegression
        5 from flask import Flask, request, jsonify, render_template
        6 import pickle
        7 import json

In [2]: 1 df = pd.read_csv(r"C:\Users\Surface Go\Dropbox\wafa\T\flask\Mall_Customers.csv")

In [3]: 1 X = df.iloc[:, :-1].values
        2 y = df.iloc[:, 4].values

ML

In [4]: 1 X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.33, random_state = 0)

In [5]: 1 regressor = LinearRegression()
        2 regressor.fit(X_train, y_train)
        3 y_pred = regressor.predict(X_test)
```

Step 2: save trained model

```
In [5]: 1 regressor = LinearRegression()
        2 regressor.fit(X_train, y_train)
        3 y_pred = regressor.predict(X_test)
```

```
In [6]: 1 pickle.dump(regressor, open('model.pkl', 'wb'))
```

## Step 3: Model deployment

```
In [8]: 1 app = Flask(__name__)
        2 model = pickle.load(open('model.pkl', 'rb'))
```

```
In [9]: 1 @app.route("/")
        2 def home():
        3     return render_template("index.html")
```

```
In [10]: 1
        2 @app.route('/', methods=['POST'])
        3 def predict():
        4     data = request.get_json(force=True)
        5     prediction = model.predict([[np.array(data['exp'])]])
        6     output = prediction[0]
        7     return jsonify(output)
```

```
In [*]: 1 if __name__ == '__main__':
        2     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)
```

## Step 4: python app in CMD

```
(base) C:\Users\Surface Go\Dropbox\wafa\T\flask>python app.py
* 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
* Restarting with windowsapi reloader
* Debugger is active!
* Debugger PIN: 303-442-999
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```

Thank you