tdnzxp3dl

July 21, 2023

[1]:
```python
import pandas as pd
import numpy as np
data = pd.read_csv("https://raw.githubusercontent.com/amankharwal/Website-data/
↪master/supplement.csv")
data
```

[1]:
```
             ID  Store_id Store_Type Location_Type Region_Code        Date  \
0       T1000001         1         S1            L3          R1  2018-01-01
1       T1000002       253         S4            L2          R1  2018-01-01
2       T1000003       252         S3            L2          R1  2018-01-01
3       T1000004       251         S2            L3          R1  2018-01-01
4       T1000005       250         S2            L3          R4  2018-01-01
…             …       …          …            …           …          …
188335  T1188336       149         S2            L3          R2  2019-05-31
188336  T1188337       153         S4            L2          R1  2019-05-31
188337  T1188338       154         S1            L3          R2  2019-05-31
188338  T1188339       155         S3            L1          R2  2019-05-31
188339  T1188340       152         S2            L1          R1  2019-05-31

        Holiday Discount  #Order     Sales
0             1      Yes       9   7011.84
1             1      Yes      60  51789.12
2             1      Yes      42  36868.20
3             1      Yes      23  19715.16
4             1      Yes      62  45614.52
…           …      …       …         …
188335        1      Yes      51  37272.00
188336        1       No      90  54572.64
188337        1       No      56  31624.56
188338        1      Yes      70  49162.41
188339        1       No      47  37977.00

[188340 rows x 10 columns]
```

[2]:
```python
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 188340 entries, 0 to 188339
```

```
Data columns (total 10 columns):
 #   Column         Non-Null Count   Dtype
---  ------         --------------   -----
 0   ID             188340 non-null  object
 1   Store_id       188340 non-null  int64
 2   Store_Type     188340 non-null  object
 3   Location_Type  188340 non-null  object
 4   Region_Code    188340 non-null  object
 5   Date           188340 non-null  object
 6   Holiday        188340 non-null  int64
 7   Discount       188340 non-null  object
 8   #Order         188340 non-null  int64
 9   Sales          188340 non-null  float64
dtypes: float64(1), int64(3), object(6)
memory usage: 14.4+ MB
```

[3]:
```
data.isnull().sum()
```

[3]:
```
ID               0
Store_id         0
Store_Type       0
Location_Type    0
Region_Code      0
Date             0
Holiday          0
Discount         0
#Order           0
Sales            0
dtype: int64
```

[4]:
```
data.describe()
```

[4]:

	Store_id	Holiday	#Order	Sales
count	188340.000000	188340.000000	188340.000000	188340.000000
mean	183.000000	0.131783	68.205692	42784.327982
std	105.366308	0.338256	30.467415	18456.708302
min	1.000000	0.000000	0.000000	0.000000
25%	92.000000	0.000000	48.000000	30426.000000
50%	183.000000	0.000000	63.000000	39678.000000
75%	274.000000	0.000000	82.000000	51909.000000
max	365.000000	1.000000	371.000000	247215.000000

[5]:
```
import plotly.express as px
pie = data["Store_Type"].value_counts()
store = pie.index
orders = pie.values
```

```
fig = px.pie(data, values=orders, names=store)
fig.show()
```

[6]:
```
pie2 = data["Location_Type"].value_counts()
location = pie2.index
orders = pie2.values

fig = px.pie(data, values=orders, names=location)
fig.show()
```

[7]:
```
pie3 = data["Discount"].value_counts()
discount = pie3.index
orders = pie3.values

fig = px.pie(data, values=orders, names=discount)
fig.show()
```

[8]:
```
pie4 = data["Holiday"].value_counts()
holiday = pie4.index
orders = pie4.values

fig = px.pie(data, values=orders, names=holiday)
fig.show()
```

[9]:
```
data["Discount"] = data["Discount"].map({"No": 0, "Yes": 1})
data
```

[9]:
```
              ID  Store_id Store_Type Location_Type Region_Code        Date  \
0       T1000001         1         S1            L3          R1  2018-01-01
1       T1000002       253         S4            L2          R1  2018-01-01
2       T1000003       252         S3            L2          R1  2018-01-01
3       T1000004       251         S2            L3          R1  2018-01-01
4       T1000005       250         S2            L3          R4  2018-01-01
…            …       …          …             …          …           …
188335  T1188336       149         S2            L3          R2  2019-05-31
188336  T1188337       153         S4            L2          R1  2019-05-31
188337  T1188338       154         S1            L3          R2  2019-05-31
188338  T1188339       155         S3            L1          R2  2019-05-31
188339  T1188340       152         S2            L1          R1  2019-05-31

        Holiday  Discount  #Order     Sales
0             1         1       9   7011.84
1             1         1      60  51789.12
2             1         1      42  36868.20
3             1         1      23  19715.16
4             1         1      62  45614.52
…          …         …      …        …
```

```
188335         1         1       51   37272.00
188336         1         0       90   54572.64
188337         1         0       56   31624.56
188338         1         1       70   49162.41
188339         1         0       47   37977.00

[188340 rows x 10 columns]
```

[10]:
```
data["Store_Type"] = data["Store_Type"].map({"S1": 1, "S2": 2,"S3": 3, "S4": 4})
data
```

[10]:
```
             ID  Store_id  Store_Type Location_Type Region_Code        Date  \
0      T1000001         1           1            L3          R1  2018-01-01
1      T1000002       253           4            L2          R1  2018-01-01
2      T1000003       252           3            L2          R1  2018-01-01
3      T1000004       251           2            L3          R1  2018-01-01
4      T1000005       250           2            L3          R4  2018-01-01
…             …       …           …             …           …           …
188335 T1188336       149           2            L3          R2  2019-05-31
188336 T1188337       153           4            L2          R1  2019-05-31
188337 T1188338       154           1            L3          R2  2019-05-31
188338 T1188339       155           3            L1          R2  2019-05-31
188339 T1188340       152           2            L1          R1  2019-05-31

        Holiday  Discount  #Order     Sales
0             1         1       9   7011.84
1             1         1      60  51789.12
2             1         1      42  36868.20
3             1         1      23  19715.16
4             1         1      62  45614.52
…           …         …       …         …
188335        1         1      51  37272.00
188336        1         0      90  54572.64
188337        1         0      56  31624.56
188338        1         1      70  49162.41
188339        1         0      47  37977.00

[188340 rows x 10 columns]
```

[11]:
```
data["Location_Type"] = data["Location_Type"].map({"L1": 1, "L2": 2,"L3": 3,␣
↪"L4": 4,"L5": 5})
data
```

[11]:
```
             ID  Store_id  Store_Type  Location_Type Region_Code        Date  \
0      T1000001         1           1              3          R1  2018-01-01
1      T1000002       253           4              2          R1  2018-01-01
2      T1000003       252           3              2          R1  2018-01-01
```

```
3       T1000004      251          2           3          R1  2018-01-01
4       T1000005      250          2           3          R4  2018-01-01
…          …       …        …          …          …        …
188335  T1188336      149          2           3          R2  2019-05-31
188336  T1188337      153          4           2          R1  2019-05-31
188337  T1188338      154          1           3          R2  2019-05-31
188338  T1188339      155          3           1          R2  2019-05-31
188339  T1188340      152          2           1          R1  2019-05-31

        Holiday  Discount  #Order     Sales
0             1         1       9   7011.84
1             1         1      60  51789.12
2             1         1      42  36868.20
3             1         1      23  19715.16
4             1         1      62  45614.52
…          …       …      …       …
188335        1         1      51  37272.00
188336        1         0      90  54572.64
188337        1         0      56  31624.56
188338        1         1      70  49162.41
188339        1         0      47  37977.00

[188340 rows x 10 columns]
```

[12]:
```
X=np.array(data[["Store_Type","Location_Type","Holiday","Discount"]])
y=np.array(data["#Order"])
```

[13]:
```
X
```

[13]:
```
array([[1, 3, 1, 1],
       [4, 2, 1, 1],
       [3, 2, 1, 1],
       …,
       [1, 3, 1, 0],
       [3, 1, 1, 1],
       [2, 1, 1, 0]], dtype=int64)
```

[14]:
```
y
```

[14]:
```
array([ 9, 60, 42, …, 56, 70, 47], dtype=int64)
```

[15]:
```
from sklearn.model_selection import train_test_split
```

[16]:
```
X_train, X_test, y_train,y_test= train_test_split(X,y,test_size=0.2,␣
 ↪random_state=42)
```

[17]:
```
len(X_train)
```

[17]: 150672

[18]:
```
!pip install lightgbm
```

```
Requirement already satisfied: lightgbm in c:\users\hp\anaconda3\lib\site-
packages (4.0.0)
Requirement already satisfied: numpy in c:\users\hp\anaconda3\lib\site-packages
(from lightgbm) (1.21.5)
Requirement already satisfied: scipy in c:\users\hp\anaconda3\lib\site-packages
(from lightgbm) (1.9.1)
```

[19]:
```
import lightgbm as ltb
model=ltb.LGBMRegressor()
```

[21]:
```
model.fit(X_train,y_train)
```

```
[LightGBM] [Warning] Auto-choosing row-wise multi-threading, the overhead of
testing was 0.001399 seconds.
You can set `force_row_wise=true` to remove the overhead.
And if memory is not enough, you can set `force_col_wise=true`.
[LightGBM] [Info] Total Bins 15
[LightGBM] [Info] Number of data points in the train set: 150672, number of used
features: 4
[LightGBM] [Info] Start training from score 68.163401
```

[21]: LGBMRegressor()

[22]:
```
y_pred=model.predict(X_test)
```

[23]:
```
y_pred
```

[23]:
```
array([47.35189701, 97.06871721, 66.57778822, …, 47.35189701,
       61.74938636, 85.34103853])
```

[24]:
```
y_test
```

[24]:
```
array([ 54, 111,  59, …,  40,  69,  68], dtype=int64)
```

[26]:
```
data_pred=pd.DataFrame(data={"Predicted Orders":y_pred.flatten()})
```

[27]:
```
data
```

[27]:

	ID	Store_id	Store_Type	Location_Type	Region_Code	Date	\
0	T1000001	1	1	3	R1	2018-01-01	
1	T1000002	253	4	2	R1	2018-01-01	
2	T1000003	252	3	2	R1	2018-01-01	
3	T1000004	251	2	3	R1	2018-01-01	

```
4       T1000005       250          2             3          R4  2018-01-01
…          …       …         …            …          …         …
188335  T1188336       149          2             3          R2  2019-05-31
188336  T1188337       153          4             2          R1  2019-05-31
188337  T1188338       154          1             3          R2  2019-05-31
188338  T1188339       155          3             1          R2  2019-05-31
188339  T1188340       152          2             1          R1  2019-05-31

        Holiday  Discount  #Order     Sales
0             1         1       9   7011.84
1             1         1      60  51789.12
2             1         1      42  36868.20
3             1         1      23  19715.16
4             1         1      62  45614.52
…          …        …       …        …
188335        1         1      51  37272.00
188336        1         0      90  54572.64
188337        1         0      56  31624.56
188338        1         1      70  49162.41
188339        1         0      47  37977.00

[188340 rows x 10 columns]
```

```
[ ]:
```