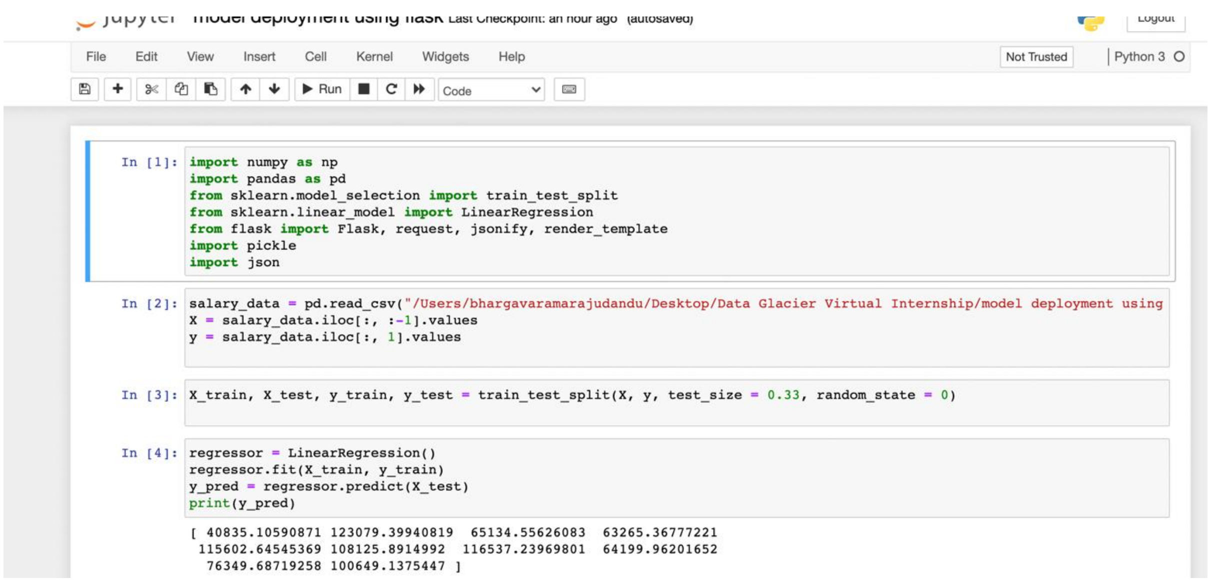
Name: Bhargava Rama Raju Dandu.
Batch code: LISUM01.
Submission Date: 10-June-2021.
Submitted to: Data Glacier.

Deployment on Flask

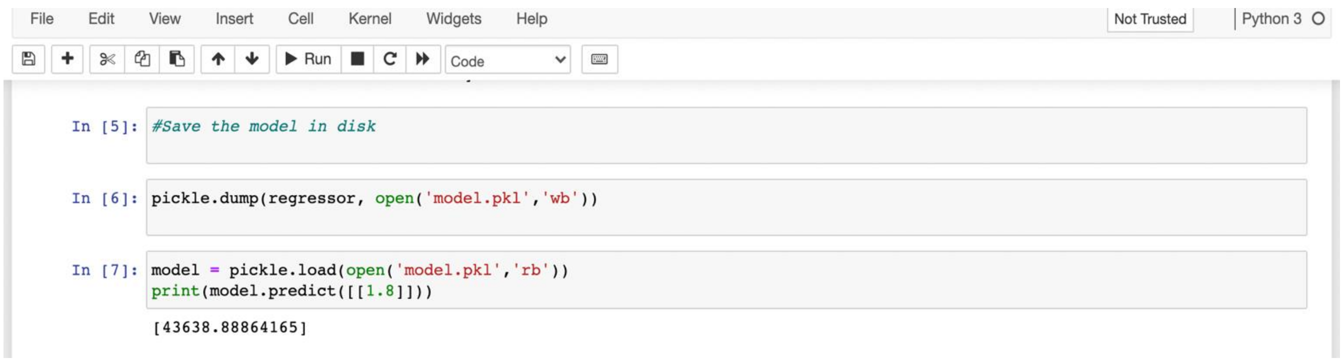
Step 1: Develop a Machine learning model.

Predict the salary of an employee using Linear Regression Model.


jupyter model deployment using flask Last Checkpoint: an hour ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3
Run
Code

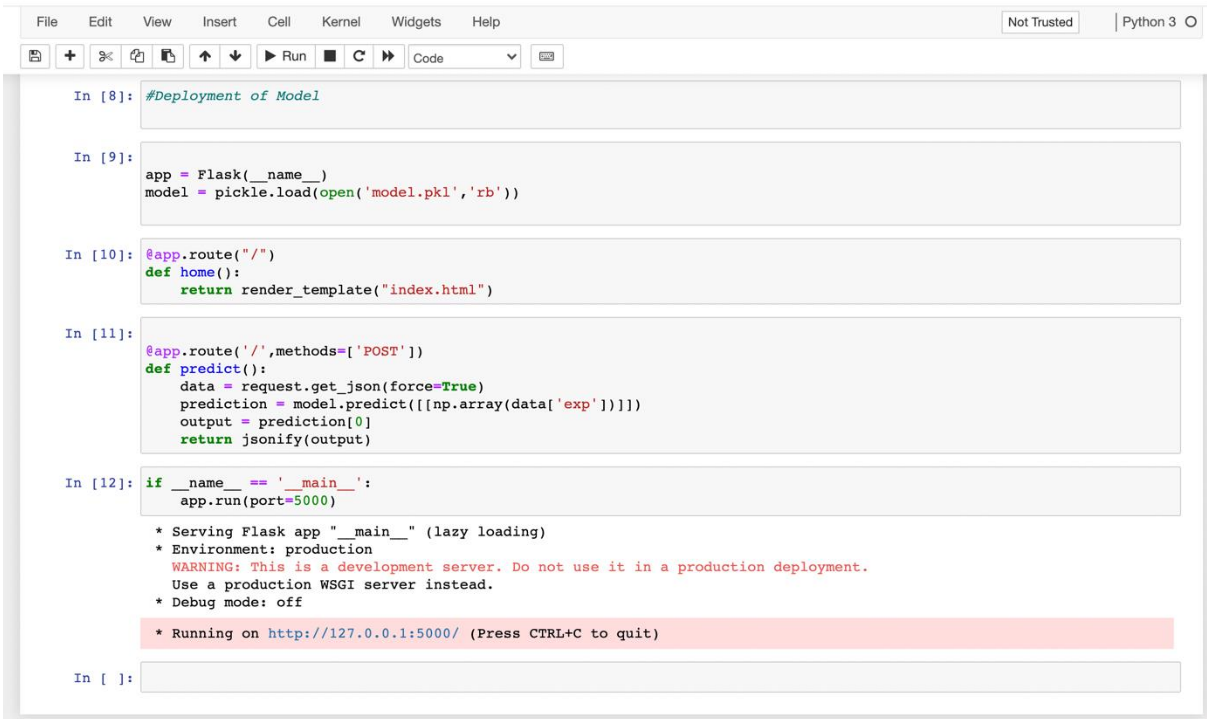
```
In [1]: import numpy as np
        import pandas as pd
        from sklearn.model_selection import train_test_split
        from sklearn.linear_model import LinearRegression
        from flask import Flask, request, jsonify, render_template
        import pickle
        import json

In [2]: salary_data = pd.read_csv("/Users/bhargavaramarajudandu/Desktop/Data Glacier Virtual Internship/model deployment using
        X = salary_data.iloc[:, :-1].values
        y = salary_data.iloc[:, 1].values

In [3]: X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.33, random_state = 0)

In [4]: regressor = LinearRegression()
        regressor.fit(X_train, y_train)
        y_pred = regressor.predict(X_test)
        print(y_pred)

        [ 40835.10590871 123079.39940819  65134.55626083  63265.36777221
         115602.64545369 108125.8914992  116537.23969801  64199.96201652
          76349.68719258 100649.1375447 ]
```

Step 2: Save the trained model by using pickle library.



```
In [5]: #Save the model in disk

In [6]: pickle.dump(regressor, open('model.pkl','wb'))

In [7]: model = pickle.load(open('model.pkl','rb'))
        print(model.predict([[1.8]]))

        [43638.88864165]
```

Step 3: Deployment of the model

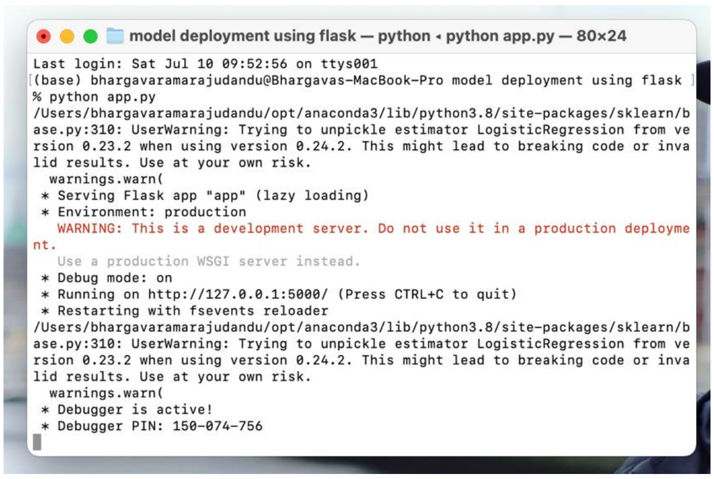


```
In [8]: #Deployment of Model

In [9]:
app = Flask(__name__)
model = pickle.load(open('model.pkl','rb'))

In [10]: @app.route("/")
def home():
    return render_template("index.html")

In [11]:
@app.route('/',methods=['POST'])
def predict():
    data = request.get_json(force=True)
    prediction = model.predict([[np.array(data['exp'])]])
    output = prediction[0]
    return jsonify(output)

In [12]: if __name__ == '__main__':
    app.run(port=5000)

 * Serving Flask app "__main__" (lazy loading)
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployment.
   Use a production WSGI server instead.
 * Debug mode: off
 * Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)

In [ ]:
```

- Created the instance of the *Flask()* and loaded the model.
- Bounded " */"* with the method *predict()* in which predict method gets the data from the json passed by the requestor.
- *model.predict()* method takes input from the json and converts it into 2D *numpy array* the results are stored into the variable named *output.*
- Return this variable after converting it into the json object using flasks *jsonify()* method.
- Run our server by following above code section and using port 5000.

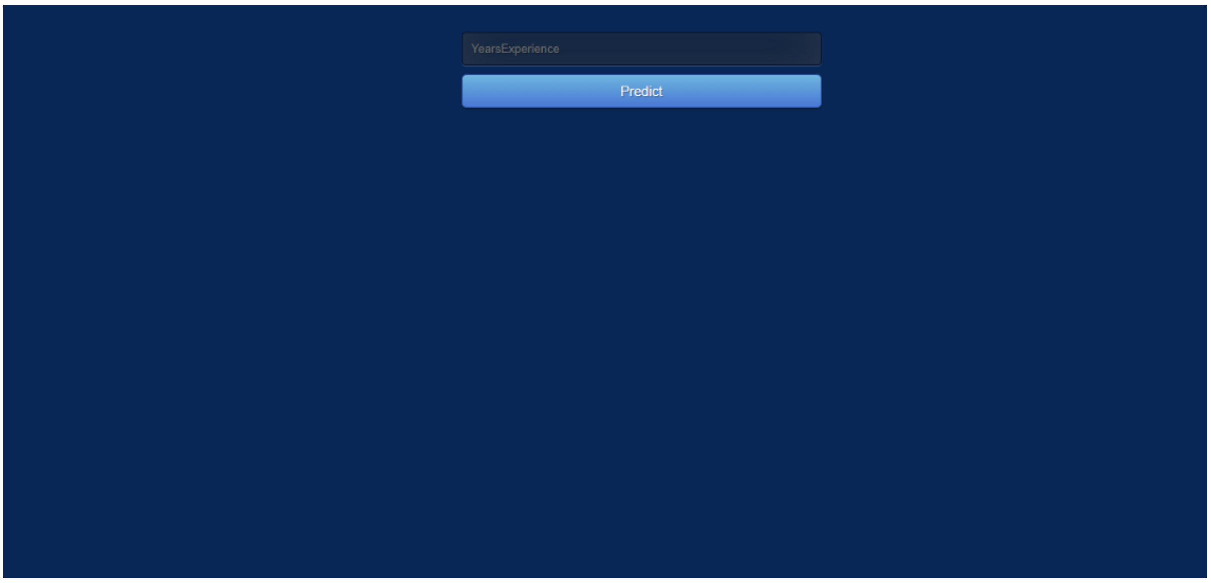
Step 4:

Checking python app.py file in terminal


model deployment using flask — python ◂ python app.py — 80×24

```
Last login: Sat Jul 10 09:52:56 on ttys001
(base) bhargavaramarajudandu@Bhargavas-MacBook-Pro model deployment using flask
% python app.py
/Users/bhargavaramarajudandu/opt/anaconda3/lib/python3.8/site-packages/sklearn/b
ase.py:310: UserWarning: Trying to unpickle estimator LogisticRegression from ve
rsion 0.23.2 when using version 0.24.2. This might lead to breaking code or inva
lid results. Use at your own risk.
  warnings.warn(
 * Serving Flask app "app" (lazy loading)
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployme
nt.
   Use a production WSGI server instead.
 * Debug mode: on
 * Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
 * Restarting with fsevents reloader
/Users/bhargavaramarajudandu/opt/anaconda3/lib/python3.8/site-packages/sklearn/b
ase.py:310: UserWarning: Trying to unpickle estimator LogisticRegression from ve
rsion 0.23.2 when using version 0.24.2. This might lead to breaking code or inva
lid results. Use at your own risk.
  warnings.warn(
 * Debugger is active!
 * Debugger PIN: 150-074-756
```

Step 5: Creating the web app using the using in browser



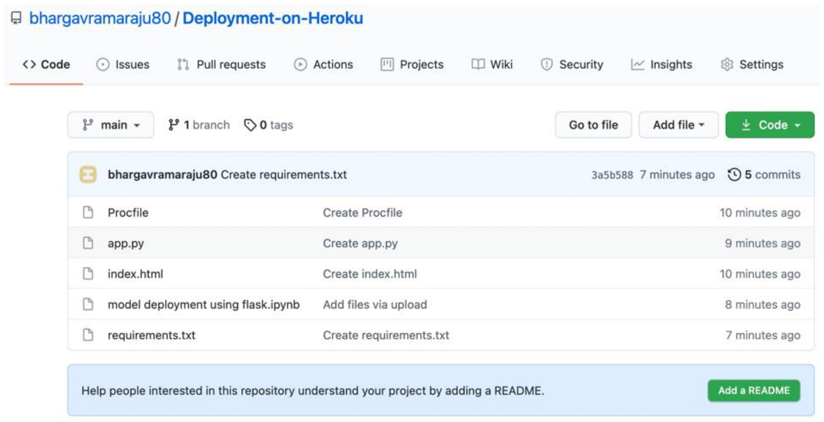
Step 6: Create Procfile which specifies the commands that are executed by the Heroku app on the startup. Web:gunicorn app:app.

Running the command pip freeze > requirements.txt in CMD for creating requirement.txt file which will contain all of the dependencies of the flask app.

Step 7:

Create the repository in Github repository and commit the code.

https://github.com/bhargavramaraju80/Deployment-on-Heroku

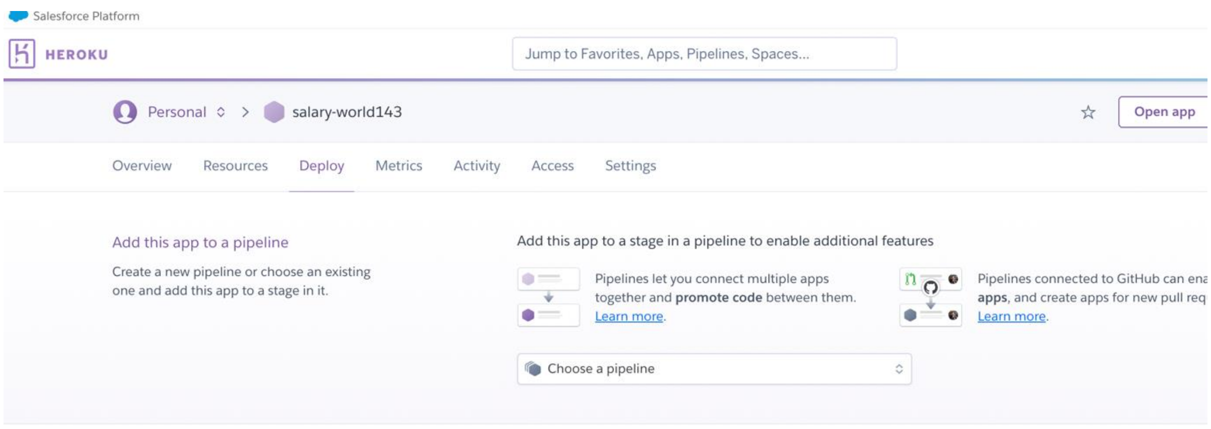


bhargavramaraju80 Create requirements.txt 3a5b588 7 minutes ago 5 commits

Procfile	Create Procfile	10 minutes ago
app.py	Create app.py	9 minutes ago
index.html	Create index.html	10 minutes ago
model deployment using flask.ipynb	Add files via upload	8 minutes ago
requirements.txt	Create requirements.txt	7 minutes ago

Help people interested in this repository understand your project by adding a README. Add a README

Step 8: Create an account in Heroku and the create an app (Salary-world143).


Salesforce Platform

HEROKU

Jump to Favorites, Apps, Pipelines, Spaces...

Personal > salary-world143

Open app

Overview Resources Deploy Metrics Activity Access Settings

Add this app to a pipeline

Create a new pipeline or choose an existing one and add this app to a stage in it.

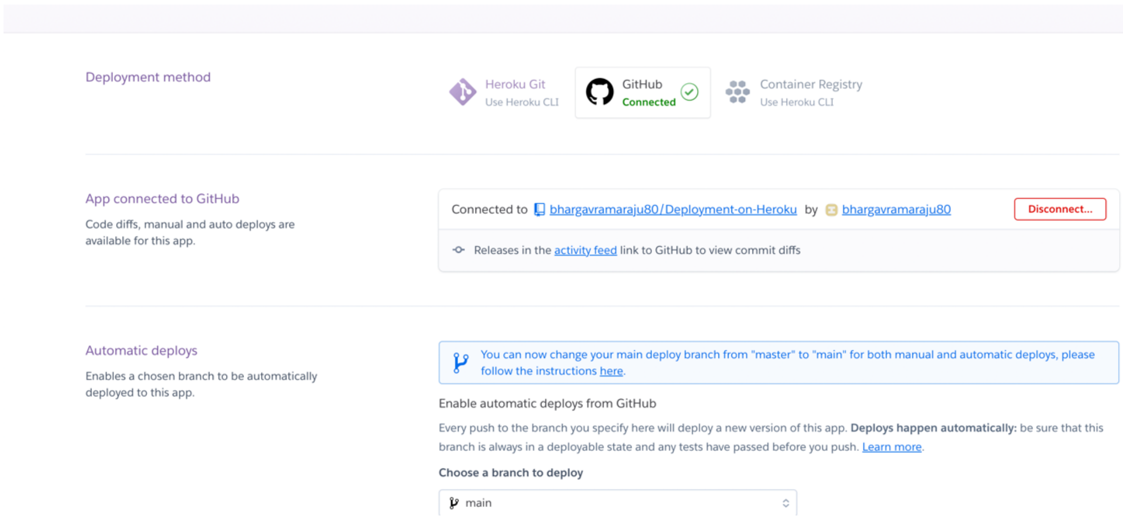
Add this app to a stage in a pipeline to enable additional features

Pipelines let you connect multiple apps together and **promote code** between them. Learn more.

Pipelines connected to GitHub can ena **apps**, and create apps for new pull req Learn more.

Choose a pipeline

Step 9: Link your Github account with your Heroku account and give access to your repository.


Deployment method

Heroku Git
Use Heroku CLI

GitHub
Connected

Container Registry
Use Heroku CLI

App connected to GitHub

Code diffs, manual and auto deploys are available for this app.

Connected to bhargavramaraju80/Deployment-on-Heroku by bhargavramaraju80

Disconnect...

Releases in the activity feed link to GitHub to view commit diffs

Automatic deploys

Enables a chosen branch to be automatically deployed to this app.

You can now change your main deploy branch from "master" to "main" for both manual and automatic deploys, please follow the instructions here.

Enable automatic deploys from GitHub

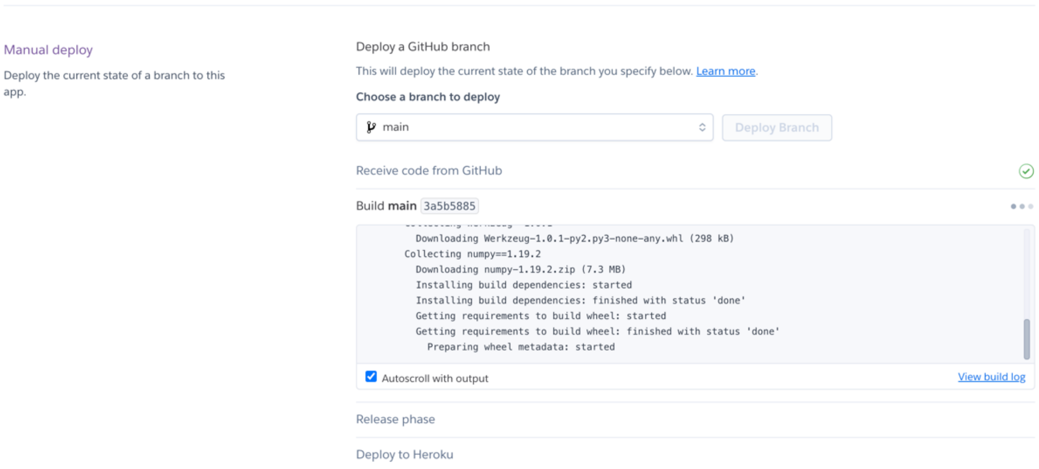
Every push to the branch you specify here will deploy a new version of this app. **Deploys happen automatically:** be sure that this branch is always in a deployable state and any tests have passed before you push. Learn more.

Choose a branch to deploy

main

Step 10:

Finally Deploy the model on the Heroku app.


Manual deploy

Deploy the current state of a branch to this app.

Deploy a GitHub branch

This will deploy the current state of the branch you specify below. Learn more.

Choose a branch to deploy

main

Deploy Branch

Receive code from GitHub

Build **main** 3a5b5885

```
    Downloading Werkzeug-1.0.1-py2.py3-none-any.whl (298 kB)
  Collecting numpy==1.19.2
    Downloading numpy-1.19.2.zip (7.3 MB)
    Installing build dependencies: started
    Installing build dependencies: finished with status 'done'
    Getting requirements to build wheel: started
    Getting requirements to build wheel: finished with status 'done'
      Preparing wheel metadata: started
```

Autoscroll with output

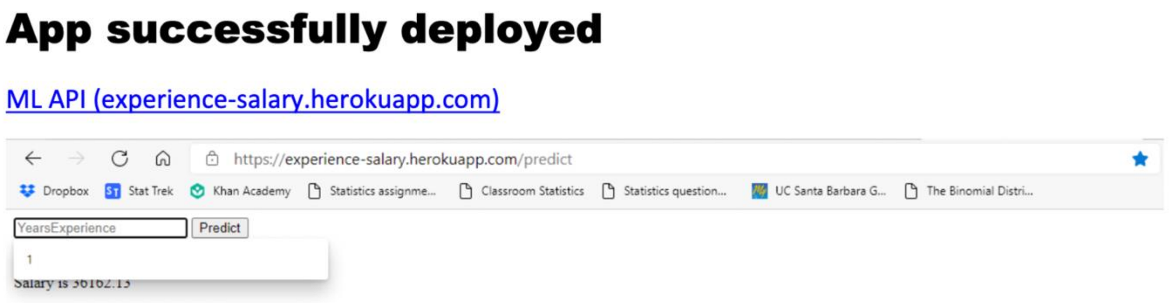
View build log

Release phase

Deploy to Heroku

Step 11:

App Sucessfully deployed.

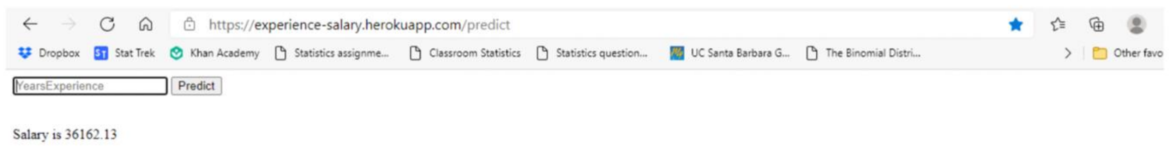


App successfully deployed

ML API (experience-salary.herokuapp.com)

https://experience-salary.herokuapp.com/predict

Salary is 36162.13



https://experience-salary.herokuapp.com/predict

Salary is 36162.13