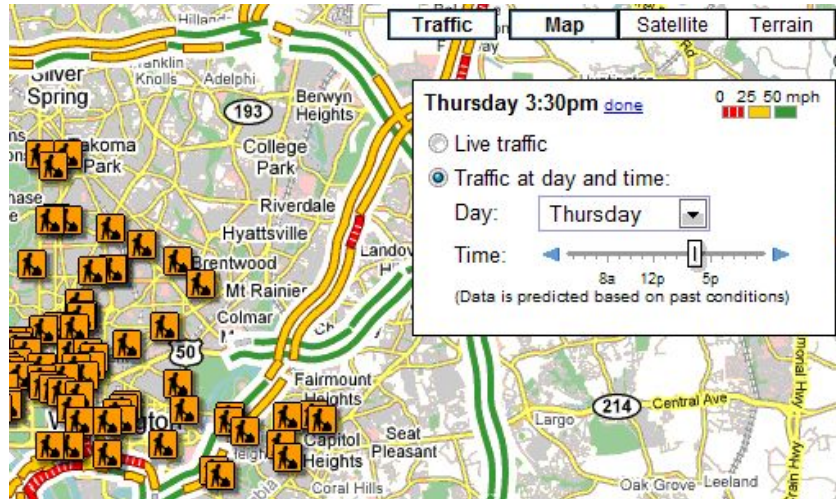


## CS499 Cloud Computing and Big Data Hackathon 3 - Google Maps Traffic Prediction



### Due Date

**SHOW US IN CLASS: Monday 3:50pm, March 6, 2017**

**SAME DAY DELIVERY: Monday 11:59pm, March 6, 2017**

**IF YOU ARE BUSY: Tuesday 11:59pm, March 7, 2017**

**FINAL DUE: Wednesday 2:00pm, March 8, 2017**

Note: Bonus/penalties will be considered based on these times/dates. Generally, the earlier you can submit, the better. The later you submit, the more likely you will lose points, and we will be picky on the functions and code quality. No submission will be allowed after Wed 2:00pm. Don't ask questions on late submissions.

*"Embrace what you don't know, especially in the beginning, because what you don't know can become your greatest asset. It ensures that you will absolutely be doing things different from everybody else."*

-- Sara Blakely

### Score

50

## **Directions**

Let's build a simple prototype to predict the traffic!

In this project you will work with one partner (or just yourself) to develop a machine learning backend to predict road traffic and a very simple frontend page to interact with your backend. You will be working with Python, Python Flask, and scikit-learn, and you will need to deploy your python web service to an EC2.

The core of this application is to develop the traffic prediction training model. Below is the suggested traffic data prediction dataset schema:

*<road id, direction, dayOfWeek, timeOfDay> => <traffic status>*

For instance,

```
<10, 0, 1, 8>  => <8>  // for I-10 west, Monday at 8am, the traffic is 8 (1-10)
<10, 0, 2, 9>  => <7>
<10, 1, 1, 13> => <5>
<5, 0, 1, 15>  => <6>
<5, 1, 5, 16>  => <9>  // for I-5 south, Friday at 4pm, the traffic is 9
<5, 1, 6, 18>  => <9>
```

You can design your own dataset schema and feel free to add more dimensions to make the model more accurate.

### **1. Create an API to receive the sample data**

This API accepts one sample data record each time. Thus, make sure you can pass all the dimensions through this API (e.g., road id, direction, day, time, status). Each time you receive a data record, you should 1) save this new record into your dataset; 2) trigger the training of the model to include the new data record.

When saving the dataset, you can decide where you want to save it. You can simply save it in the memory, or you can save it in the disk. You can also use a database which would be more professional.

You should use scikit-learn to train (i.e., fit) a model. Feel free to use any machine learning model you can find, as long as you think it is reasonable for this problem. We will not measure the accuracy of the prediction.

The final result of this API call is that every time you pass a new data record, the training model will be updated and ready to make predictions. You don't need to return anything for this API, as long as you have the correct HTTP 200 response.

## ***2. Create an API to prediction the traffic status***

This API should accept a new given data record (e.g., 10, 1, 4, 10), and it should return a prediction of the traffic status (e.g., 4) using the current trained model.

## ***3. Provide a simple HTML page for us to try***

Once you are done with Task 1 and 2, please make one simple HTML frontend page to let us play around with your service. You do NOT need a fancy page. Just one simple page that allows us to 1) input the new data record and submit it to trigger the training process (API 1); 2) input a new data record and get the predicted traffic status (API 2). You might need to use jQuery or AngularJS to implement this. You can take a look at how to serve static html file from Flask: <http://stackoverflow.com/questions/20646822/how-to-serve-static-files-in-flask>

## ***4. Deploy to EC2***

You need to deploy your web service to an EC2, and remember that you need to do this configuration to open your web service to the public:

```
app.run(host='0.0.0.0')
```

(see <http://askubuntu.com/questions/224392/how-to-allow-remote-connections-to-flask>)

## **Submission**

All the code must be checked in to GitHub, but DO NOT check in your AWS credentials.

You need to submission the following:

1. Team member names (only one member needs to submit it)
2. The URL to your GitHub repo that contains the web service source code
3. **A readme file in your GitHub repo that explains the machine learning algorithm and model you used**
4. The URL to the frontend page

Please use Google Form (<https://goo.gl/forms/xXYlgcWEDkcdMhGq2>) to submit your work.

### **After Submission**

Include this project in your resume. The keywords should be “Cloud Computing In-class Hackathon”, “Google Maps Traffic Prediction”, “Machine Learning”, “Python”, “Python Flask”, “scikit-learn”, etc.