



**Data Glacier**

Your Deep Learning Partner

## ***VIRTUAL DATA INTERNSHIP***

NAME: ONIFADE MICHAEL

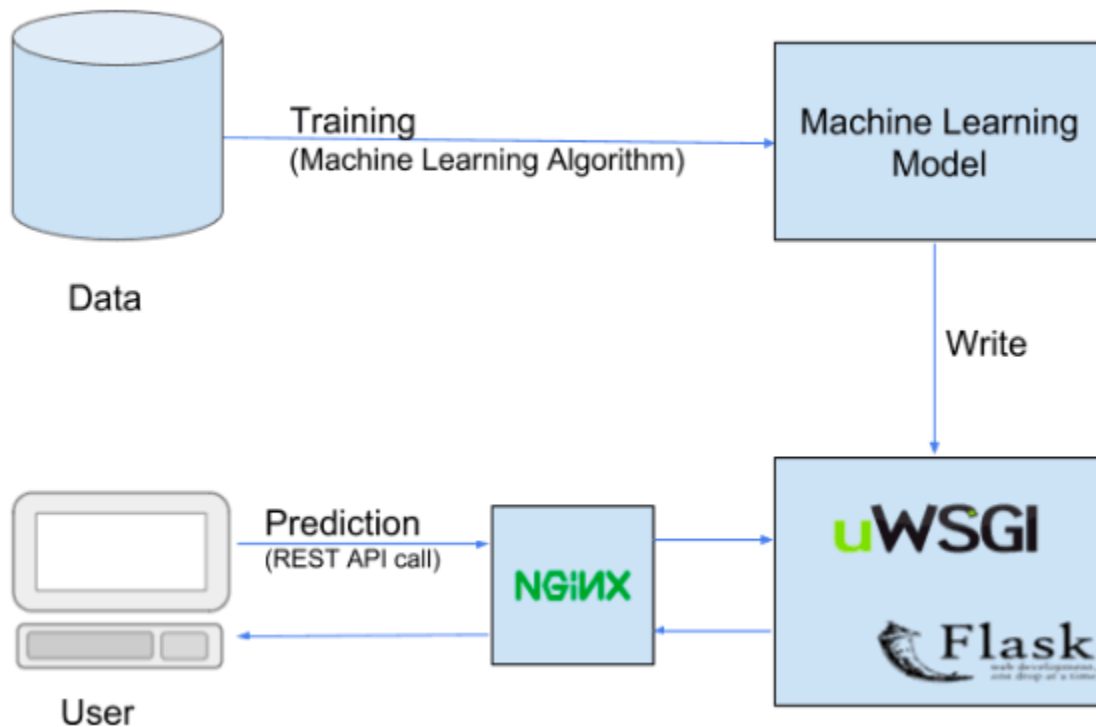
BATCH CODE: LISUM32

DATE: 29<sup>TH</sup> APRIL 2024

SUBMITTED TO: DATA GLACIER

## Introduction

This project aims at the deployment of a machine learning model for data that expresses sales according to the type of advertisement and the size of its associated cost using a Flask framework. The proposed workflow is shown in diagram 1.1 while the data information is showcased in table 1.1



**Diagram 1.1 Application Workflow**

### **Table Data Details**

Total number of observations	200
Total number of files	1
Total number of features	4
Base format of the file	Csv
Size of the data	4KB

**Table 1.1 Dataset Information**

The process for the deployment is summarized as follows

## 1) Data import

Import Libraries and Data

```
import pandas as pd
import numpy as np

advertising = pd.read_csv('https://raw.githubusercontent.com/themichaelonifade/DataGlacier_Virtual_Internship/main/Week%204/Advertising.csv')
advertising
```

	TV	Radio	Newspaper	Sales
0	230.1	37.8	69.2	22.1
1	44.5	39.3	45.1	10.4
2	17.2	45.9	69.3	9.3
3	151.5	41.3	58.5	18.5
4	180.8	10.8	58.4	12.9
...	...	...	...	...
195	38.2	3.7	13.8	7.6
196	94.2	4.9	8.1	9.7
197	177.0	9.3	6.4	12.8
198	283.6	42.0	66.2	25.5

Activate Windows  
Go to Settings to activate Windows.

## 2) Model Building

Build Model

```
from sklearn.model_selection import train_test_split
#splitting data into train and test sets

x = advertising.drop(['Sales'], axis=1)
y = advertising['Sales']

X_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2, random_state=0)
print(X_train.shape, x_test.shape)
```

(160, 3) (40, 3)

Linear Regression

```
[3] from sklearn.linear_model import LinearRegression

model = LinearRegression()
model.fit(X_train, y_train)
```

LinearRegression  
LinearRegression()

Save Model

```
[4] import pickle
    pickle.dump(model, open('model.pkl', 'wb'))
```

### 3) Turning Model into web application using flask

```
app.py > ...
1  from flask import Flask, request, render_template, jsonify
2  import numpy as np
3  import pickle
4
5  # Load the trained model
6
7  model = pickle.load(open('model.pkl', 'rb'))
8
9  app = Flask(__name__)
10
11 @app.route('/')
12 def index():
13     return render_template('index.html')
14
15 @app.route('/predict', methods=['POST'])
16 def predict_sales():
17     data = request.get_json()
18
19     tv = float(data['tv'])
20     radio = float(data['radio'])
21     newspaper = float(data['newspaper'])
22
23     # Make prediction
24     prediction = model.predict(np.array([[tv, radio, newspaper]]))
25
26     return jsonify({'projected sales revenue': prediction[0]})
27
28 if __name__ == '__main__':
29     app.run(debug=True)
```

### Final output

## Sales Revenue Generator

TV:

Radio:

Newspaper: