

Deployment Of The Model Using Flask

- When a data scientist/machine learning engineer develops a machine learning model using Scikit-Learn, TensorFlow, Keras, PyTorch etc, the ultimate goal is to make it available in production.
- Often times when working on a machine learning project, we focus a lot on Exploratory Data Analysis(EDA), Feature Engineering, tweaking with hyper-parameters etc. But we tend to forget our main goal, which is to extract real value from the model predictions.

In This Note Book I am teach you how you deploy your model in locally.

- Actually When Deployment Part comes there are two type of deployment
 1. Deploy Locally (On your computer)
 2. Deploy Globally (Use Heraku,Aws,cloud etc)
- But befor deploy globally we need to check how our model works in own system.
- So here 1st i am deploy this in own system
- then after few days i am teach you how to deploy in cloud.

So Lets Begin This amazing Part

Step 1 : 1st you need to develop the model using your jupyter notebook or any python ide (Here i Use Jupyter)

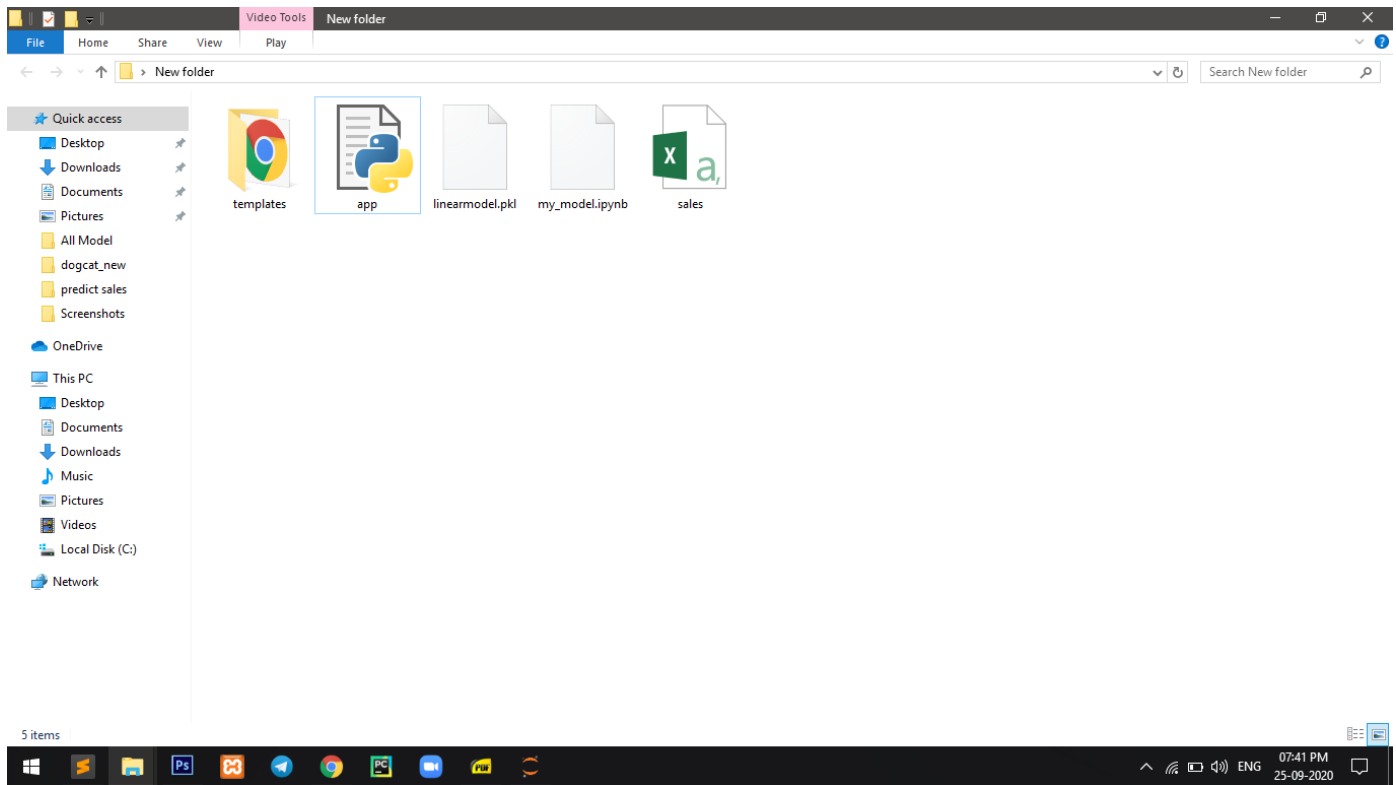
step 2 : Dumb This model to Pickle File For the production. Because in jupyter we cant deploy. so by the help of Pickle Module you need to dumb you model.

step 3 : It is most important step , because in this you build your web app.

step 1 : You Nedd to know flask framework to build this and also known as Ht ml/bootstrap to design your webpage.

step 2 : You make a folder , inside this folder you create app.py file and t emplate folder.

See Your Folder Structure:



See there are One Template Folder, One app.py, One my_model.ipynb and one linearmodel.pkl file, and sales.csv

- In Template Folder Your index.html page present
- in app.py file contains flask code for deployment on locally
- in my_model.ipynb file your model build code.
- linearmodel.pkl is the your dumb file
- sales.csv is your dataset

step 3 : all your html file you put inside on template folder otherwise it throws error.

step 4 : 1st you install flask in your system by "pip install flask" . otherwise it not work. Then you code your app.py file. (See below this notebook i add app.py file)

step 5 : Then Create your index.html, Now your work is Done (See below this notebook i add index.html file)

step 6 : You execute your app.py file in cmd type (python app.py) After that it open <http://127.0.0.1:5000/predict> this type of link.

step 7 : Hurray!! You make your 1st flask app.

Here also i demonstrate this whole step , once you go through it so that you can confidence to build any model

Step: 1 (develop the model Use jupyter)

In [1]:

```
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
import pandas as pd
import pickle
```

In [2]:

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

In [3]:

```
dataset.head(10)
```

Out[3]:

	rate	sales_in_first_month	sales_in_second_month	sales_in_third_month
0	NaN	2	500	300
1	NaN	4	300	650
2	four	600	200	400
3	nine	450	320	650
4	seven	600	250	350
5	five	550	200	700

Here our target variable is sales_in_third_month

In [4]:

```
dataset.shape
```

Out[4]:

(6, 4)

Here i am taking 6row and 4column dataset. i know this is very less dataset but the deployment part same for all dataset.

Analysis

In [5]:

```
dataset.isnull().sum()
```

Out[5]:

```
rate                2
sales_in_first_month 0
sales_in_second_month 0
sales_in_third_month 0
dtype: int64
```

Here you see there are 2 null value in rate column so we need to fix it

In [6]:

```
dataset['rate'] = dataset['rate'].fillna(dataset['rate'].mode()[0])
```

In [7]:

```
dataset.isnull().sum()
```

Out[7]:

```
rate                0
sales_in_first_month 0
sales_in_second_month 0
sales_in_third_month 0
dtype: int64
```

In [8]:

```
dataset.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 6 entries, 0 to 5
Data columns (total 4 columns):
#   Column                Non-Null Count  Dtype
---  -
0   rate                  6 non-null     object
1   sales_in_first_month  6 non-null     int64
2   sales_in_second_month 6 non-null     int64
3   sales_in_third_month  6 non-null     int64
dtypes: int64(3), object(1)
memory usage: 320.0+ bytes
```

Deal with Categorical Data

In [9]:

```
dataset['rate'].value_counts()
```

Out[9]:

```
five      3
four       1
seven      1
nine       1
Name: rate, dtype: int64
```

In [10]:

```
dataset['rate'] = dataset['rate'].replace(("five", "seven", "nine", "four"), (5, 7, 9, 4))
```

In [11]:

```
dataset['rate'].value_counts()
```

Out[11]:

```
5      3
7      1
4      1
9      1
Name: rate, dtype: int64
```

In [13]:

```
X = dataset.iloc[:, :3]
y = dataset.iloc[:, -1]
```

In [14]:

```
from sklearn.linear_model import LinearRegression
regressor = LinearRegression()
```

In [15]:

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

Out[15]:

```
LinearRegression()
```

In [16]:

```
print("Accuracy :", regressor.score(X,y))
```

```
Accuracy : 0.6029310840324749
```

Test Our Model

In [17]:

```
print("Sale in 3rd month is ", regressor.predict([[4, 200, 300]])[0].round(2))
```

```
Sale in 3rd month is 507.19
```

Step 2: (Dumb This model to Pickle)

In [18]:

```
pickle.dump(regressor, open('linearmodel.pkl', 'wb'))
```

Step 3 :

step 4: Make your app.py

app.py File

```
### app.py Code

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

app = Flask(__name__)

## See here i use linearmodel.pkl file for production

model = pickle.load(open('linearmodel.pkl', 'rb'))

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

@app.route('/predict', methods=['POST'])
def predict():

    if request.method == 'POST':

        ## Rate,sales1stmonth,sales2ndmonth all are from index form filed name so that i
        can capture the value

        rate = request.form['rate']
        sales1stmonth = request.form['sales1stmonth']
        sales2ndmonth = request.form['sales2ndmonth']

        data = [rate,sales1stmonth,sales2ndmonth]

        int_features = [int(x) for x in data]
        final_features = [np.array(int_features)]

        prediction = model.predict(final_features)

        output = round(prediction[0], 2)

        return render_template('index.html', prediction_text='Sales should be $
{}'.format(output))

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

Step 5 : Make index.html Put This file in to your Templates Folder

Index.html file

```
## Index.html code

<!DOCTYPE html>
<html >
  <head>
    <meta charset="UTF-8">
    <title>Deployment Tutorial 1</title>
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/css/bootstrap.min.css">
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js">
</script>
    <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/js/bootstrap.min.js">
</script>

  </head>
  <body>
    <div class="container-fluid" style="width: 50%">
      <h1>Sales Forecasting</h1>
      <!-- Main Input For Receiving Query to our ML -->

      <form action="/predict" method="POST">
        <div class="form-group">

          <input type="text" class="form-control" id="exampleInputEmail1" name="rate"
placeholder="Rate">

        </div>
        <div class="form-group">

          <input type="text" class="form-control" id="exampleInputEmail1"
name="sales1stmonth" placeholder="sales1stmonth">

        </div>
        <div class="form-group">

          <input type="text" class="form-control" id="exampleInputEmail1"
name="sales2ndmonth" placeholder="sales2ndmonth">

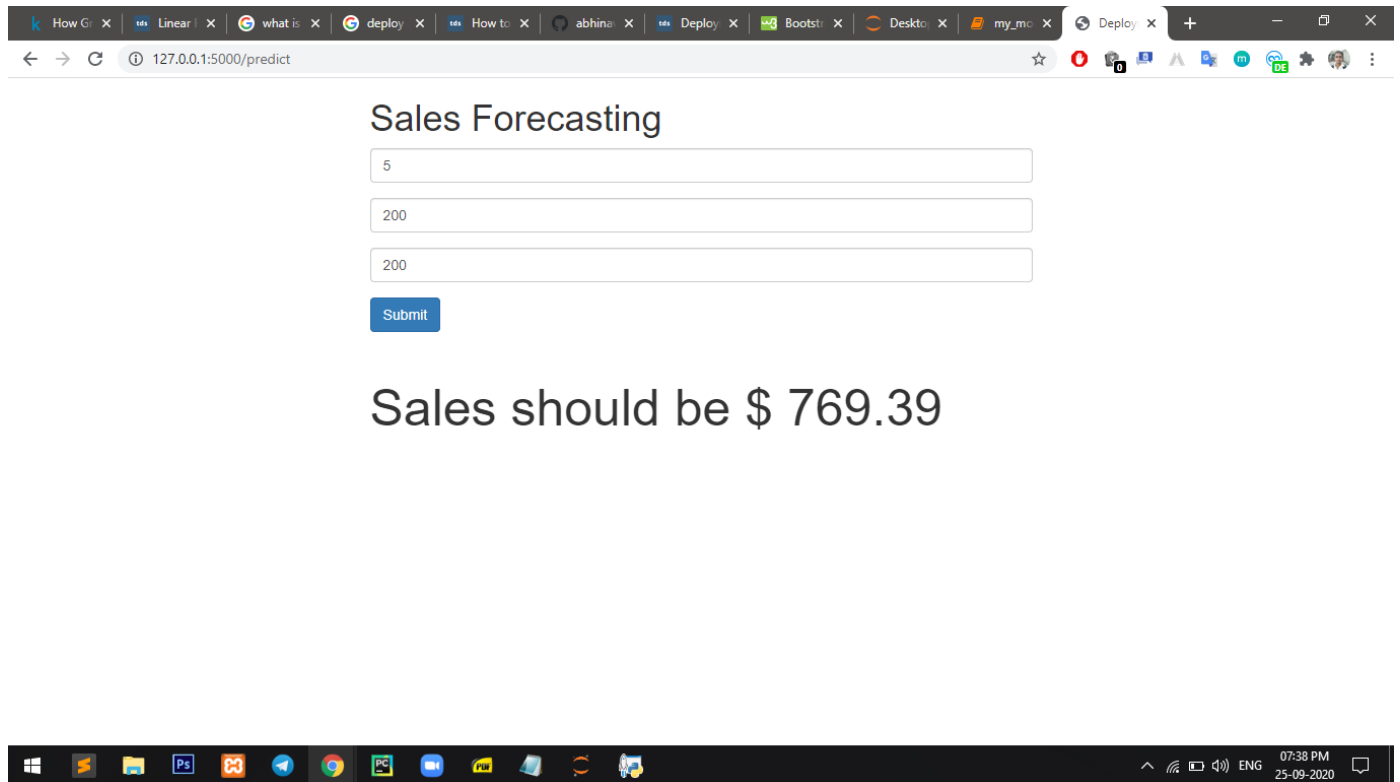
        </div>

        <button type="submit" class="btn btn-primary">Submit</button>
      </form>
      <br>
      <br>

      #here your result show
      <p style="font-size: 50px;"> {{ prediction_text }} </p>

    </div>
  </body>
</html>
```

See The Output Page



Congratulations YOu Made your First Machine Learning Website

Here i Only Demonstrate How to Deploy on Locally , There is another part that you deploy in globally like in cloud platform (Heraku,google cloud,aws etc)

But Before Going to that part you all please make your own locally flask app and share with me after that i will going to teach about how to deploy in globally.

You Make your own Web app and send me a screen shot of the output page as i shown above. Its My Humble request Because this part is more imporatatant