Week5:
Cloud and API deployment

Name: Rayhanul Islam Rumel
Batch Code: LISUM17
Submission Date: 05/02/2023
Submission To: Data Glacier

Agenda

Project Overview

Explore and Understand the Dataset

Preprocessing the dataset

Build the ML model

Building Webapp using Flask

Deploy Spam Detection web app using Render

User guide

Try Yourself - Dummy Dataset

Project Overview

Project Overview

- **Dataset:** Spam text message classification
- **No of columns:** 2 (Message, Category)
- **ML Model:** Random Forest Classifier
- **Target:** We will input a text, the model will predict whether the text is spam or not.
- **Front-end:** HTML, CSS, and JavaScript
- **Back-end:** Flask, ML Model (Random Forest Classifier)

Explore and Understand the Dataset

Explore and Understand the dataset

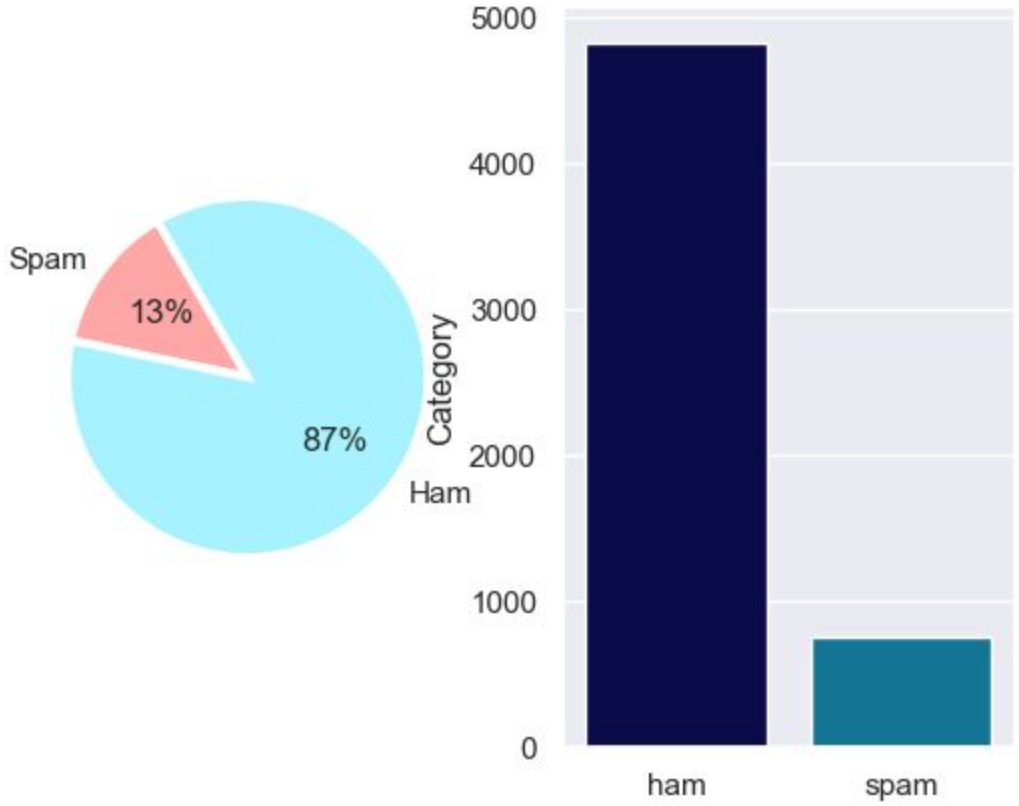
- The dataset has 2 columns: Category, Message
- The shape of the dataset is: 5572, 2
- The dataset contains no null or duplicate values

```
ds.head()
```

	Category	Message
0	ham	Go until jurong point, crazy.. Available only ...
1	ham	Ok lar... Joking wif u oni...
2	spam	Free entry in 2 a wkly comp to win FA Cup fina...
3	ham	U dun say so early hor... U c already then say...
4	ham	Nah I don't think he goes to usf, he lives aro...

Explore and Understand the dataset

- In total 13% of the total messages are spam and the rest (87%) is ham messages.



Preprocessing the dataset

Preprocessing the dataset

- Before preparing the train and test set, we processed the Message column with the following steps:
  - Standardized the messages by converting all of the text to lowercase
  - Filtered all of the message in order to remove any hyperlinks available in the message
  - Removed punctuation from the message
  - Removed stop words from the corpus
  - Stemmed the words of each text

- Then I split the dataset into training set and test set:
  - 80% of the dataset is used as Training set, 20% of the dataset is used as Test set

- With TF-IDF I extracted the features of each message of the both train and test set as vectors

Build the ML model

Build the ML model

- Then I defined random forest classifier:

```
rfc = RandomForestClassifier(n_estimators= 300)
```

- Once the model is defined, with train and test sets, I trained the model and evaluated the model accuracy
- The model accuracy is approximately 98%

Building Webapp using Flask

Building Webapp using Flask

- app.py : The controller that controls the app.
- Here I created two app routes:
  - **home**

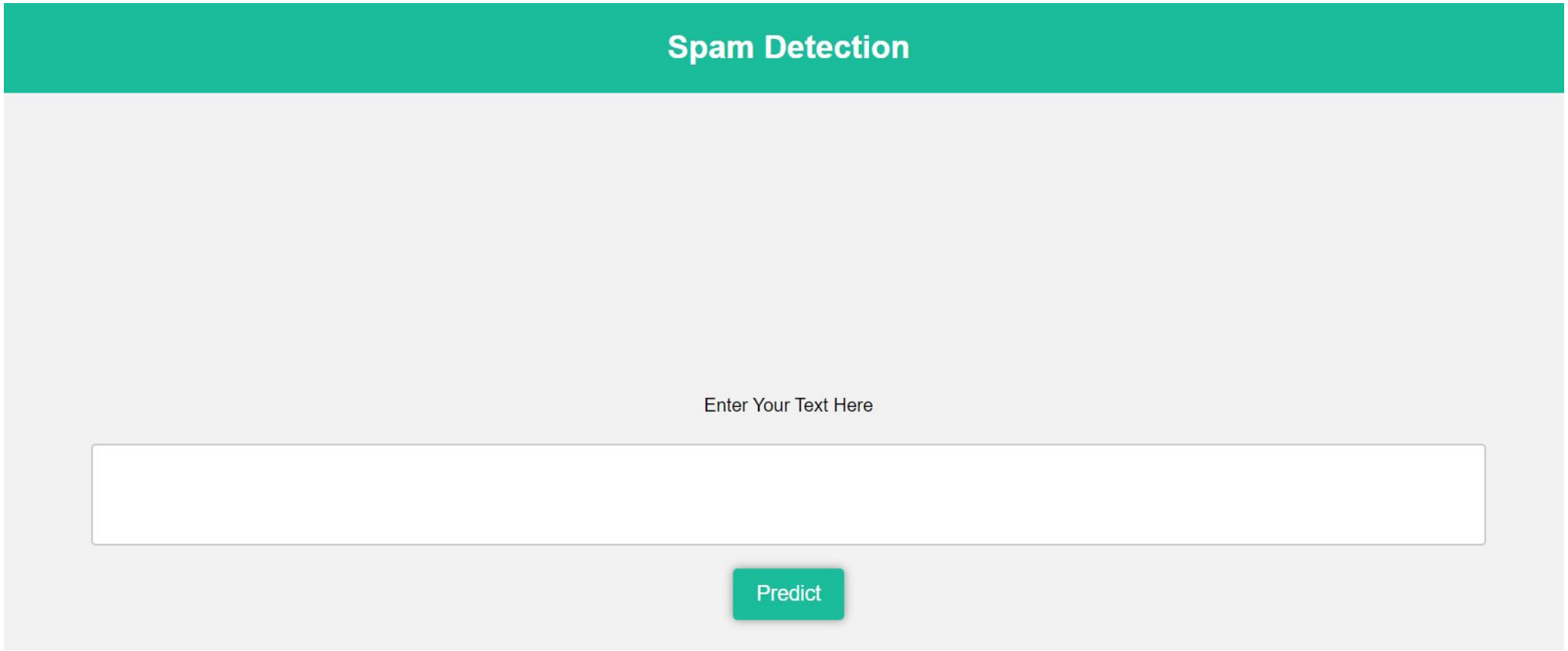
```python
@app.route('/')
def home():
    return render_template('home.html')
```

  - home renders an HTML file where I created a simple form with one text field and a submit button.
    - Users will type text message in the text field and would click on the *Predict* button

```html
<div class="ml-container">

    <form action="{{ url_for('predict')}}" method="POST">
        <p>Enter Your Text Here</p>
        <textarea name="text" rows="4" cols="50"></textarea>
        <br />

        <input type="submit" class="btn-info cool-button" value="Predict">

    </form>

</div>
```

Building Webapp using Flask

- With CSS, the home.html looks like this:



Building Webapp using Flask

- app.py : The controller that controls the app.
- Here I created two app routes:
    - **predict** route, on the other hand loads the trained random forest classifier model
    - Receives the user submitted text from home.html
    - Extracts the features as vectors from the text using TF-IDF that was used during training phase
    - With the help of the trained model, predict the text
    - Forwards the predicted result to result.html

```
@app.route('/predict',methods=['POST'])
def predict():
    with open(model_file, 'rb') as f:
        loaded_model = pickle.load(f)
    loaded_tfidf = pickle.load(open(filename, 'rb'))

    if request.method == 'POST':
        text = request.form['text']
        data = [text]
        data = np.asarray(data)
        data = loaded_tfidf.transform(data).toarray()
        my_prediction = loaded_model.predict(data)
        print(my_prediction[0])
        return render_template('result.html', prediction = my_prediction[0])
```

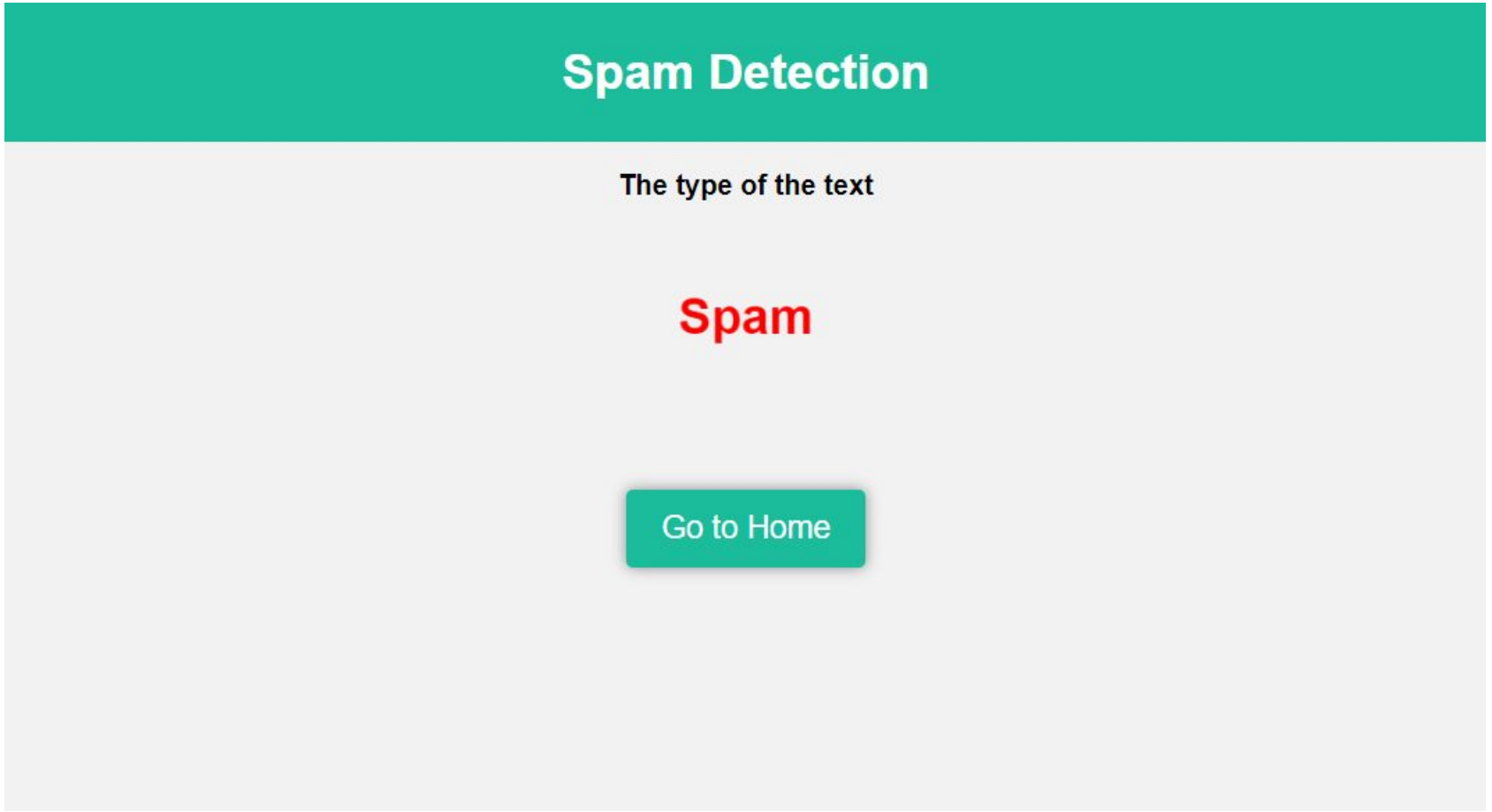
Building Webapp using Flask

- result.html:
  - It receives the value of prediction from predict and prints whether the text is spam or ham
  - Once a user submits a text, he/she views this page the next

```
<body>

    <header>
        <div class="container">

            <h2>Spam Detection</h2>

        </div>
    </header>
    <p style="color:black;font-size:20;text-align: center;"><b>The type of the text</b></p>
    <div class="results">



        {% if prediction == 'spam' %}
        <h2 class="animate-message spam-color">Spam</h2>
        {% elif prediction == 'ham' %}
        <h2 class="animate-message ham-color">It is a Ham</h2>
        {% endif %}
        <div style="margin-top: 2cm;"></div>
        <button type="button" class="cool-button" onclick="location.href='{{ url_for('home') }}'">Go to Home</button>



    </div>

</body>
```

Building Webapp using Flask

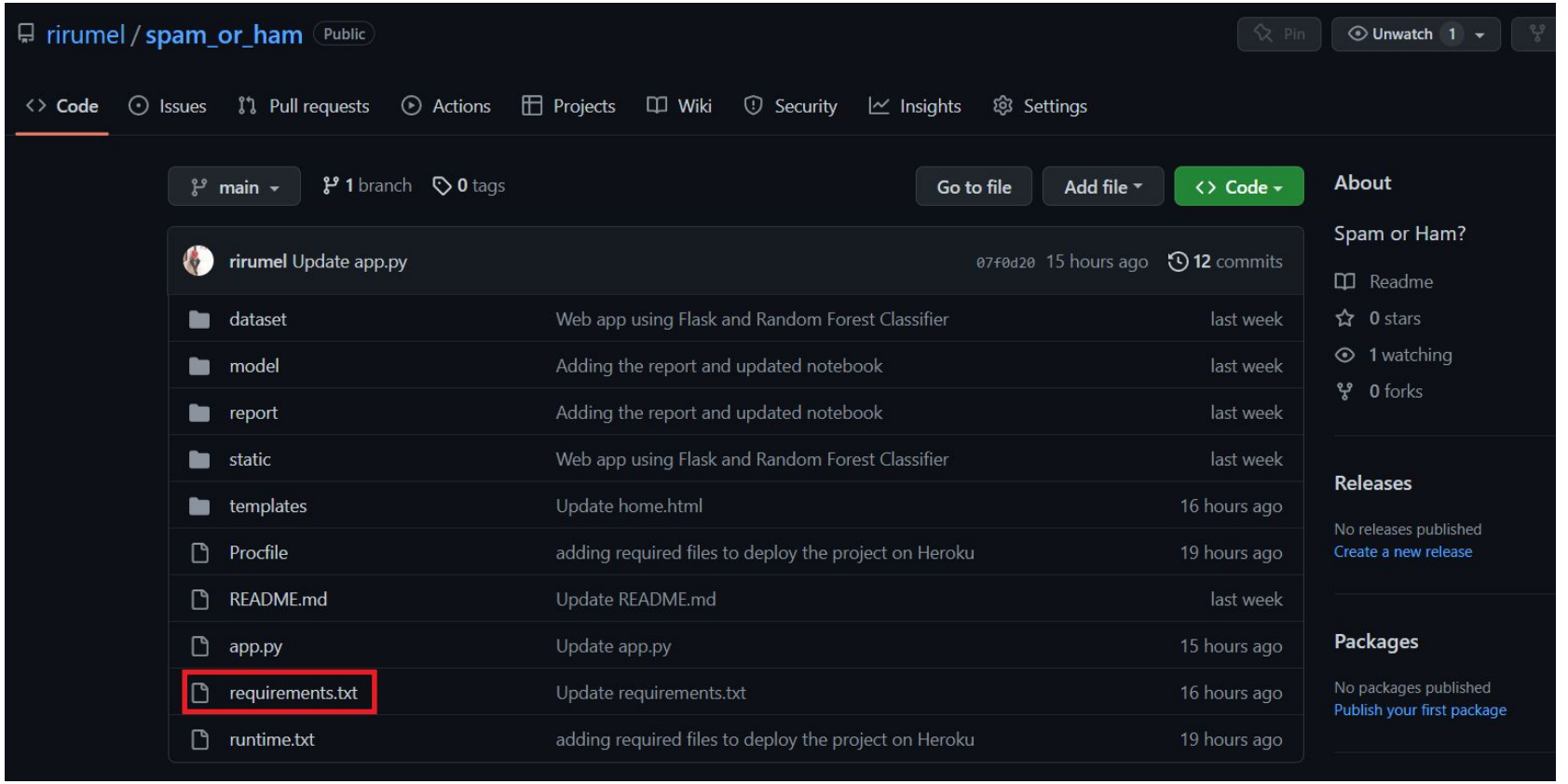
- With CSS, the result.html looks like this:



Deploying Spam Detection web app using Render

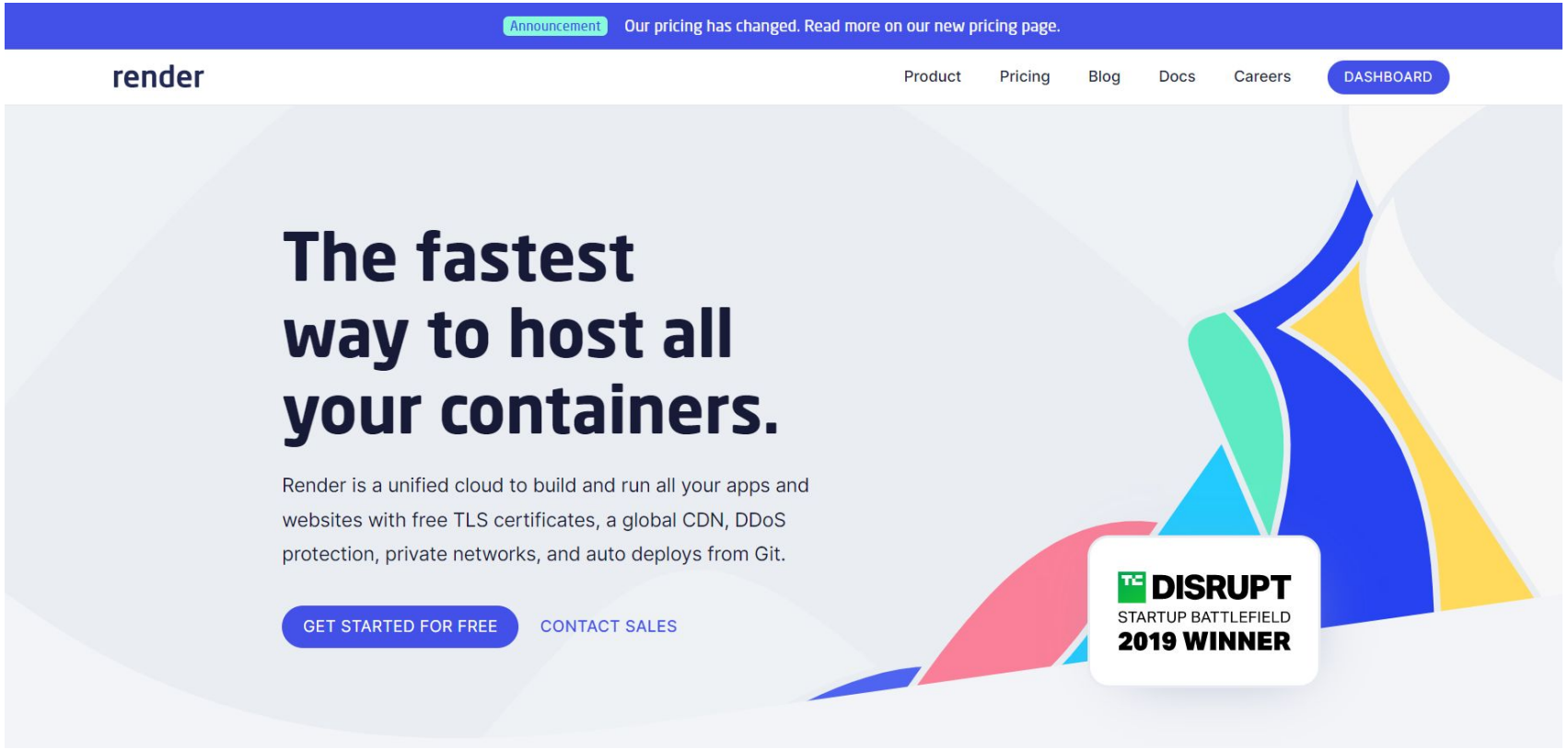
Deploy Spam Detection web app using Render

- I created a Git repository where I added project files and folders, and also added **requirements.txt** which contains a list of project dependencies which is mandatory to deploy my web app.



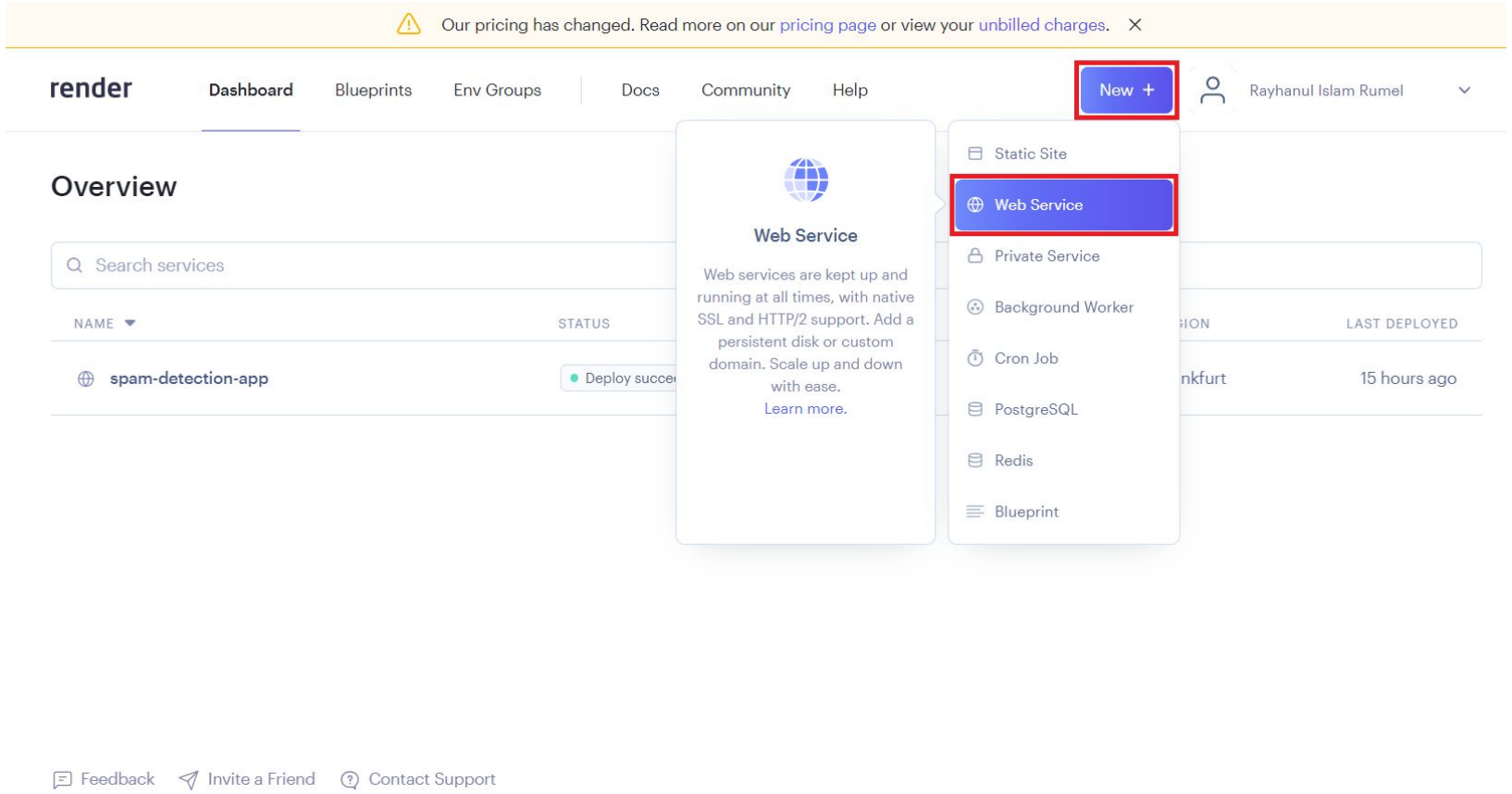
Deploy Spam Detection web app using Render

- Also, created an account at https://render.com/ and logged into the website with my created account.



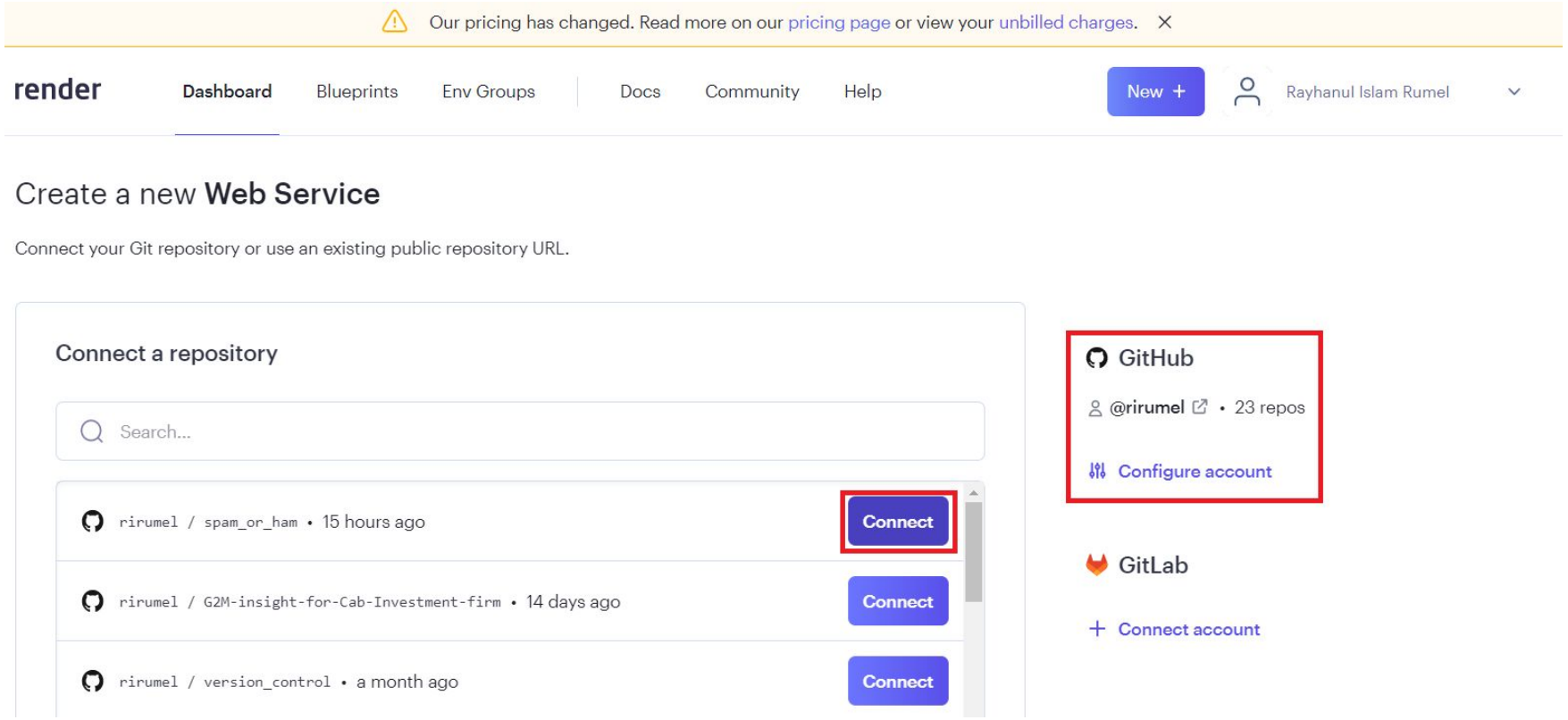
Deploy Spam Detection web app using Render

- Then, I selected the “Web Service” option by clicking on the “New” button.



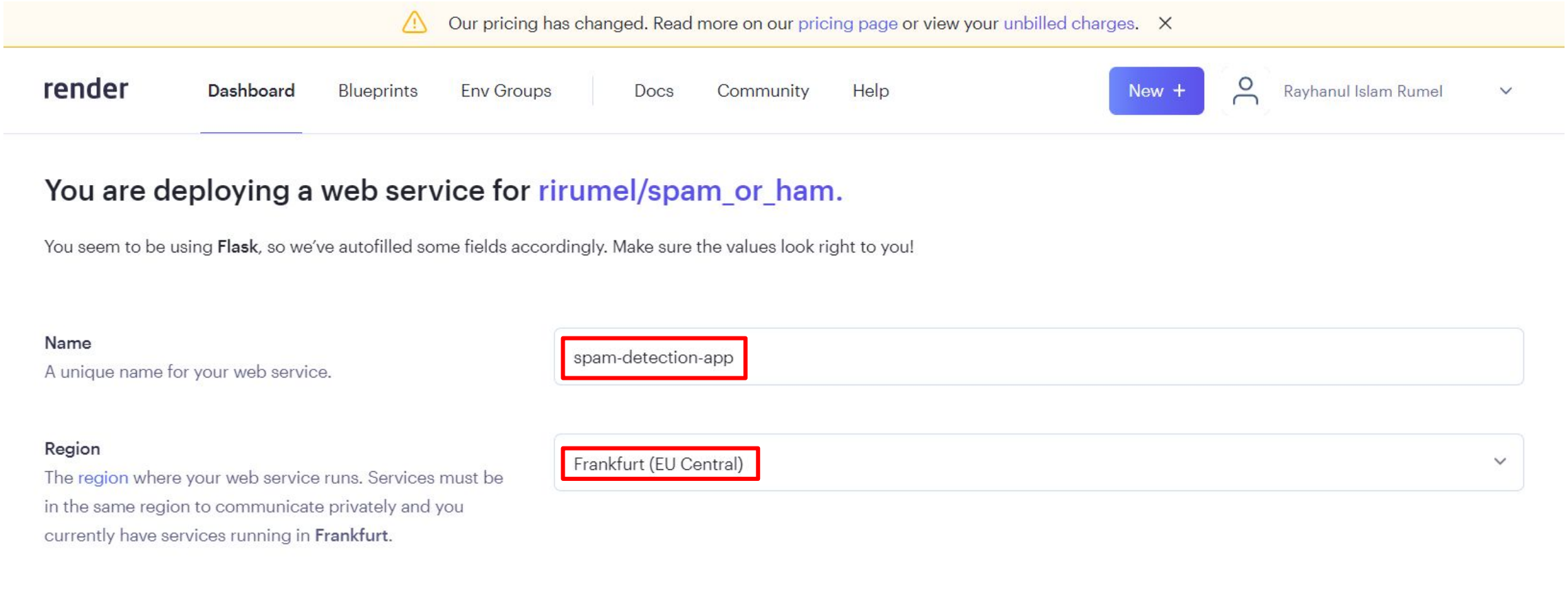
Deploy Spam Detection web app using Render

- Then, I needed to add my GitHub project repository. There are a couple of ways to do it. I signed into my GitHub account from the right panel shown in the picture. As a result, all of my projects are now showing at the center of the page. I clicked on the “Connect” available next to spam_or_ham repository.



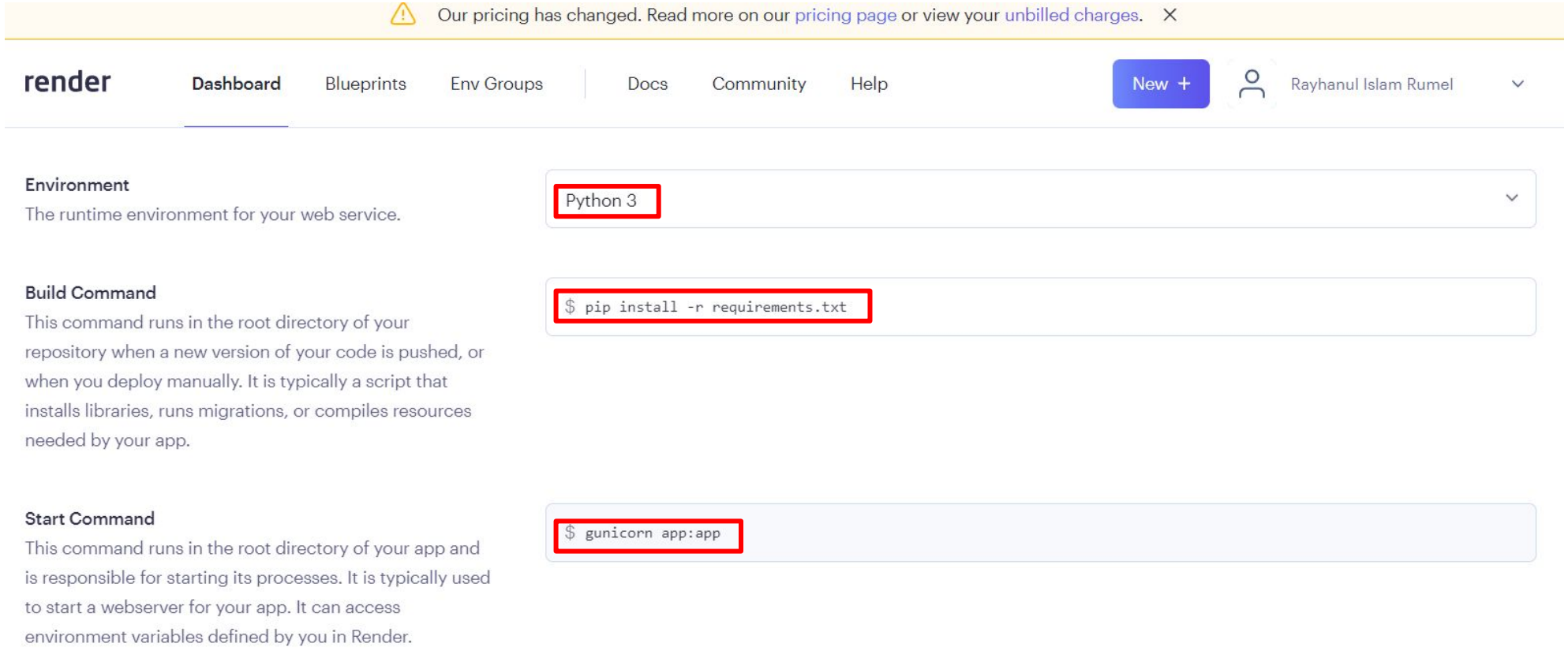
Deploy Spam Detection web app using Render

- Then, it redirected me to a page where I need to fill up a few information in order to successfully start deploying my web app. In this picture, we see that I needed to give info like the name of the web app and region where my app belongs to.



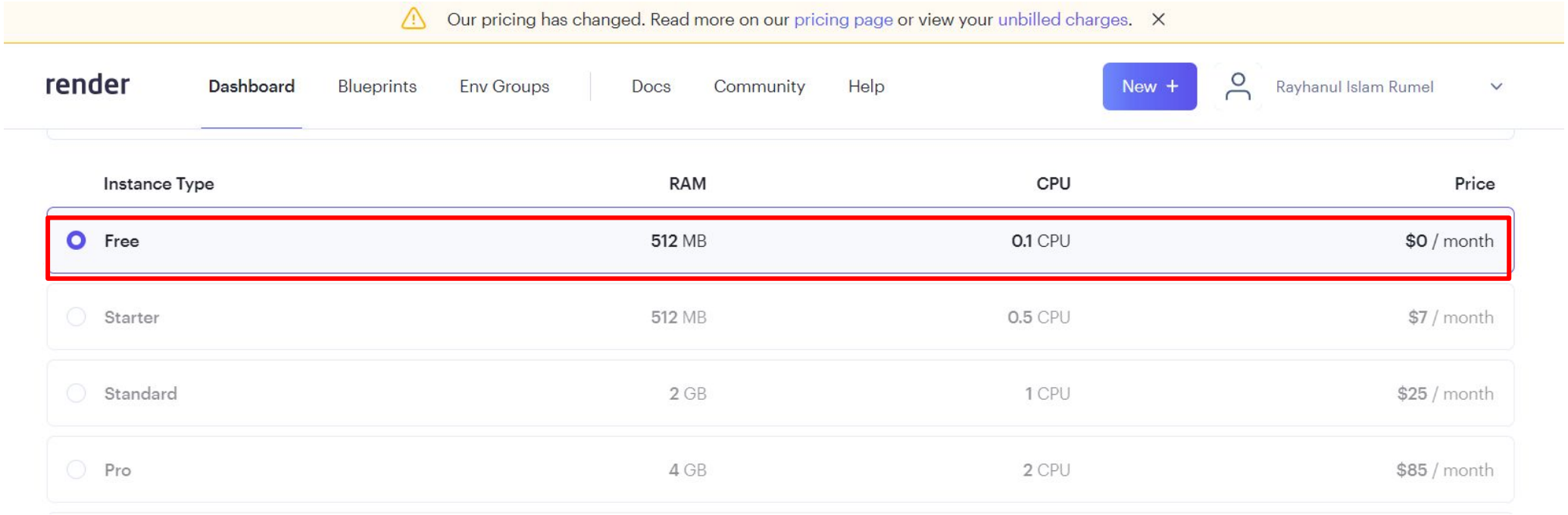
Deploy Spam Detection web app using Render

- Later, I needed to give information like *Environment*, **Python3** in my case, *Build Command,* here I mentioned to install all the packages and dependencies from **requirements.txt**, and *Start Command,* which file to execute at the beginning of the application, in my case it's app.py.



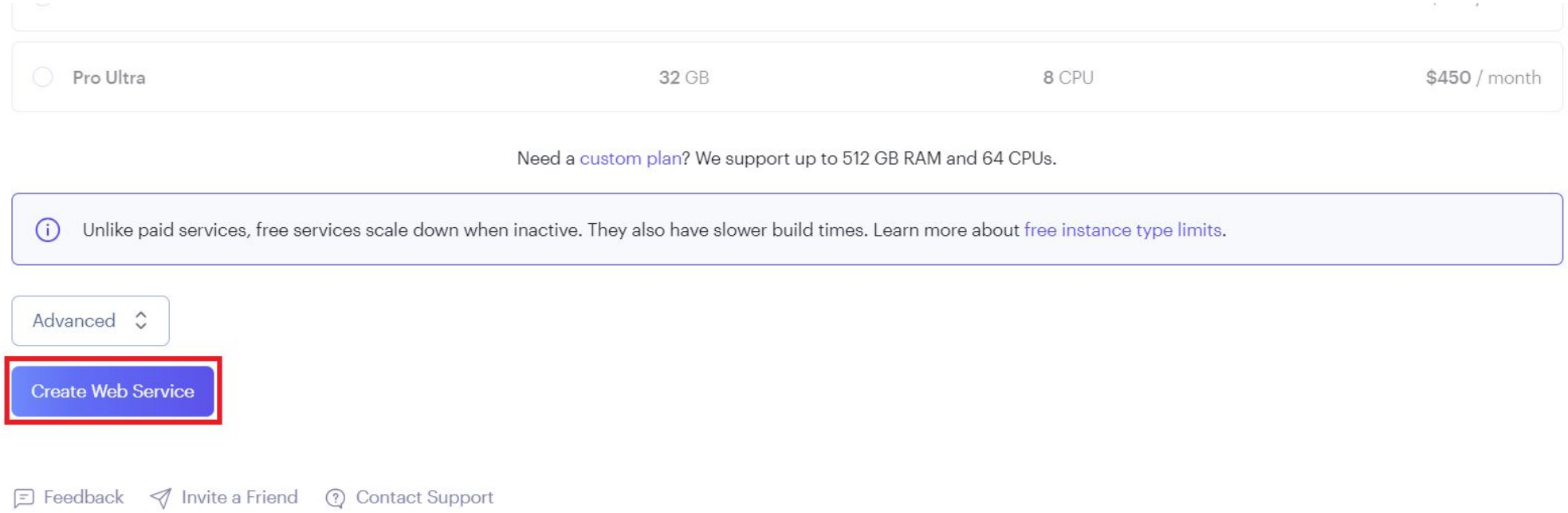
Deploy Spam Detection web app using Render

- Then I needed to choose the insurance type to successfully deploy my web app. There are different plans available. However, for this project I chose the free one.



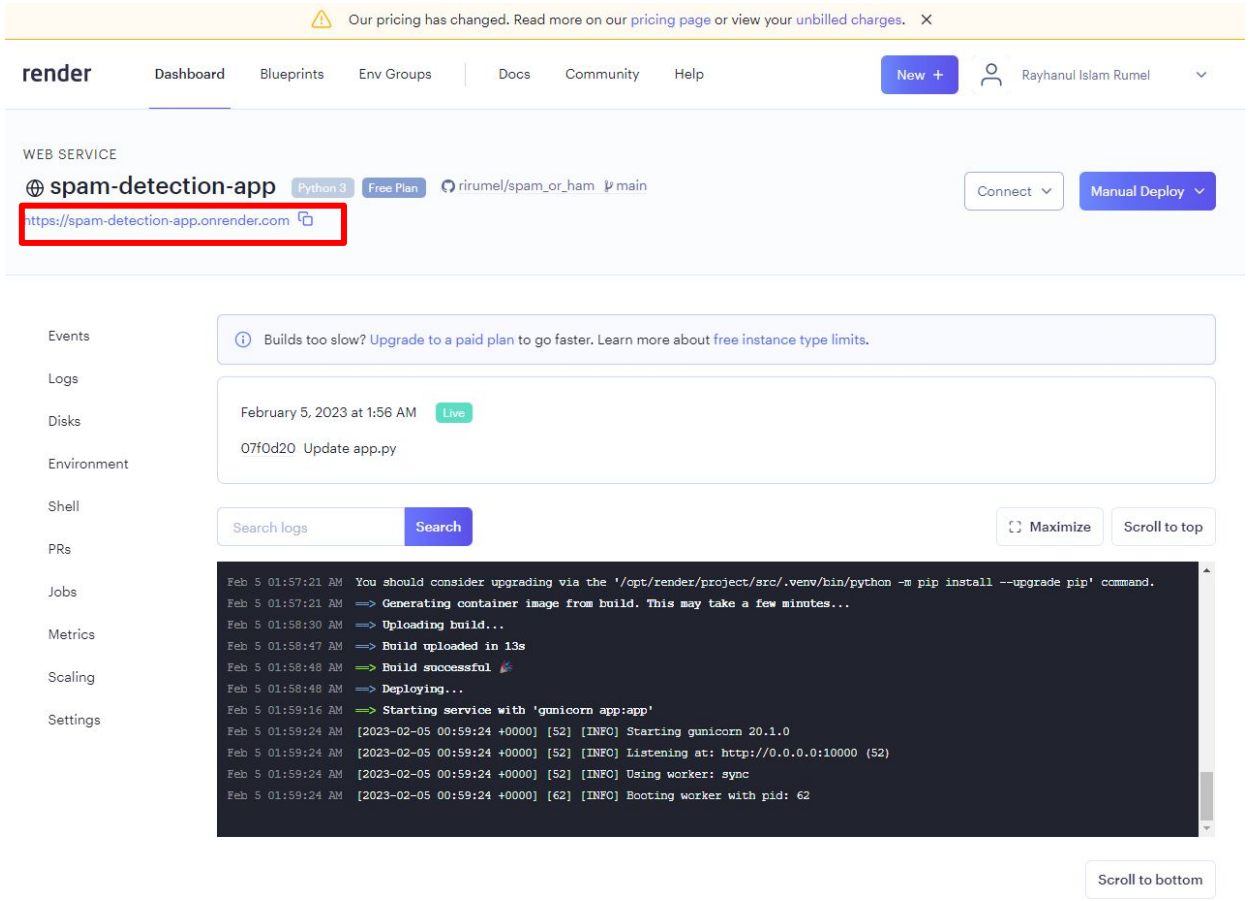
Deploy Spam Detection web app using Render

- Last but not least, I clicked on the “Create Web Service” button to start the deployment process.



Deploy Spam Detection web app using Render

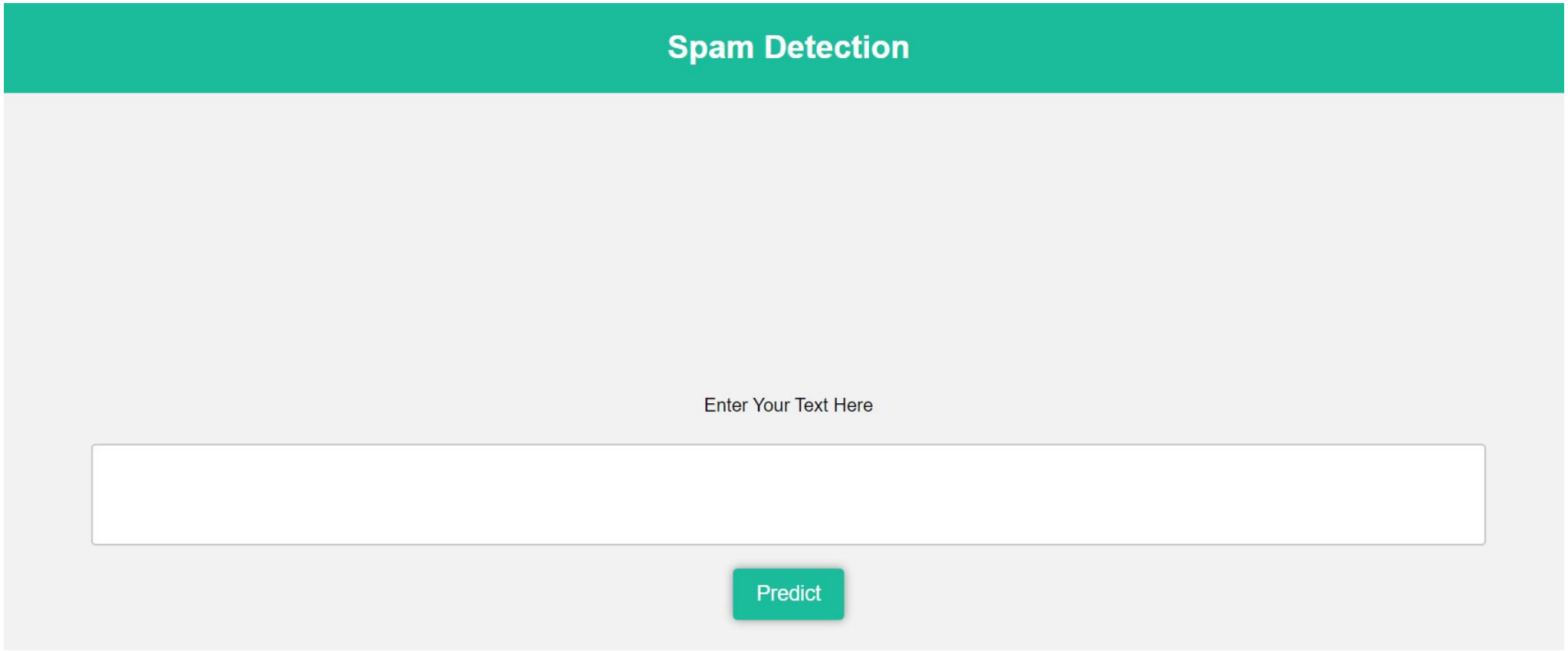
- It took a few minutes to install all the dependencies and deploy the web app. After successful deployment, the web app is now live at: https://spam-detection-app.onrender.com/



User Guide

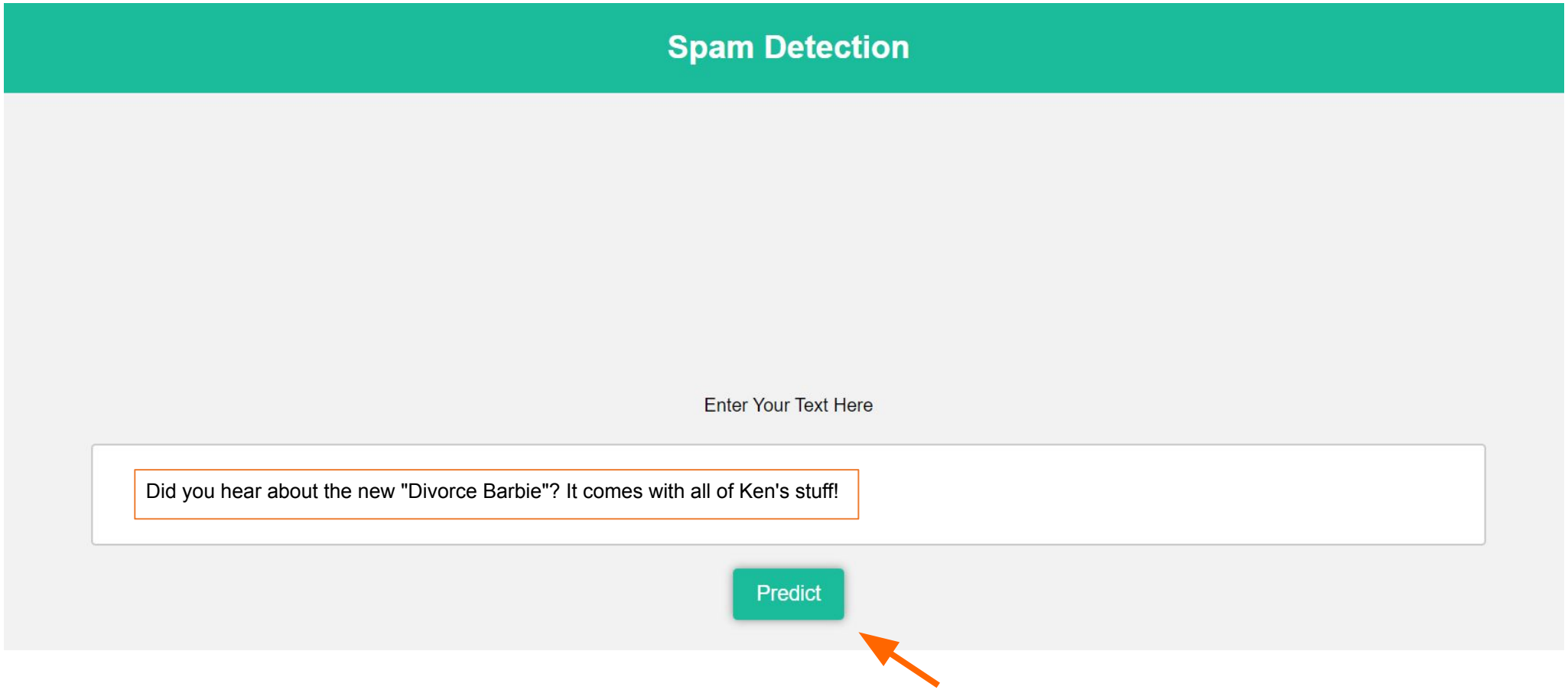
User Guide

- After visiting the homepage of Spam Detection, you would see the following user interface.



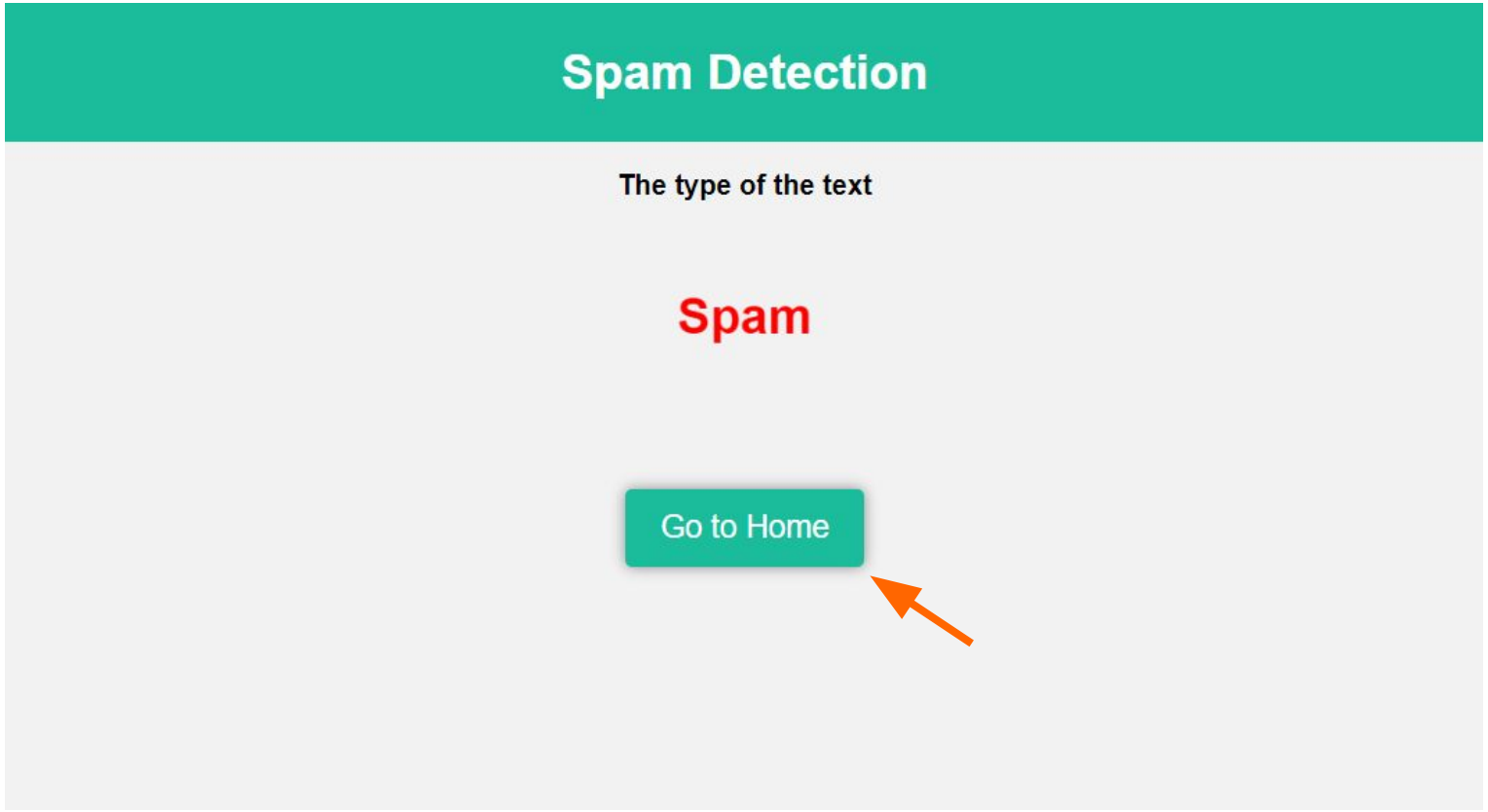
User Guide

- Type your text in the text field and click on the ‘Predict’ button.



User guide

- The app will redirect you to the next page and will show you the type of text you entered.
- You can click on ‘Go to Home’ button to visit the homepage of the app.



Try Yourself
Dummy Dataset

Try Yourself

- Since the model would be expecting unseen text comes from the similar distribution as the training set, I generated some unseen text messages, both spam and ham to check how the app performs in an unseen situation.
- You will find those messages in the next two slides.
- All the messages are grouped into two groups: Spam messages, and Ham messages
- In total 10 text messages are available. Each group contains 5 text messages.
- Copy a message from those slides and paste it on the app to see how good the app performs.

Dummy Unseen Data (Spam)

- **SPAM TEXTS:**

  1) Congratulations! You have been chosen to receive a £1000 prize reward as a valued network customer. Call 09061701461 to claim. Use code KL342. Limited time offer.

  2) Exciting news! You have won a £800 prize reward as a valued network customer. Call 09061701461 to claim your prize. Code KL343. Offer expires in 24 hours.

  3) Hurry! As a valued network customer, you have been selected to receive a £500 prize reward. Call 09061701461 to claim. Code KL344. Offer valid for only 12 hours.

  4) You're a winner! As a valued network customer, you have been selected for a £1200 prize reward. Call 09061701461 to claim. Code KL345. Time-sensitive offer.

  5) Lucky you! You have been selected to receive a £700 prize reward as a valued network customer. Call 09061701461 to claim. Use code KL346. Offer valid for limited hours only.

Dummy Unseen Data (Ham)

- **HAM TEXTS:**

  1) I am at a loss for words to express my gratitude for your kindness. I vow to repay your generosity and keep my word. You have been a constant source of support and a true blessing.

  2) I have been struggling to find the right way to say thank you for this moment of respite. I swear I will not take your generosity for granted and will live up to my promise. Your unwavering support and blessings have been invaluable.

  3) I am overwhelmed by the kindness you have shown me and I am searching for the perfect words to express my thanks. I promise to repay your help and keep my commitment. Your love and blessings have been a constant source of strength.

  4) I cannot find the words to adequately express my gratitude for this break you have given me. I assure you I will not waste your generosity and will honor my promise. You have been a wonderful friend and a constant source of blessings.

  5) My heart is filled with gratitude and I am searching for the right words to thank you for this moment of peace. I will not forget your help and will fulfill my obligations. Your selflessness and blessings have been a true blessing in my life.

Thank You