BigScale Analytics

Week 9: How to create a web application using Flask.

Flask is a lightweight WSGI web application framework. It is designed to make getting started quick and easy, with the ability to scale up to complex applications. It began as a simple wrapper around Werkzeug and Jinja and has become one of the most popular Python web application frameworks. Check out the documentation here.

In this lab we are going to make a simple web application in which we can enter the information about a customer of a telecommunication company and get the prediction as he/she will churn or not. We have used the telecom customer churn dataset from Kaggle. We have provided you a notebook called "training.ipynb" in which we have preprocessed the data and trained a logistic regression model. The trained model is then saved as a joblib file. The we application will be using this trained model to predict churn for the input parameters. In the following we will go through some basics of making a flask app.

Installing flask: you can easily install and update flask using pip:

```
pip install flask
```

Application script: the web app starts by running a Python script, let's call it "application.py". This script usually has a specific structure. As an example look at this simple "Hello World" application.

```
from flask import Flask

app = Flask(__name__)

@app.route("/")
def hello():
    return "Hello, World!"

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

By running "application.py", the web app starts at the local host address (usually 127.0.0.1:5000) and the message "Hello World" will be shown. Here, as you can see, we only have a root url, but it is also possible to have several pages with different urls within a flask app.

HTML templates: It is also possible to have html template for you application. In flask all the html templates should be in a folder called "templates". The `render_template` function helps you to render the html files.

GET and POST methods: GET and POST are two basic http methods. GET is used when you enter the url of a webpage and by hitting enter you get the content of the page. POST is used when you are sending some data to the server through the web application. Inside the flask app you can determine wether a GET or POST request is called by using the `requests` library. That is:

```
if request.method == 'POST':
     <DO SOMETHING>
else:
     <...>
```

You can also send parameters to your flask app by using a form inside the html file (more on that in the lab). More one http methods and retrieving forms data: https://www.youtube.com/watch?v=9MHYHgh4jYc&list=PLzMcBGfZo4-n4vJJybUVV3Un_NFS5EOgX&index=4

Go to the Github repository of the course under week 9. By running the script "application.py" you can start your customer churn prediction web application.