applied-data-science-phase-3-1

October 28, 2023

1 Date - 26/10/2023

2 Team ID - 3872

3 Project Title - Product Demand Prediction using ML

4 Importing Dependencies

```
[11]: import pandas as pd
      import re
      import matplotlib.pyplot as plt
      import os
      import plotly.express as px
      import numpy as np
      import seaborn as sns
      import matplotlib.pyplot as plt
```

5 Loading Dataset

```
[2]: df = pd.read_csv("F:\\Applied_dataScience_Phase4\\trainnew.csv")
```

6 Data Exploration

```
[3]: df
```

```
[3]:             date  store  item  sales
     0       01-01-2013      1     1     13
     1       02-01-2013      1     1     11
     2       03-01-2013      1     1     14
     3       04-01-2013      1     1     13
     4       05-01-2013      1     1     10
     …             …     …     …     …
     912995  27-12-2017     10    50     63
     912996  28-12-2017     10    50     59
     912997  29-12-2017     10    50     74
```

```
912998  30-12-2017     10     50      62
912999  31-12-2017     10     50      82

[913000 rows x 4 columns]
```

[4]:

```
df.set_index('date',inplace=True)
```

[5]:

```
df.head()
```

[5]:

```
            store  item  sales
date
01-01-2013      1     1     13
02-01-2013      1     1     11
03-01-2013      1     1     14
04-01-2013      1     1     13
05-01-2013      1     1     10
```

[9]:

```
df.tail()
```

[9]:

```
            store  item  sales
date
27-12-2017     10    50     63
28-12-2017     10    50     59
29-12-2017     10    50     74
30-12-2017     10    50     62
31-12-2017     10    50     82
```

[6]:

```
df.describe()
```

[6]:

```
               store           item          sales
count  913000.000000  913000.000000  913000.000000
mean        5.500000      25.500000      52.250287
std         2.872283      14.430878      28.801144
min         1.000000       1.000000       0.000000
25%         3.000000      13.000000      30.000000
50%         5.500000      25.500000      47.000000
75%         8.000000      38.000000      70.000000
max        10.000000      50.000000     231.000000
```

[7]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 913000 entries, 01-01-2013 to 31-12-2017
Data columns (total 3 columns):
 #   Column  Non-Null Count   Dtype
---  ------  --------------   -----
 0   store   913000 non-null  int64
 1   item    913000 non-null  int64
 2   sales   913000 non-null  int64
dtypes: int64(3)
memory usage: 27.9+ MB
```

[8]:
```
df.shape
```

[8]:
```
(913000, 3)
```

[7]:
```
store_sales=df.groupby(by='store')[['sales']].sum()
store_sales
```

[7]:
```
         sales
store
1      4315603
2      6120128
3      5435144
4      5012639
5      3631016
6      3627670
7      3320009
8      5856169
9      5025976
10     5360158
```

[8]:
```
store=store_sales.index
store
```

[8]:
```
Int64Index([1, 2, 3, 4, 5, 6, 7, 8, 9, 10], dtype='int64', name='store')
```

7 Pre-Processing and Visualisation of Data

[9]:
```
fig = px.bar(store_sales,color=store)
fig.show()
```

[10]:
```
fig = px.histogram(df[df.item==1][['sales']],labels=dict(value="Sales"))
fig.show()
```

[11]:
```
fig = px.line(df[(df.item==1) & (df.store==4)][['sales']],y='sales')
fig.show()
```

[12]:
```
df_1_1=df[(df.item==1) & (df.store==1)][['sales']]
df_1_1
```

[12]:
```
            sales
date
```

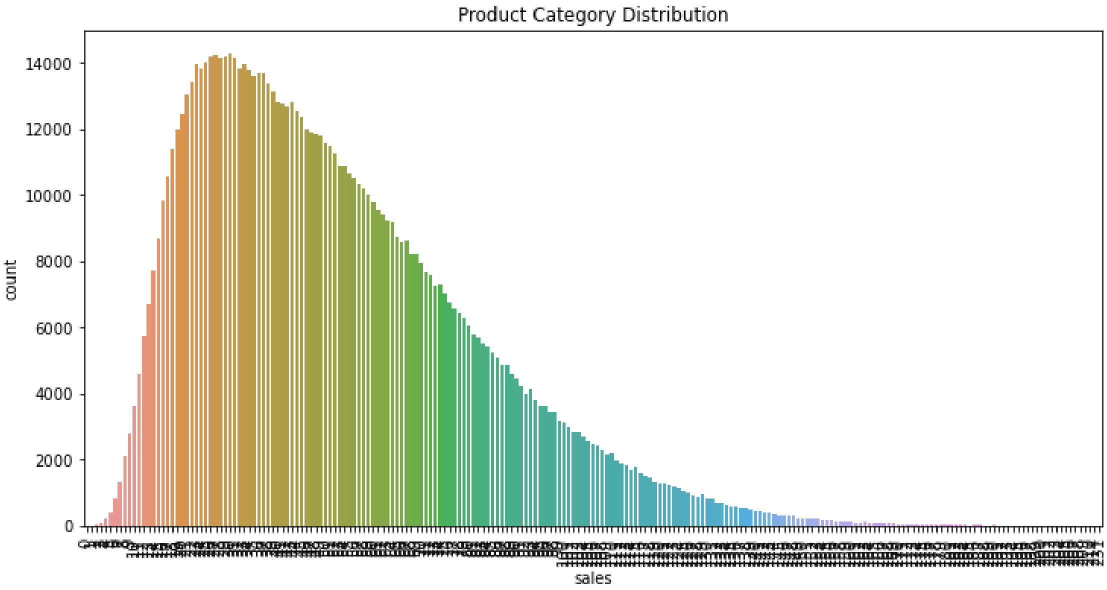
```
01-01-2013     13
02-01-2013     11
03-01-2013     14
04-01-2013     13
05-01-2013     10
…            …
27-12-2017     14
28-12-2017     19
29-12-2017     15
30-12-2017     27
31-12-2017     23

[1826 rows x 1 columns]
```

[13]:

```python
fig = px.line(df_1_1)
fig.show()
```

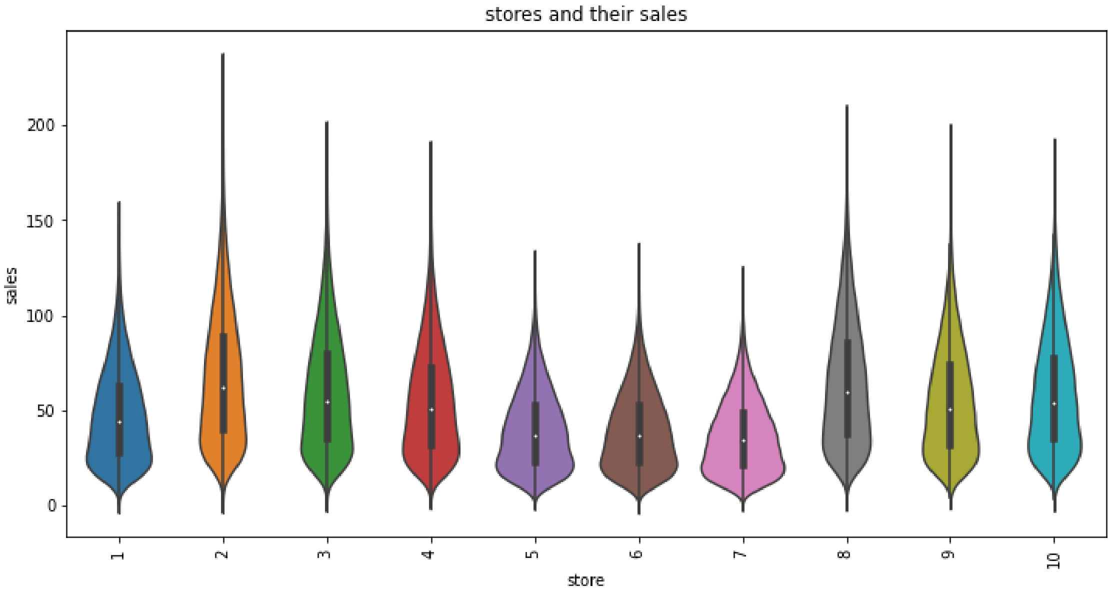
[16]:

```python
plt.figure(figsize=(12, 6))
sns.countplot(data=df, x='sales')
plt.title('Product Category Distribution')
plt.xticks(rotation=90)
plt.show()
```



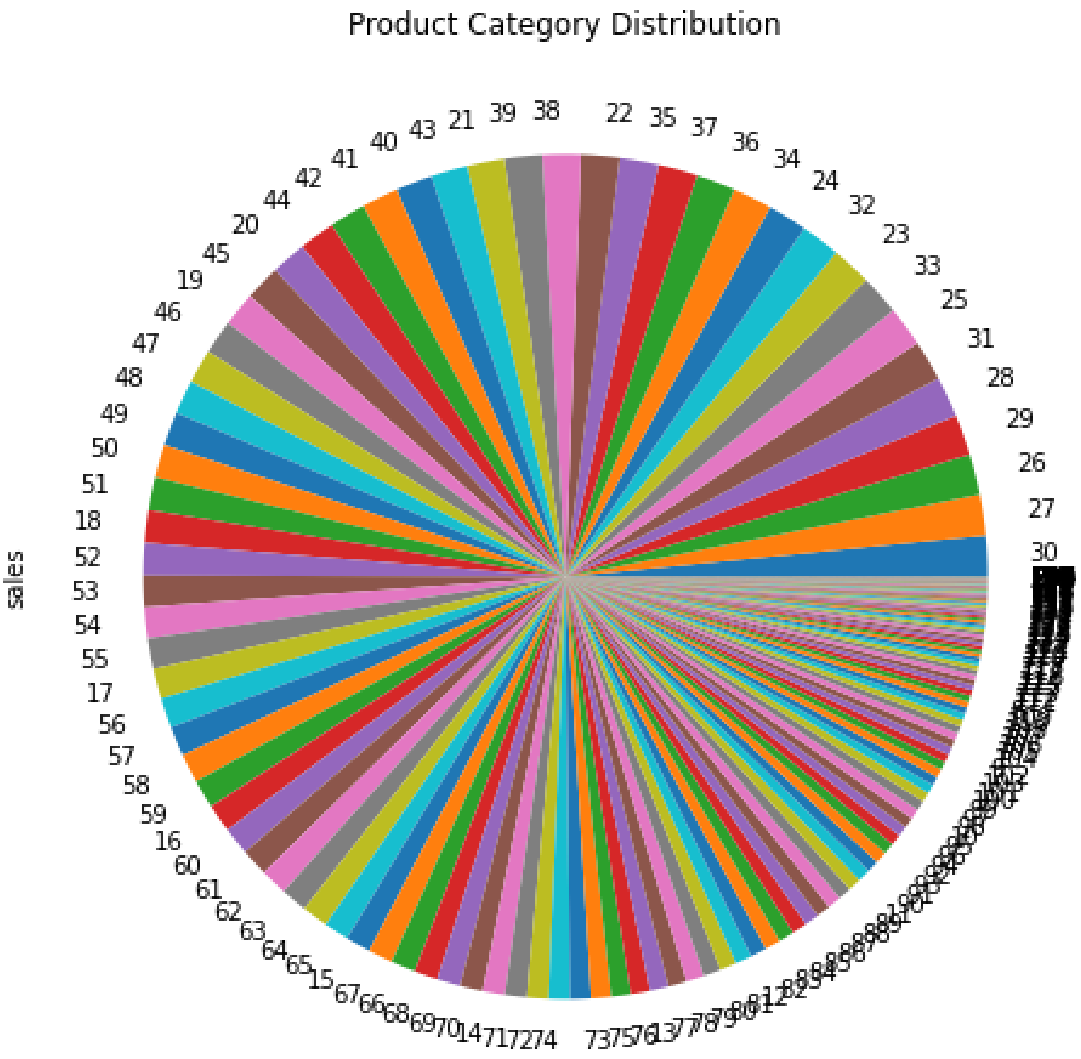
[17]:

```python
plt.figure(figsize=(12, 6))
sns.violinplot(data=df, x='store', y='sales')
plt.title('stores and their sales')
plt.xticks(rotation=90)
plt.show()
```



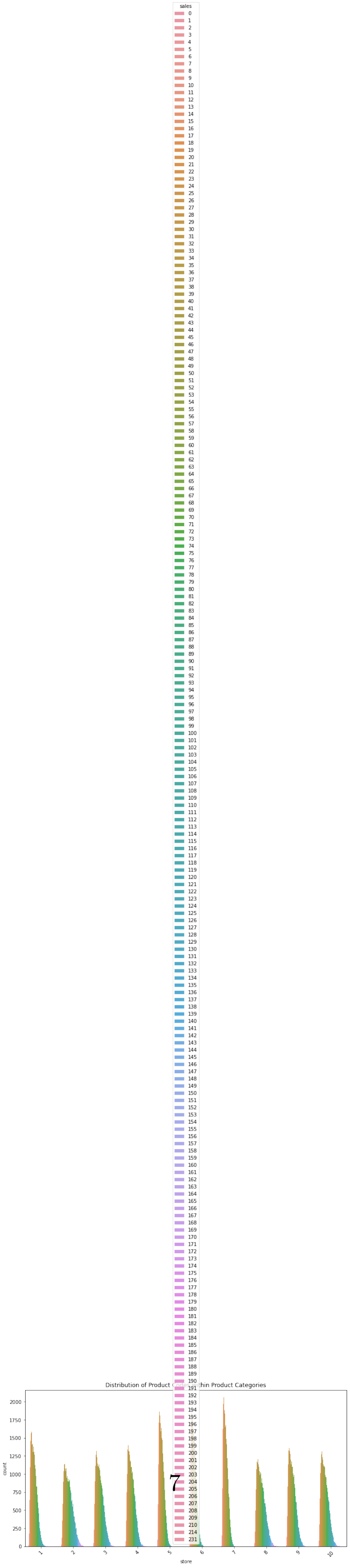
[19]:
```
plt.figure(figsize=(8, 8))
df['sales'].value_counts().plot.pie()
plt.title('Product Category Distribution')
```

[19]:
```
Text(0.5, 1.0, 'Product Category Distribution')
```



[22]:
```
plt.figure(figsize=(12, 6))
sns.countplot(data=df, x='store', hue='sales')
plt.title('Distribution of Product Codes within Product Categories')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```

```
C:\Users\Dell\AppData\Local\Temp\ipykernel_4296\2646529376.py:5: UserWarning:
Tight layout not applied. The bottom and top margins cannot be made large enough
to accommodate all axes decorations.
  plt.tight_layout()
```


Distribution of Product Codes within Product Categories
sales
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
214
231
count
2000
1750
1500
1250
1000
750
500
250
0
1
2
3
4
5
6
7
8
9
10
store

[ ]:

```
```