In [274]:
```
import pandas as pd
import numpy as np
from sklearn.tree import DecisionTreeRegressor,DecisionTreeClassifier
from sklearn.model_selection import train_test_split
from sklearn.neighbors import KNeighborsRegressor
from sklearn.impute import SimpleImputer
from sklearn.model_selection import train_test_split, GridSearchCV
from sklearn.linear_model import LogisticRegression,SGDClassifier
from sklearn.ensemble import VotingClassifier, ExtraTreesClassifier,Ra
from sklearn.svm import SVC
from sklearn.metrics import mean_squared_error
from sklearn.neural_network import MLPClassifier

from sklearn.linear_model import LinearRegression
from sklearn.preprocessing import FunctionTransformer
from sklearn.preprocessing import StandardScaler,OneHotEncoder
from sklearn.pipeline import Pipeline
from sklearn.compose import ColumnTransformer

from sklearn.tree import DecisionTreeRegressor
from sklearn.model_selection import train_test_split
from sklearn.neighbors import KNeighborsRegressor
from sklearn.model_selection import train_test_split
pd.set_option('display.max_columns', 500)
```

Part 1, extract the mean time grouby (origin,dest,weekday,timeslot)

In [275]:
```
data=pd.read_csv('https://raw.githubusercontent.com/Xiao-Wang-UCSD/dat

```

In [276]:
```
data['Origin Name'].unique()
```

Out[276]:
```
array(['The Palace Of Fine Arts, 3601 Lyon St, San Francisco, CA',
       'Embarcadero, San Francisco, CA',
       'Oracle Park, 24 Willie Mays Plaza, San Francisco, CA',
       "Fisherman's Wharf, 286-298 Jefferson St, San Francisco, CA",
       '2nd Street and Stevenson Street, San Francisco, CA',
       'Powell BART Station, Market St and Powell St, San Francisco, C
A'],
      dtype=object)
```

In [518]:
```
def cleaning2(data,name):
    col=data['Daily Mean Travel Time']
    #extract the percentile of col in col
    func=lambda a:percentileofscore(col,a)
    percent=col.apply(func)
    new=data[name].dropna()
    #apply the percentile of col to name
    func2=lambda b:np.percentile(new, b)
    value=percent.apply(func2)
    data[name]=data[name].fillna(value)
    return data
```

In [ ]:
```
<option value="The Palace Of Fine Arts, 3601 Lyon St, San Francisco, CA"
<option value="Fisherman's Wharf, 286-298 Jefferson St, San Francisco, C
<option value="Oracle Park, 24 Willie Mays Plaza, San Francisco, CA">Ora
<option value="Embarcadero, San Francisco, CA">Embarcadero Bart</option>
<option value="2nd Street and Stevenson Street, San Francisco, CA">Montg
<option value="Powell BART Station, Market St and Powell St, San Francis
```

In [586]:
```
data=pd.read_csv('https://raw.githubusercontent.com/Xiao-Wang-UCSD/dat
data['Destination Name']=data['Destination Name'].replace({'2nd Street

column_name=['AM Mean Travel Time', 'AM Range - Lower',
       'AM Range - Upper', 'PM Mean Travel Time',
       'PM Range - Lower Bound Travel Time',
       'PM Range - Upper Bound Travel Time', 'Midday Mean Travel Time'
       'Midday Range - Lower', 'Midday Range - Upper',
       'Evening Mean Travel Time', 'Evening Range - Lower',
       'Evening Range - Upper', 'Early Morning Mean Travel Time',
       'Early Morning Range - Lower', 'Early Morning Range - Upper']
for i in column_name:
    data=cleaning2(data,i)

```

In [ ]:
```

```

In [576]:
```
data1=data[['Origin Name','Destination Name','AM Mean Travel Time','PM
name=list(data1.groupby(['Origin Name','Destination Name','weekday']).
frame=data1.groupby(['Origin Name','Destination Name','weekday']).mean

```

In [577]:
```
frame=frame.reset_index()
frame['weekday']=frame['weekday'].replace({0:'Mon',1:'Tues',2:'Wed',3:
```

In [578]:
```
# temp = frame.reset_index()
# temp = temp.groupby('Origin Name')[['Destination Name','AM Mean Trav
# temp

```

In [579]:
```
1 frame
```

	Origin Name	Destination Name	Weekday	Mean Travel Time	Mean Travel Time	Mean Travel Time	Mean Travel Time	M
7	2nd Street and Stevenson Street, San Francisco...	Oracle Park, 24 Willie Mays Plaza, San Francis...	Mon	443.320000	586.480000	537.400000	411.080000	3
8	2nd Street and Stevenson Street, San Francisco...	Oracle Park, 24 Willie Mays Plaza, San Francis...	Tues	467.769231	657.961538	526.461538	443.615385	3
9	2nd Street and Stevenson Street, San Francisco...	Oracle Park, 24 Willie Mays Plaza, San Francis...	Wed	512.153846	696.769231	584.884615	451.423077	3

In [ ]:
```
1
```

In [580]:
```
1 a=frame['Origin Name'].unique()
2 b=frame['Destination Name'].unique()
3 c=['Mon','Tues','Wed','Thur','Fri','Sat','Sun']
```

In [581]:
```
dict={}
for aa in a:
    dict[aa]={}
    for bb in b:
        dict[aa][bb]={}
        for cc in c:
            dict[aa][bb][cc]=0


dict

```

Out[581]:
```
{'2nd Street and Stevenson Street, San Francisco, CA': {"Fisherman's Whar
f, 286-298 Jefferson St, San Francisco, CA": {'Mon': 0,
   'Tues': 0,
   'Wed': 0,
   'Thur': 0,
   'Fri': 0,
   'Sat': 0,
   'Sun': 0},
  'Oracle Park, 24 Willie Mays Plaza, San Francisco, CA': {'Mon': 0,
   'Tues': 0,
   'Wed': 0,
   'Thur': 0,
   'Fri': 0,
   'Sat': 0,
   'Sun': 0},
  'The Palace Of Fine Arts, 3601 Lyon St, San Francisco, CA': {'Mon': 0,
   'Tues': 0,
   'Wed': 0,
   'Thur': 0,
```

In [582]:
```
for inx in range(frame.shape[0]):
    dict[frame.iloc[inx,0]][frame.iloc[inx,1]][frame.iloc[inx,2]]=list

dict
```

```
 689.6538461538462,
 511.38461538461536],
'Wed': [875.3076923076923,
 961.7307692307693,
 782.8076923076923,
 708.9615384615385,
 529.7307692307693],
'Thur': [931.9615384615385,
 981.7307692307693,
 814.7692307692307,
 749.5,
 512.9230769230769],
'Fri': [860.1923076923077,
 950.9230769230769,
 860.0384615384615,
 753.7692307692307,
 543.8461538461538],
'Sat': [668.3846153846154,
 875.2692307692307,
```

In [ ]:
```

```

In [583]:
```
import json

t=json.dumps(dict,indent=5)
with open("mean.json", "w") as outfile:
    outfile.write(t)
```

2 machine learning

In [741]:
```
data=pd.read_csv('https://raw.githubusercontent.com/Xiao-Wang-UCSD/dat
data['month']=data['Date'].apply(lambda a:pd.to_datetime(a).month)

column_name=['AM Mean Travel Time', 'AM Range - Lower',
       'AM Range - Upper', 'PM Mean Travel Time',
       'PM Range - Lower Bound Travel Time',
       'PM Range - Upper Bound Travel Time', 'Midday Mean Travel Time'
       'Midday Range - Lower', 'Midday Range - Upper',
       'Evening Mean Travel Time', 'Evening Range - Lower',
       'Evening Range - Upper', 'Early Morning Mean Travel Time',
       'Early Morning Range - Lower', 'Early Morning Range - Upper']
for i in column_name:
    data=cleaning2(data,i)

```

In [742]:
```
data['month']=data['month'].apply(lambda a: a+1)

```

In [743]:
```
# data1=data[['Origin Name','Destination Name','AM Mean Travel Time','

# data1=data[['Origin Name','Destination Name','AM Mean Travel Time','


data['Destination Name']=data['Destination Name'].replace({'2nd Street
data['Origin Name']=data['Origin Name'].replace({'2nd Street and Steve
# data.to_csv('out.csv')



```

In [744]:
```
neww=pd.DataFrame([])
for i in range(data.shape[0]):
    neww=neww.append(data[['Date']].loc[[i,i,i,i,i ] ])
neww
```

0	2019-06-09
0	2019-06-09
0	2019-06-09
0	2019-06-09
1	2019-04-28
1	2019-04-28
1	2019-04-28
1	2019-04-28
1	2019-04-28
2	2019-05-18
2	2019-05-18
2	2019-05-18
2	2019-05-18

In [745]:

```
new=pd.DataFrame([])
for i in range(data1.shape[0]):
    print(i)
    each_row=data1.iloc[i,:].to_frame().transpose()
    interval=list(each_row.iloc[0,2:7])
    nkm=data1.loc[[i,i,i,i,i]]
    nkm['result']=interval
    nkm=nkm.drop(['AM Mean Travel Time','PM Mean Travel Time','Midday
    nkm['interval']=[0,1,2,3,4]
    nkm=nkm.reset_index(drop=True)
    new=new.append(nkm)

new.head()
# for i in range(data1.shape[0]):

```

```
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
```

In [746]:
```
new=new.reset_index(drop=True)
new
```

0	3601 Lyon St, San Fra...	Francisco, CA	6	7	1380.000000	0
1	The Palace Of Fine Arts, 3601 Lyon St, San Fra...	Embarcadero, San Francisco, CA	6	7	1252.000000	1
2	The Palace Of Fine Arts, 3601 Lyon St, San Fra...	Embarcadero, San Francisco, CA	6	7	1398.500000	2
3	The Palace Of Fine Arts, 3601 Lyon St, San Fra...	Embarcadero, San Francisco, CA	6	7	1149.500000	3
4	The Palace Of Fine Arts, 3601 Lyon St, San Fra...	Embarcadero, San Francisco, CA	6	7	1228.000000	4
5	Embarcadero, San Francisco, CA	Fisherman's Wharf, 286-298 Jefferson St, San F...	6	5	579.000000	0
6	Embarcadero, San Francisco, CA	Fisherman's Wharf, 286-298 Jefferson St, San F...	6	5	664.000000	1
7	Embarcadero, San Francisco, CA	Fisherman's Wharf, 286-298 Jefferson St, San F...	6	5	654.000000	2
8	Embarcadero, San Francisco, CA	Fisherman's Wharf, 286-298 Jefferson St, San F...	6	5	608.000000	3

In [747]:
```
import math
from sklearn.ensemble import RandomForestRegressor
from sklearn.ensemble import GradientBoostingRegressor

def train(train_df):
    train_df = train_df.copy()
    X_train, X_test, y_train, y_test = train_test_split(train_df.drop(

    col_c3 = ['Destination Name','Origin Name','weekday','month','inte
    c3_transformer = Pipeline(steps=[('onehot', OneHotEncoder())])

    preproc = ColumnTransformer(transformers=[
                                            ('c3', c3_transformer, co

    pl = Pipeline(steps=[('preprocessor', preproc), ('regressor', Deci
    pl.fit(X_train,y_train)
    pred = pl.predict(X_test)
    rms = math.sqrt(mean_squared_error(y_test, pred))
    preds = pl.predict(train_df.drop(['result'],axis=1))
    train_df['pred'] = preds
    return rms,train_df
```

In [753]:
```
# ddddd=train(new)[1]
# neww=neww.reset_index(drop=True)
# ddddd['date']=neww['Date']
# ddddd.to_csv('outpututtt.csv')
import statsmodels

```

In [ ]:
```

```