Dt:- 9/7/22
12:00 pm

FLASK

Deployment FrameWork

* Here There Problem Statement, We have data of Experience and test score and Interview Score of past data, We have to predict the Salary of new Candidates.

```
import numpy as np
import pandas as pd
# df = pd.read_csv("hiring.csv")
# df
```

Out

X_1	X_2	X_3	y
Experience	test_score	interview_Score	Salary
NaN	8.0	9	50000
NaN	8.0	6	45000
5.0	6.0	7	60000
2.0	10.0	10	65000
7.0	9.0	6	70000
3.0	7.0	10	62000
10.0	NaN	7	72000
11.0	7.0	8	80000

Data cleaning

```
# df["experience"].fillna(0, inplace = True)
# df["test_score"].fillna(df["test_score"].mean(),
                          inplace = True)
```

Out : Filled with "0" and median values.

X & y

```
# x = df.iloc[:, :3]
# y = df.iloc[:, -1]
```

modelling

```
from sklearn.linear_model import LinearRegression
# model = LinearRegression()
# model.fit(x,y)
```

Out: LinearRegression() → Assume That, This is our Final Model.

Here, We are not Focusing Machine Learning Algorithm. we are using Flask for Deployement

* For saving The model.

  we have 2 options

  *1. joblib } These two do The Same job.

  *2. pickle

\# Saving model to disk

```
import pickle
# pickle.dump(model, open("model.pkl", "wb"))
```

(model → modelname. ; "wb" → write mode ; For saving we use)

Out: Create a .pkl File → save in current working directory

* This is done by programmers

```
# Load model
# model = pickle.load(open("model.pkl", "rb"))
```

"rb" → read mode

Prediction

```
# print(model.predict([[6, 3, 3]]))
```

[x1] [x2] [x3]

6 → Experience, 3 → test_score, 3 → Interview_score

Out : 46532.8746065 → Salary predicted.

* So, How it is predicted the Values. What is The equation That "model" has used for prediction.

```
# model.coef_
Out: array([ 2827.634, 1912.938, 2196.975])

# model.intercept_
Out: 17237.330
```

So, Here is Equation

* This is done by Mathematician

$$\hat{y} = 17237.33 + 2827.63[x_1] + 1912.93[x_2] + 2196.97[x_3]$$

$$y = b_0 + b_1 * x_1 + b_2 * x_2 + b_3 * x_3$$

b_0 → y-intercept

b_1, b_2, b_3 → coefficients

So, How non-Programmer can access The Prediction.

Here, we Use Flask. (Web FrameWork)

Where, We Host our Machine learning Model

* Where Deployment will be done ?
  1. Website
  2. apps (Androids, ios)
  3. Cloud (Azure, Aws)

As, per The client requirement, We have to Deploy Our Model.

Code.

```
import numpy as np
import pickle
From flask import Flask, request, render_template

# app = Flask(__name__)  (# initalization of Flask app)
# model = pickle.load(open("model.pkl", "rb"))
```

Load The model as pickle file.

```
# @app.route("/") => In flask redirect to URL(/)
    def home(): => initial home page (URL) (or) URL (to where it has to go)
        return render_template("index.html")
```

open The template — File, web page

HTML

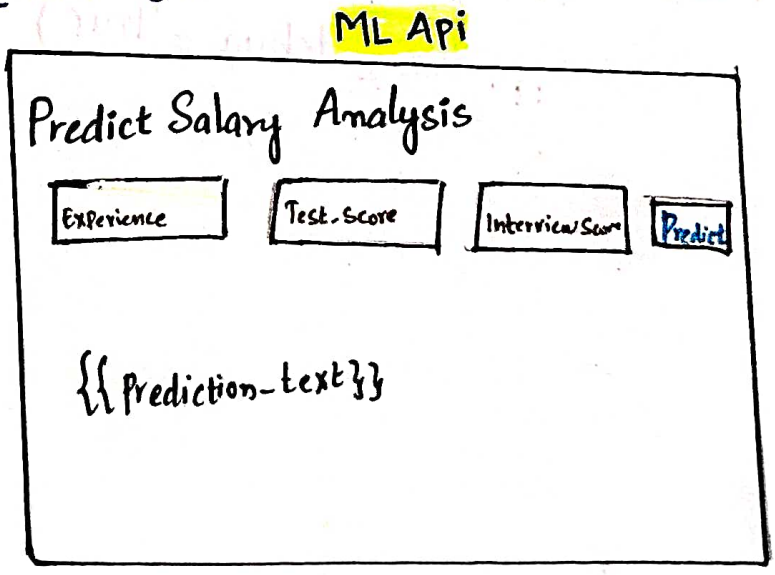
Salary Prediction Web Frame

:- Index.html

```
<!DOCTYPE html>
<html>
<!-- From https://codepen.io/frytyler/pen/EGdtg -->
<head>
    <meta charset="UTF-8">
    <title>ML API</title>
    <link href="https://fonts.googleapis.com/css?
        family=Pacifico" rel="stylesheet" type="text/cs
    <link href="https://fonts.googleapis.com/css?
        family=Arimo" rel="stylesheet" type="text/c
    <link href="https://fonts.googleapis.com/css?
        family=Hind:300" rel="stylesheet"
        type="text/css">
    <link rel="stylesheet" href="{{ url_for('stati
        filname="css/style.css")}}">
</head>

<body>
    <div class="login">
        <h1>Predict Salary Analysis</h1>
```

```
<!--Main Input For Receiving Query To our ML-->
<form action = "{{ url_for ("Predict") }}" methods = "post">
    <input type = "text" name = "experience"
           place holder = "Experience" required="required"/>
    <input type = "text" name = "test_score"
           place holder = "Test score" required="required"/>
    <input type = "text" name = "interview_score"
           placeholder = "Interview Score" required="required"/>
    <button type = "submit" class = "btn btn-primary
           btn-block-large" > predict</button>
</form>
<br>
<br>
{{ Prediction_text }}
</div>
</body>
</html>
```



```
# @app.route("/Predict", methods=["POST"])
```

It reads the values whatever you have posted in the Url.

```
def Predict():
    int_features = [int(x) for x in request.form.values()]
```

Ex: 2,6,9 → int_features

stores in

→ this is "list" comprehe

```
    final_features = [np.array(int_features)]
```

(2,6,9)

converts into array

```
    Prediction = model.predict(final_features)
```

↓ on array ⇒ model.predict([2,6,9])

```
    Output = round(Prediction[0], 2)

    return render_template("index.html", Prediction_
```

run

```
    text = "Employee Salary should be $("
    .format(output))
```

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