Objective

Explore the relationship between trader performance and market sentiment, uncover hidden patterns, and deliver insights that can drive smarter trading strategies.

Importing (library ,dataset)

- Importing historical_data
- Importing fear_greed_index

In [1]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import OneHotEncoder, StandardScaler
import seaborn as sns
```

In [2]:

```
data1=pd.read_csv("fear_greed_index.csv")
```

In [3]:

```
data2=pd.read_csv("historical_dataset.csv")
```

Exploring and merging dataset

In [4]:

```
data1.dtypes
```

Out[4]:

```
timestamp          int64
value              int64
classification    object
date              object
dtype: object
```

In [5]:

```
data1.head()
```

Out[5]:

	timestamp	value	classification	date
0	1517463000	30	Fear	01-02-2018
1	1517549400	15	Extreme Fear	02-02-2018
2	1517635800	40	Fear	03-02-2018
3	1517722200	24	Extreme Fear	04-02-2018
4	1517808600	11	Extreme Fear	05-02-2018

In [6]:

```
data2.head()
```

Out[6]:

	Account	Coin	Execution Price	Size Tokens	Size USD	Side	Timestamp IST
0	0xae5eacaf9c6b9111fd53034a602c192a04e082ed	@107	7.9769	986.87	7872.16	BUY	02-12-2024 22:50
1	0xae5eacaf9c6b9111fd53034a602c192a04e082ed	@107	7.9800	16.00	127.68	BUY	02-12-2024 22:50
2	0xae5eacaf9c6b9111fd53034a602c192a04e082ed	@107	7.9855	144.09	1150.63	BUY	02-12-2024 22:50
3	0xae5eacaf9c6b9111fd53034a602c192a04e082ed	@107	7.9874	142.98	1142.04	BUY	02-12-2024 22:50
4	0xae5eacaf9c6b9111fd53034a602c192a04e082ed	@107	7.9894	8.73	69.75	BUY	02-12-2024 22:50

In [7]:

```
data2.dtypes
```

Out[7]:

```
Account               object
Coin                  object
Execution Price      float64
Size Tokens          float64
Size USD             float64
Side                  object
Timestamp IST         object
OnlyDate              object
Matching data         object
Start Position       float64
Direction             object
Closed PnL           float64
Transaction Hash      object
Order ID               int64
Crossed                 bool
Fee                  float64
Trade ID             float64
Timestamp            float64
dtype: object
```

In [8]:

```
da_shape1=data1.shape
da_shape2=data2.shape

print(da_shape1)
print(da_shape2)
```

```
(2644, 4)
(211224, 18)
```

In [9]:

```
merged_df=pd.merge(data1,data2,left_on='date',right_on='Matching data',how='inner')
```

In [10]:

```
merged_df.shape
```

Out[10]:

```
(211218, 22)
```

In [11]:

```
merged_df.dtypes
```

Out[11]:

```
timestamp             int64
value                 int64
classification       object
date                 object
Account              object
Coin                 object
Execution Price     float64
Size Tokens         float64
Size USD            float64
Side                 object
Timestamp IST        object
OnlyDate             object
Matching data        object
Start Position      float64
Direction            object
Closed PnL          float64
Transaction Hash     object
Order ID              int64
Crossed                bool
Fee                 float64
Trade ID            float64
Timestamp           float64
dtype: object
```

In [12]:

```
merged_df.isnull().sum()
```

Out[12]:

```
timestamp           0
value               0
classification      0
date                0
Account             0
Coin                0
Execution Price     0
Size Tokens         0
Size USD            0
Side                0
Timestamp IST       0
OnlyDate            0
Matching data       0
Start Position      0
Direction           0
Closed PnL          0
Transaction Hash    0
Order ID            0
Crossed             0
Fee                 0
```

```
Trade ID            0
Timestamp           0
dtype: int64
```

In [13]:

```
text_col=merged_df.dtypes[merged_df.dtypes=='object'].index
text_col
```

Out[13]:

```
Index(['classification', 'date', 'Account', 'Coin', 'Side', 'Timestamp IST',
       'OnlyDate', 'Matching data', 'Direction', 'Transaction Hash'],
      dtype='object')
```

In [14]:

```
for i in text_col:
    counts=merged_df[i].value_counts()
    print(counts)
```

```
classification
Fear                61837
Greed               50303
Extreme Greed       39992
Neutral             37686
Extreme Fear        21400
Name: count, dtype: int64
date
25-02-2025    6246
23-04-2025    6159
24-02-2025    5616
12-03-2025    3968
09-04-2025    3967
              ...
08-09-2024       1
04-10-2024       1
11-09-2024       1
05-09-2024       1
08-07-2024       1
Name: count, Length: 479, dtype: int64
Account
0xbee1707d6b44d4d52bfe19e41f8a828645437aab      40184
0xbaaaf6571ab7d571043ff1e313a9609a10637864      21192
0xa0feb3725a9335f49874d7cd8eaad6be45b27416      15605
0x8477e447846c758f5a675856001ea72298fd9cb5      14998
0xb1231a4a2dd02f2276fa3c5e2a2f3436e6bfed23      14726
0x28736f43f1e871e6aa8b1148d38d4994275d72c4      13311
0x513b8629fe877bb581bf244e326a047b249c4ff1      12236
0x75f7eeb85dc639d5e99c78f95393aa9a5f1170d4       9893
0x47add9a56df66b524d5e2c1993a43cde53b6ed85       8519
0x4f93fead39b70a1824f981a54d4e55b278e9f760       7584
0x23e7a7f8d14b550961925fbfdaa92f5d195ba5bd       7280
0xb899e522b5715391ae1d4f137653e7906c5e2115       4838
0x8170715b3b381dffb7062c0298972d4727a0a63b       4601
0x4acb90e786d897ecffb614dc822eb231b4ffb9f4       4356
0x083384f897ee0f19899168e3b1bec365f52a9012       3818
0x271b280974205ca63b716753467d5a371de622ab       3809
0x39cef799f8b69da1995852eea189df24eb5cae3c       3589
0x2c229d22b100a7beb69122eed721cee9b24011dd       3239
0x92f17e8d81a944691c10e753af1b1baae1a2cd0d       3052
0xbd5fead7180a9c139fa51a103cb6a2ce86ddb5c3       2641
0x8381e6d82f1affd39a336e143e081ef7620a3b7f       1911
```

```
0x72743ae2822edd658c0c50608fd7c5c501b2afbd       1590
0x7f4f299f74eec87806a830e3caa9afa5f2b9db8f       1559
0x72c6a4624e1dffa724e6d00d64ceae698af892a0       1424
0x430f09841d65beb3f27765503d0f850b8bce7713       1237
0x6d6a4b953f202f8df5bed40692e7fd865318264a        975
0x3998f134d6aaa2b6a5f723806d00fd2bbbbce891        815
0xae5eacaf9c6b9111fd53034a602c192a04e082ed        563
0xaf40fdc468c30116bd3307bcbf4a451a7ebf1deb        534
0xa520ded057a32086c40e7dd6ed4eb8efb82c00e0        417
0x420ab45e0bd8863569a5efbb9c05d91f40624641        383
0x3f9a0aadc7f04a7c9d75dc1b5a6ddd6e36486cf6        332
 0xb1231a4a2dd02f2276fa3c5e2a2f3436e6bfed23         7
Name: count, dtype: int64
Coin
HYPE    68005
@107    29992
BTC     26064
ETH     11158
SOL     10691
        ...
@11         1
@48         1
@68         1
@86         1
@135        1
Name: count, Length: 246, dtype: int64
Side
SELL    108528
BUY     102690
Name: count, dtype: int64
Timestamp IST
14-02-2025 00:31    441
25-02-2025 05:23    432
07-03-2025 20:39    376
03-03-2025 12:25    366
24-02-2025 13:36    359
                   ...
20-02-2025 08:50      1
20-02-2025 08:49      1
20-02-2025 08:46      1
20-02-2025 08:41      1
20-02-2025 08:38      1
Name: count, Length: 27974, dtype: int64
OnlyDate
25-02-2025 00:00    6246
23-04-2025 00:00    6159
24-02-2025 00:00    5616
12-03-2025 00:00    3968
09-04-2025 00:00    3967
                    ...
08-09-2024 00:00       1
04-10-2024 00:00       1
11-09-2024 00:00       1
05-09-2024 00:00       1
08-07-2024 00:00       1
Name: count, Length: 479, dtype: int64
Matching data
25-02-2025    6246
23-04-2025    6159
```

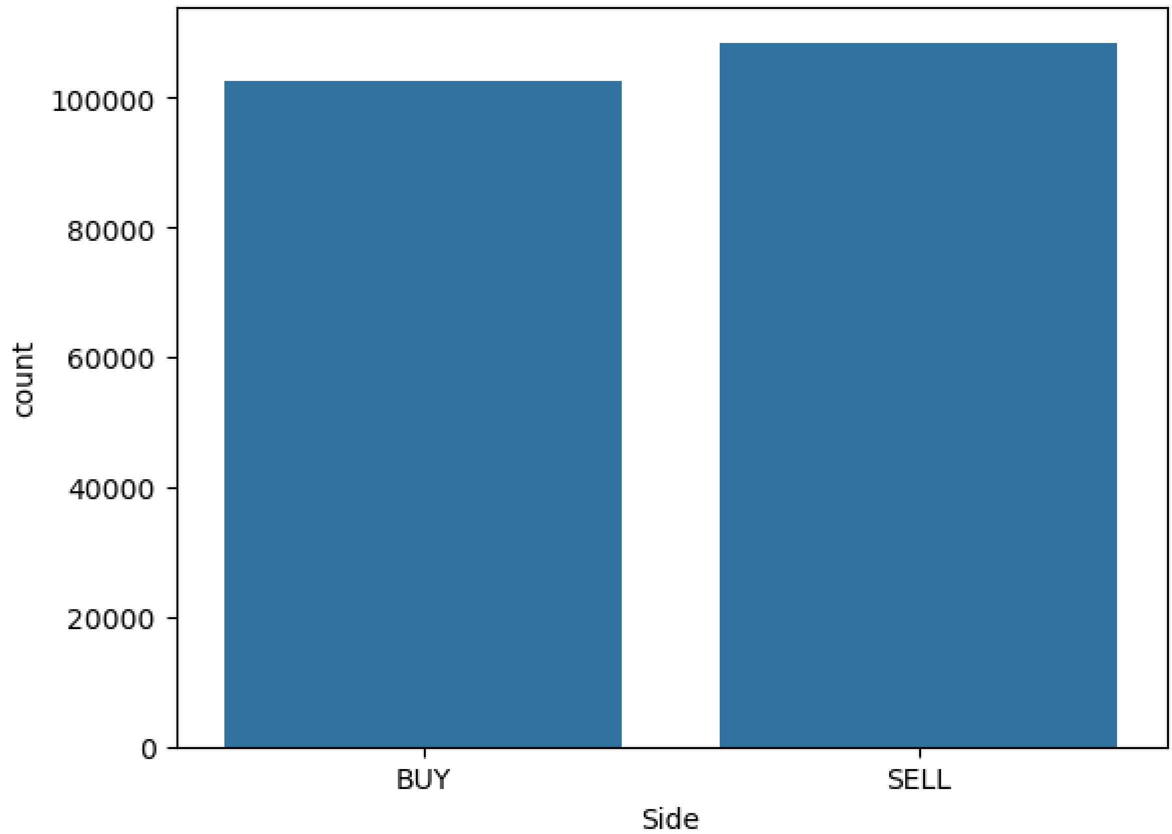
```
24-02-2025    5616
12-03-2025    3968
09-04-2025    3967
              ... 
08-09-2024       1
04-10-2024       1
11-09-2024       1
05-09-2024       1
08-07-2024       1
Name: count, Length: 479, dtype: int64
Direction
Open Long                    49895
Close Long                   48678
Open Short                   39741
Close Short                  36007
Sell                         19902
Buy                          16716
Spot Dust Conversion           142
Short > Long                    70
Long > Short                    57
Auto-Deleveraging                8
Settlement                       1
Liquidated Isolated Short        1
Name: count, dtype: int64
Transaction Hash
0x0000000000000000000000000000000000000000000000000000000000000000    9032
0x8543ffeb4fdab50bfa0304222f29b702012700b5ee6b85c21ed9215e3bc8ba45     298
0xb3a9e6e4f5293d501bb3041f5c49ec02018b00ecdfb8798414bcd103222f2c30     247
0xa67acabfc24d0cb5d05b041f1704590204f000525423d8cded7ec70825f7b658     240
0x6b313e029807198fba0f04215de94801ac001092edd85e7c214059795735b0c6     223
                                                                      ... 
0x88df8d5e724ec55d6f6c04229153250201c9002a7384cfad9147f071a651600c       1
0xd578ed6cae24e29907e40417235b68018900e6e9d87320a79b84dc8c5af41036       1
0xa9f8d39b874e99359b5c0422908bc802017e004a3d83956940fed4c2010c30da       1
0xf6213ee8e9d4b2b5dc300417235f1501680039666f82a9baa0da457d12647df5       1
0xa11f8a8e570005e41d0e042198a4910206e1006c2426d119448ef5cb64e963d7       1
Name: count, Length: 101181, dtype: int64
```

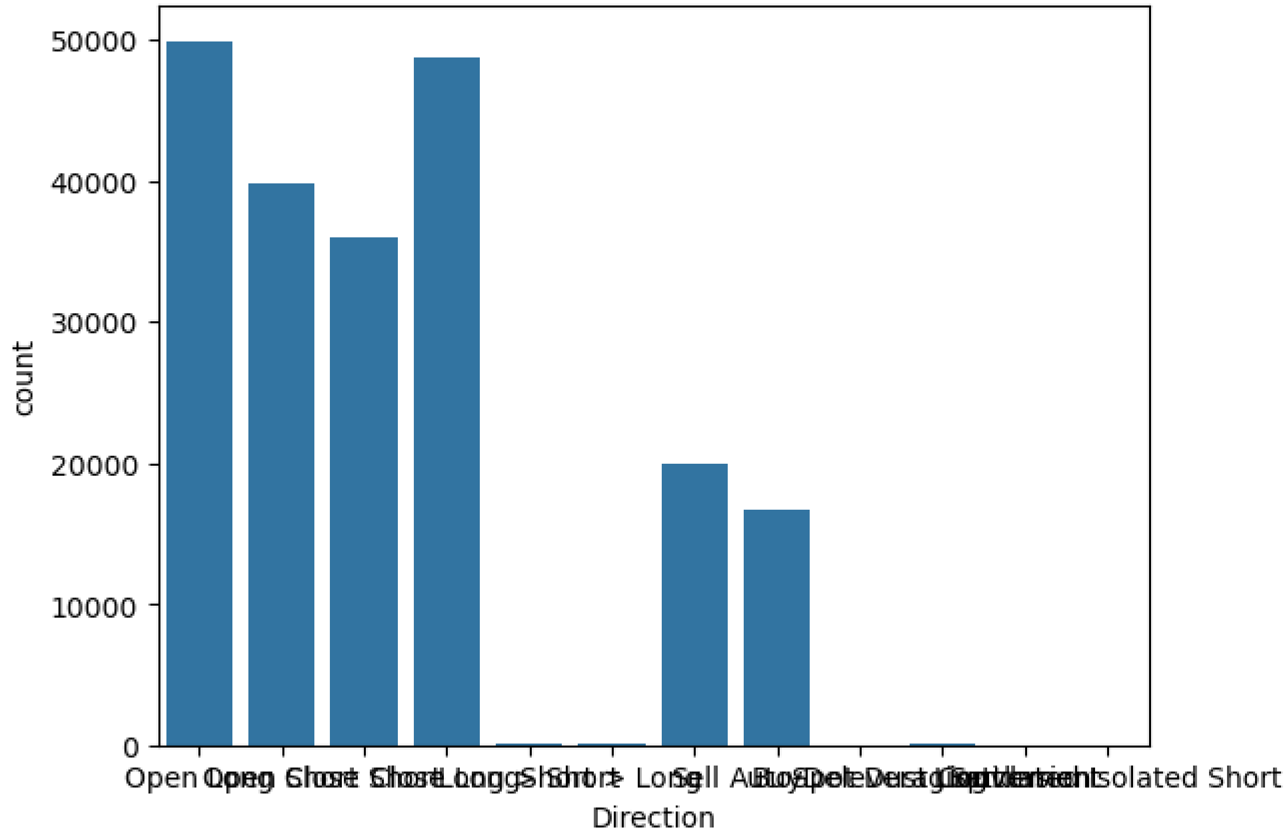
In [15]:

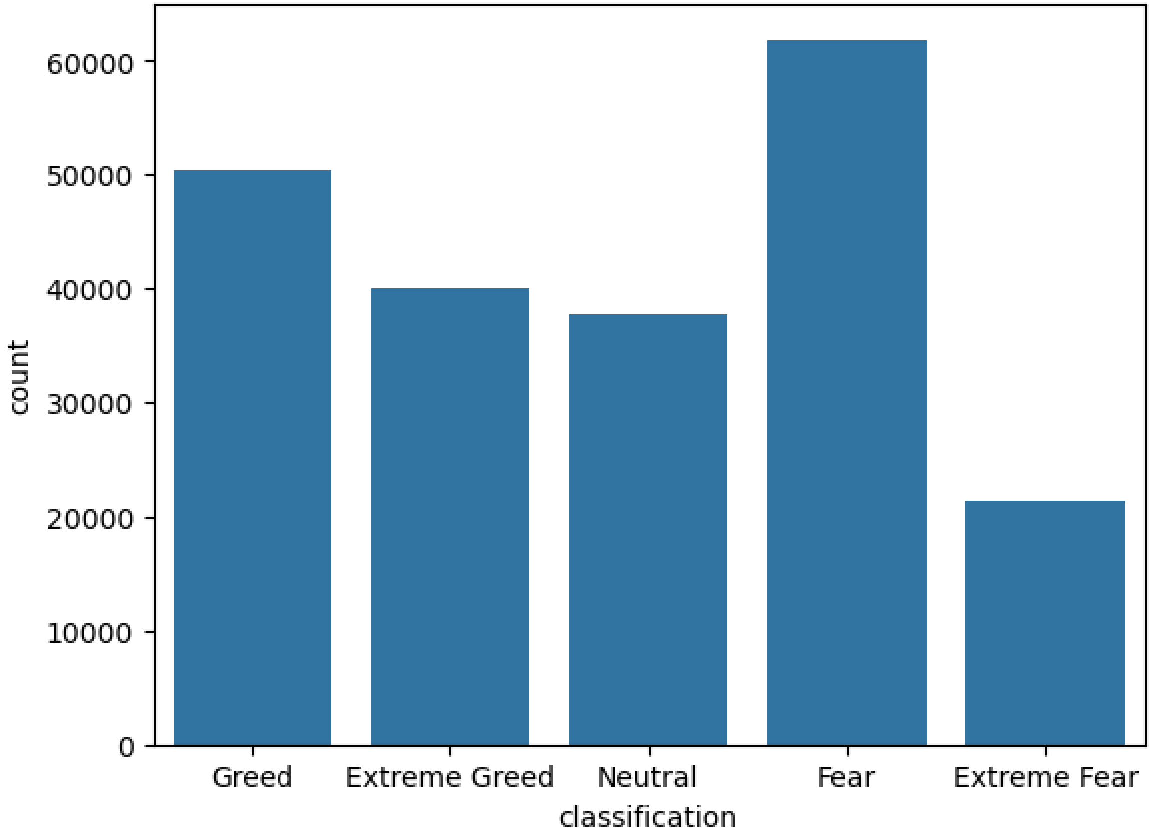
```
check_count=['Side', 'Direction','classification','Account','Trade ID']
```

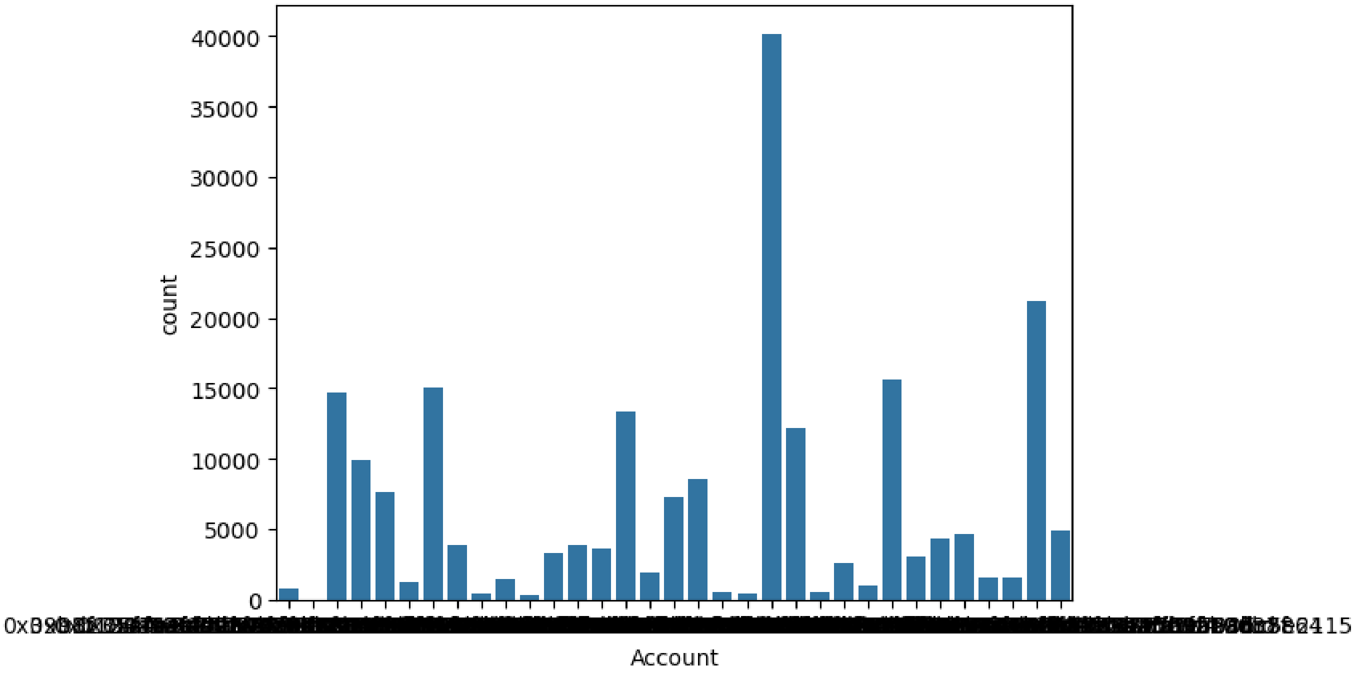
In [16]:

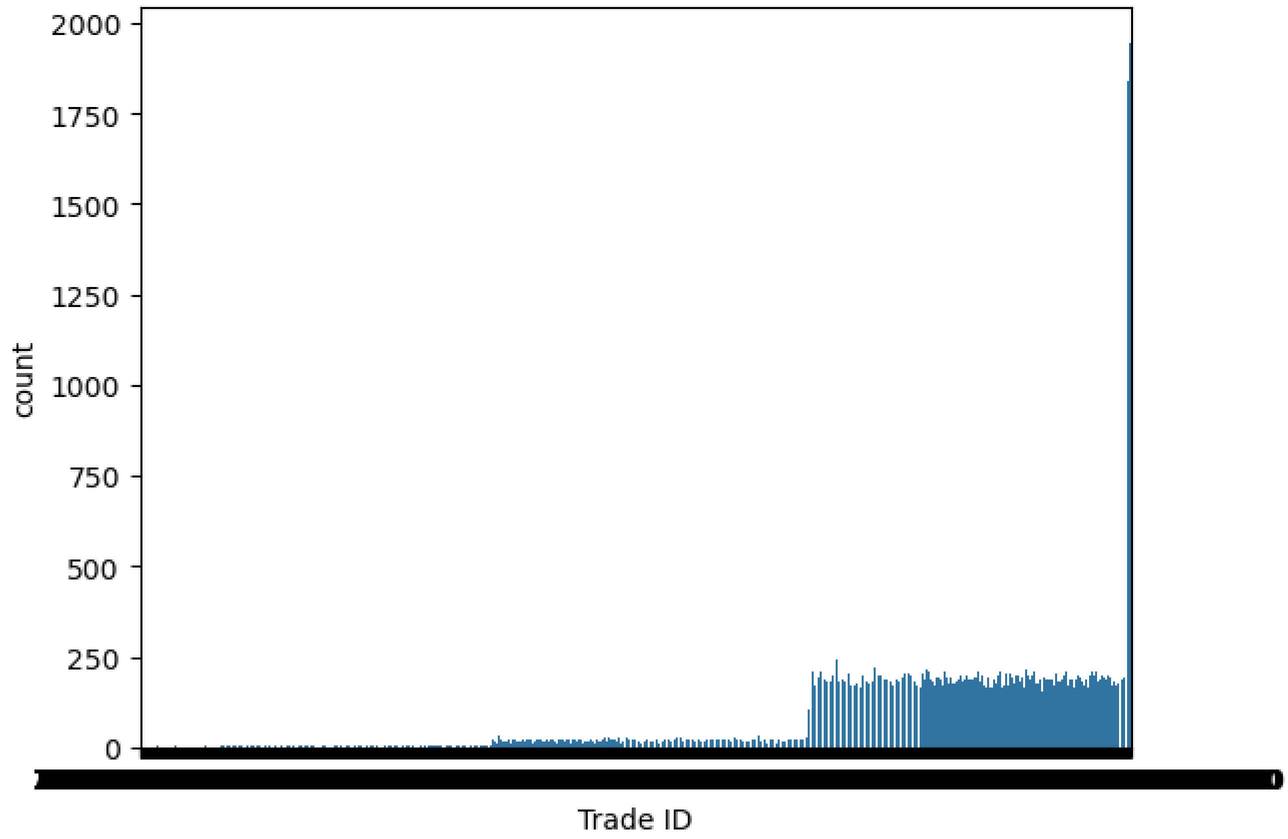
```
for i in check_count:
    sns.countplot(x=merged_df[i])
    plt.yticks(fontsize=10)
    plt.show()
```


count
100000
80000
60000
40000
20000
0
BUY
SELL
Side


count
50000
40000
30000
20000
10000
0
Direction


60000
50000
40000
30000
20000
10000
0
count
Greed
Extreme Greed
Neutral
Fear
Extreme Fear
classification

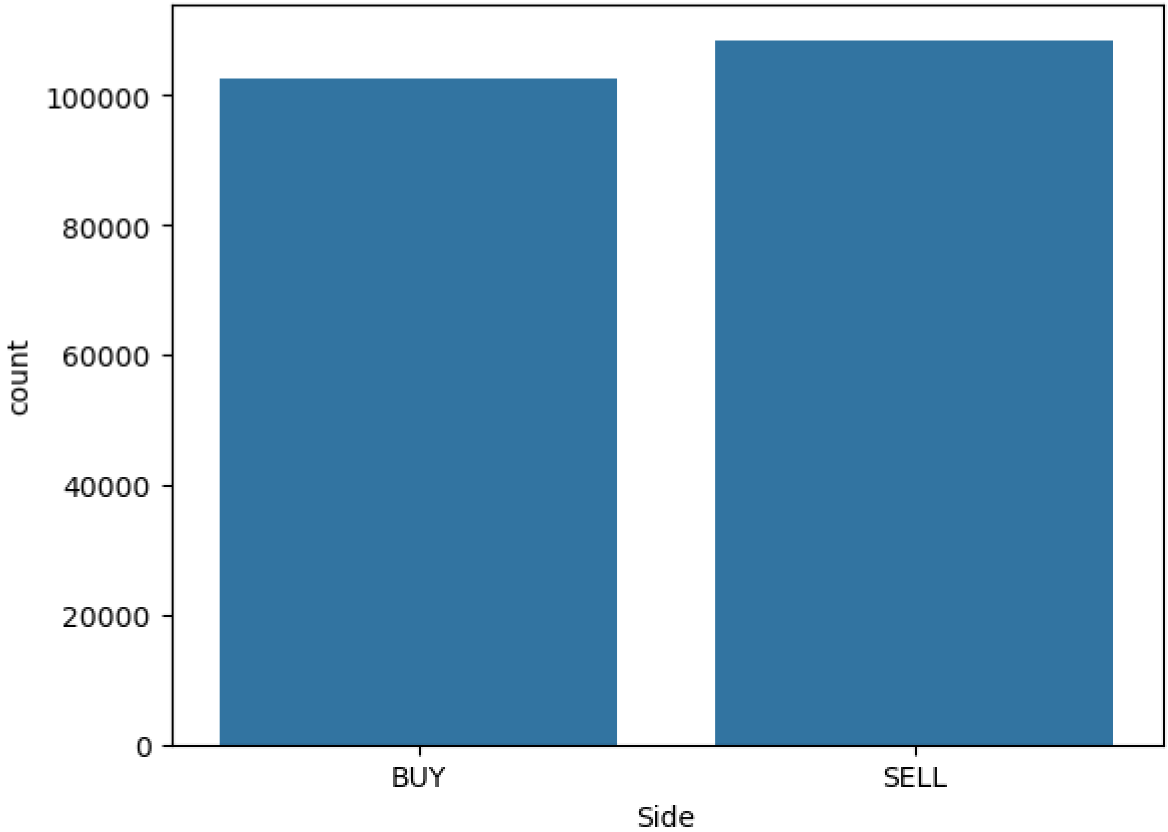

40000
35000
30000
25000
20000
15000
10000
5000
0
count
Account



In [ ]:

In [17]:

```
sns.countplot(x=merged_df["Side"])
plt.yticks(fontsize=10)
plt.show()
```

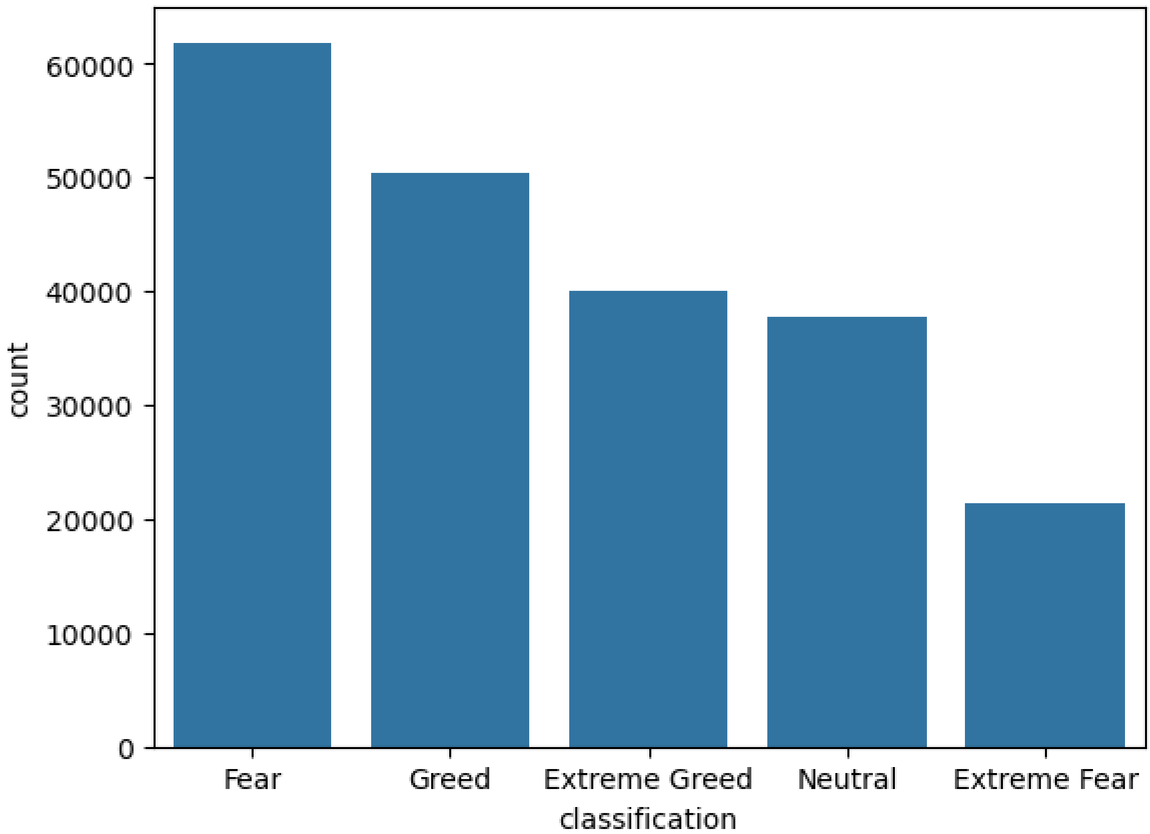


In [18]:

```
cla_order=merged_df['classification'].value_counts().index
```

In [19]:

```
sns.countplot(x=merged_df["classification"],order=cla_order)
plt.yticks(fontsize=10)
plt.show()
```



In [20]:

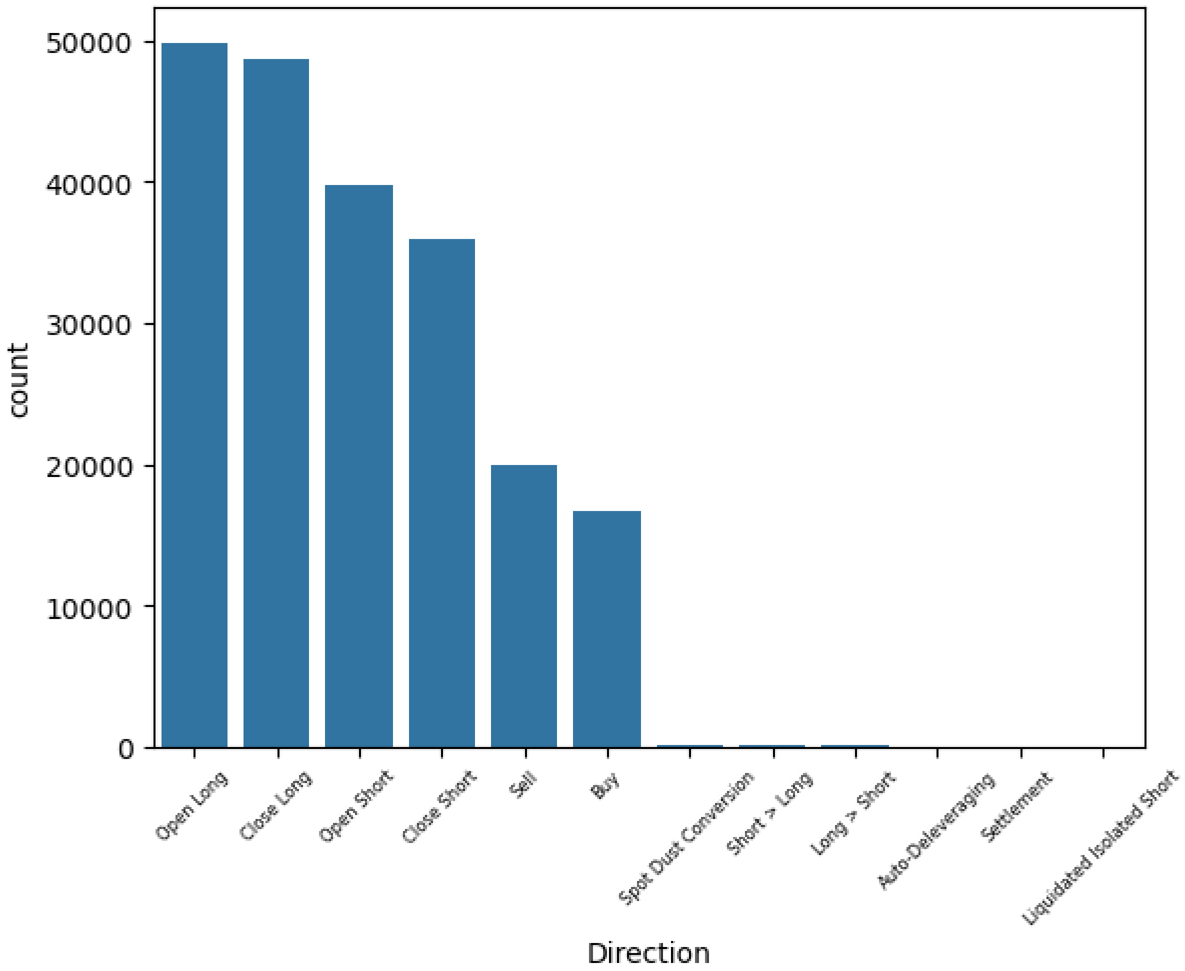
```
direction_order = merged_df['Direction'].value_counts().index
direction_order
```

Out[20]:

```
Index(['Open Long', 'Close Long', 'Open Short', 'Close Short', 'Sell', 'Buy',
       'Spot Dust Conversion', 'Short > Long', 'Long > Short',
       'Auto-Deleveraging', 'Settlement', 'Liquidated Isolated Short'],
      dtype='object', name='Direction')
```

In [21]:

```
sns.countplot(x=merged_df['Direction'],order=direction_order)
plt.yticks(fontsize=10)
plt.xticks(fontsize=6,rotation=45)
plt.show()
```

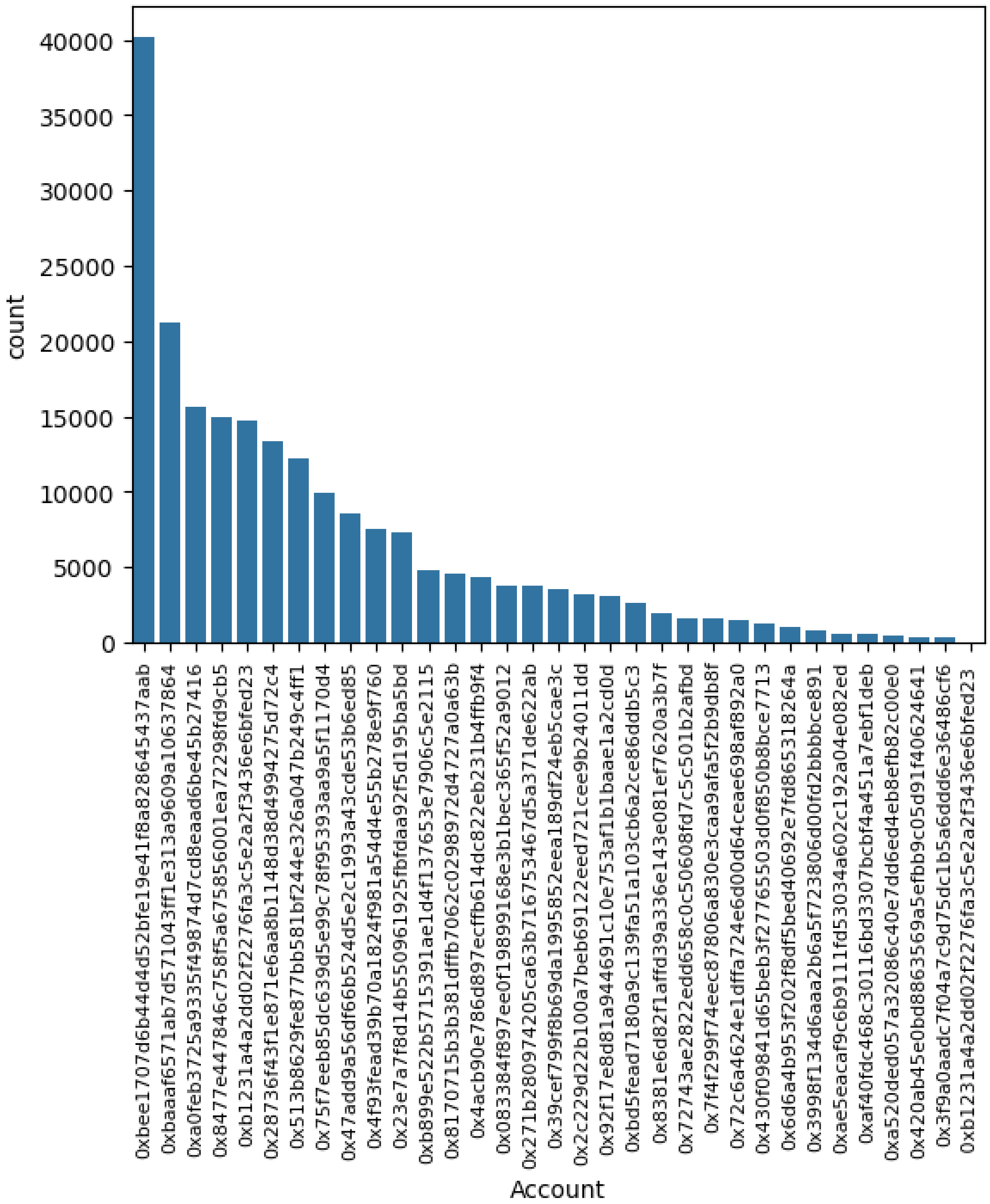


In [22]:

```
acc_order=merged_df['Account'].value_counts().index
```

In [23]:

```
sns.countplot(x=merged_df['Account'],order=acc_order)
plt.xticks(fontsize=8,rotation=90)
plt.show()
```



- There are more than 40k records of same Trader account

Sentiment Analysis with Trader Performance data

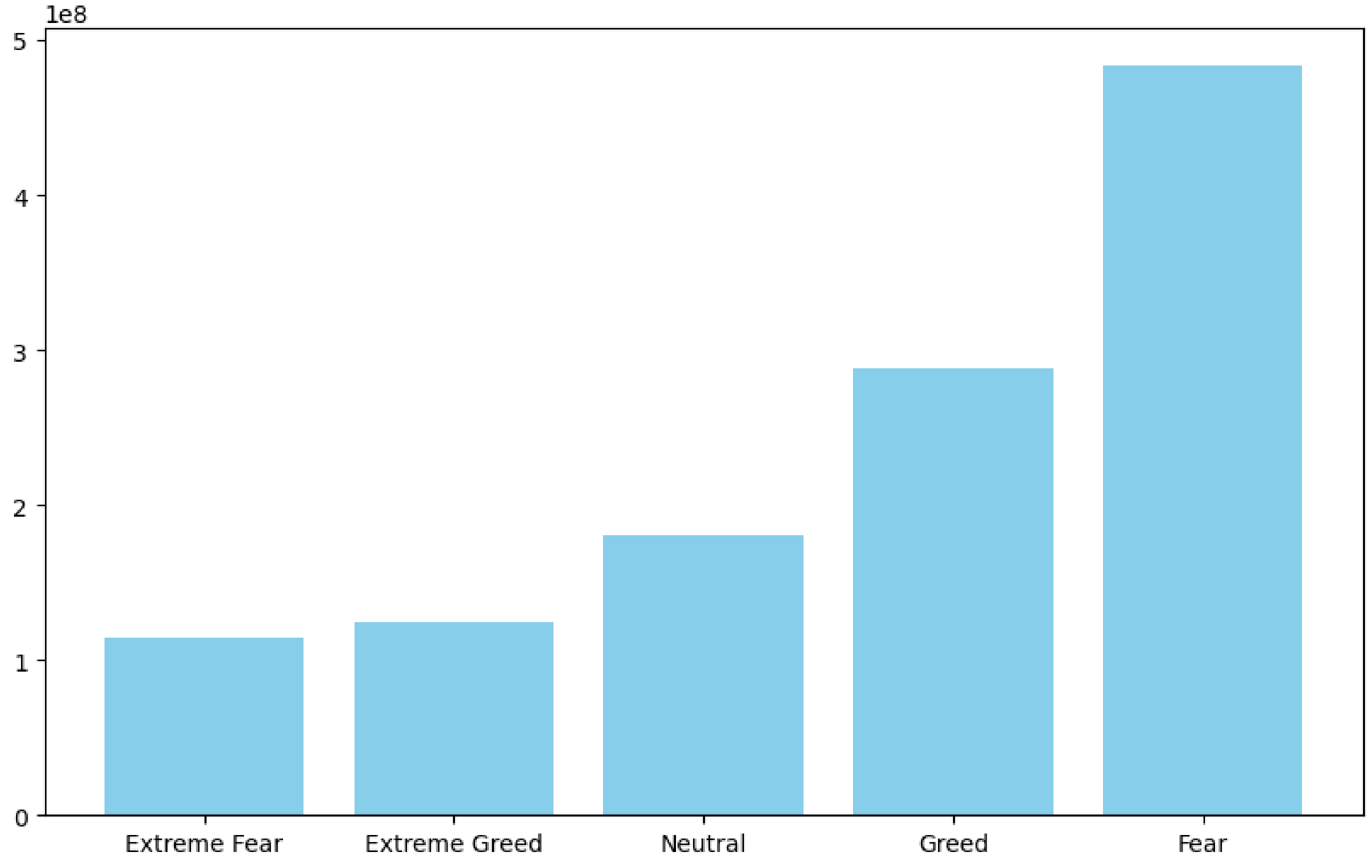
Total traded amount for each classification

In [24]:

```
group=merged_df.groupby("classification")["Size USD"].sum().reset_index().sort_values(by
plt.figure(figsize=(10,6))
plt.bar(group["classification"],group["Size USD"],color='skyblue')
group
```

Out[24]:

	classification	Size USD
0	Extreme Fear	1.144843e+08
1	Extreme Greed	1.244652e+08
4	Neutral	1.802421e+08
3	Greed	2.885825e+08
2	Fear	4.833248e+08



- from this chart we get traded buy or sell coin more on the day pf fear and greed
- Still it is incomplete parameter as we don't know how much person sales and buy coin

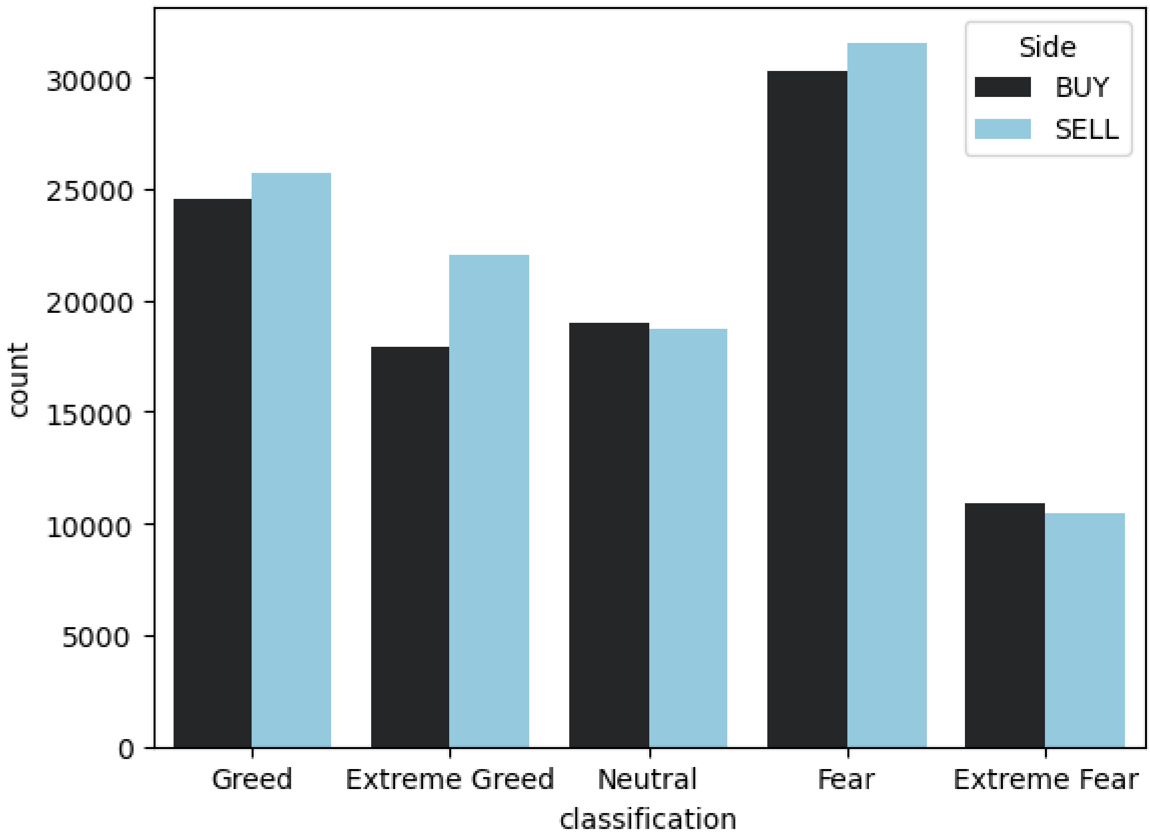
In [25]:

```
sns.countplot(x=merged_df["classification"],color='skyblue',data=merged_df,hue='Side')
plt.show()
```

```
C:\Users\swapn\AppData\Local\Temp\ipykernel_22420\1132237548.py:1: FutureWarning:

Setting a gradient palette using color= is deprecated and will be removed in v0.14.0. Se
t `palette='dark:skyblue'` for the same effect.

  sns.countplot(x=merged_df["classification"],color='skyblue',data=merged_df,hue='Side')
```



- We can see overall total of fear and greedy records are high i.e. trading operation are higher compaire to other classification category
- So it leads to covey that total sum is less reliable madtrics
- So for better analysis we have to consider average
- Also buy and sell operation are nearly eqlual and there is vary small difference in there count for every classification

Average traded amount for each classification

In [26]:

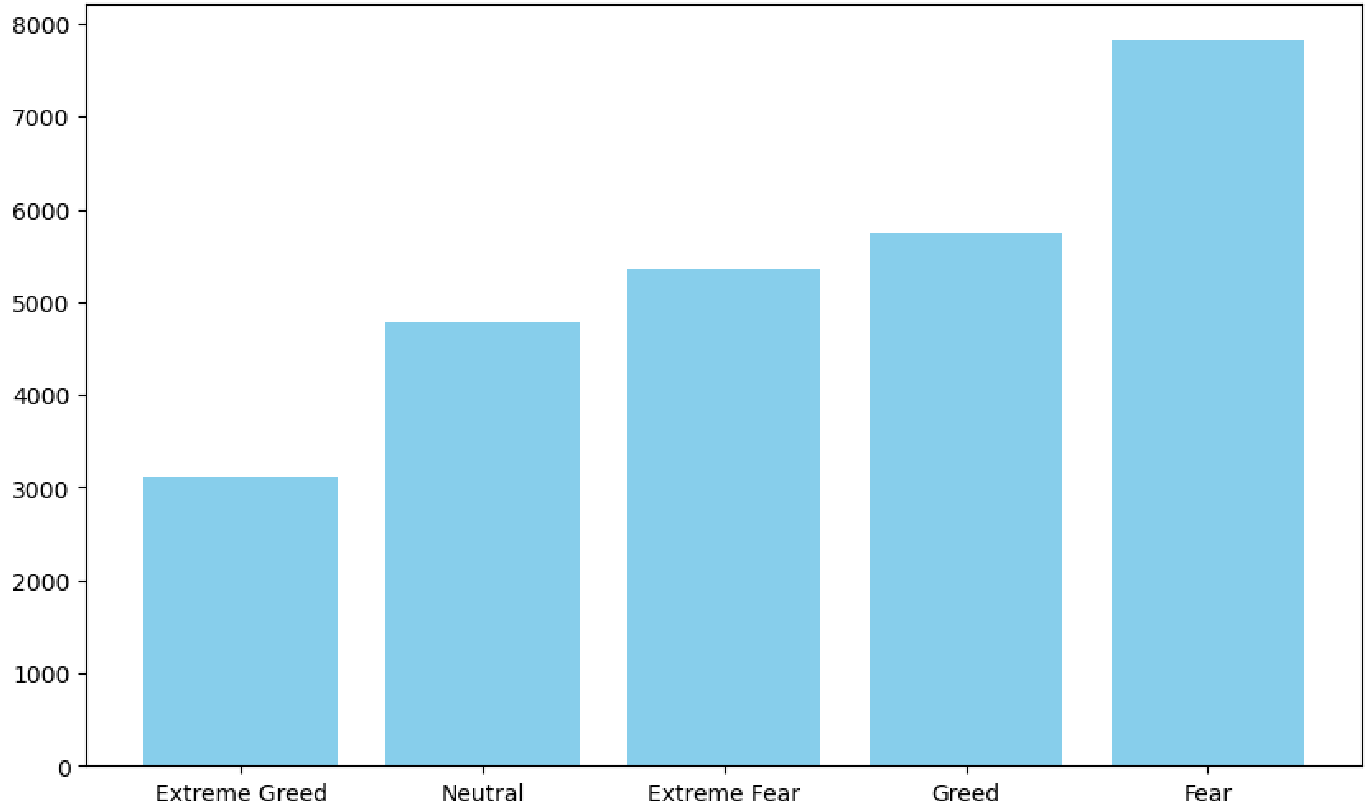
```
group_avg=merged_df.groupby("classification")["Size USD"].mean().reset_index().sort_valu

plt.figure(figsize=(10,6))
plt.bar(group_avg["classification"],group_avg["Size USD"],color='skyblue')
group_avg
```

Out[26]:

	classification	Size USD
1	Extreme Greed	3112.251565
4	Neutral	4782.732661
0	Extreme Fear	5349.731843
3	Greed	5736.884375

	classification	Size USD
2	Fear	7816.109931



- We can conclude from above chart there are more trade operation occur on fear and greed sentiment
- Also there is possibilty that high amount of trade size hapeens on this sentiment

Total Profit & Loss of that trade for each Classification

In [27]:

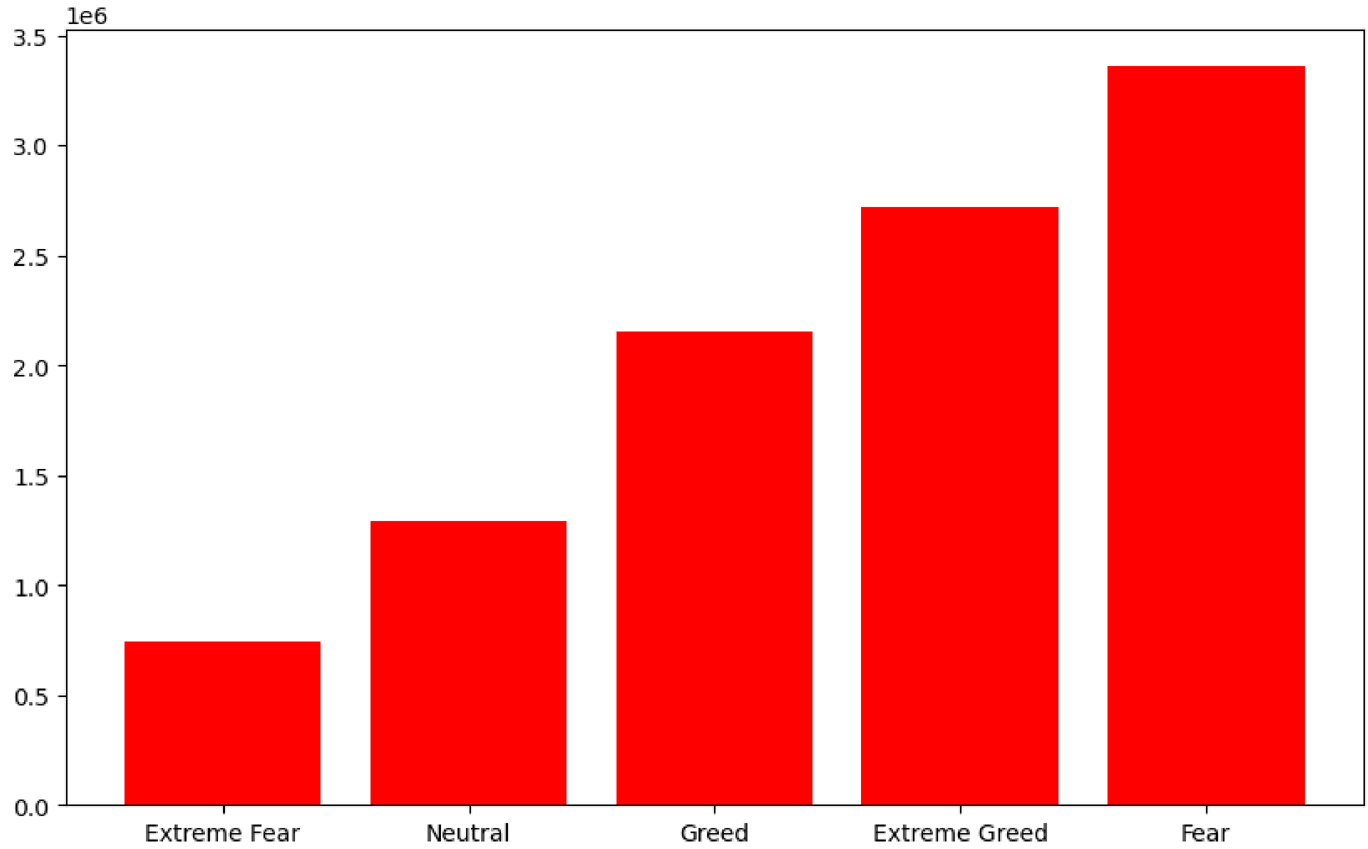
```
group_1=merged_df.groupby("classification")["Closed PnL"].sum().reset_index().sort_value

plt.figure(figsize=(10,6))
plt.bar(group_1["classification"],group_1["Closed PnL"],color='red')

group_1
```

Out[27]:

	classification	Closed PnL
0	Extreme Fear	7.391102e+05
4	Neutral	1.292921e+06
3	Greed	2.150129e+06
1	Extreme Greed	2.715171e+06
2	Fear	3.357155e+06



- Here we get that total closed profit after operation is better for fear and extreme greed
- But sum is not reliable matrics so we will consider average next for better understanding

Average Profit & Loss of that trade for each Classification

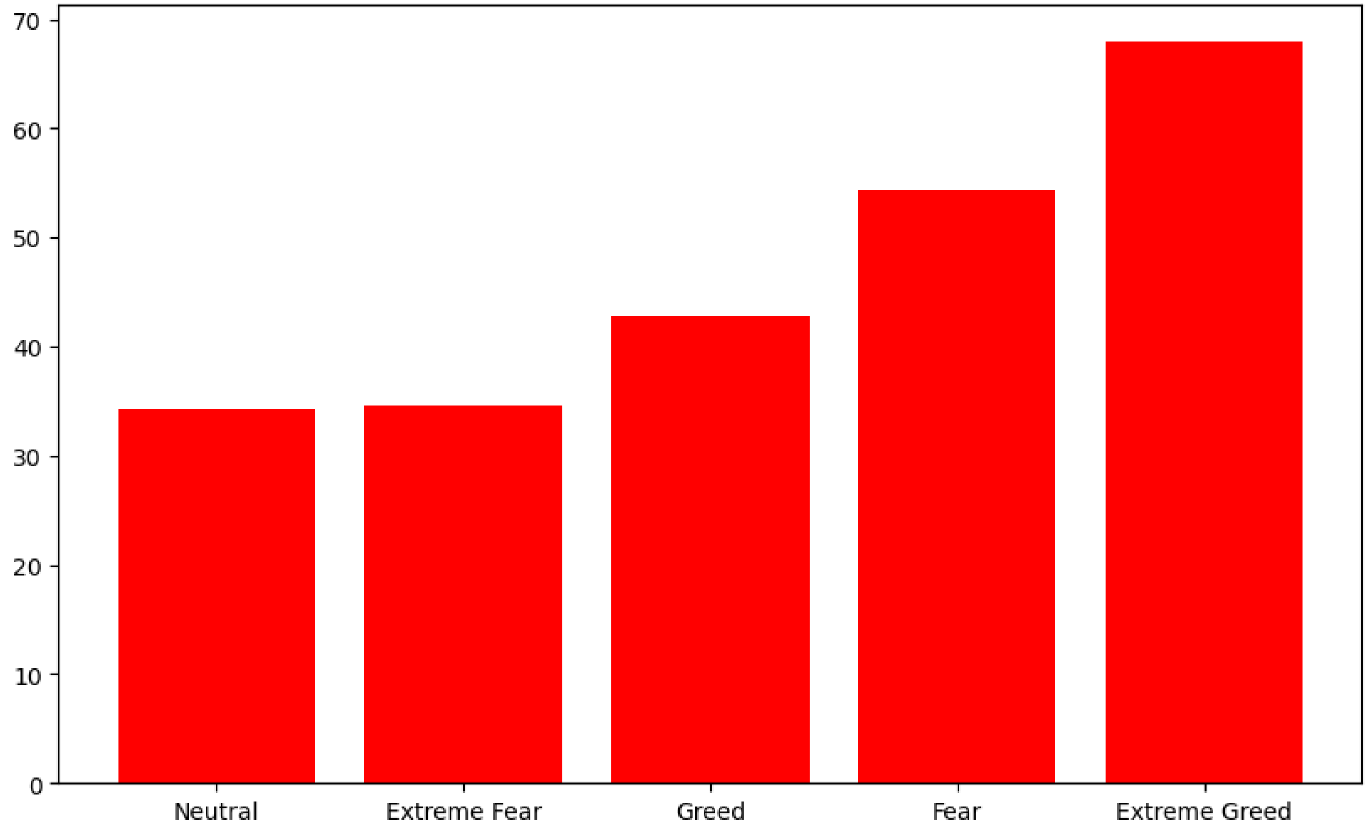
In [39]:

```
group_1avg=merged_df.groupby("classification")["Closed PnL"].mean().reset_index().sort_v

plt.figure(figsize=(10,6))
plt.bar(group_1avg["classification"],group_1avg["Closed PnL"],color='red')
group_1avg
```

Out[39]:

	classification	Closed PnL
4	Neutral	34.307718
0	Extreme Fear	34.537862
3	Greed	42.743559
2	Fear	54.290400
1	Extreme Greed	67.892861



- Average profit for closed PnL is better for Extreme Greed and Fear
- So we can say that marget is getting good profit in etreme greed and fear sentiment days
- Also for neutral days it making worst performance followed by Extreme fear days

By Side(buy/sell) metric

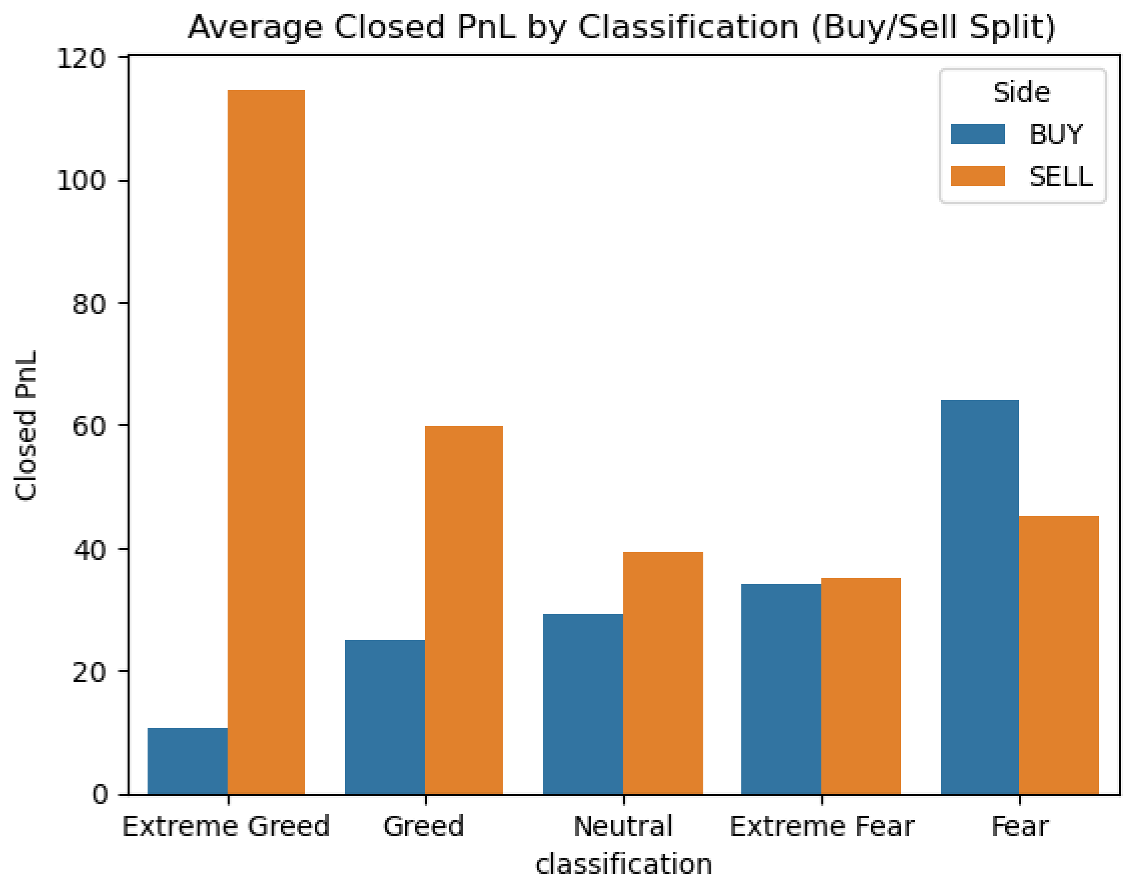
In [57]:

```
group_1avg_side=merged_df.groupby(["classification","Side"])["Closed PnL"].mean().reset_
print(group_1avg_side)

sns.barplot(data=group_1avg_side, x="classification", y="Closed PnL", hue="Side")

plt.title("Average Closed PnL by Classification (Buy/Sell Split)")
plt.show()
```

```
  classification  Side  Closed PnL
2  Extreme Greed   BUY   10.498927
6          Greed   BUY   25.002302
8        Neutral   BUY   29.227429
0   Extreme Fear   BUY   34.114627
1   Extreme Fear  SELL   34.980106
9        Neutral  SELL   39.456408
5           Fear  SELL   45.049641
7          Greed  SELL   59.691091
4           Fear   BUY   63.927104
3  Extreme Greed  SELL  114.584643
```



- average value of sell in greed and extreme greed is high
- which lead to intersting conclusion : **when whole market is in emotion of buying (as market greedy or extremely greedy) those who are selling coins really making profit which can be shown by high bars of greed and extreme greed**

Analysis for positive PnL

In [29]:

```
merged_df["win_PnL"]=merged_df["Closed PnL"]>0
```

In [30]:

```
win_avg=merged_df[merged_df["win_PnL"]==True].groupby("classification")["Closed PnL"].me
win_avg
```

Out[30]:

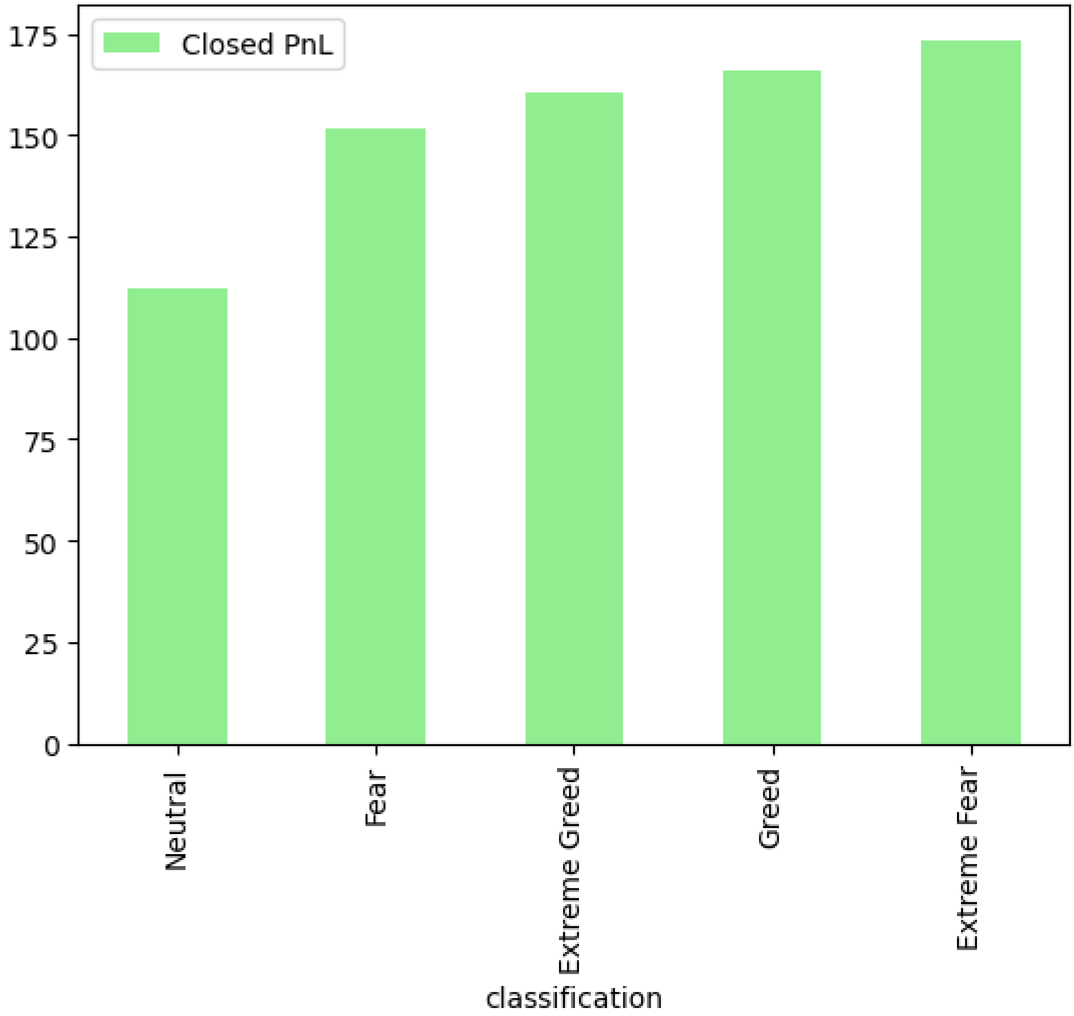
	classification	Closed PnL
4	Neutral	112.439432
2	Fear	151.840935
1	Extreme Greed	160.593269
3	Greed	165.761711
0	Extreme Fear	173.424767

In [31]:

```
win_avg.plot(kind='bar',x="classification",color="lightgreen")
```

Out[31]:

<Axes: xlabel='classification'>



- for positive value of end PnL(i.e. profit),classification of average profit values on each trading operations showing highest for Extreme Fear sentiment followed by Greed
- But we can see highly deflection in average when only considering win values in case of Extreme Fear
- **This lead to conclusion that Extreme Fear days sentiment has high loss compaire to other also Greed sentiment is second loss taker**

By Side(buy/sell) metric

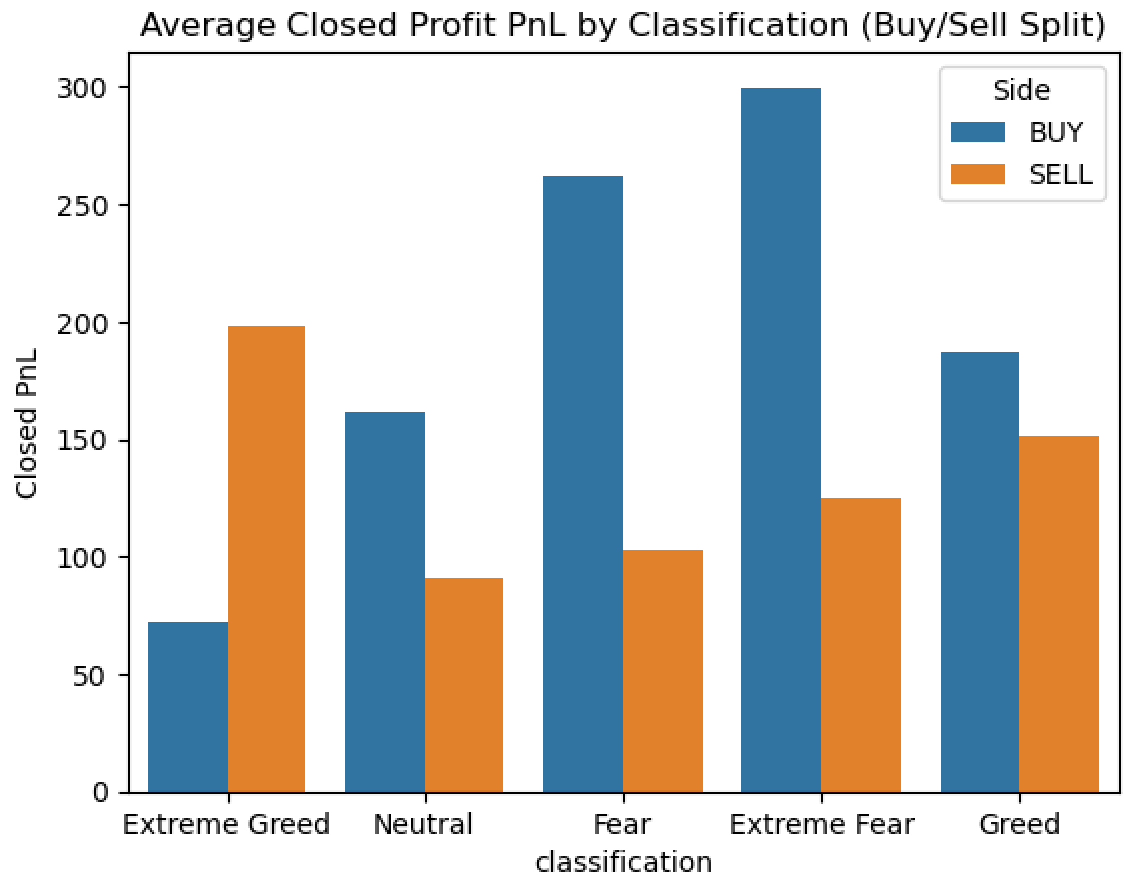
In [58]:

```
win_PnL_side=merged_df[merged_df["win_PnL"]==True].groupby(["classification","Side"])["C
print(win_PnL_side)

sns.barplot(data=win_PnL_side, x="classification", y="Closed PnL", hue="Side")

plt.title("Average Closed Profit PnL by Classification (Buy/Sell Split)")
plt.show()
```

```
  classification  Side   Closed PnL
2  Extreme Greed   BUY    72.408762
9        Neutral  SELL    90.896085
5           Fear  SELL   103.221162
1   Extreme Fear  SELL   124.779958
7          Greed  SELL   151.264036
8        Neutral   BUY   161.686775
6          Greed   BUY   187.166042
3  Extreme Greed  SELL   198.471861
4           Fear   BUY   262.125553
0   Extreme Fear   BUY   299.746845
```



- average value of buy in fear and extreme fear is high and positive
- which lead to intersting conclusion : when whole market is **in emotion of sailing (as market in fear or extremely fear) those who are buying coins really making profit which can be shown by high bars of fear and extreme fear**

Analysis for negative PnL

In [32]:

```
merged_df["loss_PnL"]=merged_df["Closed PnL"]<0
```

In [33]:

```
loss_avg=merged_df[merged_df["loss_PnL"]==True].groupby("classification")["Closed PnL"].
loss_avg
```

Out[33]:

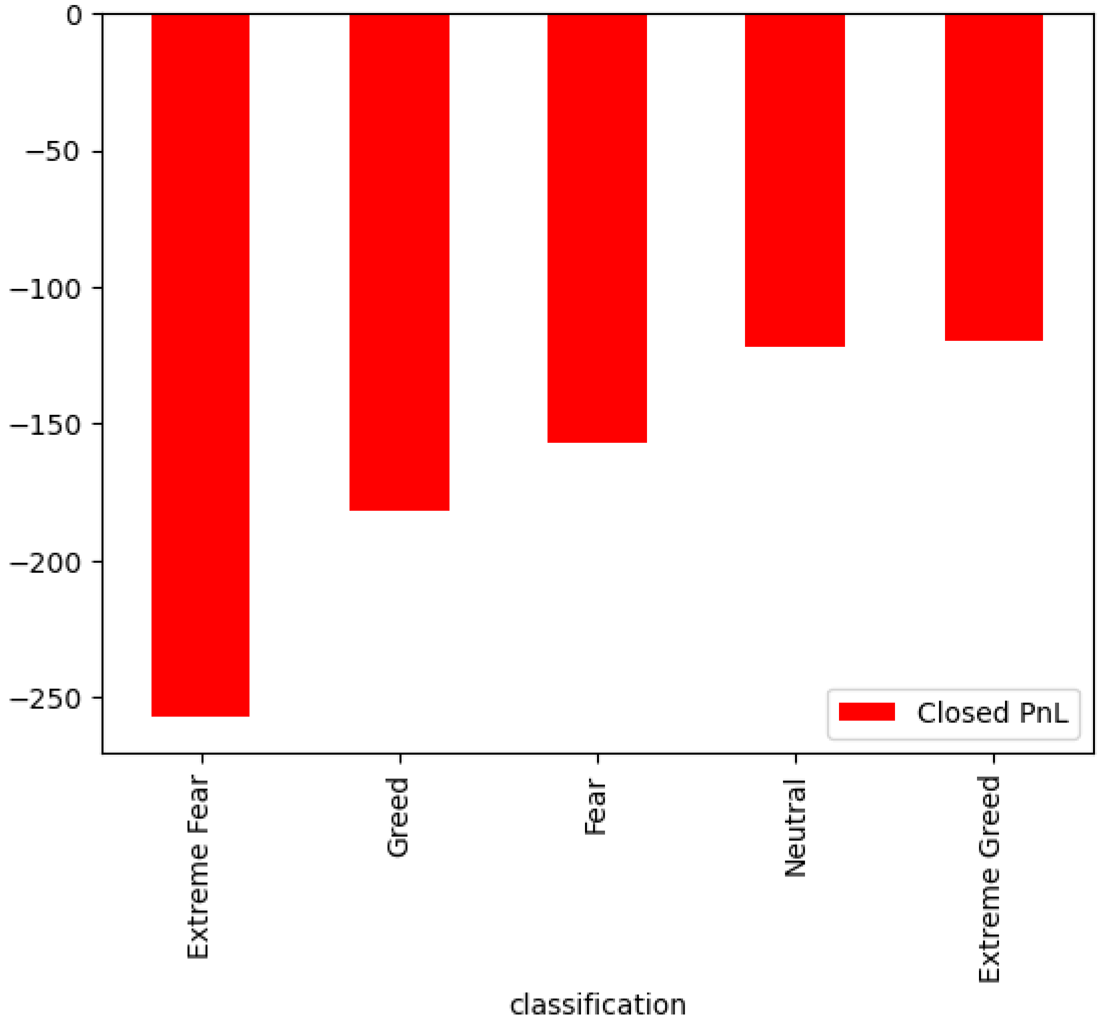
	classification	Closed PnL
0	Extreme Fear	-257.099629
3	Greed	-181.967329
2	Fear	-156.662401
4	Neutral	-121.727849
1	Extreme Greed	-119.920289

In [34]:

```
loss_avg.plot(kind='bar',x="classification",color="red")
```

Out[34]:

```
<Axes: xlabel='classification'>
```



- conclusion of Analysis for positive PnL is varified by above chart of highest loss of Extreme Fear and Greed sentiment

By Side(buy/sell) metric

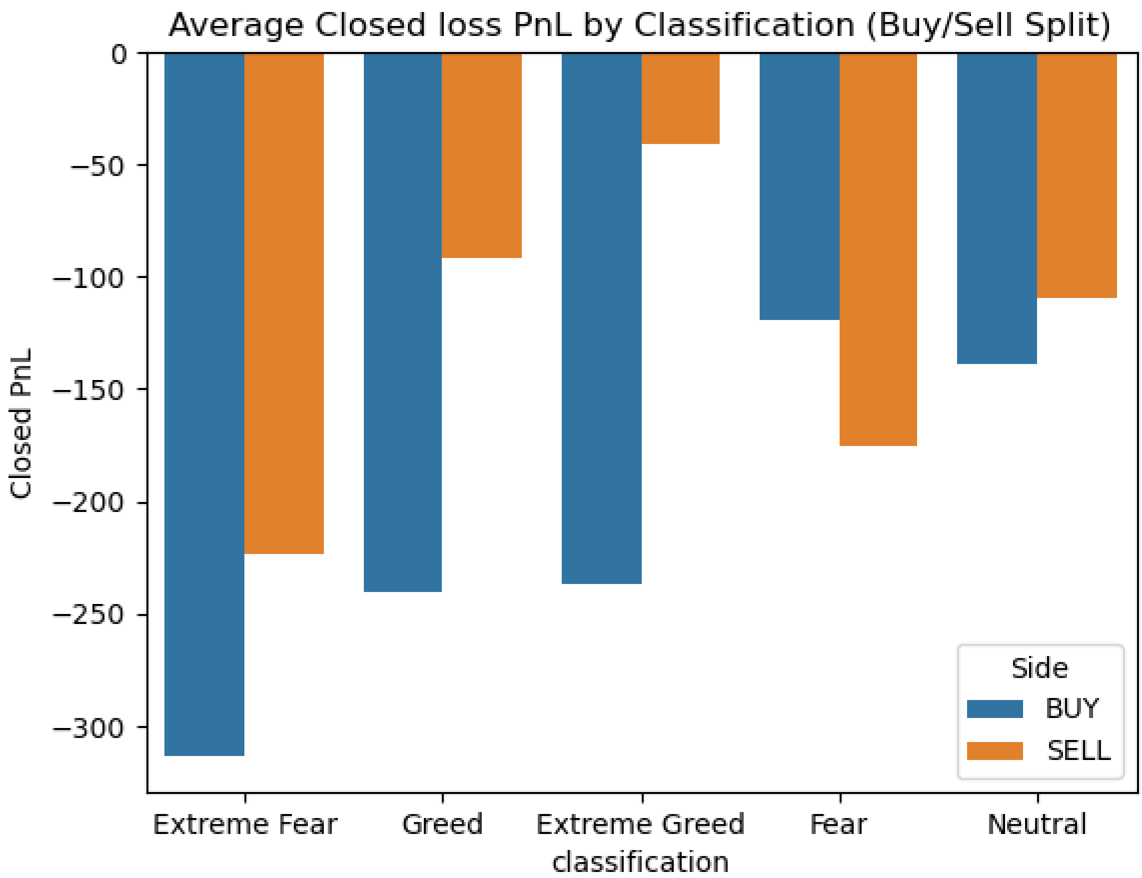
In [59]:

```
loss_PnL_side=merged_df[merged_df["loss_PnL"]==True].groupby(["classification","Side"])[
print(loss_PnL_side)
```

```
sns.barplot(data=loss_PnL_side, x="classification", y="Closed PnL", hue="Side")

plt.title("Average Closed loss PnL by Classification (Buy/Sell Split)")
plt.show()
```

```
  classification  Side  Closed PnL
0   Extreme Fear   BUY -313.614762
6          Greed   BUY -240.675089
2  Extreme Greed   BUY -236.798463
1   Extreme Fear  SELL -223.778568
5           Fear  SELL -175.351487
8        Neutral   BUY -138.948620
4           Fear   BUY -119.549323
9        Neutral  SELL -109.809981
7          Greed  SELL  -91.651638
3  Extreme Greed  SELL  -40.641111
```



- While in extreme fear and greed sentiment which are highest loss taker, the more loss cause by buying coin
- **loss is least in case of extreme greed but in that loss most conceive loss trade side was buy** we can check by difference of bar of buy and sell in extreme greed

Analysis for zero PnL

In [35]:

```
merged_df["zero_PnL"]=merged_df["Closed PnL"]==0
```

In [36]:

```
zero_PnL_count=merged_df[merged_df["zero_PnL"]==True].groupby("classification")["zero_Pn
zero_PnL_count
```

Out[36]:

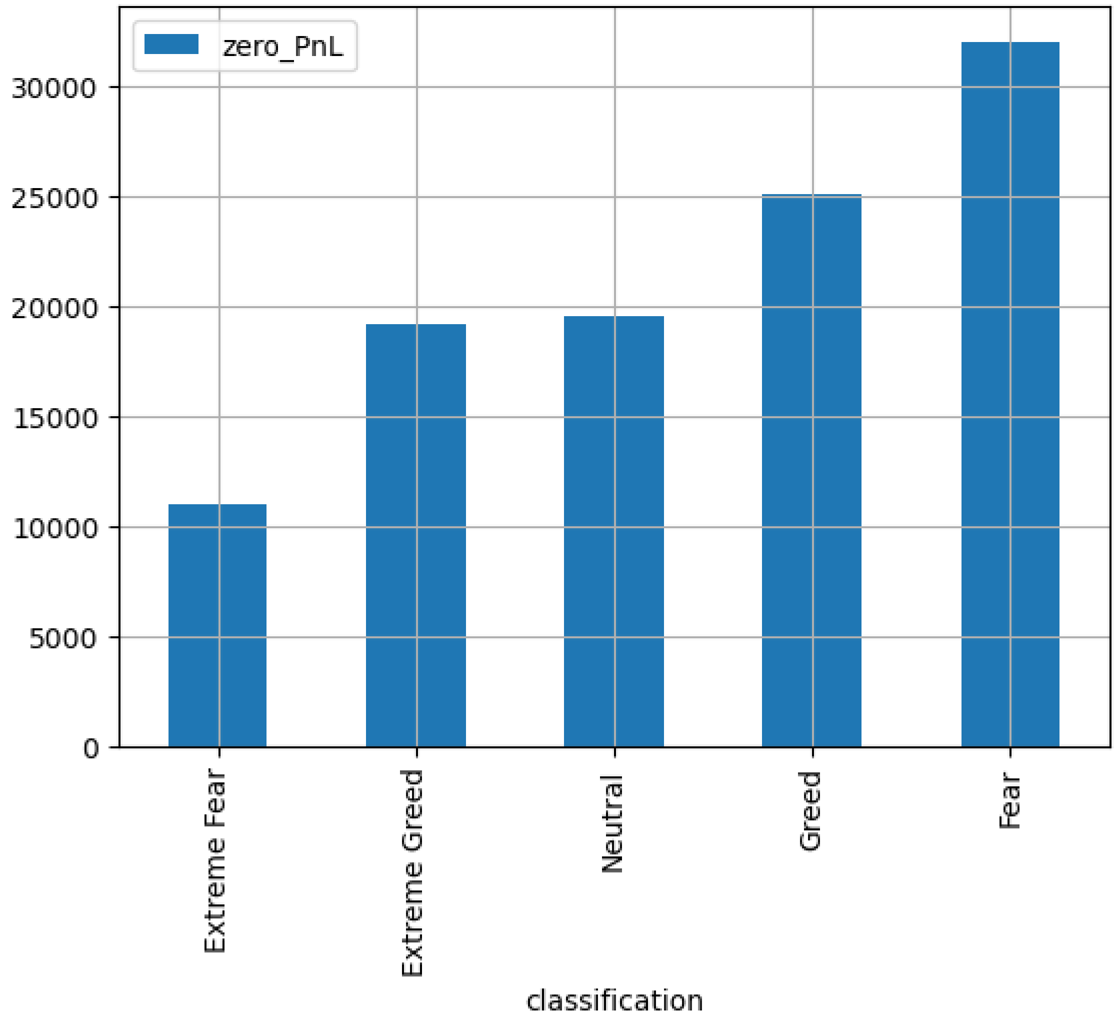
	classification	zero_PnL
0	Extreme Fear	10994
1	Extreme Greed	19139
4	Neutral	19527
3	Greed	25127
2	Fear	32029

In [37]:

```
zero_PnL_count.plot(kind='bar',x="classification",grid="Axis grid lines")
```

Out[37]:

```
<Axes: xlabel='classification'>
```



- For fear and then greed end PnL zero count is highest

By Side(buy/sell) metric

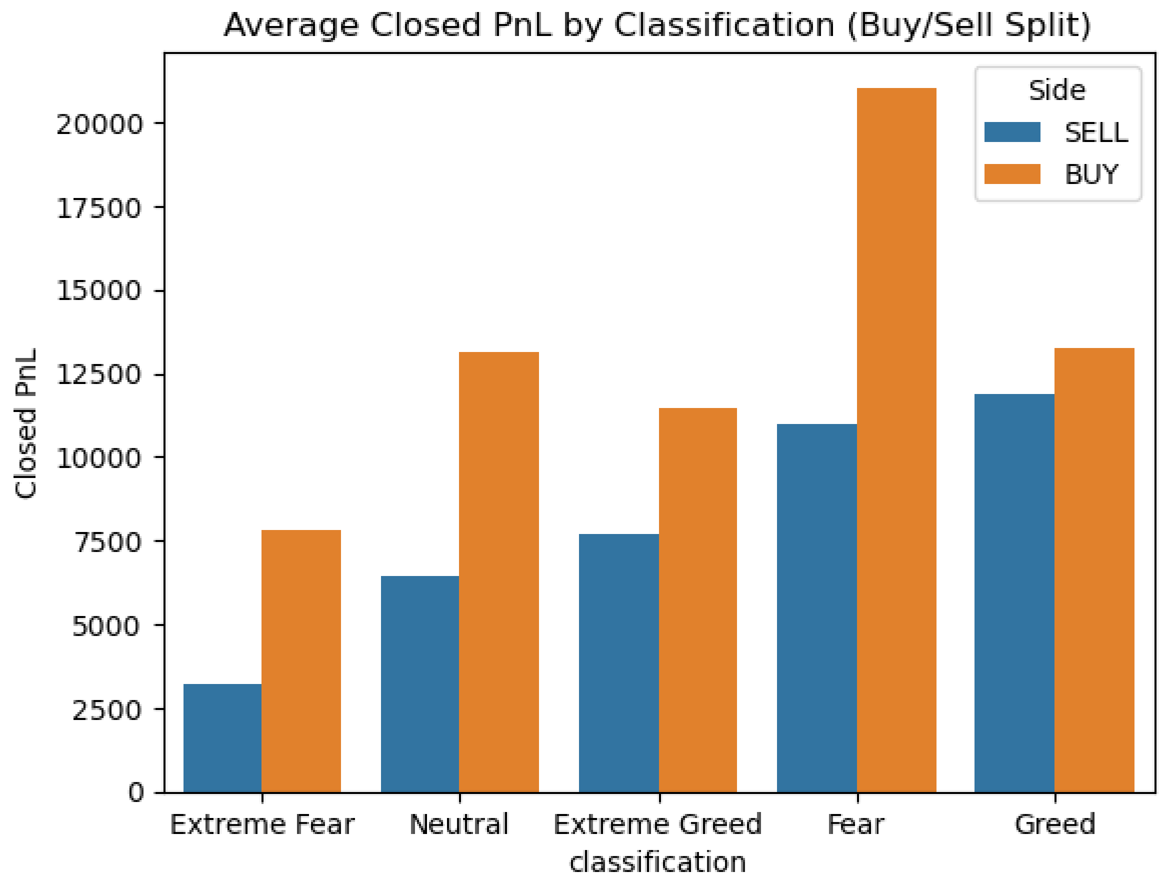
In [60]:

```
zero_PnL_side=merged_df[merged_df["zero_PnL"]==True].groupby(["classification","Side"])[
print(zero_PnL_side)

sns.barplot(data=zero_PnL_side, x="classification", y="Closed PnL", hue="Side")

plt.title("Average Closed PnL by Classification (Buy/Sell Split)")
plt.show()
```

```
  classification  Side  Closed PnL
1   Extreme Fear  SELL        3182
9        Neutral  SELL        6419
3  Extreme Greed  SELL        7699
0   Extreme Fear   BUY        7812
5           Fear  SELL       10989
2  Extreme Greed   BUY       11440
7          Greed  SELL       11894
8        Neutral   BUY       13108
6          Greed   BUY       13233
4           Fear   BUY       21040
```



- We can see more buying for end PnL zero trading operation for every sentiment.
- Also above situation is enhance in case of fear and extreme fear.
- Zero-PnL Buy trades spike during Fear and Extreme Fear, indicating **strong trader hesitation and lack of conviction in bearish sentiment**.

Overall Analysis

- **when whole market is in emotion of buying (as market greedy or extremely greedy) those who are selling coins really making profit**
- **Also VICEVARSA chances of geting end PnL remain positive or high by buying coins in selling sentiment market**
- **Extreme Fear days sentiment has high loss compaire to other also Greed sentiment is second loss taker**
- **Avoid trade operation of buying in extreme greedy sentiment days to avoid PnL loss**
- **There is strong trader hesitation and lack of conviction in bearish sentiment(Fear,Extreme Fear)**