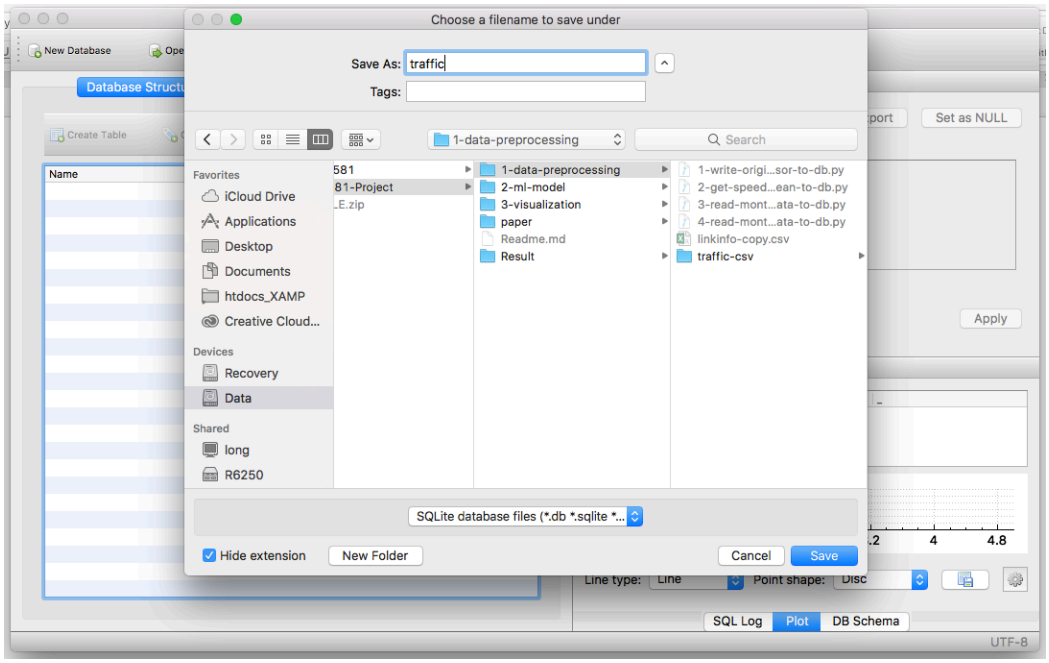
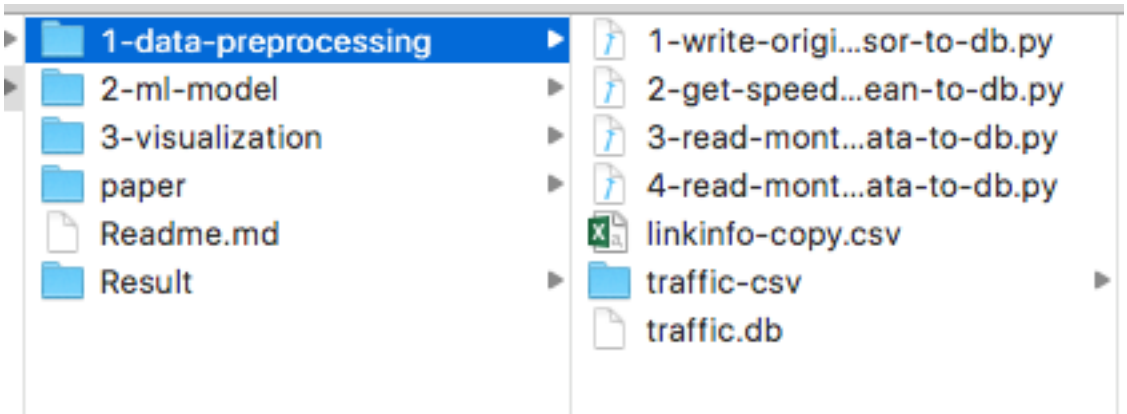
Software demo

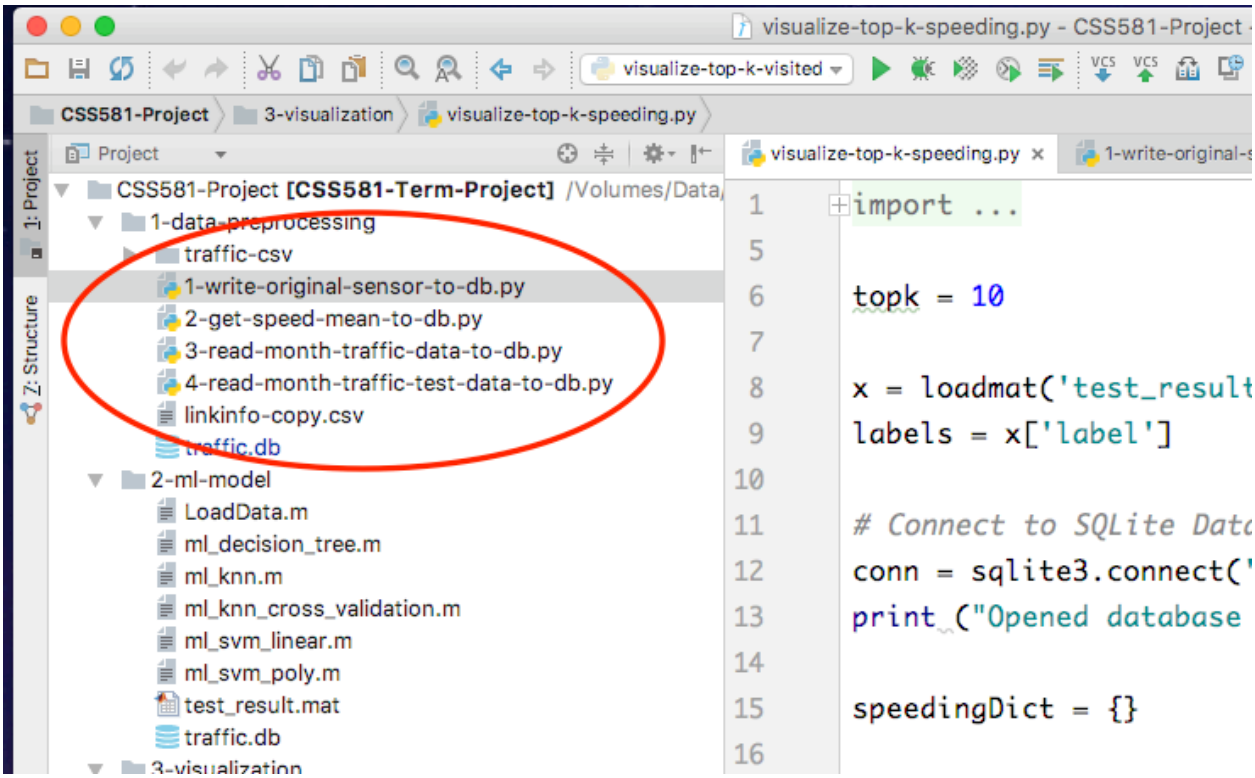
Steps in Data Pre-processing:
Objective: Re-organize data and write them into the SQLite DB

1. Use DB Browser or command line to create the empty database and save as “traffic.db” to the folder '1-data-preprocessing'



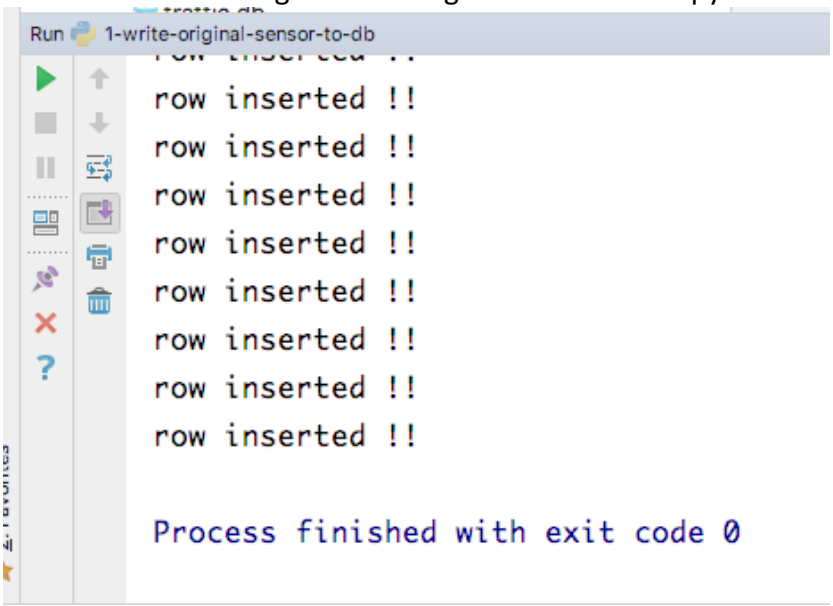


2. Run Python scripts in the folder '1-data-preprocessing':
   2.1: Run '1-write-original-sensor-to-db.py' to store the sensor info into the db
   2.2: Run '2-get-speed-mean-to-db.py' to store the speed limit of set sensors into the db
   2.3: Run '3-read-month-traffic-data-to-db.py' to store the training data into the db
   2.4: Run '4-read-month-traffic-test-data-to-db.py' to store the testing data into the db

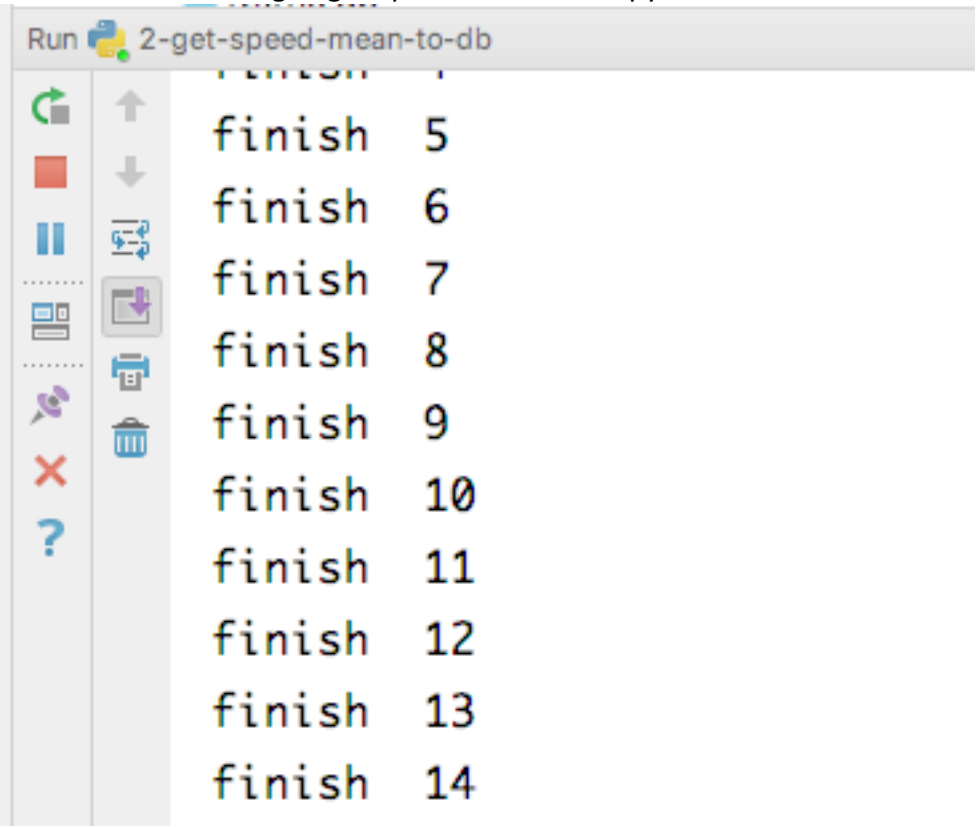


Run those scripts by the ascending order marked as part of the script name: 1 -> 2 -> 3 -> 4

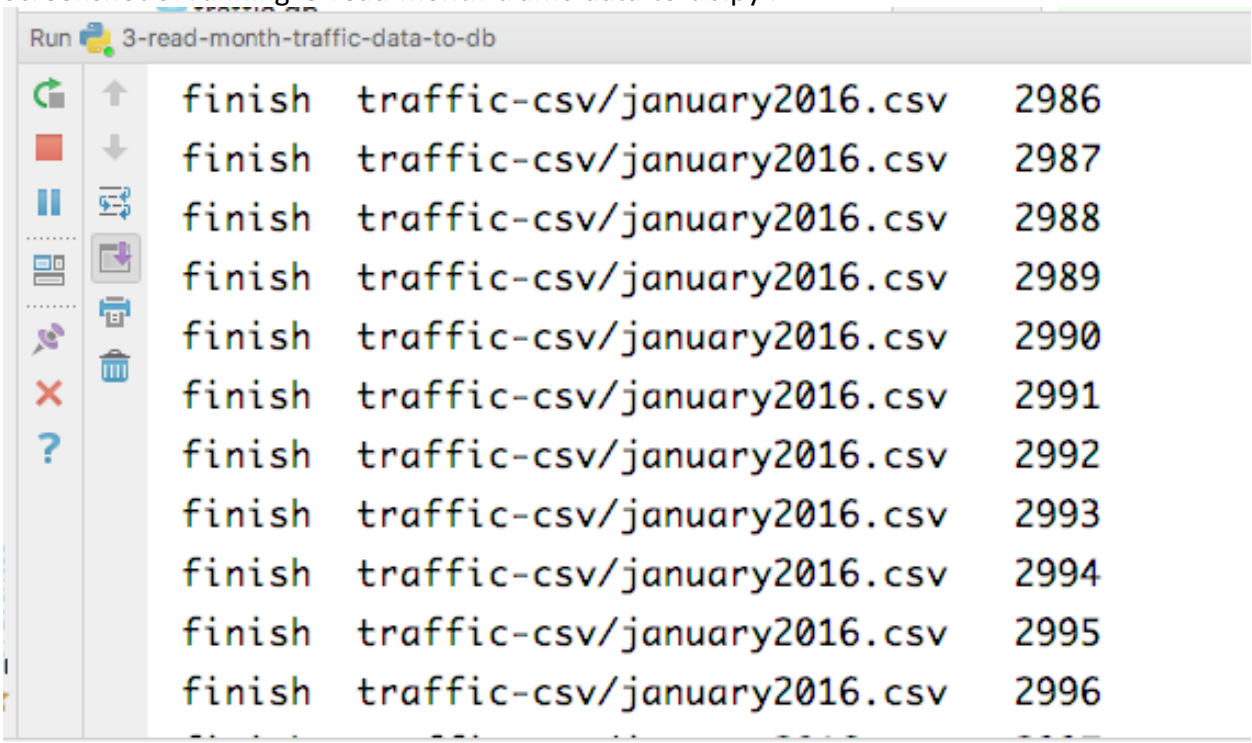
Screenshot of running '1-write-original-sensor-to-db.py':



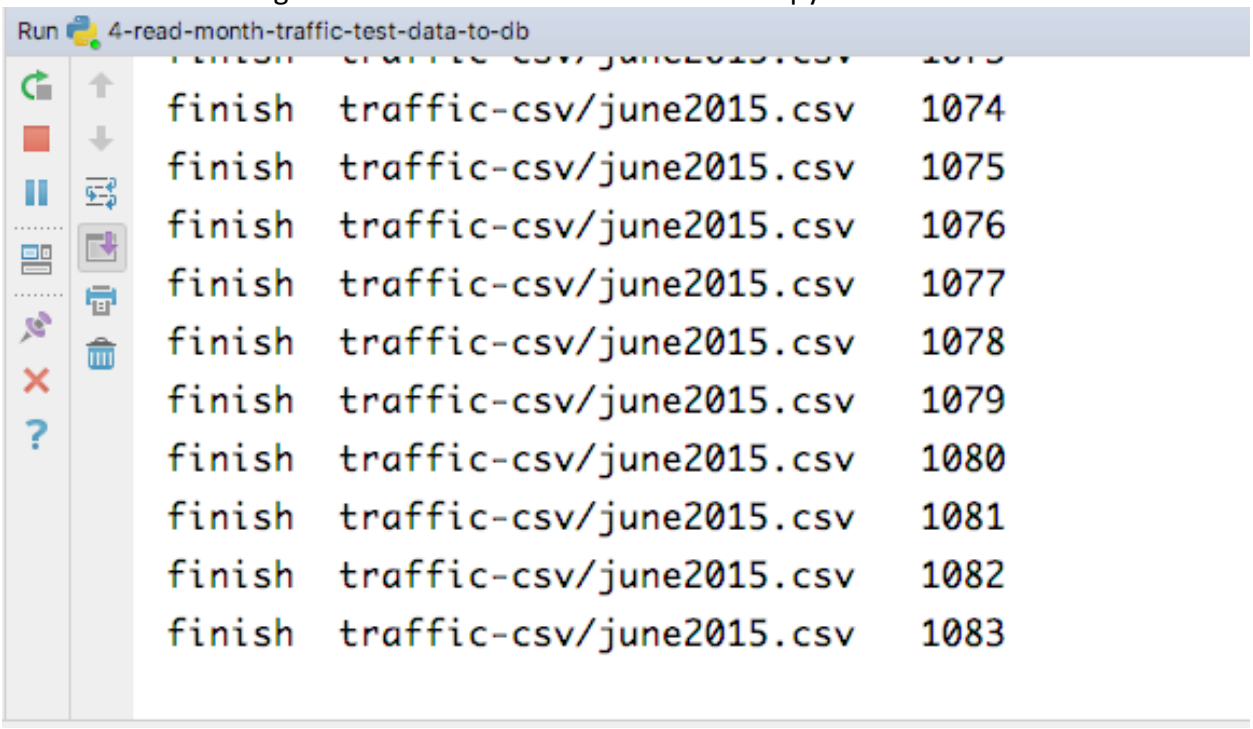
Screenshot of running '2-get-speed-mean-to-db.py':



Screenshot of running '3-read-month-traffic-data-to-db.py':

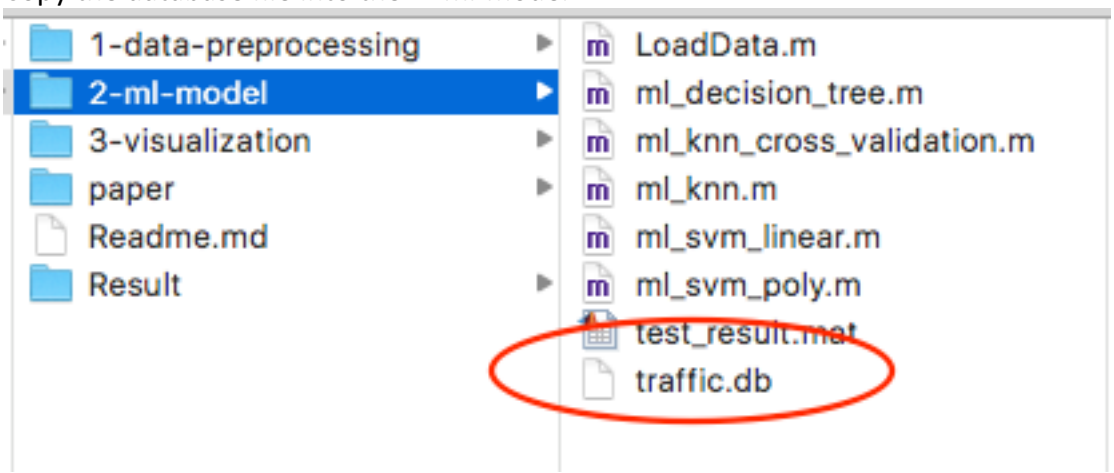


Screenshot of running ‘4-read-month-traffic-test-data-to-db.py’:



Steps in ML prediction:
Objective: Prediction records’ location

3. Copy the database file into the ‘2-ml-model’



4. Open Matlab, then:
    4.1: Run 'ml_decision_tree.m' (Decision Tree model)
    4.2: Run 'ml_knn.m' (KNN algorithm, where parameters are given in the 10 - fold cross validation)
    4.3: Run 'ml_svm_linear.m' (SVM with linear kernel)
    4.4: Run 'ml_svm_poly.m' (Optional, my paper doesn't include it because it takes long time for training)

Result of decision tree:

```
Decision Tree Accuracy: 0.943000
Decision Tree F1-Score (Mean): 0.923890
Decision Tree Precision (Mean): 0.938776
Decision Tree Recall (Mean): 0.918303
>>
```

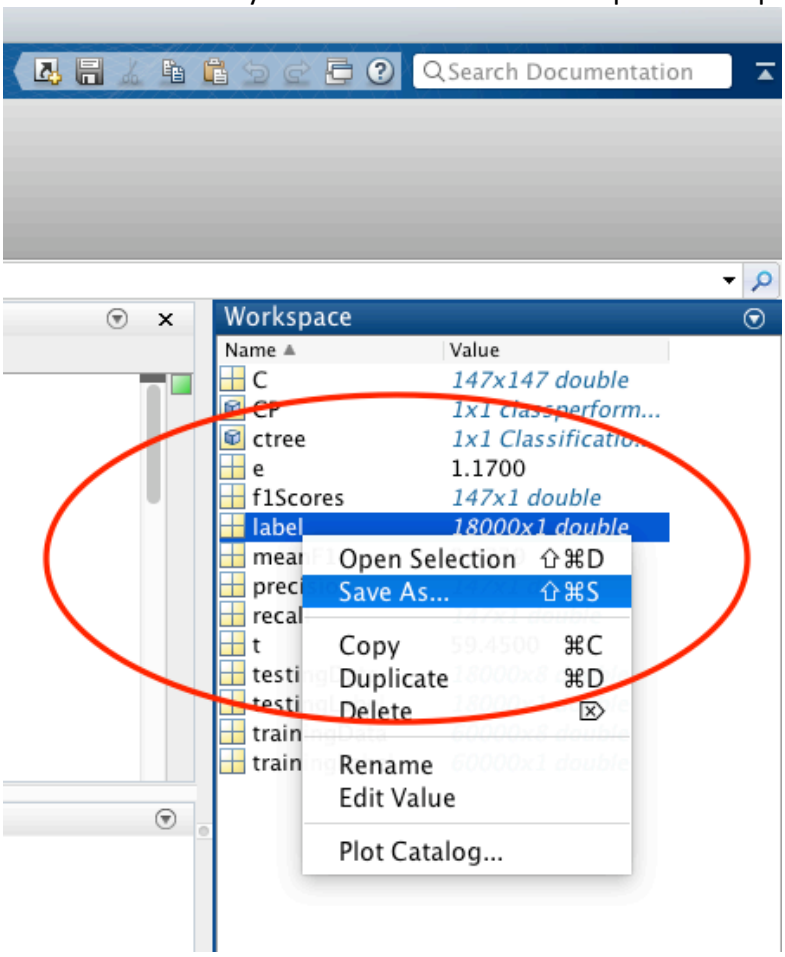
Result of KNN:

```
KNN Accuracy: 0.852167
KNN Accuracy F1-Score (Mean): 0.821214
KNN Precision (Mean): 0.841200
KNN Recall (Mean): 0.817189
>>
```

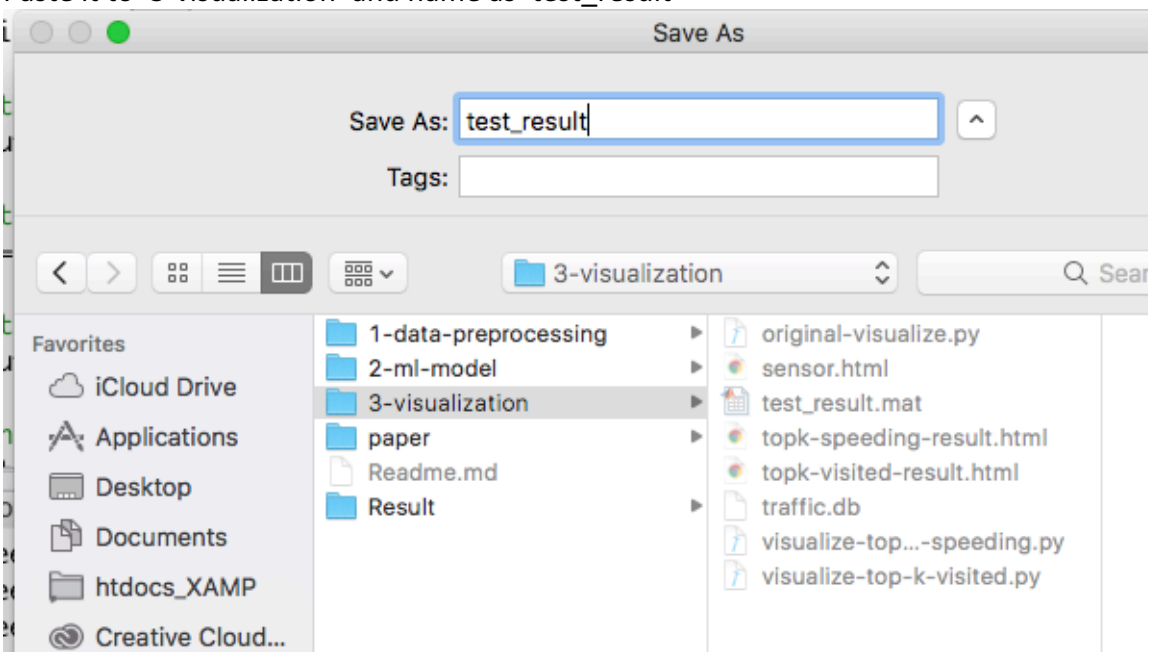
Result of Linear SVM:

```
SVM Linear Accuracy: 0.854722
SVM Linear F1-Score (Mean): 0.821214
SVM Linear Precision (Mean): 0.841200
SVM Linear Recall (Mean): 0.817189
>>
```

5. Choose the model you want to visualize and export its output

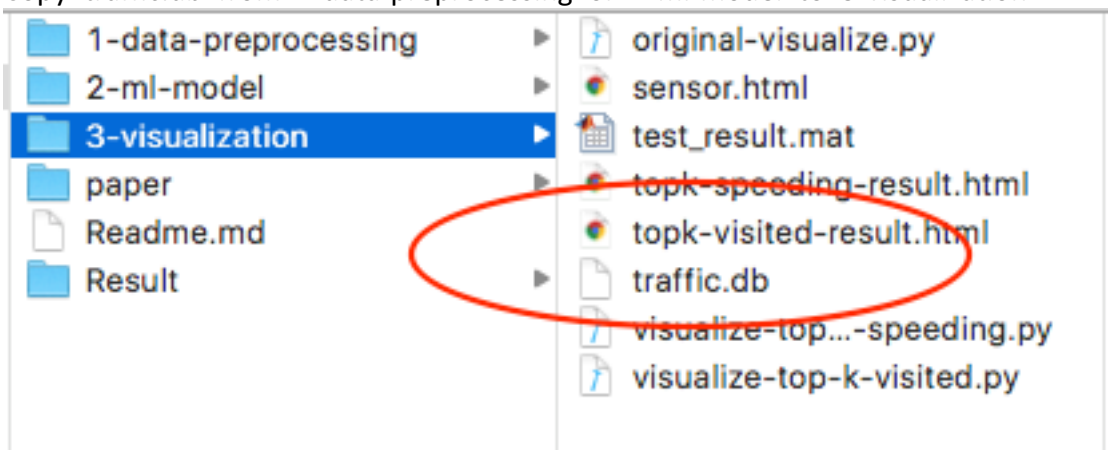


6. Paste it to '3-visualization' and name as 'test_result'



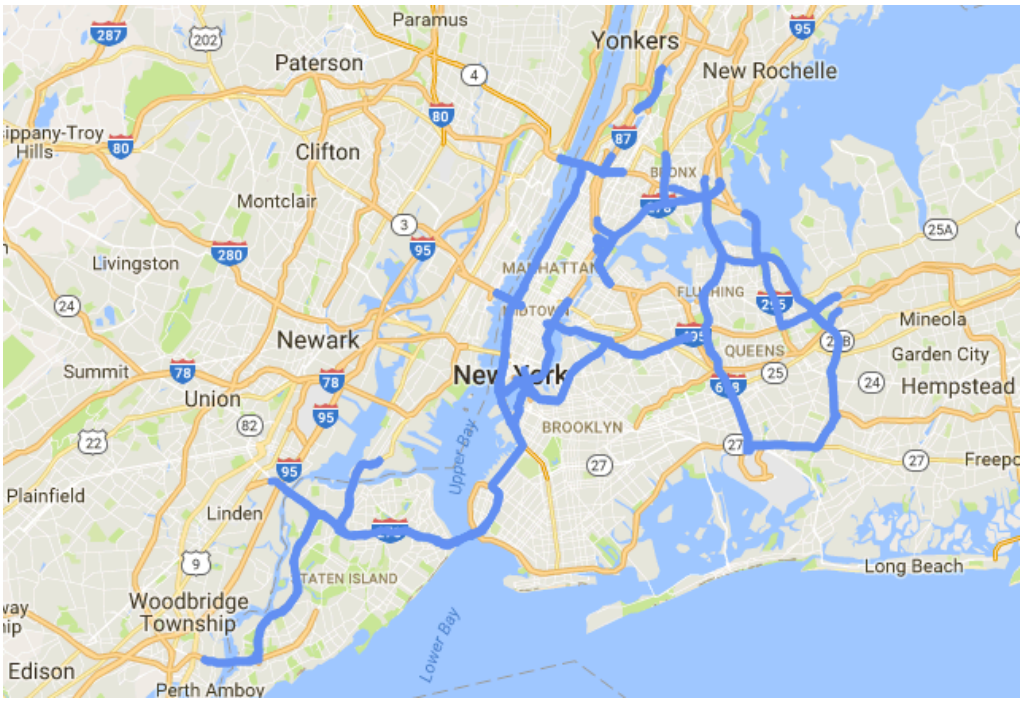
Steps in data visualization: (Show the result)

7. Copy ‘traffic.db’ from ‘1-data-preprocessing’ or ‘2-ml-model’ to ‘3-visualization’

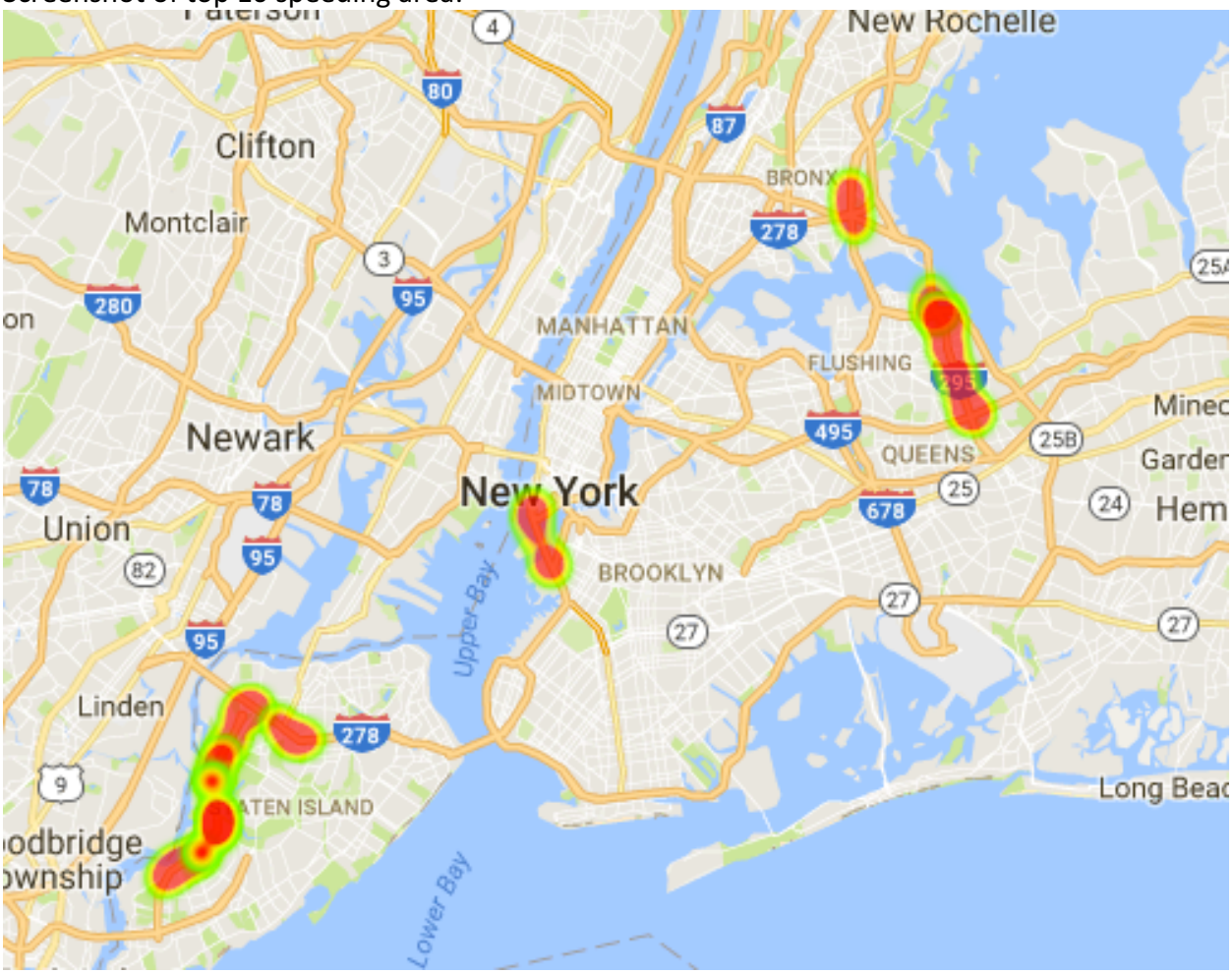


8. Run following scripts:
   8.1: Run 'original-visualize.py' and it will export 'sensor.html' including the sensor locations
   8.2: Run 'visualize-top-k-speeding.py' and it will export 'topk-speeding-result.html' including the top 10 speeding area
   8.3: Run 'visualize-top-k-visited.py' and it will export 'topk-visited-result.html' including the top 10 places that vehicles visited

   Screenshot of sensors’ location:



Screenshot of top 10 speeding area:



Screenshot of top 10 places visited: