

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

df=pd.read_csv("train_data.csv")
df_test=pd.read_csv("validation_data.csv")

In [ ]: df['DateID'] = pd.to_datetime(df['DateID'])

In [ ]: df['month']= df['DateID'].dt.month
df['day']= df['DateID'].dt.day
df['week_of_month']=(df['day']/7)+1

In [ ]: df['week_of_year'] = df['DateID'].dt.week

/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:1: FutureWarning: Series.dt.week
of year and Series.dt.week have been deprecated. Please use Series.dt.isocalendar().week inst
ead.
"""Entry point for launching an IPython kernel.

In [ ]:
```

```
In [ ]: cleanup_week_of_year = {"week_of_year":          {39: 1, 40: 2,41:3,42:4,43:5,44:6,45:7,46:8,47:9
,48:10,49:11,50:12,51:13,52:14,1:15,2:16,3:17,4:18,5:19,6:20,7:21,8:22,9:23,10:24}}
df = df.replace(cleanup_week_of_year)

In [ ]: cleanup_month = {"month":          {10: 1, 11: 2,12:3,1:4,2:5,3:6}}
df = df.replace(cleanup_month)

In [ ]: df.drop('day', axis=1, inplace=True)

Manipulating the validation dataset

In [ ]: df["month"].value_counts()
df_test['weekOG']=df_test['Week']
y_test=df_test['WeeklySales']
df_test.drop('WeeklySales', axis=1, inplace=True)

cleanup_month_test = {"Week":          {"w1": 5, "w2": 5, "w3": 6, "w4": 6}}
df_test = df_test.replace(cleanup_month_test)
df_test = df_test.rename(columns={'Week': 'month'})

df_test['weekOG2']=df_test['WeekOG']
cleanup_week_of_month_test = {"WeekOG":          {"w1": 3, "w2": 4, "w3": 1, "w4": 2}}
df_test = df_test.replace(cleanup_week_of_month_test)
df_test = df_test.rename(columns={'WeekOG': 'week_of_month'})

df_test['weekOG3']=df_test['WeekOG2']
cleanup_week_of_year_test = {"WeekOG2":          {"w1": 21, "w2": 22, "w3": 23, "w4": 24}}
df_test = df_test.replace(cleanup_week_of_year_test)
df_test = df_test.rename(columns={'weekOG2': 'week_of_year'})
df_test.drop('WeekOG3', axis=1, inplace=True)

cleanup_category_code = {"CategoryCode":          {"category_1": 1, "category_2": 2, "category_3":
3, "category_4": 4}}
df_test = df_test.replace(cleanup_category_code)
df = df.replace(cleanup_category_code)

df['WeeklySales']=df.groupby(['CategoryCode', 'ItemCode', 'week_of_year'])['DailySales'].trans
form(sum)
y_train=df['WeeklySales']
df.drop('DailySales', axis=1, inplace=True)
df.drop('WeeklySales', axis=1, inplace=True)
df
```

Out []:

	CategoryCode	ItemCode	DateID	month	week_of_month	week_of_year
0	2	117610	2021-11-06	2	1	6
1	4	836584	2021-11-18	2	3	8
2	1	370195	2022-01-24	4	4	18
3	2	172582	2021-10-30	1	5	5
4	2	1006009	2021-10-30	1	5	5
...
19916	2	225259	2021-10-04	1	1	2
19917	2	111436	2021-10-12	1	2	3
19918	2	1098502	2021-10-01	1	1	1
19919	2	20824	2021-10-01	1	1	1
19920	1	371104	2021-10-04	1	1	2

19921 rows x 6 columns

```
In [ ]: df.drop('DateID', axis=1, inplace=True)
df

Out [ ]:
```

	CategoryCode	ItemCode	month	week_of_month	week_of_year
0	2	117610	2	1	6
1	4	836584	2	3	8
2	1	370195	4	4	18
3	2	172582	1	5	5
4	2	1006009	1	5	5
...
19916	2	225259	1	1	2
19917	2	111436	1	2	3
19918	2	1098502	1	1	1
19919	2	20824	1	1	1
19920	1	371104	1	1	2

19921 rows x 5 columns

```
In [ ]:

In [ ]: x_train=df
x_test=df_test
y_train

Out [ ]:
```

0	65
1	123
2	37
3	26
4	13
...	...
19916	17
19917	11
19918	3
19919	53
19920	6

Name: WeeklySales, Length: 19921, dtype: int64

LightGBM

```
In [ ]: df_realtest=pd.read_csv("test_data.csv")

df_realtest = df_realtest.replace(cleanup_category_code)

cleanup_month_test = {"Week":          {"w1": 5, "w2": 5, "w3": 6, "w4": 6}}
df_realtest = df_realtest.replace(cleanup_month_test)
df_realtest = df_realtest.rename(columns={'Week': 'month'})
df_realtest.drop("PredictedSales",axis=1, inplace=True)
df_realtest

Out [ ]:
```

	CategoryCode	ItemCode	month
0	1	43738	6
1	2	1006090	5
2	2	1076929	6
3	1	1081321	6
4	2	216151	6
...
372	2	1101571	5
373	2	1090258	6
374	2	906595	5
375	2	32245	5
376	2	1006090	5

377 rows x 3 columns

```
In [ ]: x_train.drop('week_of_month', axis=1, inplace=True)
x_test.drop('week_of_month', axis=1, inplace=True)

x_train.drop('week_of_year', axis=1, inplace=True)
x_test.drop('week_of_year', axis=1, inplace=True)
x_train

Out [ ]:
```

	CategoryCode	ItemCode	month
0	2	117610	2
1	4	836584	2
2	1	370195	4
3	2	172582	1
4	2	1006009	1
...
19916	2	225259	1
19917	2	111436	1
19918	2	1098502	1
19919	2	20824	1
19920	1	371104	1

19921 rows x 3 columns

```
In [ ]: x_test

Out [ ]:
```

	CategoryCode	ItemCode	month
0	2	1044502	5
1	2	1105009	5
2	2	913561	6
3	1	1048975	6
4	1	17287	5
...
365	2	124954	5
366	2	40759	5
367	1	1090303	5
368	2	1090276	6
369	1	3418	6

370 rows x 3 columns

```
In [ ]: import lightgbm as lgb
lgbm=lgb.LGBMRegressor(n_estimators=2000,learning_rate=0.3)

lgbm.fit(x_train,y_train)
# y_final_pred=(lgbm.predict(df_realtest))

Out [ ]: LGBMRegressor(learning_rate=0.3, n_estimators=2000)
```

```
In [ ]: import lightgbm as lgb
from sklearn.metrics import mean_absolute_percentage_error
lgbm=lgb.LGBMRegressor(n_estimators=2000,learning_rate=0.3)

lgbm.fit(x_train,y_train)
y_pred=(lgbm.predict(x_test))
mean_absolute_percentage_error(y_test,y_pred)
print("mean_absolute_percentage_error",mean_absolute_percentage_error(y_test,y_pred))
from sklearn.metrics import mean_squared_error,r2_score
print("r2_score",r2_score(y_test,y_pred))
print((abs(y_pred-y_test).sum()/(y_test.sum()))

mean_absolute_percentage_error 0.6021795578373091
r2_score 0.6985989405684123
0.4273651837431775
```

```
In [ ]:

In [ ]: print("mean_absolute_percentage_error",mean_absolute_percentage_error(y_test,y_pred))
from sklearn.metrics import mean_squared_error,r2_score
print("r2_score",r2_score(y_test,y_pred))
print(abs(y_pred-y_test).sum()/(y_test.sum()))
d1 = pd.Series(y_pred,name='PredictedSales')
df=pd.concat([y_test, d1], axis=1)
df

mean_absolute_percentage_error 0.6021795578373091
r2_score 0.6985989405684123
0.4273651837431775
```

Out []:

	WeeklySales	PredictedSales
0	11	16.364021
1	11	9.998328
2	5	11.798610
3	30	48.382069
4	60	19.625502
...
365	43	29.693255
366	48	39.504478
367	19	9.224322
368	6	10.821454
369	69	89.074630

370 rows x 2 columns

```
In [ ]: df=pd.read_csv("test_data.csv")
df['random']= df['ItemCode'].astype(str)
df['ID']=df['CategoryCode']+'_'+df['random']+'_'+df['Week']
df
df2=df["ID"]
df3=pd.DataFrame(df2)
df3["WeeklySales"]=y_final_pred
df3.to_csv("submission8.csv",index=False)
```

```
In [ ]:

In [ ]:
```

```
In [ ]: import xgboost as xgb

xgbr = xgb.XGBRegressor(learning_rate=0.3,n_estimators=4300,objective='reg:squarederror')

xgbr.fit(x_train, y_train)
ypred = xgbr.predict(df_realtest)
df_ypred=pd.DataFrame(ypred)

df=pd.read_csv("test_data.csv")
df['random']= df['ItemCode'].astype(str)
df['ID']=df['CategoryCode']+'_'+df['random']+'_'+df['Week']
df
df2=df["ID"]
df3=pd.DataFrame(df2)
df3["WeeklySales"]=df_ypred
df3.to_csv("Submission8.csv",index=False)
```

```
In [ ]: df3

Out [ ]:
```

	ID	WeeklySales
0	category_1_43738_w4	24.808975
1	category_2_1006090_w1	19.345791
2	category_2_1076929_w4	4.309184
3	category_1_1081321_w3	7.761096
4	category_2_216151_w4	13.905788
...
372	category_2_1101571_w1	14.060581
373	category_2_1090258_w4	27.882864
374	category_2_906595_w1	10.763870
375	category_2_32245_w1	51.652836
376	category_2_1006090_w2	19.345791

377 rows x 2 columns

```
In [ ]: from sklearn.metrics import mean_squared_error,r2_score
r2_score(y_test,df_y_pred2)
```

```
Out [ ]: 0.7002779533170033
```

```
In [ ]: (abs(y_pred2-y_test).sum()/(y_test.sum()))

Out [ ]: 0.42587982234637767
```