

## Deploying a ML web app

**Name:** Juan Carlos Gutiérrez

**Batch code:** NLP01

**Submission date:** November 15th

**Submitted to:** Data Glacier

I decided to generate my own data on a spam classification problem. It is composed of three variables only, but it works as an example upon which I applied Naïve Bayes and let me to learn about this kind of algorithms.

I coded a small app that deployed as a web app using the framework Flask.

These are snapshots to follow up on the process.

1. Script to generate data and save it to a csv file.

```
import numpy as np
import pandas as pd

my_array = np.random.choice(('spam', 'not spam'), size=1000, p=(0.5, 0.5))
tags = [['ad', 'not an ad'], ['phishing', 'not phishing'], ['unknown', 'contact']]

for tag_list in range(len(tags)):
    my_array = np.column_stack((my_array, np.random.choice(tags[tag_list], size=1000, p=(0.5, 0.5))))

df = pd.DataFrame(my_array, columns=['filter', 'tag_1', 'tag_2', 'tag_3'])
df.to_csv('Generated_data.csv', index=False, mode='w')
print('A dataframe has been generated and saved to a csv file.')
```

2. Modelling data applying a Naïve Bayes classifier .

```
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.naive_bayes import CategoricalNB
from sklearn.metrics import classification_report, confusion_matrix
import pickle

df = pd.read_csv('Generated_data.csv')
print(df.head())

for col in df.columns:
    plt.hist(df[col])
    plt.title(f'Distribution of {col}')
    plt.savefig(f'Distribution of {col}.png')
    plt.close()

df.replace({'ad':1, 'not an ad':2,
            'phishing':1, 'not phishing':2,
            'unknown':1, 'contact':2},
           inplace=True)
train = df[:int(0.7*len(df))]
test = df[int(0.7*len(df)):]
```

## Deploying a ML web app

```
X_train = train.drop('filter', axis=1)
y_train = train['filter']
X_test = test.drop('filter', axis=1)
y_test = test['filter']

catNB = CategoricalNB()
model = catNB.fit(X_train, y_train)

predictions = model.predict(X_test)

clf_report = classification_report(y_test, predictions)
conf_matrix = confusion_matrix(y_test, predictions)
print(clf_report)
print(conf_matrix)

with open('Naive Bayes model.pkl', 'wb') as f:
    pickle.dump(model, f)
    print('Pickling completed.')
```

### 3. Building the app using Flask.

```
from flask import Flask, request, render_template
import pickle
import numpy as np

app = Flask(__name__, template_folder='templates')
model = pickle.load(open('Naive Bayes model.pkl', 'rb'))

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

@app.route('/', methods=['POST'])
def predict():
    user_input = int(request.form.get('ad')), int(request.form.get('phishing')), int(request.form.get('sender'))
    prediction = model.predict(np.array([user_input]))
    return 'Your email will be categorized as ' + str(prediction[0])

if __name__ == '__main__':
    app.run(port=5000, debug=True)
```

## Deploying a ML web app

4. Designing the HTML template to show on the home page.

```
<strong style="font-size:40px">Categorize your email!</strong>

<form method="post">

  <select name="ad">
    <option value="1">It's an ad</option>
    <option value="2">It's not an ad</option>
  </select>

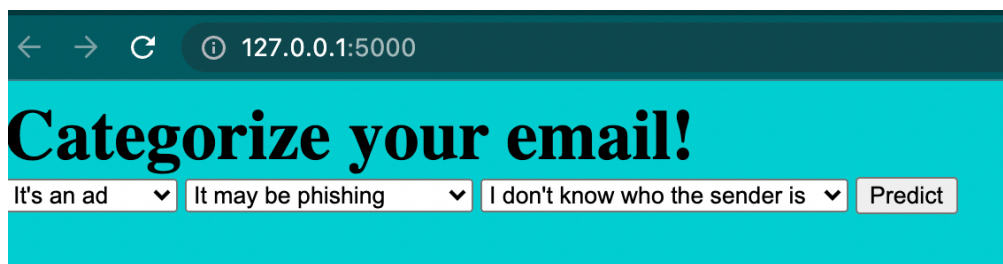
  <select name="phishing">
    <option value="1">It may be phishing</option>
    <option value="2">Not phishing apparently</option>
  </select>

  <select name="sender">
    <option value="1">I don't know who the sender is</option>
    <option value="2">The sender is a contact of mine</option>
  </select>

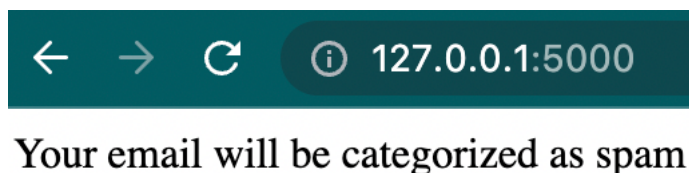
  <input type="submit" value="Predict">
</form>

<body style="background-color:rgb(0, 206, 209);">
</body>
```

5. Running the app locally.



A screenshot of a web browser window. The address bar shows '127.0.0.1:5000'. The page has a teal background. At the top, the text 'Categorize your email!' is displayed in a large, bold, black serif font. Below this, there are three dropdown menus. The first dropdown is labeled 'It's an ad' and has a downward arrow. The second dropdown is labeled 'It may be phishing' and has a downward arrow. The third dropdown is labeled 'I don't know who the sender is' and has a downward arrow. To the right of these dropdowns is a white button with the text 'Predict' in black.



A screenshot of a web browser window. The address bar shows '127.0.0.1:5000'. Below the address bar, the text 'Your email will be categorized as spam' is displayed in a black serif font.