Model Deployment

In this milestone, we will see the model deployment

Save The Best Model

Saving the best model after comparing its performance using different evaluation metrics means selecting the model with the highest performance and saving its weights and configuration. This can be useful in avoiding the need to retrain the model every time it is needed and also to be able to use it in the future.

```
In [61]: model.save('spam.h5')
```

Integrate With Web Framework

In this section, we will be building a web application that is integrated to the model we built. A UI is provided for the uses where he has to enter the values for predictions. The enter values are given to the saved model and prediction is showcased on the UI.

This section has the following tasks

- Building HTML Pages
- Building server side script
- Run the web application

Building Html Pages

For this project create two HTML files namely

- index.html
- spam.html
- result.html

and save them in the templates folder.

Build Python Code

Import the libraries

In [69]:
```
from flask import Flask,render_template,request
import pickle
import numpy as np
import re
import nltk
from nltk.corpus import stopwords
from nltk.stem import PorterStemmer
from tensorflow.keras.models import load_model
loaded_model = load_model('spam.h5')
cv = pickle.load(open('cv1.pk1','rb'))
app = Flask(__name__)
@app.route('/')
def home():
    return render_template('home.html')
@app.route('/Spam',methods=['POST','GET'])
def prediction():
    return render_template('spam.html')
@app.route('/predict',methods=['POST'])
def predict():
    if request.method == 'POST':
        message = request.form['message']
        data = message
```

```
    new_review = str(data)
    print(new_review)
    new_review = re.sub('[^a-zA-Z]', ' ',new_review)
    new_review = new_review.lower()
    new_review = new_review.split()
    ps = PorterStemmer()
    all_stopwords = stopwords.words('english')
    all_stopwords.remove('not')
    new_review = [ps.stem(word) for word in new_review if not word in set(all_stopwords)]
    new_review = ' '.join(new_review)
    new_corpus = [new_review]
    new_X_test = cv.transform(new_corpus).toarray()
    print(new_X_test)
    new_y_pred =  loaded_model.predict(new_X_test)
    new_X_pred = np.where(new_y_pred>0.5,1,0)
    print(new_X_pred)
    if new_review[0][0]==1:
        return render_template('result.html', prediction="Spam")
    else :
        return render_template('result.html', prediction="Not a Spam")
```

Load the saved model. Importing the flask module in the project is mandatory. An object of Flask class is our WSGI application. Flask constructor takes the name of the current module (__name__) as argument.

Here we will be using a declared constructor to route to the HTML page which we have created earlier.

In the above example, '/' URL is bound with the home.html function. Hence, when the home page of the web server is opened in the browser, the html page will be rendered. Whenever you enter the values from the html page the values can be retrieved using POST Method.

Here we are routing our app to predict() function. This function retrieves all the values from the HTML page using Post request. That is stored in an array. This array is passed to the model.predict() function. This function returns the prediction. And this prediction value will be rendered to the text that we have mentioned in the submit.html page earlier.

```
In [71]: import os
         if __name__=="__main__":

             port=int(os.environ.get('PORT',5000))
             app.run(debug=False)
```

```
* Serving Flask app '__main__'
* Debug mode: off
```

Run The Web Application

- Open anaconda prompt from the start menu
- Navigate to the folder where your python script is.
- Now type "python app.py" command
- Navigate to the localhost where you can view your web page.
- Click on the predict button from the top left corner, enter the inputs, click on the submit button, and see the result/prediction on the web.

```
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
 * Running on http://127.0.0.1:5000
Press CTRL+C to quit
[2023-04-21 12:41:32,537] ERROR in app: Exception on / [GET]
Traceback (most recent call last):
  File "C:\Users\GASCCS23\anaconda3\lib\site-packages\flask\app.py", line 2525, in wsgi_app
    response = self.full_dispatch_request()
  File "C:\Users\GASCCS23\anaconda3\lib\site-packages\flask\app.py", line 1822, in full_dispatch_request
    rv = self.handle_user_exception(e)
  File "C:\Users\GASCCS23\anaconda3\lib\site-packages\flask\app.py", line 1820, in full_dispatch_request
    rv = self.dispatch_request()
  File "C:\Users\GASCCS23\anaconda3\lib\site-packages\flask\app.py", line 1796, in dispatch_request
    return self.ensure_sync(self.view_functions[rule.endpoint])(**view_args)
  File "C:\Users\GASCCS23\AppData\Local\Temp\ipykernel_1416\2211322710.py", line 14, in home
    return render_template('home.html')
  File "C:\Users\GASCCS23\anaconda3\lib\site-packages\flask\templating.py", line 146, in render_template
    template = app.jinja_env.get_or_select_template(template_name_or_list)
  File "C:\Users\GASCCS23\anaconda3\lib\site-packages\jinja2\environment.py", line 1081, in get_or_select_template
    return self.get_template(template_name_or_list, parent, globals)
  File "C:\Users\GASCCS23\anaconda3\lib\site-packages\jinja2\environment.py", line 1010, in get_template
    return self._load_template(name, globals)
  File "C:\Users\GASCCS23\anaconda3\lib\site-packages\jinja2\environment.py", line 969, in _load_template
    template = self.loader.load(self, name, self.make_globals(globals))
  File "C:\Users\GASCCS23\anaconda3\lib\site-packages\jinja2\loaders.py", line 126, in load
    source, filename, uptodate = self.get_source(environment, name)
  File "C:\Users\GASCCS23\anaconda3\lib\site-packages\flask\templating.py", line 62, in get_source
    return self._get_source_fast(environment, template)
  File "C:\Users\GASCCS23\anaconda3\lib\site-packages\flask\templating.py", line 98, in _get_source_fast
    raise TemplateNotFound(template)
jinja2.exceptions.TemplateNotFound: home.html
127.0.0.1 - - [21/Apr/2023 12:41:33] "GET / HTTP/1.1" 500 -
127.0.0.1 - - [21/Apr/2023 12:41:33] "GET /favicon.ico HTTP/1.1" 404 -
```

- **This is the main page of Spam Detection , where you can know about the project and also from this page users can click onto the spam button and they will redirect onto the spam/ prediction page for providing the inputs.**