

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':2}
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 `convert_to_int()`
- This function assigns integers to words (male = 1 , female = 2 ...)

Allocate vector Y

```

24 #We will train our model with all available data.
25
26 from sklearn.linear_model import LinearRegression
27 regressor = LinearRegression()
28
29 #Fitting model with training data
30
31 regressor.fit(X.values, y)
32
33 # Saving model to disk
34 pickle.dump(regressor, open('model.pkl','wb'))
35
36 # Loading model to compare the results
37 model = pickle.load(open('model.pkl','rb'))
38

```

Train model and get the regressor

Get file model.pkl

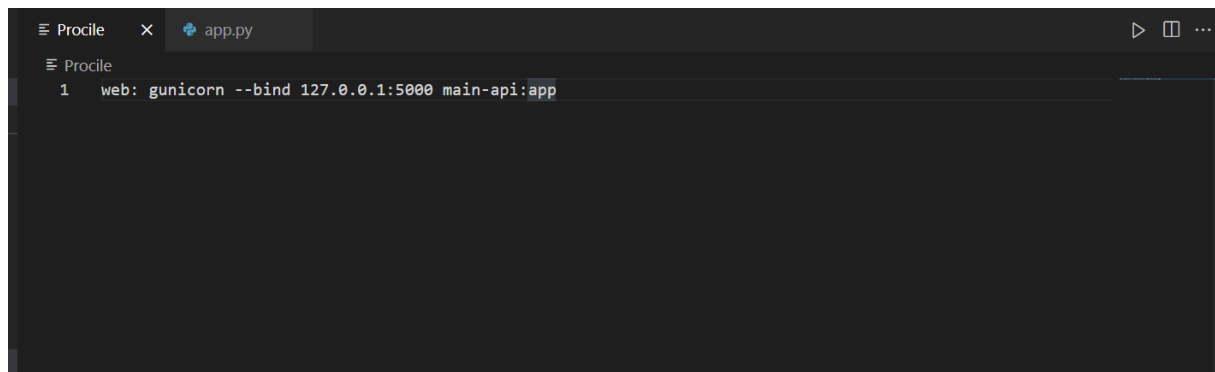
```

4
5 app = Flask(__name__)
6 model = pickle.load(open('model.pkl', 'rb'))
7
8 @app.route('/')
9 def home():
10     return render_template('index.html')
11
12 @app.route('/predict',methods=['POST'])
13 def predict():
14     '''
15     For rendering results on HTML GUI
16     '''
17     int_features = [float(x) for x in request.form.values()]
18     final_features = [np.array(int_features)]
19     prediction = model.predict(final_features)
20
21     output = round(prediction[0], 2)
22
23     return render_template('index.html', prediction_text='Insurance should charge {}'.format(output))

```

Set up the app using Flask

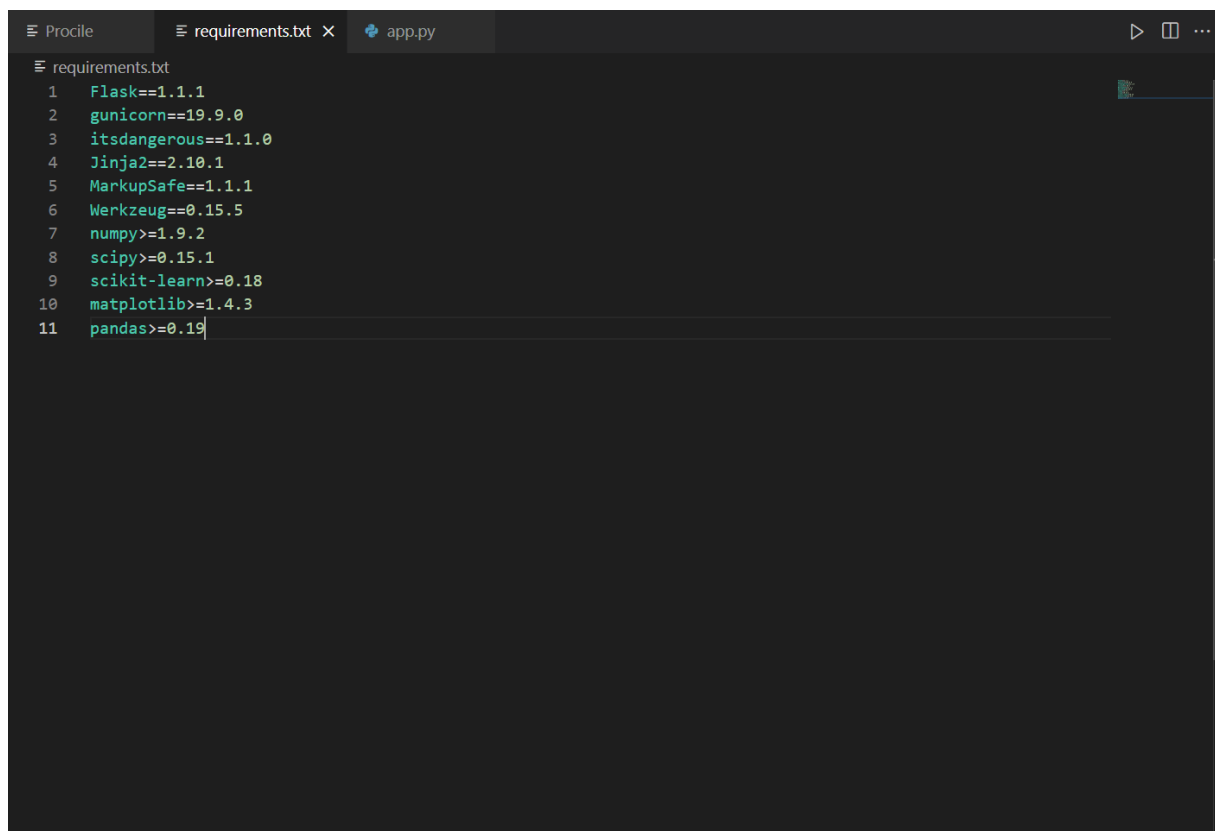
Make a Proctile file



A screenshot of a code editor window with a dark theme. The top bar shows a tab labeled 'Procfile' and another tab labeled 'app.py'. The 'Procfile' tab is active, and the text 'Procfile' is visible in the top left corner of the editor area. The main content of the file is a single line of text: '1 web: gunicorn --bind 127.0.0.1:5000 main-api:app'. The text is white on a dark background.

```
1 web: gunicorn --bind 127.0.0.1:5000 main-api:app
```

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

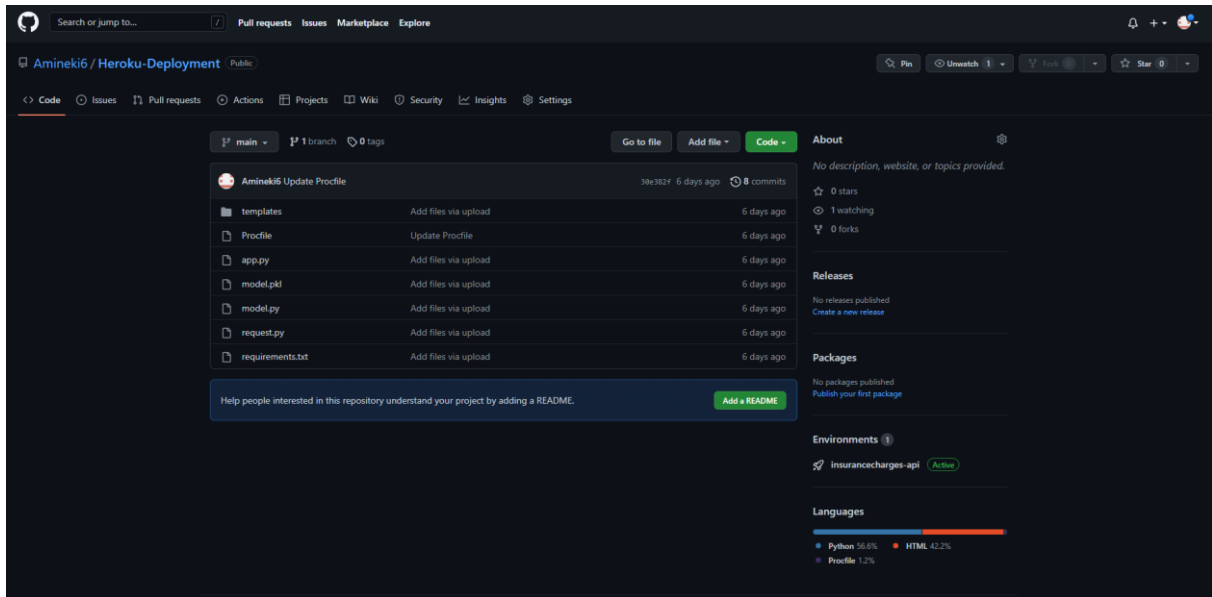


A screenshot of a code editor window with a dark theme. The top bar shows three tabs: 'Procfile', 'requirements.txt', and 'app.py'. The 'requirements.txt' tab is active, and the text 'requirements.txt' is visible in the top left corner of the editor area. The main content of the file is a list of 11 lines of text, each representing a library and its version constraint. The text is white on a dark background.

```
1 Flask==1.1.1
2 gunicorn==19.9.0
3 itsdangerous==1.1.0
4 Jinja2==2.10.1
5 MarkupSafe==1.1.1
6 Werkzeug==0.15.5
7 numpy>=1.9.2
8 scipy>=0.15.1
9 scikit-learn>=0.18
10 matplotlib>=1.4.3
11 pandas>=0.19
```

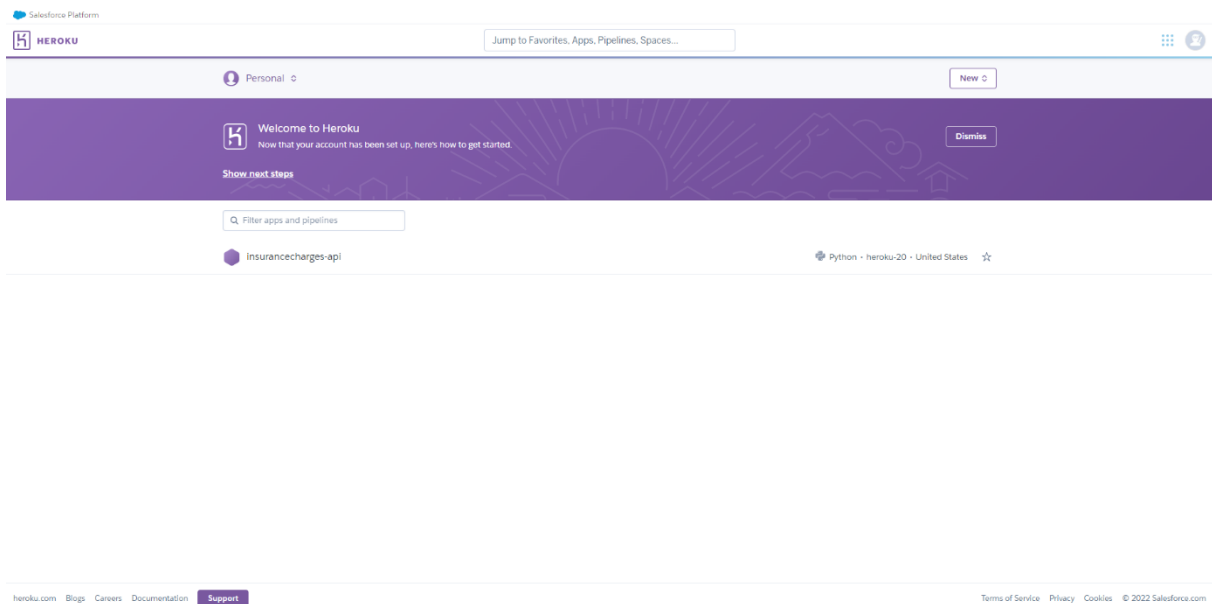
Set up a new GitHub Repository

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