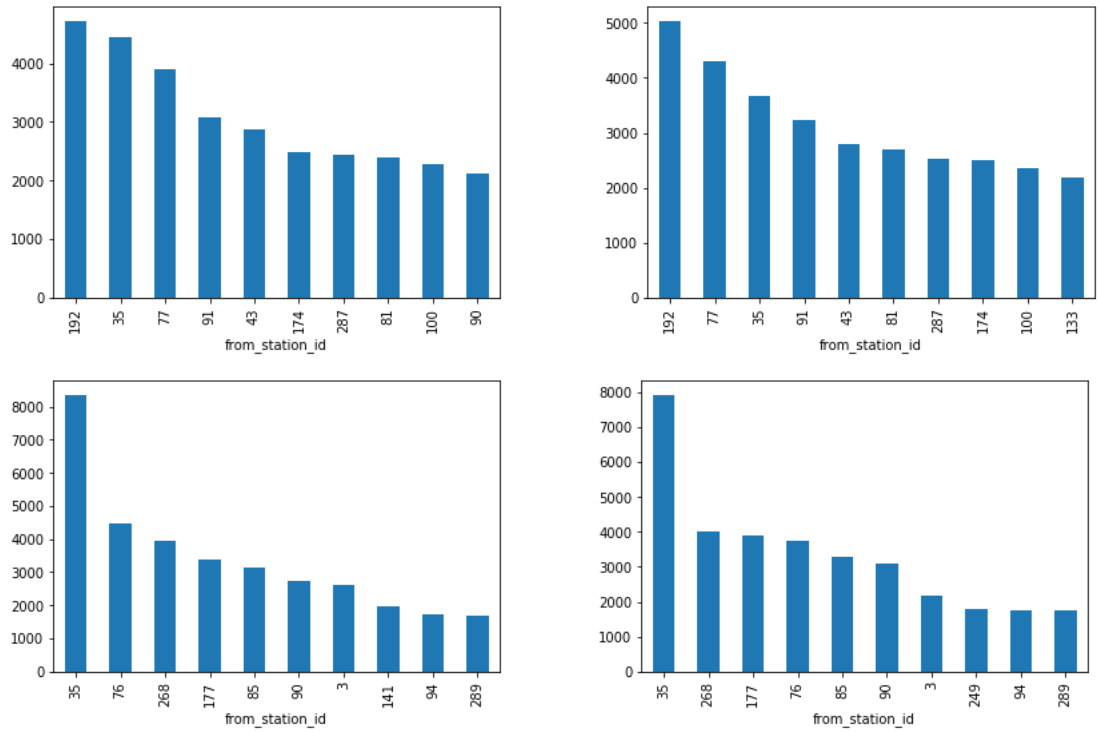
We first try to improve the data.

Each sample have 8 attributes and 1 labels.

id	start_time	tripduration	from_station_id	to_station_id	usertype	gender	age	month	weekday
0	0	1436	140	106	-1	0	0	7	7
1	0	6	153	250	1	1	6	7	7
2	0	23	76	301	1	-1	6	7	7

We try to make changes to 4 of these attributes.(weekday, start_time, tripduration, age)

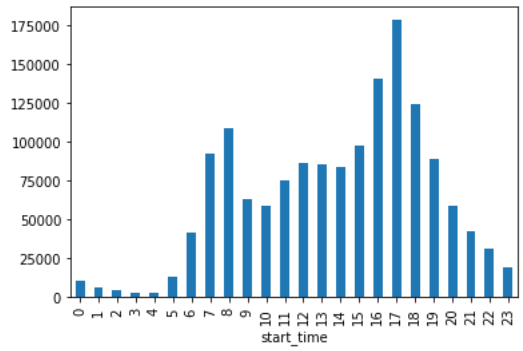
As to the weekday,



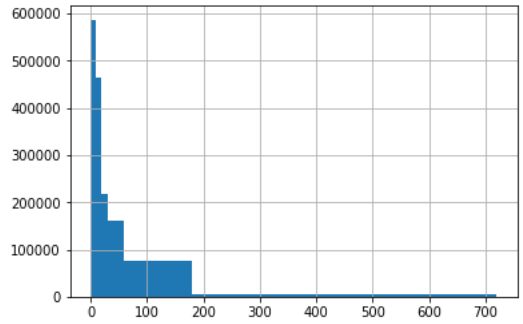
The above two pictures display top 10 to_station_id on Monday and Tuesday.
The below two picutres display top 10 to_station_id on Saturday and Sunday.
We find that the top10 to_station_id are almost same in weekday and they are quite different from weekend.
So we try to change the attribute weekday to weekend, 1 means weekend and 0 means weekday.

Then is the start_time. we divide each day into 24 hours and count the trips taken at each hour, where the trips time denotes their starting time. As we can see, most of the trip are taken during the daytime from 7AM to 8PM and 15PM to 19PM.

So we try to add two attributes called morning_rush_hour and night_rush_hour. Means whether the trip taken during the morning rush hour and night rush hour

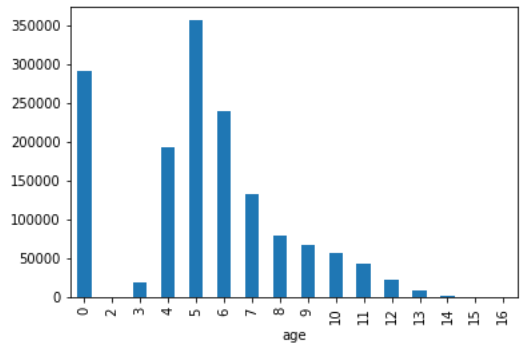


Then is the tripduration.
Let look at the tripduration distribution figure.



We first remove trips with duration larger than 600 minutes.
But in connection with the actual phenomenon, most of the people who ride the longest are tourists, and the tourists may borrow the bike for whole week and most people won't travel longer than 1 week, so we think that the duration more than one week is unreasonable.
So we delete the samples with a duration greater than 10,000 minutes

At last is the age. The distribution of age shows below: we first set unknow age as 0



We can see that, people from 5 to 6 are more likely to ride bike. And the age of customers are unknow, so we set the age of customers to 5 and 6 proportionally.

And we think that some attributes may have more weight than others. Through the experiment, we finally decide to copy some of the attributes again.
The final data looks like that: we have 14 attributes totally.

	start_time	tripduration	from_station_id	usertype	gender	age	month	weekend	morning_rush_hour	night_rush_hour	from_station_id	from_station_id	start_time	usertype
0	0	1436	140	-1	0	0	7	1	0	0	140	140	0	-1
1	0	6	153	1	1	6	7	1	0	0	153	153	0	1
2	0	23	76	1	-1	6	7	1	0	0	76	76	0	1
3	0	23	76	1	1	6	7	1	0	0	76	76	0	1
4	0	10	60	1	1	11	7	1	0	0	60	60	0	1

And this is the result after data improvement.

```
this is the 1 th round
 training accuracy is:  28.162859980139025
 test1 accuracy is:  25.900000000000002
 test2 accuracy is:  16.900000000000002
this is the 2 th round
 training accuracy is:  27.46110559417411
 test1 accuracy is:  26.1
 test2 accuracy is:  17.7
this is the 3 th round
 training accuracy is:  27.851704733531946
 test1 accuracy is:  26.3
 test2 accuracy is:  17.4
this is the 4 th round
 training accuracy is:  27.851704733531946
 test1 accuracy is:  25.8
 test2 accuracy is:  16.900000000000002
```

There are total 8 parameters in KNeighborsClassifier. And we try to change below parameters:
n_neighbors,: Number of neighbors to use by default for kneighbors queries.
weights: weight function used in prediction.
Algorithm: Algorithm used to compute the nearest neighbors:
leaf_size: Leaf size passed to BallTree or KDTree.
p : Power parameter for the Minkowski metric.

And the other 3 parameters we decide to use the default value.

Through the experiment, we found that when weight= ‘distance’ , algorithm=‘kd_tree’ , p=1(manhattan_distance), the model performs much better.

Then we use GridSearchCV from sklearn to search the best n_neighbors and leaf_size.
It’s a exhaustive search over specified parameter values for an estimator.
But it’s pretty slow

And the final result is that:

```
##knn model
estimator=KNeighborsClassifier(n_neighbors=9, weights='distance', algorithm='kd_tree', leaf_size=80, p=1)
estimator.fit(X_train,Y_train)
```

This is the best result we get .

```
this is the 1 th round
 training accuracy is:  34.2469380999669
 test1 accuracy is:  31.2
 test2 accuracy is:  21.4
this is the 2 th round
 training accuracy is:  34.207216153591524
 test1 accuracy is:  31.3
 test2 accuracy is:  21.5
this is the 3 th round
 training accuracy is:  33.995365772922874
 test1 accuracy is:  30.599999999999998
 test2 accuracy is:  20.8
this is the 4 th round
 training accuracy is:  34.47864945382324
 test1 accuracy is:  30.7
 test2 accuracy is:  20.7
```

And the final performance is still not good. I think bad performance in this forecast is due to two factors, one is that the months of the data in training set and test set are different, different months will have different climates. And the number of target stations is too large, which is another factor that leads to inaccurate predictions

That's all! Thanks for listening.