

Linear Regression And Random Forest Regressor

Name : Satbhai Anket Satbhai

Roll No. : 4286, Batch : B7

```
In [1]: import pandas as pd
df=pd.read_csv('C:\\Users\\DELL\\Desktop\\uber.csv')
df.head()
```

```
Out[1]:
```

	Unnamed: 0	key	fare_amount	pickup_datetime	pickup_longitude	pickup_latitude
0	24238194	2015-05-07 19:52:06.0000003	7.5	2015-05-07 19:52:06 UTC	-73.999817	40.738354
1	27835199	2009-07-17 20:04:56.0000002	7.7	2009-07-17 20:04:56 UTC	-73.994355	40.728225
2	44984355	2009-08-24 21:45:00.00000061	12.9	2009-08-24 21:45:00 UTC	-74.005043	40.740770
3	25894730	2009-06-26 08:22:21.0000001	5.3	2009-06-26 08:22:21 UTC	-73.976124	40.790844
4	17610152	2014-08-28 17:47:00.000000188	16.0	2014-08-28 17:47:00 UTC	-73.925023	40.744085

```
In [2]: df.shape
```

```
Out[2]: (200000, 9)
```

```
In [4]: df.isna().sum()
```

```
Out[4]: Unnamed: 0      0
key      0
fare_amount      0
pickup_datetime      0
pickup_longitude      0
pickup_latitude      0
dropoff_longitude    1
dropoff_latitude     1
passenger_count      0
dtype: int64
```

```
In [5]: df=df.dropna(axis=0)
```

```
In [6]: df.isna().sum()
```

```
Out[6]: Unnamed: 0      0
         key          0
         fare_amount  0
         pickup_datetime  0
         pickup_longitude  0
         pickup_latitude  0
         dropoff_longitude  0
         dropoff_latitude  0
         passenger_count  0
         dtype: int64
```

```
In [7]: df.shape
```

```
Out[7]: (199999, 9)
```

```
In [8]: df.dtypes
```

```
Out[8]: Unnamed: 0      int64
         key          object
         fare_amount  float64
         pickup_datetime  object
         pickup_longitude  float64
         pickup_latitude  float64
         dropoff_longitude  float64
         dropoff_latitude  float64
         passenger_count  int64
         dtype: object
```

```
In [9]: df['pickup_datetime']=pd.to_datetime(df['pickup_datetime'])
         df.dtypes
```

```
Out[9]: Unnamed: 0      int64
         key          object
         fare_amount  float64
         pickup_datetime  datetime64[ns, UTC]
         pickup_longitude  float64
         pickup_latitude  float64
         dropoff_longitude  float64
         dropoff_latitude  float64
         passenger_count  int64
         dtype: object
```

```
In [10]: df=df.assign(hour=df.pickup_datetime.dt.hour,
                       day=df.pickup_datetime.dt.day,
                       month=df.pickup_datetime.dt.month,
                       year=df.pickup_datetime.dt.year,
                       dayofweek=df.pickup_datetime.dt.dayofweek)
```

```
In [11]: df.shape
```

```
Out[11]: (199999, 14)
```

```
In [12]: df.columns
```

```
Out[12]: Index(['Unnamed: 0', 'key', 'fare_amount', 'pickup_datetime',  
              'pickup_longitude', 'pickup_latitude', 'dropoff_longitude',  
              'dropoff_latitude', 'passenger_count', 'hour', 'day', 'month', 'year',  
              'dayofweek'],  
              dtype='object')
```

```
In [13]: x=df.drop(['Unnamed: 0','key','fare_amount','pickup_datetime'],axis=1)  
        y=df['fare_amount']
```

```
In [14]: from sklearn.model_selection import train_test_split  
        x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.30,random_state=51)
```

```
In [15]: x_train.shape
```

```
Out[15]: (139999, 10)
```

```
In [16]: x_test.shape
```

```
Out[16]: (60000, 10)
```

Linear Regression

```
In [17]: from sklearn.linear_model import LinearRegression  
        lr=LinearRegression()
```

```
In [18]: lr.fit(x_train,y_train)
```

```
Out[18]: LinearRegression()
```

```
In [19]: y_pred=lr.predict(x_test,)
```

```
In [20]: from sklearn import metrics as mt
```

```
In [22]: #MAE  
        mt.mean_absolute_error(y_pred,y_test)
```

```
Out[22]: 5.930443606838259
```

```
In [23]: #MSE  
        a=mt.mean_squared_error(y_pred,y_test)  
        a
```

```
Out[23]: 91.85349772689274
```

```
In [24]: #RMSE  
a**(1/2)
```

Out[24]: 9.584023044989653

Random Forest Regressor

```
In [25]: from sklearn.ensemble import RandomForestRegressor  
rf=RandomForestRegressor(n_estimators=100)
```

```
In [26]: rf.fit(x_train,y_train)
```

Out[26]: RandomForestRegressor()

```
In [27]: y_pred=rf.predict(x_test)
```

```
In [28]: #MAE  
mt.mean_absolute_error(y_pred,y_test)
```

Out[28]: 1.9943375981746032

```
In [29]: #MSE  
a=mt.mean_squared_error(y_pred,y_test)  
a
```

Out[29]: 20.383110788803346

```
In [30]: #RMSE  
a**(1/2)
```

Out[30]: 4.5147658620135935