Untitled

April 22, 2019

```
In [1]: import numpy as np
        import pandas as pd

In [2]: train=pd.read_csv('train.csv')
        test=pd.read_csv('test.csv')

In [3]: print (train.shape)
        print (test.shape)

(4998181, 8)
(9914, 7)


In [4]: train.tail(10)

Out[4]:                                   key  fare_amount          pickup_datetime  \
        4998171    2010-08-29 05:20:49.0000002         11.7  2010-08-29 05:20:49 UTC
        4998172  2011-07-12 22:26:00.000000112         12.1  2011-07-12 22:26:00 UTC
        4998173   2013-02-26 12:01:00.00000013          5.5  2013-02-26 12:01:00 UTC
        4998174    2014-06-02 15:57:37.0000001          7.5  2014-06-02 15:57:37 UTC
        4998175    2011-09-08 17:16:06.0000004          6.1  2011-09-08 17:16:06 UTC
        4998176  2012-04-13 19:42:00.000000161          4.9  2012-04-13 19:42:00 UTC
        4998177    2015-03-01 01:17:59.0000003          5.5  2015-03-01 01:17:59 UTC
        4998178   2010-09-25 19:02:00.00000058          4.9  2010-09-25 19:02:00 UTC
        4998179    2010-02-12 23:47:15.0000001          5.7  2010-02-12 23:47:15 UTC
        4998180    2011-09-01 21:07:12.0000004          5.7  2011-09-01 21:07:12 UTC

                 pickup_longitude  pickup_latitude  dropoff_longitude  \
        4998171        -73.991126        40.765717         -74.009531
        4998172        -73.981595        40.772818         -73.934812
        4998173        -73.975162        40.755982         -74.011527
        4998174        -74.013305        40.715278         -74.002228
        4998175        -73.990385        40.737763         -73.996019
        4998176        -73.956597        40.771295         -73.976495
        4998177        -73.985558        40.758202         -73.994881
        4998178        -73.972405        40.786398         -73.976293
        4998179        -73.992380        40.718933         -73.982690
        4998180        -73.971000        40.788289          -7.000000
```

```
         dropoff_latitude  passenger_count
4998171         40.715764              2.0
4998172         40.798258              3.0
4998173         40.703047              1.0
4998174         40.724782              1.0
4998175         40.725152              2.0
4998176         40.745060              1.0
4998177         40.761951              1.0
4998178         40.775907              3.0
4998179         40.723067              4.0
4998180               NaN              NaN
```

```
In [5]: new_df=pd.concat([train,test],axis=0)
```

```
/opt/anaconda3/lib/python3.7/site-packages/ipykernel_launcher.py:1: FutureWarning: Sorting becau
of pandas will change to not sort by default.

To accept the future behavior, pass 'sort=False'.

To retain the current behavior and silence the warning, pass 'sort=True'.

  """Entry point for launching an IPython kernel.
```

```
In [6]: new_df.head()
```

```
Out[6]:    dropoff_latitude  dropoff_longitude  fare_amount  \
        0         40.712278         -73.841610          4.5
        1         40.782004         -73.979268         16.9
        2         40.750562         -73.991242          5.7
        3         40.758092         -73.991567          7.7
        4         40.783762         -73.956655          5.3

                                     key  passenger_count          pickup_datetime  \
        0    2009-06-15 17:26:21.0000001              1.0  2009-06-15 17:26:21 UTC
        1    2010-01-05 16:52:16.0000002              1.0  2010-01-05 16:52:16 UTC
        2   2011-08-18 00:35:00.00000049              2.0  2011-08-18 00:35:00 UTC
        3    2012-04-21 04:30:42.0000001              1.0  2012-04-21 04:30:42 UTC
        4  2010-03-09 07:51:00.000000135              1.0  2010-03-09 07:51:00 UTC

           pickup_latitude  pickup_longitude
        0        40.721319        -73.844311
        1        40.711303        -74.016048
        2        40.761270        -73.982738
        3        40.733143        -73.987130
        4        40.768008        -73.968095
```

```
In [7]: temp=new_df.pickup_datetime.apply(lambda x : x[:-4])
In [8]: temp = pd.to_datetime(temp)
```

```
In [9]: new_df['pickup_datetime']=temp
```

```
In [10]: new_df['key']=pd.to_datetime(train['key'])
```

```
In [11]: new_df.head()
```

```
Out[11]:    dropoff_latitude  dropoff_longitude  fare_amount  \
         0         40.712278         -73.841610          4.5
         1         40.782004         -73.979268         16.9
         2         40.750562         -73.991242          5.7
         3         40.758092         -73.991567          7.7
         4         40.783762         -73.956655          5.3

                                     key  passenger_count     pickup_datetime  \
         0 2009-06-15 17:26:21.000000100              1.0 2009-06-15 17:26:21
         1 2010-01-05 16:52:16.000000200              1.0 2010-01-05 16:52:16
         2 2011-08-18 00:35:00.000000490              2.0 2011-08-18 00:35:00
         3 2012-04-21 04:30:42.000000100              1.0 2012-04-21 04:30:42
         4 2010-03-09 07:51:00.000000135              1.0 2010-03-09 07:51:00

            pickup_latitude  pickup_longitude
         0        40.721319        -73.844311
         1        40.711303        -74.016048
         2        40.761270        -73.982738
         3        40.733143        -73.987130
         4        40.768008        -73.968095
```

```
In [12]: new_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 5008095 entries, 0 to 9913
Data columns (total 8 columns):
dropoff_latitude     float64
dropoff_longitude    float64
fare_amount          float64
key                  datetime64[ns]
passenger_count      float64
pickup_datetime      datetime64[ns]
pickup_latitude      float64
pickup_longitude     float64
dtypes: datetime64[ns](2), float64(6)
memory usage: 343.9 MB
```

```
In [13]: pd.DataFrame(new_df.isnull().sum())
```

```
Out[13]:                       0
         dropoff_latitude     37
```

```
dropoff_longitude    36
fare_amount        9914
key                   0
passenger_count       1
pickup_datetime       0
pickup_latitude       0
pickup_longitude      0
```

```
In [14]: from sklearn.preprocessing import Imputer
         imputer_mean = Imputer(missing_values='NaN',strategy='mean',axis=0)
         new_df['dropoff_latitude']=imputer_mean.fit_transform(new_df[['dropoff_latitude']])
         new_df['dropoff_latitude']=imputer_mean.fit_transform(new_df[['dropoff_latitude']])
         new_df['passenger_count']=imputer_mean.fit_transform(new_df[['passenger_count']])
         new_df['dropoff_longitude']=imputer_mean.fit_transform(new_df[['dropoff_longitude']])
```

```
/opt/anaconda3/lib/python3.7/site-packages/sklearn/utils/deprecation.py:58: DeprecationWarning:
  warnings.warn(msg, category=DeprecationWarning)
```

```
In [15]: new_df.describe()
```

```
Out[15]:        dropoff_latitude  dropoff_longitude   fare_amount  passenger_count  \
         count      5.008095e+06       5.008095e+06  4.998181e+06     5.008095e+06
         mean       3.991897e+01      -7.250950e+01  1.134080e+01     1.684654e+00
         std        9.478195e+00       1.283571e+01  9.820255e+00     1.331747e+00
         min       -3.488080e+03      -3.412653e+03 -1.000000e+02     0.000000e+00
         25%        4.073404e+01      -7.399139e+01  6.000000e+00     1.000000e+00
         50%        4.075315e+01      -7.398016e+01  8.500000e+00     1.000000e+00
         75%        4.076811e+01      -7.396367e+01  1.250000e+01     2.000000e+00
         max        3.345917e+03       3.457622e+03  1.273310e+03     2.080000e+02

                pickup_latitude  pickup_longitude
         count     5.008095e+06      5.008095e+06
         mean      3.992148e+01     -7.250980e+01
         std       8.955292e+00      1.279760e+01
         min      -3.488080e+03     -3.426609e+03
         25%       4.073491e+01     -7.399206e+01
         50%       4.075264e+01     -7.398181e+01
         75%       4.076712e+01     -7.396711e+01
         max       3.310364e+03      3.439426e+03
```

```
In [16]: new_df[new_df['fare_amount']<0]
```

```
Out[16]:          dropoff_latitude  dropoff_longitude  fare_amount  \
         2039            40.641952         -73.788665        -2.90
         2486            40.720539         -73.999809        -2.50
         13032           40.741357         -73.995885        -3.00
         28839            0.000000           0.000000        -2.50
         36722           40.792839         -73.950043        -2.50
```

```
42337           40.759869         -73.980820          -5.00
56748           40.737240         -73.981216          -5.00
58937           40.786890         -73.676533         -44.90
97838           40.764065         -73.914963          -3.00
102938          40.779775         -73.973443          -2.90
165147          40.773811         -73.982094          -2.50
179311          40.795979         -73.945160          -3.00
182341          40.757858         -73.982193          -5.00
288960          40.738804         -74.009827          -4.50
298412          40.749307         -73.975762          -6.50
301356          40.730198         -74.006699          -3.00
323637          40.754877         -73.966578          -6.50
399785          40.773613         -73.964104          -2.50
427602          40.711573         -74.010582         -18.10
443469          40.744591         -73.956757          -2.50
481419          40.770851         -73.988449          -5.00
512494           0.000000           0.000000         -20.00
519532          40.757504         -73.971680          -2.50
534751          40.749783         -73.975303         -29.87
549210          40.717247         -73.994370          -2.50
577725          40.742802         -73.980347          -6.50
605427          40.764503         -73.971466          -2.50
698287          40.735535         -73.985428          -2.50
738404          40.772838         -73.885178          -2.50
740842           0.000000           0.000000          -2.50
...                   ...                ...            ...
4220311         40.725700         -74.000969          -2.50
4241056         40.781841         -73.945969          -2.50
4249245         40.761552         -73.963462          -2.90
4278807         40.691509         -73.809174         -40.00
4293610         40.743233         -73.991035          -3.00
4296875         40.757095         -73.998474          -3.50
4356535         40.751751         -73.976517          -2.50
4364879         40.767372         -73.983208          -2.50
4369986         40.743550         -73.990418          -4.90
4378164         40.771708         -73.866992          -3.70
4415931         40.768322         -73.953590          -5.50
4464104         40.769588         -73.956650          -6.50
4469688         40.742737         -73.990601          -3.50
4470814         40.750057         -73.971947         -57.33
4496200         40.720220         -73.979427          -4.10
4498716         40.763062         -73.981461          -2.50
4614892         40.706291         -74.006264         -12.50
4625530          0.000000           0.000000         -45.00
4692009         40.758251         -73.937378          -2.50
4734044         40.760057         -73.971245          -2.50
4762323         40.644975         -73.789952         -17.90
4765875         40.755527         -73.962127          -2.50
```

```
4781925         40.755829         -73.998070         -4.50
4868952         40.744965         -73.978691         -2.50
4903392         40.753247         -73.992663         -2.50
4919498         40.762989         -73.989410         -3.00
4966889         40.745268         -73.980460         -2.50
4973568         40.790779         -73.942749         -3.00
4983087         40.642865         -73.786122         -2.50
4987995         40.751469         -73.974251         -2.50

                                  key  passenger_count     pickup_datetime  \
2039    2010-03-09 23:37:10.000000500              1.0 2010-03-09 23:37:10
2486    2015-03-22 05:14:27.000000100              1.0 2015-03-22 05:14:27
13032   2013-08-30 08:57:10.000000200              4.0 2013-08-30 08:57:10
28839   2013-08-11 13:39:10.000000100              1.0 2013-08-11 13:39:10
36722   2015-04-30 15:19:45.000000300              1.0 2015-04-30 15:19:45
42337   2015-03-09 10:29:46.000000400              1.0 2015-03-09 10:29:46
56748   2015-06-26 01:13:18.000000200              6.0 2015-06-26 01:13:18
58937   2010-02-19 23:47:10.000000200              1.0 2010-02-19 23:47:10
97838   2015-06-07 02:54:14.000000400              5.0 2015-06-07 02:54:14
102938  2010-02-10 12:33:10.000000400              1.0 2010-02-10 12:33:10
165147  2015-05-31 10:23:50.000000300              2.0 2015-05-31 10:23:50
179311  2015-04-21 22:45:11.000000200              1.0 2015-04-21 22:45:11
182341  2015-02-10 01:34:08.000000200              3.0 2015-02-10 01:34:08
288960  2015-02-13 00:36:36.000000200              1.0 2015-02-13 00:36:36
298412  2010-03-12 12:17:10.000000100              1.0 2010-03-12 12:17:10
301356  2015-03-28 01:42:18.000000800              1.0 2015-03-28 01:42:18
323637  2010-03-13 00:10:10.000000500              1.0 2010-03-13 00:10:10
399785  2015-03-28 09:50:44.000000400              1.0 2015-03-28 09:50:44
427602  2010-03-14 14:07:10.000000400              3.0 2010-03-14 14:07:10
443469  2015-01-17 19:13:41.000000600              1.0 2015-01-17 19:13:41
481419  2015-02-18 07:22:53.000000500              1.0 2015-02-18 07:22:53
512494  2015-04-19 15:57:05.000000700              5.0 2015-04-19 15:57:05
519532  2015-03-10 23:26:19.000000400              1.0 2015-03-10 23:26:19
534751  2010-02-08 09:04:10.000000300              1.0 2010-02-08 09:04:10
549210  2015-06-19 03:35:46.000000300              1.0 2015-06-19 03:35:46
577725  2015-01-01 04:35:09.000000400              1.0 2015-01-01 04:35:09
605427  2015-02-26 15:12:43.000000600              1.0 2015-02-26 15:12:43
698287  2015-01-11 14:34:09.000000400              1.0 2015-01-11 14:34:09
738404  2010-03-09 18:18:10.000000600              1.0 2010-03-09 18:18:10
740842  2015-01-14 15:57:46.000000400              1.0 2015-01-14 15:57:46
...                               ...              ...                 ...
4220311 2015-04-09 17:33:00.000000200              1.0 2015-04-09 17:33:00
4241056 2015-02-15 12:37:47.000000100              1.0 2015-02-15 12:37:47
4249245 2010-02-08 21:47:10.000000300              1.0 2010-02-08 21:47:10
4278807 2015-06-16 18:29:16.000000300              1.0 2015-06-16 18:29:16
4293610 2015-05-29 10:26:28.000000700              2.0 2015-05-29 10:26:28
4296875 2015-03-11 16:14:09.000000400              5.0 2015-03-11 16:14:09
4356535 2015-01-06 06:46:10.000000500              1.0 2015-01-06 06:46:10
```

```
4364879 2015-04-08 21:55:10.000000600              1.0 2015-04-08 21:55:10
4369986 2010-03-18 11:01:10.000000500              1.0 2010-03-18 11:01:10
4378164 2010-02-17 14:06:10.000000500              1.0 2010-02-17 14:06:10
4415931 2015-01-30 14:15:26.000000700              2.0 2015-01-30 14:15:26
4464104 2015-03-11 12:43:36.000000600              1.0 2015-03-11 12:43:36
4469688 2015-04-08 23:32:30.000000900              1.0 2015-04-08 23:32:30
4470814 2015-02-02 11:22:11.000000600              4.0 2015-02-02 11:22:11
4496200 2010-03-30 09:59:10.000000100              1.0 2010-03-30 09:59:10
4498716 2015-02-23 14:45:18.000000600              4.0 2015-02-23 14:45:18
4614892 2015-03-10 23:47:37.000000500              0.0 2015-03-10 23:47:37
4625530 2010-03-28 18:14:10.000000200              1.0 2010-03-28 18:14:10
4692009 2015-03-27 16:58:47.000000400              1.0 2015-03-27 16:58:47
4734044 2013-08-16 18:35:10.000000100              2.0 2013-08-16 18:35:10
4762323 2010-03-19 22:42:10.000000300              1.0 2010-03-19 22:42:10
4765875 2010-03-16 15:42:10.000000400              1.0 2010-03-16 15:42:10
4781925 2015-02-06 23:41:47.000000600              1.0 2015-02-06 23:41:47
4868952 2015-04-20 15:24:20.000000500              1.0 2015-04-20 15:24:20
4903392 2013-08-22 22:13:10.000000600              2.0 2013-08-22 22:13:10
4919498 2015-01-23 00:03:54.000000400              2.0 2015-01-23 00:03:54
4966889 2010-02-22 13:34:10.000000600              1.0 2010-02-22 13:34:10
4973568 2015-04-05 16:33:00.000000300              2.0 2015-04-05 16:33:00
4983087 2010-03-22 00:01:10.000000300              1.0 2010-03-22 00:01:10
4987995 2015-05-07 08:57:53.000000400              1.0 2015-05-07 08:57:53

        pickup_latitude  pickup_longitude
2039          40.643498        -73.789450
2486          40.720631        -74.000031
13032         40.740755        -73.995062
28839         40.648442        -73.785260
36722         40.790112        -73.952187
42337         40.755985        -73.990974
56748         40.743240        -73.979797
58937         40.773902        -73.871120
97838         40.766212        -73.913246
102938        40.783425        -73.970775
165147        40.773621        -73.982162
179311        40.791683        -73.944504
182341        40.750374        -73.990974
288960        40.740425        -74.006142
298412        40.730085        -73.989493
301356        40.730244        -74.004997
323637        40.728967        -73.984355
399785        40.773609        -73.964104
427602        40.760293        -73.958278
443469        40.745770        -73.956787
481419        40.765751        -73.980362
512494         0.000000          0.000000
519532        40.757504        -73.971680
```

```
534751         40.769440       -73.863158
549210         40.717243       -73.994331
577725         40.746986       -73.993660
605427         40.764351       -73.970779
698287         40.735886       -73.985229
738404         40.772840       -73.885183
740842         40.648655       -73.783615
...                  ...              ...
4220311        40.725700       -74.000969
4241056        40.782600       -73.947678
4249245        40.765085       -73.961020
4278807        40.691509       -73.809174
4293610        40.743114       -73.991112
4296875        40.756901       -73.998466
4356535        40.751759       -73.976509
4364879        40.768055       -73.984718
4369986        40.745813       -73.980107
4378164        40.768837       -73.862783
4415931        40.772469       -73.963196
4464104        40.778740       -73.951218
4469688        40.743366       -73.992058
4470814        40.750149       -73.971931
4496200        40.707898       -73.999640
4498716        40.763062       -73.981461
4614892        40.706596       -74.006264
4625530         0.000000         0.000000
4692009        40.758190       -73.937431
4734044        40.759987       -73.971240
4762323        40.645165       -73.786772
4765875        40.755495       -73.962140
4781925        40.760960       -73.990891
4868952        40.744953       -73.978691
4903392        40.752582       -73.993177
4919498        40.765190       -73.987808
4966889        40.745328       -73.980443
4973568        40.792641       -73.940979
4983087        40.642332       -73.786072
4987995        40.751400       -73.974472

[211 rows x 8 columns]
```

```
In [17]: new_df[new_df['fare_amount']<0].index
```

```
Out[17]: Int64Index([   2039,    2486,   13032,   28839,   36722,   42337,   56748,
                         58937,   97838,  102938,
                       ...
                       4762323, 4765875, 4781925, 4868952, 4903392, 4919498, 4966889,
                       4973568, 4983087, 4987995],
                      dtype='int64', length=211)
```

```
In [18]: new_df = new_df.drop(new_df[new_df['fare_amount']<0].index,axis=0)
```

```
In [19]: new_df[new_df['passenger_count']>6]
```

```
Out[19]:          dropoff_latitude  dropoff_longitude  fare_amount  \
         929022           0.000000           0.000000         3.30
         1007609         40.708340         -74.170280       104.00
         2154045          0.000000           0.000000         3.30
         2198549          0.000000           0.000000         3.30
         2910347          0.000000           0.000000         4.50
         3107489         40.758250         -73.937827         2.70
         3323791         40.752413         -74.000682         8.50
         4095440         40.774506         -73.872482        37.04
         4103745          0.000000           0.000000        23.70
         4432483         40.758273         -73.937737        11.10
         4467314         40.735100         -73.988045         8.50
         4679603         28.121107       -2069.478952         9.30

                                           key  passenger_count     pickup_datetime  \
         929022  2009-07-30 11:54:00.000000193            208.0 2009-07-30 11:54:00
         1007609 2014-06-24 15:13:00.000000400              9.0 2014-06-24 15:13:00
         2154045 2010-12-16 11:21:00.000000209            208.0 2010-12-16 11:21:00
         2198549 2010-12-15 14:20:00.000000100            208.0 2010-12-15 14:20:00
         2910347 2010-12-16 06:44:00.000000390            208.0 2010-12-16 06:44:00
         3107489 2009-05-12 14:50:00.000000175            208.0 2009-05-12 14:50:00
         3323791 2011-08-27 01:24:00.000000168            129.0 2011-08-27 01:24:00
         4095440 2015-06-14 08:56:16.000000100              9.0 2015-06-14 08:56:16
         4103745 2010-12-22 12:11:00.000000230            208.0 2010-12-22 12:11:00
         4432483 2009-05-11 13:56:00.000000880            208.0 2009-05-11 13:56:00
         4467314 2015-01-01 21:32:16.000000700              7.0 2015-01-01 21:32:16
         4679603 2010-02-20 01:53:00.000000370             51.0 2010-02-20 01:53:00

                  pickup_latitude  pickup_longitude
         929022          0.000000          0.000000
         1007609        40.715420        -74.015780
         2154045         0.000000          0.000000
         2198549         0.000000          0.000000
         2910347         0.000000          0.000000
         3107489        40.758260        -73.937818
         3323791        40.760340        -73.987858
         4095440        40.756252        -73.982094
         4103745         0.000000          0.000000
         4432483        40.758267        -73.937733
         4467314        40.740643        -74.005867
         4679603         0.000000          0.554830
```

```
In [20]: new_df[new_df['fare_amount']<0]
```

```
Out[20]: Empty DataFrame
```

```
        Columns: [dropoff_latitude, dropoff_longitude, fare_amount, key, passenger_count, picku
        Index: []

In [21]: new_df[new_df['passenger_count']>6].index

Out[21]: Int64Index([ 929022, 1007609, 2154045, 2198549, 2910347, 3107489, 3323791,
                     4095440, 4103745, 4432483, 4467314, 4679603],
                    dtype='int64')

In [22]: new_df=new_df.drop(new_df[new_df['passenger_count']>6].index,axis=0)

In [23]: new_df[new_df['passenger_count']>6]

Out[23]: Empty DataFrame
         Columns: [dropoff_latitude, dropoff_longitude, fare_amount, key, passenger_count, picku
         Index: []

In [24]: new_df.describe()

Out[24]:        dropoff_latitude  dropoff_longitude   fare_amount  passenger_count  \
         count      5.007870e+06       5.007870e+06  4.997958e+06     5.007870e+06
         mean       3.991914e+01      -7.250942e+01  1.134162e+01     1.684322e+00
         std        9.477968e+00       1.280387e+01  9.819102e+00     1.307784e+00
         min       -3.488080e+03      -3.412653e+03  0.000000e+00     0.000000e+00
         25%        4.073405e+01      -7.399139e+01  6.000000e+00     1.000000e+00
         50%        4.075315e+01      -7.398016e+01  8.500000e+00     1.000000e+00
         75%        4.076811e+01      -7.396367e+01  1.250000e+01     2.000000e+00
         max        3.345917e+03       3.457622e+03  1.273310e+03     6.000000e+00

                pickup_latitude  pickup_longitude
         count     5.007870e+06      5.007870e+06
         mean      3.992163e+01     -7.251008e+01
         std       8.955083e+00      1.279694e+01
         min      -3.488080e+03     -3.426609e+03
         25%       4.073491e+01     -7.399206e+01
         50%       4.075264e+01     -7.398181e+01
         75%       4.076712e+01     -7.396711e+01
         max       3.310364e+03      3.439426e+03

In [25]: new_df.info()

<class 'pandas.core.frame.DataFrame'>
Int64Index: 5007870 entries, 0 to 9913
Data columns (total 8 columns):
dropoff_latitude     float64
dropoff_longitude    float64
fare_amount          float64
key                  datetime64[ns]
passenger_count      float64
```

```
pickup_datetime      datetime64[ns]
pickup_latitude      float64
pickup_longitude     float64
dtypes: datetime64[ns](2), float64(6)
memory usage: 343.9 MB
```

```
In [26]: import calendar
         new_df['day']=new_df['pickup_datetime'].apply(lambda x:x.day)
         new_df['hour']=new_df['pickup_datetime'].apply(lambda x:x.hour)
         new_df['weekday']=new_df['pickup_datetime'].apply(lambda x:calendar.day_name[x.weekday(
         new_df['month']=new_df['pickup_datetime'].apply(lambda x:x.month)
         new_df['year']=new_df['pickup_datetime'].apply(lambda x:x.year)

In [27]: #here we can see that week are in monday , tuesday and so on. So we need convert them i
         new_df.weekday = new_df.weekday.map({'Sunday':0,'Monday':1,'Tuesday':2,'Wednesday':3,'T

In [28]: new_df.drop(["key","pickup_datetime"], axis=1, inplace=True)

In [29]: new_df.head()

Out[29]:    dropoff_latitude  dropoff_longitude  fare_amount  passenger_count  \
         0         40.712278         -73.841610          4.5              1.0
         1         40.782004         -73.979268         16.9              1.0
         2         40.750562         -73.991242          5.7              2.0
         3         40.758092         -73.991567          7.7              1.0
         4         40.783762         -73.956655          5.3              1.0

            pickup_latitude  pickup_longitude  day  hour  weekday  month  year
         0        40.721319        -73.844311   15    17        1      6  2009
         1        40.711303        -74.016048    5    16        2      1  2010
         2        40.761270        -73.982738   18     0        4      8  2011
         3        40.733143        -73.987130   21     4        6      4  2012
         4        40.768008        -73.968095    9     7        2      3  2010

In [30]: ## Train Test Split
         train = new_df.iloc[:4998181,:]
         test = new_df.iloc[4998182:,:]

In [31]: train.isna().sum()

Out[31]: dropoff_latitude       0
         dropoff_longitude      0
         fare_amount          223
         passenger_count        0
         pickup_latitude        0
         pickup_longitude       0
         day                    0
         hour                   0
```

```
weekday              0
month                0
year                 0
dtype: int64
```

In [32]:
```
train[train['fare_amount'].isna()]
```

Out[32]:
```
     dropoff_latitude  dropoff_longitude  fare_amount  passenger_count  \
0           40.743835         -73.981430          NaN              1.0
1           40.739201         -73.998886          NaN              1.0
2           40.746139         -73.979654          NaN              1.0
3           40.751635         -73.990448          NaN              1.0
4           40.744427         -73.988565          NaN              1.0
5           40.740053         -73.979177          NaN              1.0
6           40.770893         -73.959622          NaN              1.0
7           40.759368         -73.985083          NaN              1.0
8           40.741365         -73.995106          NaN              1.0
9           40.770725         -73.980686          NaN              1.0
10          40.722534         -73.999300          NaN              1.0
11          40.767437         -73.956387          NaN              1.0
12          40.735269         -73.997166          NaN              1.0
13          40.761545         -74.001867          NaN              1.0
14          40.750149         -73.983397          NaN              1.0
15          40.785903         -73.951178          NaN              1.0
16          40.732318         -74.010204          NaN              1.0
17          40.772121         -73.952220          NaN              1.0
18          40.756417         -73.972096          NaN              1.0
19          40.755563         -73.975954          NaN              1.0
20          40.714789         -73.941741          NaN              1.0
21          40.771800         -73.949132          NaN              1.0
22          40.747767         -73.992767          NaN              1.0
23          40.762960         -73.991520          NaN              1.0
24          40.711682         -74.015665          NaN              1.0
25          40.762705         -74.000850          NaN              1.0
26          40.689945         -73.741922          NaN              1.0
27          40.808220         -73.938852          NaN              1.0
28          40.768810         -73.950277          NaN              1.0
29          40.710192         -74.007125          NaN              1.0
..                ...                ...          ...              ...
193         40.776920         -73.982949          NaN              1.0
194         40.758526         -73.968612          NaN              1.0
195         40.744953         -74.005576          NaN              1.0
196         40.731559         -73.983230          NaN              1.0
197         40.761097         -73.979219          NaN              1.0
198         40.728989         -73.998192          NaN              1.0
199         40.755837         -73.991014          NaN              1.0
200         40.743739         -73.979618          NaN              1.0
201         40.747326         -73.972162          NaN              1.0
```

```
202         40.743944         -73.983810          NaN              1.0
203         40.776222         -73.977747          NaN              1.0
204         40.745231         -73.990333          NaN              1.0
205         40.757360         -73.975898          NaN              1.0
206         40.696673         -73.996243          NaN              1.0
207         40.725886         -74.005707          NaN              1.0
208         40.754099         -73.991643          NaN              1.0
209         40.706195         -74.009293          NaN              1.0
210         40.735961         -73.985410          NaN              1.0
211         40.762818         -73.959567          NaN              1.0
212         40.763811         -73.967179          NaN              1.0
213         40.738394         -74.002170          NaN              1.0
214         40.702387         -74.013773          NaN              1.0
215         40.759490         -73.995598          NaN              1.0
216         40.794797         -73.936200          NaN              1.0
217         40.730305         -73.953957          NaN              1.0
218         40.638197         -73.964767          NaN              1.0
219         40.776243         -73.985015          NaN              1.0
220         40.705207         -74.007737          NaN              1.0
221         40.726593         -74.006707          NaN              1.0
222         40.734665         -74.006975          NaN              1.0

     pickup_latitude  pickup_longitude  day  hour  weekday  month  year
0          40.763805        -73.973320   27    13        2      1  2015
1          40.719383        -73.986862   27    13        2      1  2015
2          40.751260        -73.982524    8    11        6     10  2011
3          40.767807        -73.981160    1    21        6     12  2012
4          40.789775        -73.966046    1    21        6     12  2012
5          40.765547        -73.960983    1    21        6     12  2012
6          40.773204        -73.949013    6    12        4     10  2011
7          40.646636        -73.777282    6    12        4     10  2011
8          40.709638        -74.014099    6    12        4     10  2011
9          40.765519        -73.969582   18    15        2      2  2014
10         40.741973        -73.989374   18    15        2      2  2014
11         40.740893        -74.001614   18    15        2      2  2014
12         40.739937        -73.991198   29    20        1      3  2010
13         40.762723        -73.982034   29    20        1      3  2010
14         40.728701        -73.992455    6     3        4     10  2011
15         40.746993        -73.983583    6     3        4     10  2011
16         40.731721        -74.006746   15    16        0      7  2012
17         40.785598        -73.976446   15    16        0      7  2012
18         40.763349        -73.973548   15    16        0      7  2012
19         40.756025        -73.970918   15    16        0      7  2012
20         40.705866        -73.926071   29     2        3     10  2014
21         40.764702        -73.970555   14    13        6      6  2014
22         40.736360        -73.989102   14    13        6      6  2014
23         40.748480        -74.003525   14    13        6      6  2014
24         40.759992        -73.990352   14    13        6      6  2014
```

```
25          40.757450        -73.989482   14    13         6      6  2014
26          40.773722        -73.870785   14    13         6      6  2014
27          40.733877        -73.992682   14    13         6      6  2014
28          40.778705        -73.954020   14    13         6      6  2014
29          40.743432        -73.972742   14    13         6      6  2014
..                ...               ...  ...   ...       ...    ...   ...
193         40.767420        -73.989861   21    13         6      5  2011
194         40.751283        -73.980611   21    13         6      5  2011
195         40.721409        -73.997552   21    13         6      5  2011
196         40.740982        -73.993943   25    16         5     11  2011
197         40.730506        -74.000256   25    16         5     11  2011
198         40.737057        -74.001373   31    20         6      1  2015
199         40.737529        -73.996852    1    23         1     11  2010
200         40.767036        -73.959735    1    23         1     11  2010
201         40.724698        -73.998827    1    23         1     11  2010
202         40.746472        -73.982587   24    15         4      3  2011
203         40.767553        -73.959245   24    15         4      3  2011
204         40.749266        -73.976646   10    10         1      5  2010
205         40.755650        -73.985901   10    10         1      5  2010
206         40.715461        -74.011108   10    10         1      5  2010
207         40.749447        -73.991668   10    10         1      5  2010
208         40.763702        -73.973334    9    16         6      6  2012
209         40.765684        -73.983765   22    22         3      8  2012
210         40.744983        -73.978400   22    22         3      8  2012
211         40.731949        -74.003236   23     2         0     12  2012
212         40.746582        -74.009518   30    15         6      6  2012
213         40.750902        -73.990302   30    15         6      6  2012
214         40.737562        -74.000283   30    15         6      6  2012
215         40.747890        -74.003962   30    15         6      6  2012
216         40.753840        -73.966230    2    23         0     10  2011
217         40.679922        -73.974375    2    23         0     10  2011
218         40.687278        -73.973798    2    23         0     10  2011
219         40.775892        -73.979785    2    23         0     10  2011
220         40.739613        -73.979770    2    23         0     10  2011
221         40.730533        -74.004512    2    23         0     10  2011
222         40.773378        -73.982370    2    23         0     10  2011

[223 rows x 11 columns]
```

```
In [33]: train= train.drop(train[train['fare_amount'].isna()].index,axis=0)
         #new_df=new_df.drop(new_df[new_df['passenger_count']>6].index,)
```

```
In [34]: train.isna().sum()
```

```
Out[34]: dropoff_latitude     0
         dropoff_longitude    0
         fare_amount          0
         passenger_count      0
```

```
        pickup_latitude      0
        pickup_longitude     0
        day                  0
        hour                 0
        weekday              0
        month                0
        year                 0
        dtype: int64
```

```
In [35]: print (train.shape)
         print (test.shape)
```

```
(4997735, 11)
(9688, 11)
```

```
In [36]: train.columns
```

```
Out[36]: Index(['dropoff_latitude', 'dropoff_longitude', 'fare_amount',
                'passenger_count', 'pickup_latitude', 'pickup_longitude', 'day', 'hour',
                'weekday', 'month', 'year'],
               dtype='object')
```

```
In [37]: iv=train[['dropoff_latitude', 'dropoff_longitude',
                'passenger_count', 'pickup_latitude', 'pickup_longitude', 'day', 'hour',
                'weekday', 'month', 'year']]
         dv=train[['fare_amount']]
```

```
In [38]: from sklearn.model_selection import train_test_split
         iv_train,iv_test,dv_train,dv_test = train_test_split(iv,dv,test_size=0.2,random_state=0
```

```
In [39]: from sklearn.preprocessing import StandardScaler
         sc = StandardScaler()
         iv_train=sc.fit_transform(iv_train)
         iv_test=sc.transform(iv_test)
```

```
/opt/anaconda3/lib/python3.7/site-packages/sklearn/preprocessing/data.py:625: DataConversionWarn
  return self.partial_fit(X, y)
/opt/anaconda3/lib/python3.7/site-packages/sklearn/base.py:462: DataConversionWarning: Data with
  return self.fit(X, **fit_params).transform(X)
/opt/anaconda3/lib/python3.7/site-packages/ipykernel_launcher.py:4: DataConversionWarning: Data
  after removing the cwd from sys.path.
```

```
In [40]: ## Logistic Regression Model
         from sklearn.linear_model import LinearRegression
         lin_reg = LinearRegression()
         lin_reg.fit(iv_train,dv_train)
```

```
Out[40]: LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None,
                  normalize=False)
```

```
In [41]: predictedvalues = lin_reg.predict(iv_test)

In [42]: #lets calculate rmse for linear Regression model
         from sklearn.metrics import mean_squared_error
         lrmodelrmse = np.sqrt(mean_squared_error(predictedvalues, dv_test))
         print("RMSE value for Linear regression is", lrmodelrmse)

RMSE value for Linear regression is 9.702547188137356


In [43]: #Lets see with Random Forest and calculate its rmse
         from sklearn.ensemble import RandomForestRegressor
         rfrmodel = RandomForestRegressor()

In [44]: # import warnings filter
         from warnings import simplefilter
         # ignore all future warnings
         simplefilter(action='ignore', category=FutureWarning)

In [45]: rfrmodel.fit(iv_train , dv_train.values.ravel())

Out[45]: RandomForestRegressor(bootstrap=True, criterion='mse', max_depth=None,
                    max_features='auto', max_leaf_nodes=None,
                    min_impurity_decrease=0.0, min_impurity_split=None,
                    min_samples_leaf=1, min_samples_split=2,
                    min_weight_fraction_leaf=0.0, n_estimators=10, n_jobs=None,
                    oob_score=False, random_state=None, verbose=0, warm_start=False)

In [57]: rfrmodel_pred= rfrmodel.predict(iv_test)

In [58]: rfrmodel_rmse=np.sqrt(mean_squared_error(rfrmodel_pred, dv_test))
         print("RMSE value for Random forest regression is ",rfrmodel_rmse)

RMSE value for Random forest regression is  4.4694011065534776


In [59]: rfrmodel.score(iv_train , dv_train)

Out[59]: 0.9551707955565059

In [60]: rfrmodel.score(iv_test , dv_test)

Out[60]: 0.7911548915822194

In [ ]:
```