

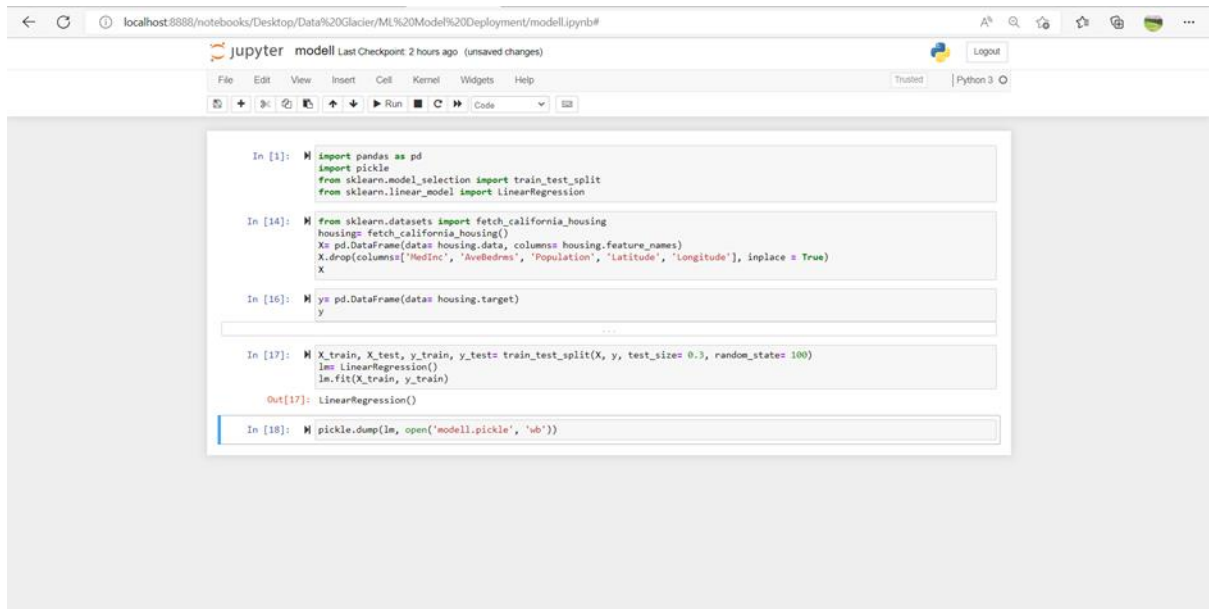
Name: Andrew Kojo Mensah-Onumah

Batch code: LISUM10

Submission date: 29/06/2022

Submitted to: Data Glacier

Week 4: Deployment on Flask



The screenshot shows a Jupyter Notebook interface with the title 'modell'. The notebook is running on a local host at 8888. The code in the notebook is as follows:

```
In [1]: import pandas as pd
import pickle
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression

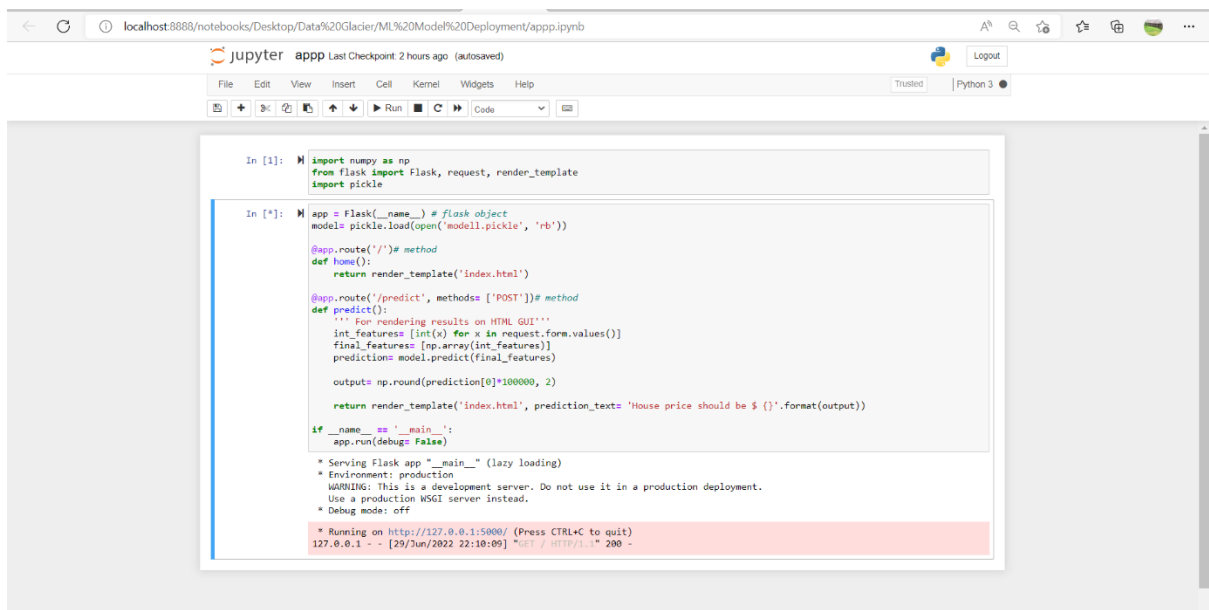
In [14]: from sklearn.datasets import fetch_california_housing
housing = fetch_california_housing()
X = pd.DataFrame(data=housing.data, columns=housing.feature_names)
X.drop(columns=['MedInc', 'AveBedrms', 'Population', 'Latitude', 'Longitude'], inplace=True)
X

In [16]: y = pd.DataFrame(data=housing.target)
y

In [17]: X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=100)
lm = LinearRegression()
lm.fit(X_train, y_train)

Out[17]: LinearRegression()

In [18]: pickle.dump(lm, open('modell.pickle', 'wb'))
```



The screenshot shows a Jupyter Notebook interface with the title 'app'. The notebook is running on a local host at 8888. The code in the notebook is as follows:

```
In [1]: import numpy as np
from flask import Flask, request, render_template
import pickle

In [*]: app = Flask(__name__) # flask object
model = pickle.load(open('modell.pickle', 'rb'))

@app.route('/') # method
def home():
    return render_template('index.html')

@app.route('/predict', methods=['POST']) # method
def predict():
    ''' For rendering results on HTML GUI'''
    int_features = [int(x) for x in request.form.values()]
    final_features = np.array(int_features)
    predictions = model.predict(final_features)

    output = np.round(predictions[0]*100000, 2)

    return render_template('index.html', prediction_text= 'House price should be $ {}'.format(output))

if __name__ == '__main__':
    app.run(debug=False)

* 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)
127.0.0.1 - - [29/Jun/2022 22:10:09] "GET / HTTP/1.1" 200 -
```

