• Summary and problem statement

There are about 173m taxi ride in NYC in 2013. The study is trying to analysis the tip category based on the Taxi data requested by Chris Whong from New York City Taxi & Limousine.

The ability to predict taxi tips could present valuable insights to taxi drivers and taxi companies, especially nowadays there are a lot competitions between taxi and Uber , Lyft.

• Description of data and collection method

The data was requested through A Freedom of Information Law request by Chris Whong and hosted on <http://www.andresmh.com/nyctaxitrips/> by andresmh.

The data are included two different parts, one is trip data which includes driver details((e.g. medallion, hack license and vendor ID), passenger count, pickup location, date and time, drop off location, date and time, trip time in seconds and trip distance. The data examples are below:

Example:  
89D227B655E5C82AECF13C3F540D4CF4,BA96DE419E711691B9445D6A6307C170,CMT,1,N,2013-01-01 15:11:48,2013-01-01 15:18:10,4,382,1.00,-73.978165,40.757977,-73.989838,40.751171

Here is the explanation of each column of the trip data:

* **medallion:** a permit to operate a yellow taxi cab in New York City, it is effectively a (randomly assigned) car ID.
* **hack license:** a license to drive the vehicle, it is effectively a (randomly assigned) driver ID.
* **vender id:**  e.g., Verifone Transportation Systems (VTS), or Mobile Knowledge Systems Inc (CMT), implemented as part of the Technology Passenger Enhancements Project.
* **rate\_code:**taximeter rate.
* **store\_and\_fwd\_flag:** unknown attribute.
* **pickup datetime:** start time of the trip, mm-dd-yyyy hh24:mm:ss EDT.
* **dropoff datetime:** end time of the trip, mm-dd-yyyy hh24:mm:ss EDT.
* **passenger count:** number of passengers on the trip, default value is one.
* t**rip time in secs:** trip time measured by the taximeter in seconds.
* **trip distance:** trip distance measured by the taximeter in miles.
* **pickup\_longitude and pickup\_latitude:** GPS coordinates at the start of the trip.
* **dropoff longitude and dropoff latitude:** GPS coordinates at the end of the trip.

The second part is the fare data. Fare data has information on the trip fare, relevant tolls and taxes, and tip amount.

Example:

89D227B655E5C82AECF13C3F540D4CF4,BA96DE419E711691B9445D6A6307C170,CMT,2013-01-01 15:11:48,CSH,6.5,0,0.5,0,0,7

• Exploration, pre-processing, and feature analysis, including visualizations

1. Join the two files to one data frame.
2. select the study area:

A small number of taxi pickups in this dataset originate from well outside the NYC are, in order to constrain our problem to NYC as well as to reduce the size to Manhattan area.

I selected the study area between (40.70, -74.024) and (40.82, -73)

taxi=taxi[(taxi['dropoff\_latitude']>40.70)&(taxi['dropoff\_latitude']<40.82)&(taxi['dropoff\_longitude']>-74.024)&(taxi['dropoff\_longitude']<-73)] taxi=taxi[(taxi['pickup\_latitude']>40.70)&(taxi['pickup\_latitude']<40.82)&(taxi['pickup\_longitude']>-74.024)&(taxi['pickup\_longitude']<-73)]

1. data cleaning
2. number of customers

The unique number of customers is from 1 to 9, for counting the regular taxi only, the record whose number of customers is more than 9 are removed.

1. Speed

Calculate the speed by using distance/trip\_in\_secs, some speed are extremely high, since we analyze the Manhattan area, I removed all the records whose speed are higher than 70 miles per hour.

1. Unit price per sec, unit price per mile

Use fare amount /trip\_in\_secs and fare amount /distance, remove all the extremely high and low records.

1. Metered Fare information of NYC

The initial charge is $2.50.

Plus 50 cents per 1/5 mile or 50 cents per 60 seconds in slow traffic or when the vehicle is stopped.

Remove all the records whose fare is less than 2.5 and whose fare is less than 3.5, but distance more than 0.2 mile or trips short than 1 mins.

1. Remove the extremely long trips, eg more than 100 miles or 2hours.select trips longer than 3 minutes
2. Date selection

Since there are 2 million records in the dataset, it is impossible to process it based on my computer processing ability, I chose only six 6 Saturdays, from January to February 2013.

taxi\_credit\_data = taxi\_credit[(taxi\_credit['date']=='2013-02-09') |(taxi\_credit['date']=='2013-02-02')|(taxi\_credit['date']=='2013-01-26')|(taxi\_credit['date']=='2013-01-19')|(taxi\_credit['date']=='2013-01-12')|(taxi\_credit['date']=='2013-01-05')]

1. Location Clustering

For better classifying the result, I clustered all the pickup and drop off location to 1 km \* 1km grids.

1. Tip classification

Classify the tips into two different categaries, less than 15% is low rate, more than 15% is high rate.

• Modeling process, testing, and validation

The features of the data are not only numerical numbers, but also time, location, number of people, distance and price. So I think the Random forest will be a good fit.

from sklearn.ensemble import RandomForestClassifier

clf = RandomForestClassifier(n\_estimators=100, n\_jobs=-1)

I think the tips percentage is really depends on the service of the taxi driver, to use the medallion information , it is necessary to preprocessing the information, by using LabelEncoder, it can transform this field.

from sklearn.ensemble import RandomForestClassifier

le = LabelEncoder()

le.fit(taxi['medallion'])

LabelEncoder()

Taxi[‘id’] = le.transform(taxi[''medallion '])

1. User features 'id','passenger\_count',' fare\_amount','price\_per\_mile'

clf.score(X\_train,y\_train): 0.99048561190617512

clf.score(X\_test,y\_test): 0.70888034962144864

confusion\_matrix(y\_test,clf.predict(X\_test))

array([[ 4817, 20256],

[12851, 75799]])

1. Add Hour of day , try to add the pickup hours to see if the score going up.

clf.score(X\_train,y\_train): 0.99920420321396386

clf.score(X\_test,y\_test): 0.74243556712362491

confusion\_matrix(y\_test,clf.predict(X\_test))

array([[ 3138, 21935],

[ 7356, 81294]])

1. Adding pick up Location

Training score is : 0.99992745498911828

Testing score: 0.76160495238430226

array([[ 1823, 23250],

[ 3861, 84789]])

1. Adding Drop off location :

0.76915839364068839 is the new testing score

5. Adding Distance:

clf.score(X\_train,y\_train)

0.99999120666534769

clf.score(X\_test,y\_test)

0.77295709750885921

confusion\_matrix(y\_test,clf.predict(X\_test))

array([[ 1254, 23819],

[ 2001, 86649]])