Mohamed Amine Kina

Flask Deployment

Case: Medical Insurance Costs

- Look for a simple dataset:

URL: https://www.kaggle.com/datasets/mirichoi0218/insurance

This dataset consists of 1337 data points with 6 features:

- age: age of primary beneficiary
- sex: insurance contractor gender, female, male
- bmi: Body mass index, providing an understanding of body, weights that are relatively high or low relative to height,
  objective index of body weight (kg / m ^ 2) using the ratio of height to weight, ideally 18.5 to 24.9
- children: Number of children covered by health insurance / Number of dependents
- smoker: Smoking
- region: the beneficiary's residential area in the US, northeast, southeast, southwest, northwest.

Output:

- charges: Individual medical costs billed by health insurance

- Make model (model.py)

```
6
7   dataset = pd.read_csv('insurance.csv')
8
9
10
11  X = dataset.iloc[:, :6]
12
13  #Converting words to integer values
14  def convert_to_int(word):
15      word_dict = {'male':1,'female':2, 'northeast':1,'southeast':2,'southwest':3,'northwest':4, 'yes':1,'no
16      return word_dict[word]
17
18  X['sex'] = X['sex'].apply(lambda x : convert_to_int(x))
19  X['region'] = X['region'].apply(lambda x : convert_to_int(x))
20  X['smoker'] = X['smoker'].apply(lambda x : convert_to_int(x))
21
22  y = dataset.iloc[:, -1]
```

Read the dataset file ‘insurance.csv’.

Allocate matrix X:

- We have to make all inputs to be numbers, that’s why we define the function conver_to_int()
- This function assigns integers to words (male = 1 , female = 2 …)

Allocate vector Y

```
#We will train our model with all availabe data.

from sklearn.linear_model import LinearRegression
regressor = LinearRegression()

#Fitting model with trainig data

regressor.fit(X.values, y)

# Saving model to disk
pickle.dump(regressor, open('model.pkl','wb'))

# Loading model to compare the results
model = pickle.load(open('model.pkl','rb'))

```

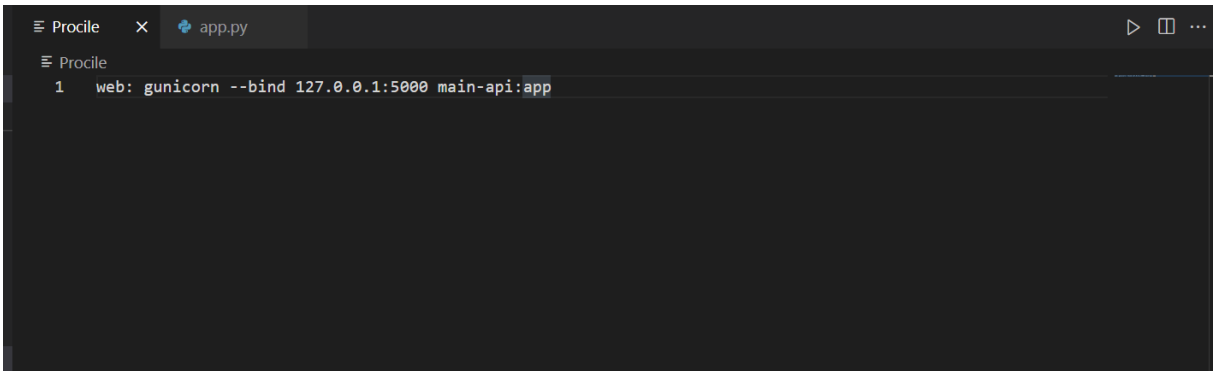
Train model and get the regressor

Get file model.pkl

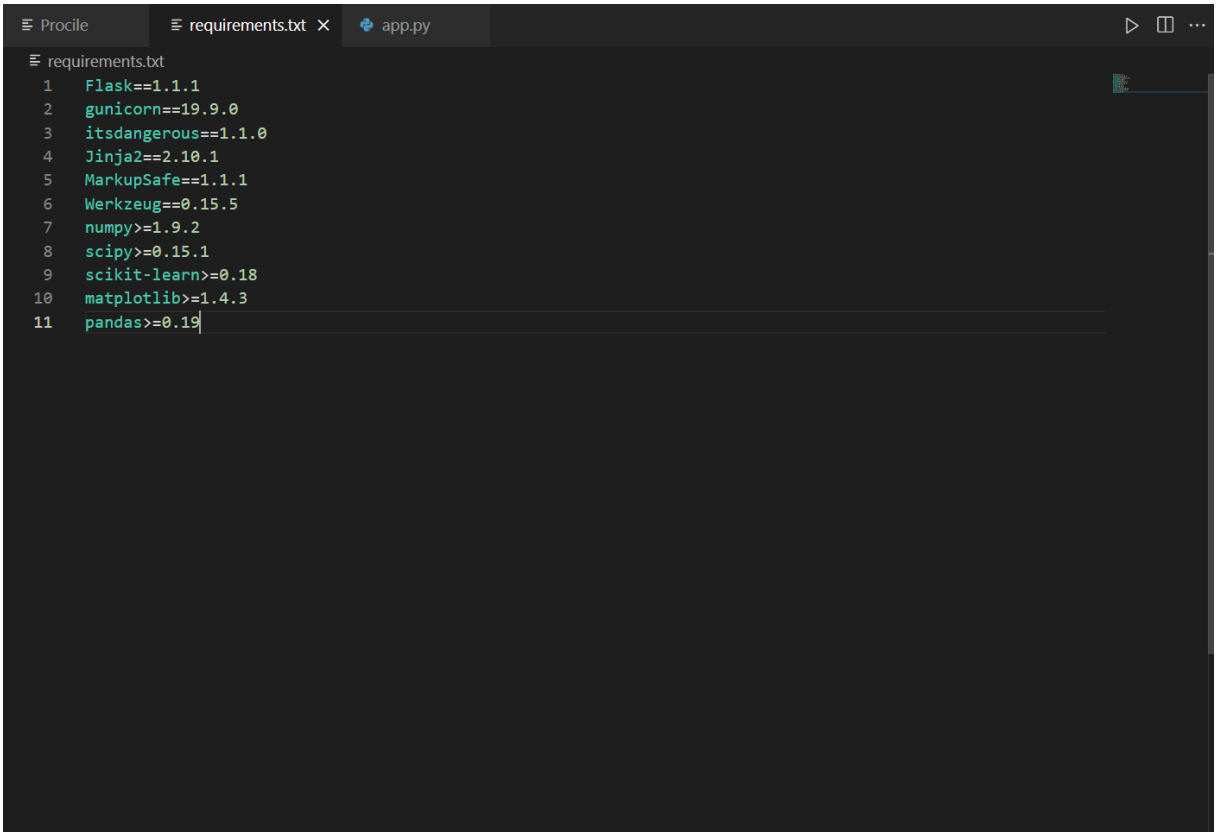
```
app = Flask(__name__)
model = pickle.load(open('model.pkl', 'rb'))

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

@app.route('/predict',methods=['POST'])
def predict():
    '''
    For rendering results on HTML GUI
    '''
    int_features = [float(x) for x in request.form.values()]
    final_features = [np.array(int_features)]
    prediction = model.predict(final_features)

    output = round(prediction[0], 2)

    return render_template('index.html', prediction_text='Insurance should charge ${}'.format(output))
```

Set up the app using Flask

Make a Proctile file

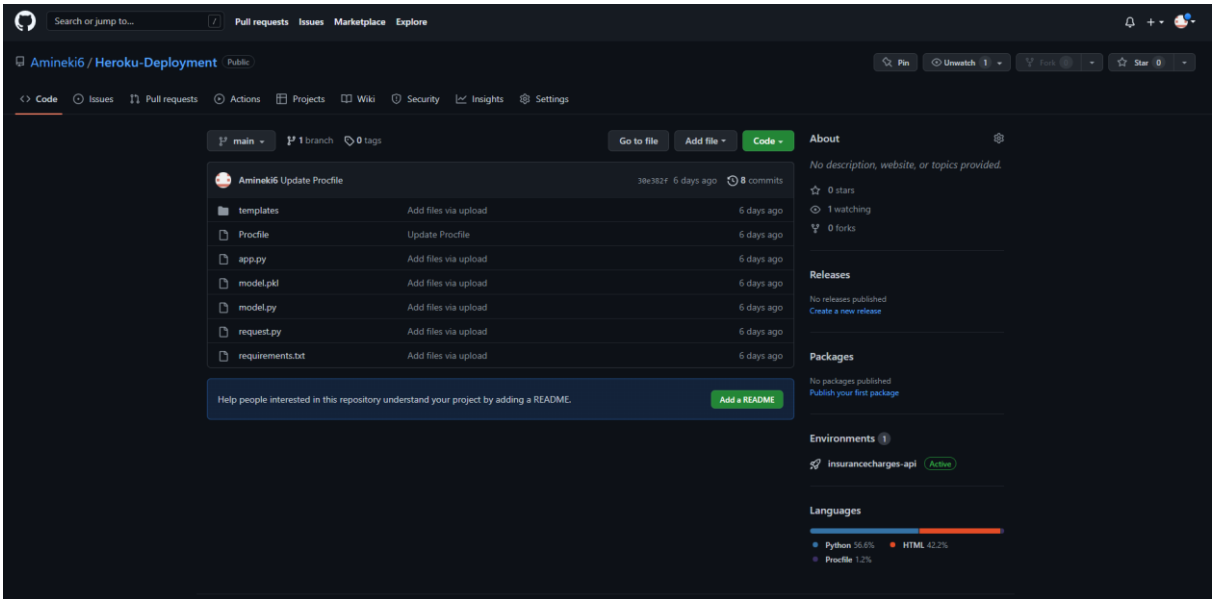


Make a Requirements file where you mention all the libraries needed to run the app.



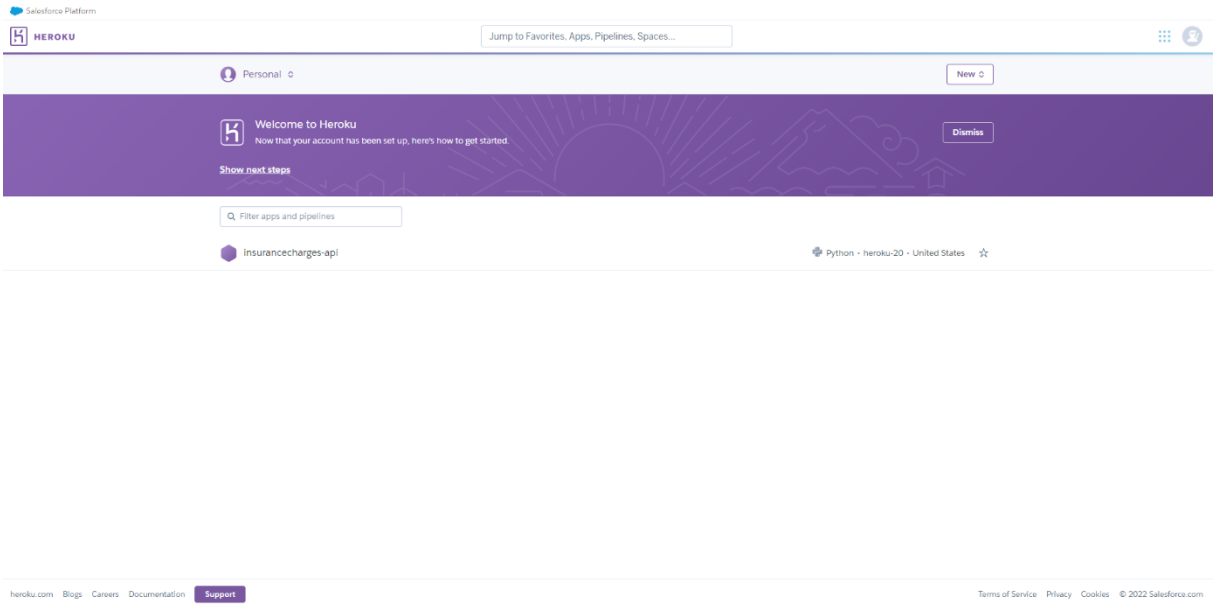
Set up a new GitHub Repository

Commit the files to git Hub



Set up Heroku

Deploy the branch manually from GitHub



View the app on:

https://insurancecharges-api.herokuapp.com

Submission Date: 10/08/2022