

Name: Shreya Ramachandra

Batch No: LISUM16

Submission Date: 26<sup>th</sup> December 2022

Submitted to: Data Glacier

Title: Model Deployment on Flask: Prediction Salary Analysis

## Writing program for Flask Application

```
D:\Internship\Data Glacier\Week 4\app.py

model.py × app.py* × style.css × index.html × request.py ×

1  import numpy as np
2  from flask import Flask, request, jsonify, render_template
3  import pickle
4
5  app = Flask(__name__)
6  model = pickle.load(open('model.pkl', 'rb'))
7
8  @app.route('/')
9  def home():
10     return render_template('index.html')
11
12  @app.route('/predict',methods=['POST'])
13  def predict():
14     '''
15     For rendering results on HTML GUI
16     '''
17     int_features = [int(x) for x in request.form.values()]
18     final_features = [np.array(int_features)]
19     prediction = model.predict(final_features)
20
21     output = round(prediction[0], 2)
22
23     return render_template('index.html', prediction_text='Employee Salary should be $ {}'.format(output))
24
25  @app.route('/predict_api',methods=['POST'])
26  def predict_api():
27     '''
28     For direct API calls trough request
29     '''
30     data = request.get_json(force=True)
31     prediction = model.predict([np.array(list(data.values()))])
32
33     output = prediction[0]
34     return jsonify(output)
35
36  if __name__ == "__main__":
37     app.run(debug=True)
38
```

## Saving the model in the Flask

Model is built for predicting the salaries of the employees based on the experience, test score and interview score

```
D:\Internship\Data Glacier\Week 4\model.py
model.py X app.py X style.css X index.html X request.py X
1 # Importing the libraries
2 import numpy as np
3 import matplotlib.pyplot as plt
4 import pandas as pd
5 import pickle
6
7 dataset = pd.read_csv('https://raw.githubusercontent.com/ShreyaRamachandra/Deploy-machine-Learn
8
9 dataset['experience'].fillna(0, inplace=True)
10
11 dataset['test_score'].fillna(dataset['test_score'].mean(), inplace=True)
12
13 X = dataset.iloc[:, :3]
14
15 #Converting words to integer values
16 def convert_to_int(word):
17     word_dict = {'one':1, 'two':2, 'three':3, 'four':4, 'five':5, 'six':6, 'seven':7, 'eight':8,
18                 'nine':9, 'ten':10, 'eleven':11, 'twelve':12, 'zero':0, 0: 0}
19     return word_dict[word]
20
21 X['experience'] = X['experience'].apply(lambda x : convert_to_int(x))
22
23 y = dataset.iloc[:, -1]
24
25 #Splitting Training and Test Set
26 #Since we have a very small dataset, we will train our model with all available data.
27
28 from sklearn.linear_model import LinearRegression
29 regressor = LinearRegression()
30
31 #Fitting model with training data
32 regressor.fit(X, y)
33
34 # Saving model to disk
35 pickle.dump(regressor, open('model.pkl','wb'))
36
37 # Loading model to compare the results
38 model = pickle.load(open('model.pkl','rb'))
39 print(model.predict([[2, 9, 6]]))
```

## HTML file

```
D:\Internship\Data Glacier\Week 4\templates\index.html
model.py × app.py* × style.css × index.html × request.py ×

1 <!DOCTYPE html>
2 <html >
3 <!-- From https://codepen.io/frytyler/pen/EGdtg-->
4 <head>
5   <meta charset="UTF-8">
6   <title>ML API</title>
7   <link href='https://fonts.googleapis.com/css?family=Pacifico' rel='stylesheet' type='text/css'>
8   <link href='https://fonts.googleapis.com/css?family=Arimo' rel='stylesheet' type='text/css'>
9   <link href='https://fonts.googleapis.com/css?family=Hind:300' rel='stylesheet' type='text/css'>
10  <link href='https://fonts.googleapis.com/css?family=Open+Sans+Condensed:300' rel='stylesheet' type='text/css'>
11  <link rel="stylesheet" href="{{ url_for('static', filename='css/style.css') }}">
12
13 </head>
14
15 <body>
16   <div class="login">
17     <h1>Predict Salary Analysis</h1>
18
19     <!-- Main Input For Receiving Query to our ML -->
20     <form action="{{ url_for('predict') }}" method="post">
21       <input type="text" name="experience" placeholder="Experience" required="required" />
22       <input type="text" name="test_score" placeholder="Test Score" required="required" />
23       <input type="text" name="interview_score" placeholder="Interview Score" required="required" />
24
25       <button type="submit" class="btn btn-primary btn-block btn-large">Predict</button>
26     </form>
27
28     <br>
29     <br>
30     {{ prediction_text }}
31
32   </div>
33
34
35 </body>
36 </html>
37
```

# Running the Flask Application in Anaconda Prompt

```

Anaconda Prompt (Anaconda) - python app.py

(base) C:\Users\Shreya>D:

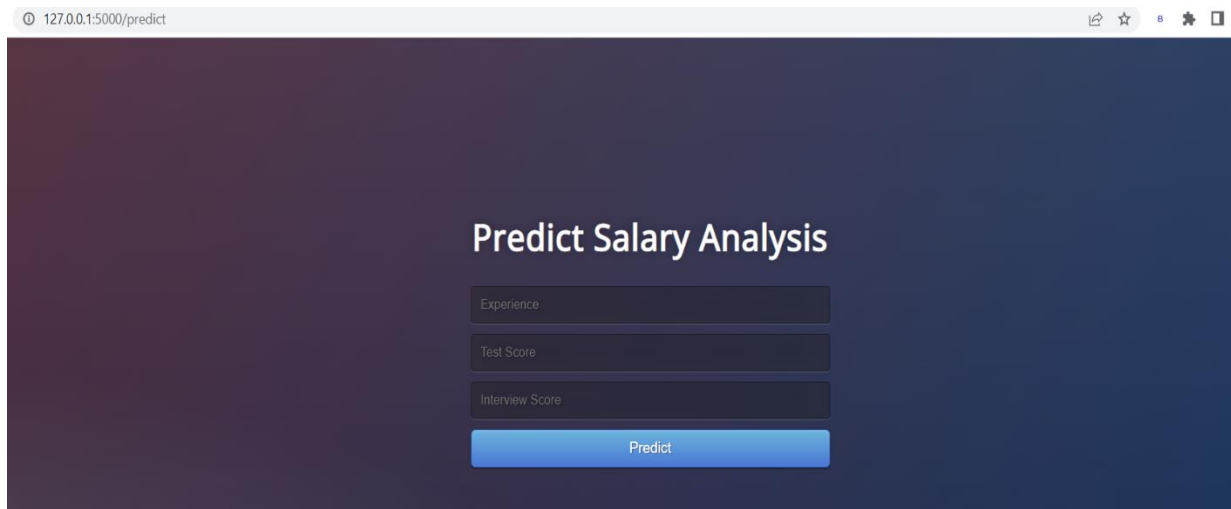
(base) D:\>cd Internship

(base) D:\Internship>cd Data Glacier

(base) D:\Internship\Data Glacier>cd Week 4

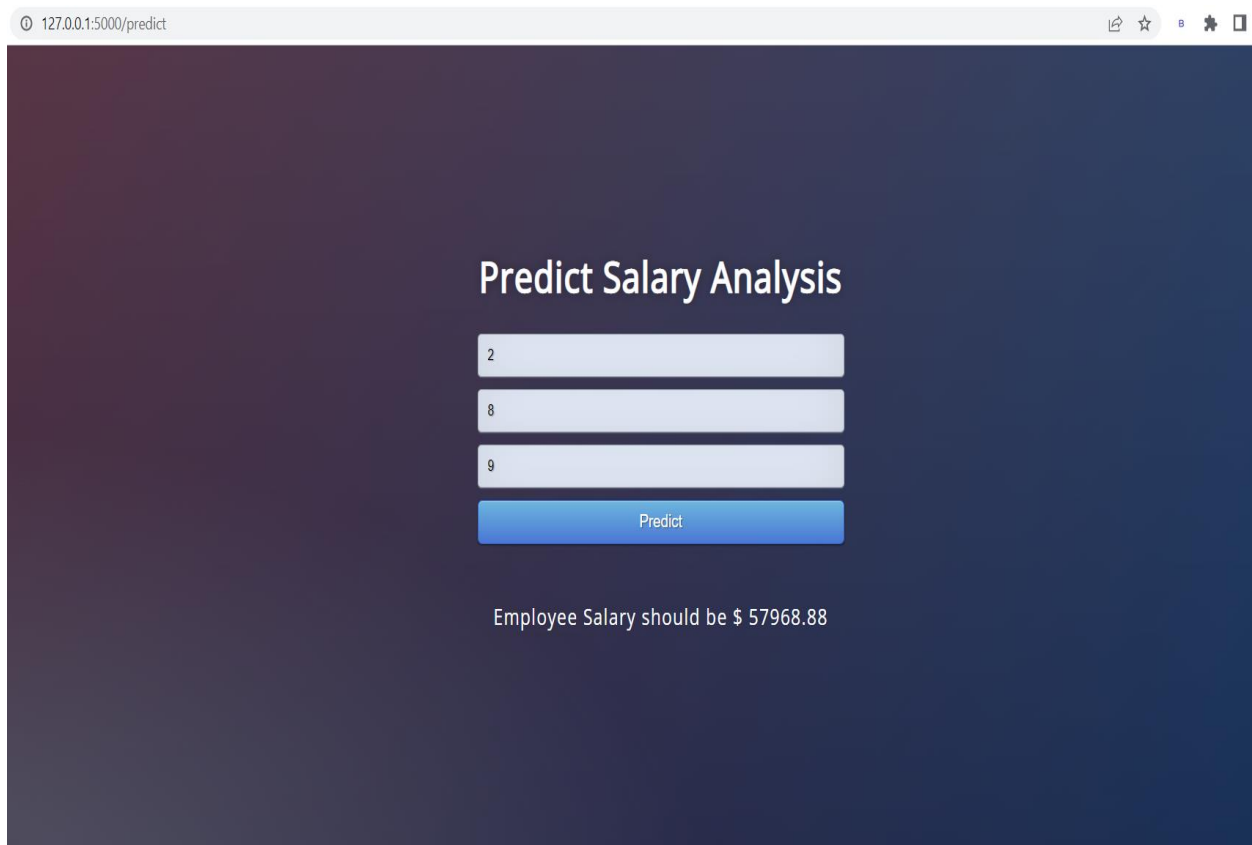
(base) D:\Internship\Data Glacier\Week 4>python app.py
* Serving Flask app 'app'
* Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on http://127.0.0.1:5000
Press CTRL+C to quit
* Restarting with watchdog (windowsapi)
* Debugger is active!
* Debugger PIN: 115-641-974
127.0.0.1 - - [25/Dec/2022 01:00:32] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [25/Dec/2022 01:00:32] "GET /static/css/style.css HTTP/1.1" 200 -
127.0.0.1 - - [25/Dec/2022 01:00:32] "GET /favicon.ico HTTP/1.1" 404 -
C:\Users\Shreya\AppData\Roaming\Python\Python39\site-packages\sklearn\base.py:409: UserWarning: X does not have valid feature names, but LinearRegression was fitted with feature names
  warnings.warn(
* Detected change in 'C:\Users\Shreya\AppData\Roaming\Python\Python39\site-packages\sklearn\base.py', reloading
127.0.0.1 - - [25/Dec/2022 01:00:42] "POST /predict HTTP/1.1" 200 -
127.0.0.1 - - [25/Dec/2022 01:00:42] "GET /static/css/style.css HTTP/1.1" 304 -
* Restarting with watchdog (windowsapi)
* Debugger is active!
* Debugger PIN: 115-641-974
C:\Users\Shreya\AppData\Roaming\Python\Python39\site-packages\sklearn\base.py:409: UserWarning: X does not have valid feature names, but LinearRegression was fitted with feature names
  warnings.warn(
127.0.0.1 - - [25/Dec/2022 01:01:28] "POST /predict HTTP/1.1" 200 -
127.0.0.1 - - [25/Dec/2022 01:01:28] "GET /static/css/style.css HTTP/1.1" 304 -
C:\Users\Shreya\AppData\Roaming\Python\Python39\site-packages\sklearn\base.py:409: UserWarning: X does not have valid feature names, but LinearRegression was fitted with feature names
  warnings.warn(
127.0.0.1 - - [25/Dec/2022 01:01:43] "POST /predict HTTP/1.1" 200 -
127.0.0.1 - - [25/Dec/2022 01:01:43] "GET /static/css/style.css HTTP/1.1" 304 -
* Detected change in 'D:\Internship\Data Glacier\Week 4\request.py', reloading
* Restarting with watchdog (windowsapi)
* Debugger is active!
* Debugger PIN: 115-641-974
C:\Users\Shreya\AppData\Roaming\Python\Python39\site-packages\sklearn\base.py:409: UserWarning: X does not have valid feature names, but LinearRegression was fitted with feature names
  warnings.warn(
* Detected change in 'C:\Users\Shreya\AppData\Roaming\Python\Python39\site-packages\sklearn\base.py', reloading
127.0.0.1 - - [25/Dec/2022 12:43:45] "POST /predict HTTP/1.1" 200 -
127.0.0.1 - - [25/Dec/2022 12:43:46] "GET /static/css/style.css HTTP/1.1" 304 -
* Restarting with watchdog (windowsapi)
* Debugger is active!
* Debugger PIN: 115-641-974
```

## Testing the model



A screenshot of a web browser showing a web application titled "Predict Salary Analysis". The browser's address bar displays "127.0.0.1:5000/predict". The application has a dark blue gradient background. It features three input fields labeled "Experience", "Test Score", and "Interview Score", each with a light blue border. Below these fields is a blue button labeled "Predict".

## Result: Predict Salary Analysis Model



A screenshot of the same web application, but now showing the result of a prediction. The input fields contain the values "2", "8", and "9". The "Predict" button is still present. Below the button, the text "Employee Salary should be \$ 57968.88" is displayed in white. The browser's address bar remains "127.0.0.1:5000/predict".