DSCI 551 Midterm Report -- Los Angeles traffic collision and weather

Xinrui Ying, Houqingchen Zhu, Mingliao Xu

• Check List

Timeline	Task	Detailed Task	Status
Week 5	Data Storage and Processing	Preprocess both weather and traffic collision data and conduct data cleaning. Store the data in Firebase and MySQL.	Complete
Week 7	Modeling and Analysis	Build data models for both datasets and find relationships between them.	Complete

So far we are on track to achieve our milestones.

• Data Challenges/Problems

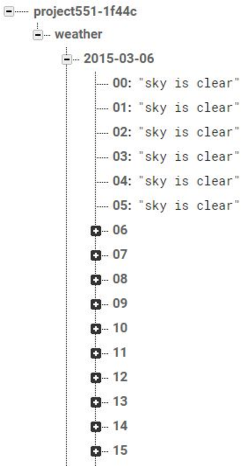
1. Data Processing:
    a. Los Angeles traffic collision data retrieved from Kaggle:
    https://www.kaggle.com/cityofLA/los-angeles-traffic-collision-data?select=traffic-collision-data-from-2010-to-present.csv
    The original data was in .csv format. It was written case by case given specific traffic occurred time with details like crime code, victim information, report dates, etc. However the only data that was related to achieving our goal wereD Date Occurred, Time Occurred, and the Area Name where traffic happened. We cleaned the data at 2012-10-01 as weather data had missing values.
    So we processed the data and wrote a python script that converted the data into a json file.

    b. Weather conditions retrieved from Kaggle:
    https://www.kaggle.com/selfishgene/historical-hourly-weather-data?select=weather_description.csv
    The original data was in .csv format. This dataset included the weather description for more than 30 countries while we only need hourly weather data for Los Angeles. We also cleaned the data at 2012-10-01 as weather data had missing values. We processed the data in the original csv file and wrote a python script that converted the data into a json file as we decided to store it both in Firebase and MySQL.

2. Data Storage:
    a. Los Angeles traffic collision data retrieved from Kaggle:
    We uploaded it to Firebase Realtime Database at
    https://project551-1f44c.firebaseio.com/traffic.json

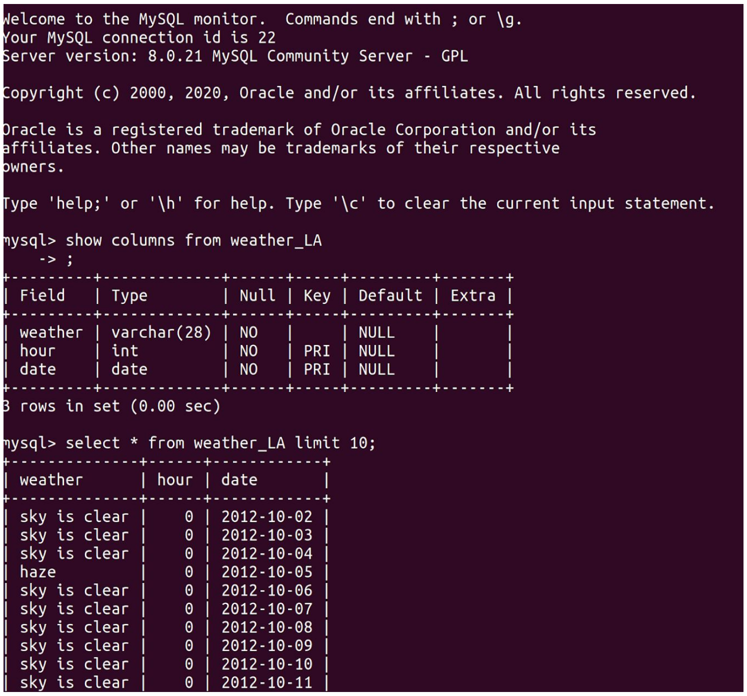
```
{"2010-01-01":{"00":{"77th
Street":3,"Foothill":1,"Harbor":1,"Hollenbeck":1,"Southwest":1,"Topanga":1},"01":
{"Harbor":1,"Hollenbeck":1,"Mission":1,"N
Hollywood":1,"Olympic":1,"Southwest":2,"Topanga":1},"02":{"77th
Street":1,"Hollywood":2,"Mission":1,"Southeast":1,"Southwest":1,"West LA":1,"Wilshire":2},"03":
{"Hollenbeck":1,"Hollywood":3,"N Hollywood":1,"Newton":1,"Olympic":1,"Southeast":1,"West
LA":2},"04":{"77th Street":1,"Hollenbeck":1,"Southeast":1,"West LA":1},"05":{"77th
Street":1,"Devonshire":1,"Southwest":2},"06":{"77th Street":1,"N Hollywood":1,"Olympic":1},"07":
{"Mission":1},"08":{"Rampart":1,"Van Nuys":1},"09":{"77th
Street":1,"Northeast":1,"Southwest":1},"10":
{"Central":1,"Foothill":1,"Northeast":2,"Southwest":1},"11":
{"Hollywood":1,"Pacific":1,"Rampart":1,"Southwest":1},"12":
{"Central":1,"Newton":3,"Southeast":1,"Topanga":1},"13":{"Devonshire":1,"N
Hollywood":1,"Southeast":1,"Topanga":1},"14":
{"Central":1,"Devonshire":1,"Hollywood":1,"Northeast":1,"Pacific":1,"West LA":1},"15":
{"Central":1,"Devonshire":1,"Hollenbeck":1,"Olympic":1,"Pacific":1,"Southeast":1,"Van
Nuys":1},"16":
```

b. Weather conditions retrieved from Kaggle:
We uploaded it to Firebase Realtime Database at:
https://project551-1f44c.firebaseio.com/weather.json



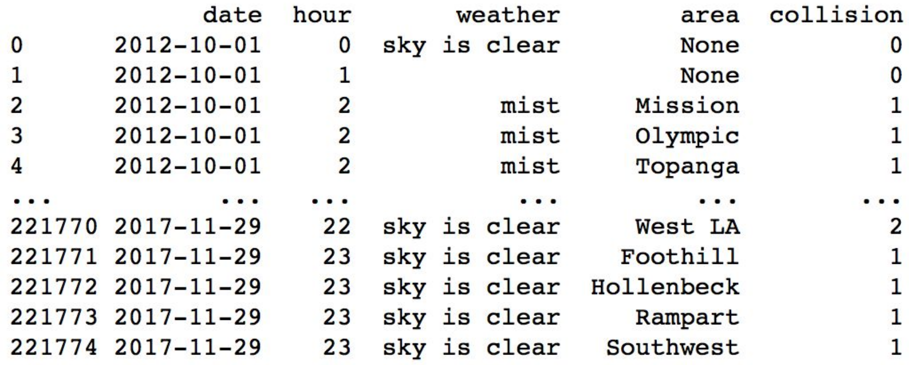
```
project551-1f44c
  weather
    2015-03-06
      00: "sky is clear"
      01: "sky is clear"
      02: "sky is clear"
      03: "sky is clear"
      04: "sky is clear"
      05: "sky is clear"
      06
      07
      08
      09
      10
      11
      12
      13
      14
      15
```

We also imported the processed data into MySQL database.



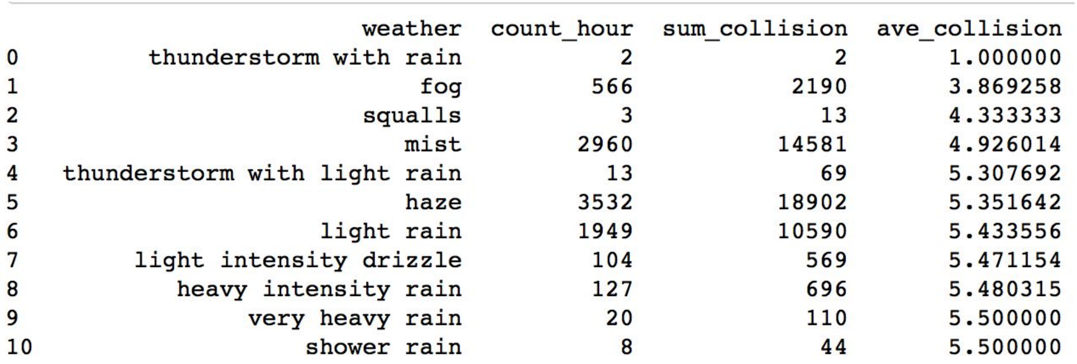
```
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 22
Server version: 8.0.21 MySQL Community Server - GPL

Copyright (c) 2000, 2020, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> show columns from weather_LA
    -> ;
+---------+-------------+------+-----+---------+-------+
| Field   | Type        | Null | Key | Default | Extra |
+---------+-------------+------+-----+---------+-------+
| weather | varchar(28) | NO   |     | NULL    |       |
| hour    | int         | NO   | PRI | NULL    |       |
| date    | date        | NO   | PRI | NULL    |       |
+---------+-------------+------+-----+---------+-------+
3 rows in set (0.00 sec)

mysql> select * from weather_LA limit 10;
+--------------+------+------------+
| weather      | hour | date       |
+--------------+------+------------+
| sky is clear |    0 | 2012-10-02 |
| sky is clear |    0 | 2012-10-03 |
| sky is clear |    0 | 2012-10-04 |
| haze         |    0 | 2012-10-05 |
| sky is clear |    0 | 2012-10-06 |
| sky is clear |    0 | 2012-10-07 |
| sky is clear |    0 | 2012-10-08 |
| sky is clear |    0 | 2012-10-09 |
| sky is clear |    0 | 2012-10-10 |
| sky is clear |    0 | 2012-10-11 |
```

3. Data Integration:
   In our project, the relation between two datasets was quite clear -- date and hour.



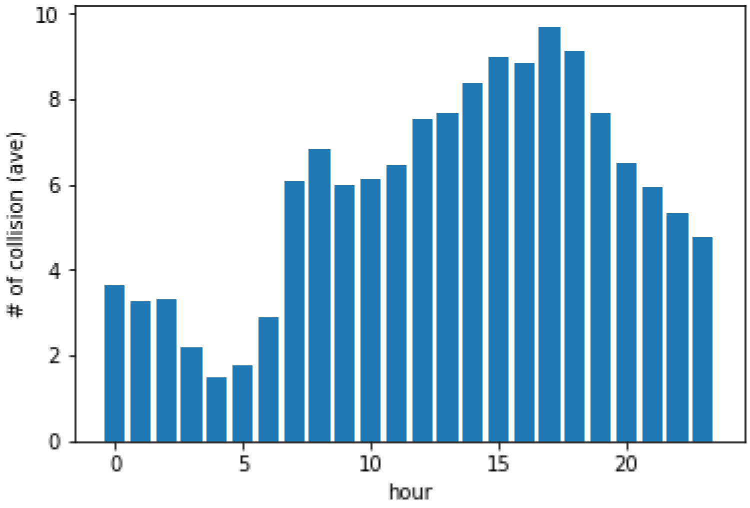
```
              date  hour       weather        area  collision
0       2012-10-01     0  sky is clear        None          0
1       2012-10-01     1                      None          0
2       2012-10-01     2          mist     Mission          1
3       2012-10-01     2          mist     Olympic          1
4       2012-10-01     2          mist     Topanga          1
...            ...   ...           ...         ...        ...
221770  2017-11-29    22  sky is clear     West LA          2
221771  2017-11-29    23  sky is clear    Foothill          1
221772  2017-11-29    23  sky is clear  Hollenbeck          1
221773  2017-11-29    23  sky is clear     Rampart          1
221774  2017-11-29    23  sky is clear   Southwest          1
```

4. Data Analysis/Understanding:
   Since we were still on the way to learning SQL, so for now most of our analysis used python and json files.
   Average collison/hour under different weather conditions:

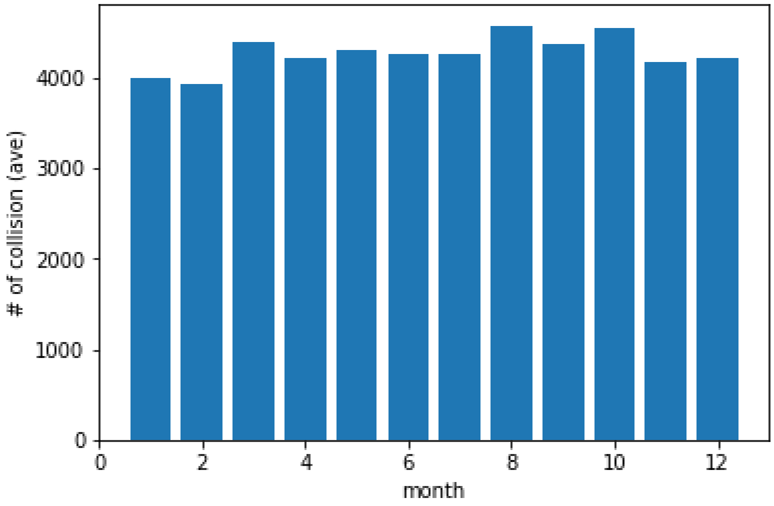


```
                        weather  count_hour  sum_collision  ave_collision
0        thunderstorm with rain           2              2       1.000000
1                           fog         566           2190       3.869258
2                       squalls           3             13       4.333333
3                          mist        2960          14581       4.926014
4  thunderstorm with light rain          13             69       5.307692
5                          haze        3532          18902       5.351642
6                    light rain        1949          10590       5.433556
7       light intensity drizzle         104            569       5.471154
8          heavy intensity rain         127            696       5.480315
9               very heavy rain          20            110       5.500000
10                  shower rain           8             44       5.500000
```

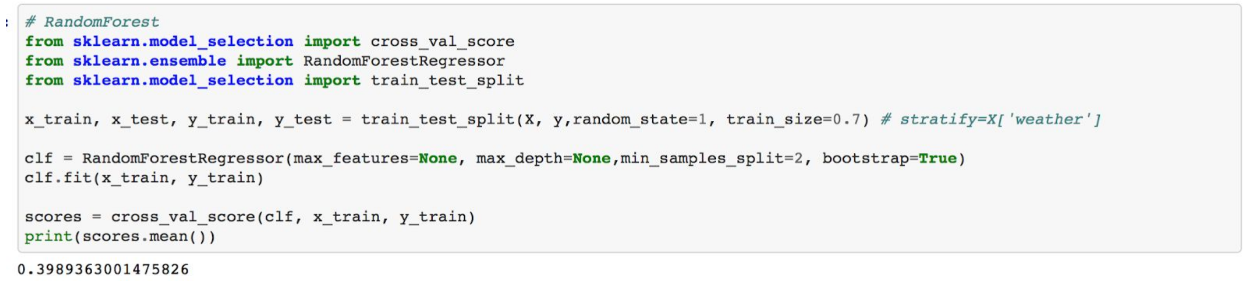
   Average collision happened in same time of different date:



Average collision happened in different month:



5. Data Aggregation:
   We would like to summarize the weather data and traffic data so that users can be forewarned to drive carefully at certain locations and weather conditions



```
# RandomForest
from sklearn.model_selection import cross_val_score
from sklearn.ensemble import RandomForestRegressor
from sklearn.model_selection import train_test_split

x_train, x_test, y_train, y_test = train_test_split(X, y,random_state=1, train_size=0.7) # stratify=X['weather']

clf = RandomForestRegressor(max_features=None, max_depth=None,min_samples_split=2, bootstrap=True)
clf.fit(x_train, y_train)

scores = cross_val_score(clf, x_train, y_train)
print(scores.mean())
```

```
0.3989363001475826
```

• Problems encountered

We tried to predict the user to be careful at certain locations under some weather conditions, however, the model performance fell short of expectations. We decided to apply what we learned in class about SQL to see if we could encounter that problem.

We should finish web development and debugging by Week 10. However, we did not have a lot of experience with javascript, so we were still struggling to build our website.