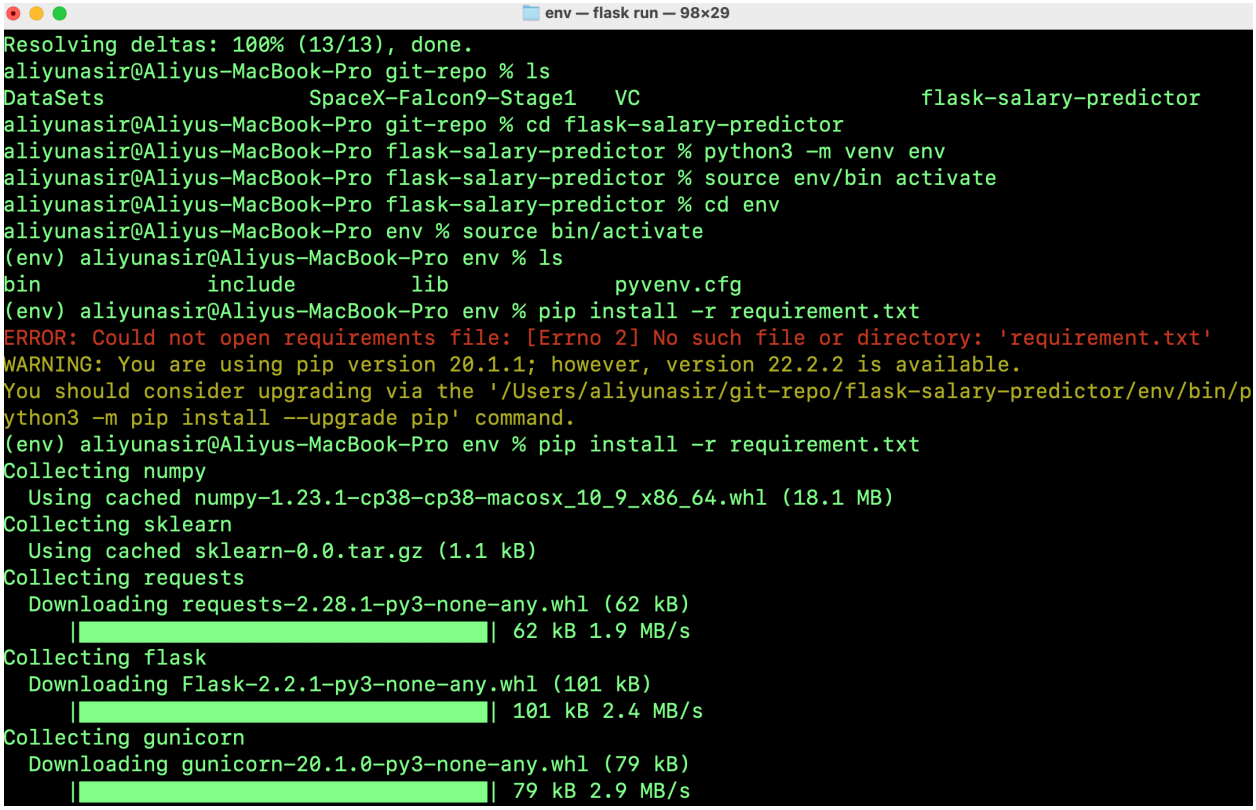
Data Intake Report

Name: Flask App Deployment
Report date: 2-08-2022
Internship Batch: LIMSUM11: 30
Version:1.0
Data intake by: Aliyu Nasir Maharaz
Data intake reviewer:
Data storage location: https://github.com/MaharazAliyu/Simple_Flask_Deployment

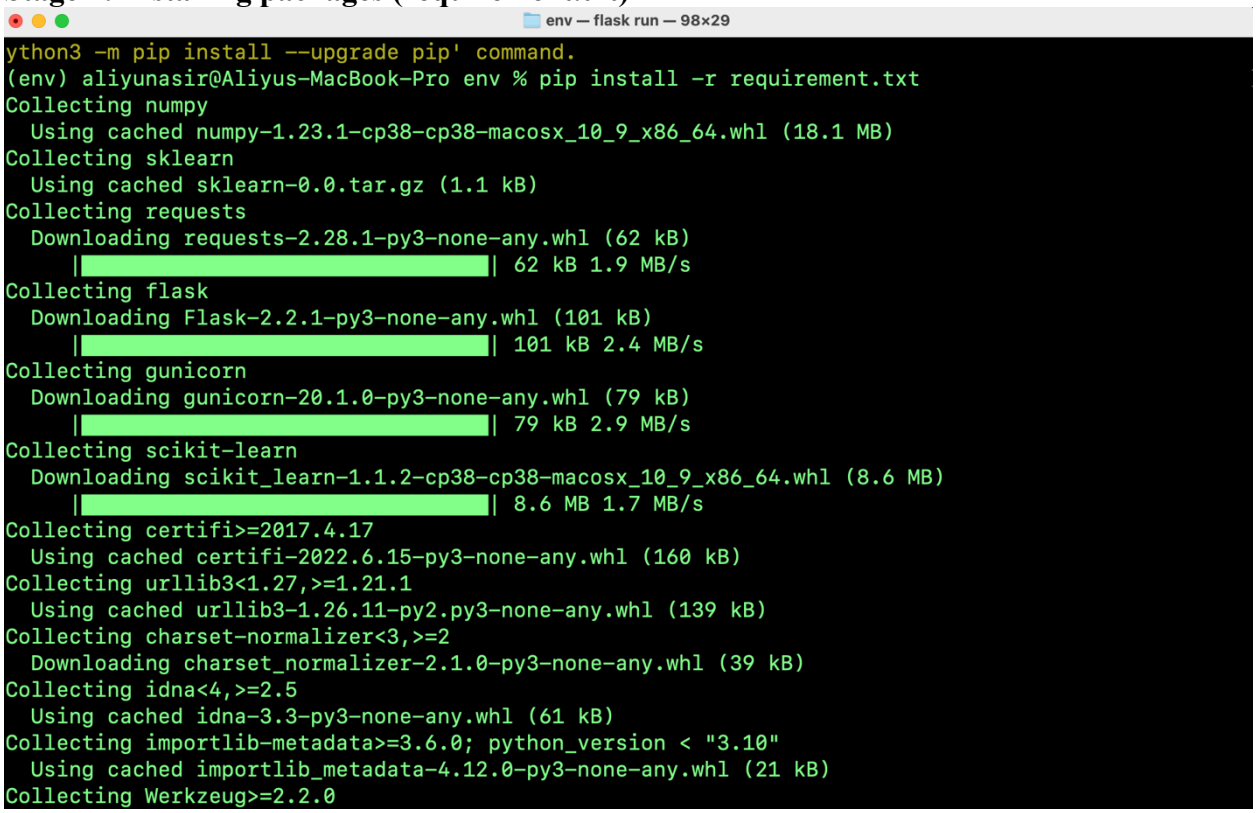
Tabular data details:

Total number of observations	29
Total number of files	1
Total number of features	2
Base format of the file	.csv
Size of the data	4 KB

Stage 1: Creating a virtual environment



Stage 2: Installing packages (requirement.txt)



Stage 3:Model.py

```python
'''
This model predicts the salary of the employ based on experience using simple linear regression model.
'''

# Importing the libraries
import numpy as np
#import matplotlib.pyplot as plt
import pandas as pd
import pickle
import requests
import json

# Importing the dataset
dataset = pd.read_csv('Salary_Data.csv')
X = dataset.iloc[:, :-1].values
y = dataset.iloc[:, 1].values

# Splitting the dataset into the Training set and Test set
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 1/3, random_state = 0)

# Fitting Simple Linear Regression to the Training set
from sklearn.linear_model import LinearRegression
regressor = LinearRegression()
regressor.fit(X_train, y_train)

# Predicting the Test set results
y_pred = regressor.predict(X_test)

# Saving model using pickle
pickle.dump(regressor, open('model.pkl','wb'))

# Loading model to compare the results
model = pickle.load(open('model.pkl','rb'))
print(model.predict([[1.8]]))

'''
    This model was forked from https://github.com/vvashemang/flask-salary-predictor/
```

Stage 4 app.y

ME.md | index.html | app.py | pyvenv.cfg | model.py

```python
import numpy as np
from flask import Flask, request, jsonify, render_template
import pickle

app = Flask(__name__)
model = pickle.load(open('model.pkl', 'rb'))

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

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

    output = round(prediction[0], 2)

    return render_template('index.html', prediction_text='Salary is {}'.format(output))


if __name__ == "__main__":
    app.run(debug=True)
```

Stage 5: Export FLASK_APP as app.py

env — flask run — 98×29

```
ython3 -m pip install --upgrade pip' command.
(env) aliyunasir@Aliyus-MacBook-Pro env % export FLASK_APP=app.py
(env) aliyunasir@Aliyus-MacBook-Pro env % flask run
Traceback (most recent call last):
  File "/Users/aliyunasir/git-repo/flask-salary-predictor/env/bin/flask", line 8, in <module>
    sys.exit(main())
  File "/Users/aliyunasir/git-repo/flask-salary-predictor/env/lib/python3.8/site-packages/flask/cl
i.py", line 1054, in main
    cli.main()
```

Stage 6: Deploy App

```
(env) aliyunasir@Aliyus-MacBook-Pro env % flask run
WARNING: This is a development server. Do not use it in a production deployment. Use a production
WSGI server instead.
 * Serving Flask app 'app.py'
 * Debug mode: off
 * Running on http://127.0.0.1:5000 (Press CTRL+C to quit)
127.0.0.1 - - [07/Aug/2022 13:56:30] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [07/Aug/2022 13:56:30] "GET /static/style.css HTTP/1.1" 304 -
127.0.0.1 - - [07/Aug/2022 13:56:30] "GET /static/bootstrap.min.css HTTP/1.1" 304 -
127.0.0.1 - - [07/Aug/2022 13:56:35] "POST /salary HTTP/1.1" 200 -
127.0.0.1 - - [07/Aug/2022 13:56:35] "GET /static/style.css HTTP/1.1" 304 -
127.0.0.1 - - [07/Aug/2022 13:56:35] "GET /static/bootstrap.min.css HTTP/1.1" 304 -
```