

Week4: Deployment on Flask

Name: Sri Harsha, Nimmagadda

Batch Code: LISUM27

Submission date: 29/11/2023

Submitted to: Data Glacier

Data Set up:

```
dataset = pd.read_csv('hiring.csv')
```

```
# Handle missing values
```

```
dataset['experience'].fillna(0, inplace=True)
```

```
dataset['test_score'].fillna(dataset['test_score'].mean(), inplace=True)
```

```
X = dataset.iloc[:, :3]
```

Data Training:

```
# Convert words to integer values consistently
```

```
def convert_to_int(word):
```

```
    word_dict = {'one':1, 'two':2, 'three':3, 'four':4, 'five':5, 'six':6, 'seven':7, 'eight':8,  
                'nine':9, 'ten':10, 'eleven':11, 'twelve':12, 'zero':0, 0: 0}
```

```
    return word_dict[word]
```

```
X['experience'] = X['experience'].apply(lambda x : convert_to_int(x))
```

```
y = dataset.iloc[:, -1]
```

```
# Splitting Training and Test Set
```

```
from sklearn.linear_model import LinearRegression
```

```
regressor = LinearRegression()
```

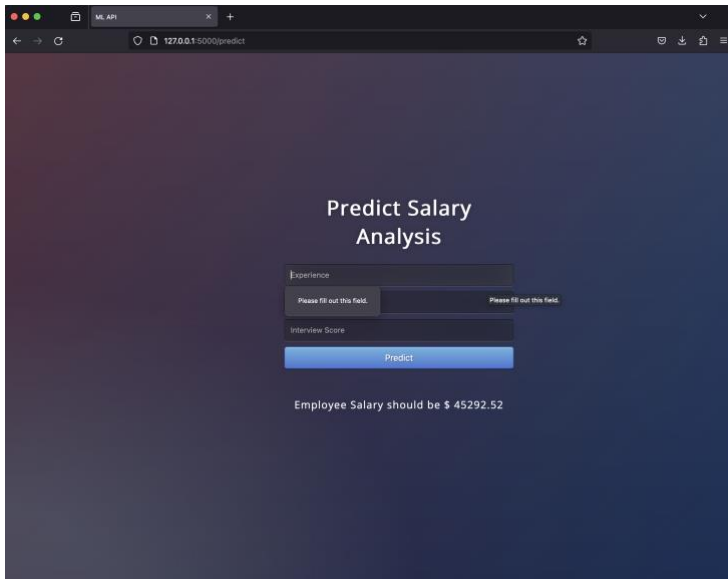
```
# Fitting model with training data
```

```
regressor.fit(X, y)
```

Predicting in app.py:

```
def predict_api():  
    data = request.get_json(force=True)  
    features = [data['experience'], data['test_score'], data['interview_score']]  
    prediction = model.predict([np.array(features)])  
  
    output = prediction[0]  
    return jsonify(output)
```

Deployed Model:



Predict Salary Analysis

Experience

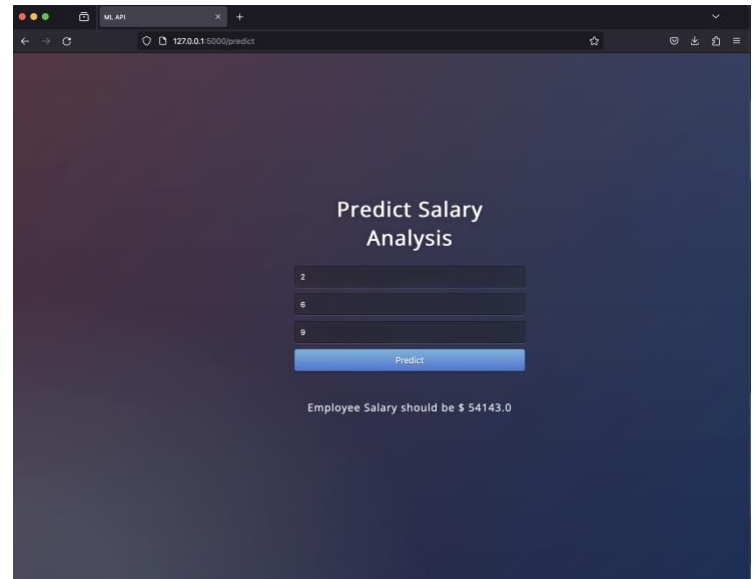
Please fill out this field.

Interview Score

Please fill out this field.

Predict

Employee Salary should be \$ 45292.52



Predict Salary Analysis

2

9

Predict

Employee Salary should be \$ 54143.0