



```
from google.colab import drive
drive.mount('/content/drive')
```

 Mounted at /content/drive

```
import os

folder_path = '/content/drive/MyDrive'
files = os.listdir(folder_path)

print("📁 Files in MyDrive:")
for f in files:
    print(f)
```


 📁 Files in MyDrive:

- Classroom
- Yashika Wedding
- Untitled document.gdoc
- Admit Card
- 54214244897286\_54214254266869.jpg
- Rishali Resume.docx
- assignment\_strategy stimulate (1).gdoc
- Cover\_letter (1).pdf
- Cover\_letter.pdf
- Rishali\_2025 .pdf
- Rishali\_25.pdf
- updated\_file.py
- DS\_ASSIGNMENT
- Colab Notebooks
- Copy of historical\_data.csv

```
import os

folder_path = '/content/drive/MyDrive/DS_ASSIGNMENT'
files = os.listdir(folder_path)

print("📁 Files inside DS_ASSIGNMENT folder:")
for f in files:
    print(f)
```


 📁 Files inside DS\_ASSIGNMENT folder:

- historical\_data.csv
- fear\_greed\_index.csv

```
import os

folder_path = '/content/drive/MyDrive/DS_ASSIGNMENT'
files = os.listdir(folder_path)

print("📁 Files inside DS_ASSIGNMENT folder:")
for f in files:
    print(f)
```

 📁 Files inside DS\_ASSIGNMENT folder:

- historical\_data.csv
- fear\_greed\_index.csv

```
import pandas as pd

# File paths
trader_path = '/content/drive/MyDrive/DS_ASSIGNMENT/historical_data.csv'
sentiment_path = '/content/drive/MyDrive/DS_ASSIGNMENT/fear_greed_index.csv'

# Load CSVs
trader_df = pd.read_csv(trader_path)
sentiment_df = pd.read_csv(sentiment_path)

# Preview data
print("📄 Trader Data:")
print(trader_df.head())
```

```
print("\n📄 Sentiment Data:")
```

```
print("\n Sentiment Data: ")
print(sentiment_df.head())
```

↻

☰

Trader Data:

		Account	Coin	Execution Price	\
0	0xae5eacaf9c6b9111fd53034a602c192a04e082ed	@107		7.9769	
1	0xae5eacaf9c6b9111fd53034a602c192a04e082ed	@107		7.9800	
2	0xae5eacaf9c6b9111fd53034a602c192a04e082ed	@107		7.9855	
3	0xae5eacaf9c6b9111fd53034a602c192a04e082ed	@107		7.9874	
4	0xae5eacaf9c6b9111fd53034a602c192a04e082ed	@107		7.9894	

	Size	Tokens	Size USD	Side	Timestamp IST	Start Position	Direction	\
0	986.87		7872.16	BUY	02-12-2024 22:50	0.000000	Buy	
1	16.00		127.68	BUY	02-12-2024 22:50	986.524596	Buy	
2	144.09		1150.63	BUY	02-12-2024 22:50	1002.518996	Buy	
3	142.98		1142.04	BUY	02-12-2024 22:50	1146.558564	Buy	
4	8.73		69.75	BUY	02-12-2024 22:50	1289.488521	Buy	

	Closed PnL		Transaction Hash	Order ID	\
0	0.0	0xec09451986a1874e3a980418412fcd0201f500c95bac...	52017706630		
1	0.0	0xec09451986a1874e3a980418412fcd0201f500c95bac...	52017706630		
2	0.0	0xec09451986a1874e3a980418412fcd0201f500c95bac...	52017706630		
3	0.0	0xec09451986a1874e3a980418412fcd0201f500c95bac...	52017706630		
4	0.0	0xec09451986a1874e3a980418412fcd0201f500c95bac...	52017706630		

	Crossed	Fee	Trade ID	Timestamp
0	True	0.345404	8.950000e+14	1.730000e+12
1	True	0.005600	4.430000e+14	1.730000e+12
2	True	0.050431	6.600000e+14	1.730000e+12
3	True	0.050043	1.080000e+15	1.730000e+12
4	True	0.003055	1.050000e+15	1.730000e+12

☰

Sentiment Data:

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

```
# Convert sentiment 'date' column to datetime
sentiment_df['date'] = pd.to_datetime(sentiment_df['date']).dt.date

# Convert trader 'Timestamp IST' to date
trader_df['Timestamp IST'] = pd.to_datetime(trader_df['Timestamp IST'], format='%d-%m-%Y %H:%M')
trader_df['date'] = trader_df['Timestamp IST'].dt.date

# Merge both DataFrames on 'date'
merged_df = pd.merge(trader_df, sentiment_df[['date', 'classification']], on='date', how='left')

# Rename for clarity
merged_df.rename(columns={'classification': 'Market Sentiment'}, inplace=True)

# Show merged data
merged_df[['date', 'Side', 'Size USD', 'Closed PnL', 'Market Sentiment']].head()
```

↻

	date	Side	Size USD	Closed PnL	Market Sentiment
0	2024-12-02	BUY	7872.16	0.0	Extreme Greed
1	2024-12-02	BUY	127.68	0.0	Extreme Greed
2	2024-12-02	BUY	1150.63	0.0	Extreme Greed
3	2024-12-02	BUY	1142.04	0.0	Extreme Greed
4	2024-12-02	BUY	69.75	0.0	Extreme Greed

```
merged_df.columns = merged_df.columns.str.strip().str.lower().str.replace(' ', '_')

# Check missing sentiment data
missing_sentiment = merged_df['market_sentiment'].isnull().sum()
print(f"Rows with missing sentiment: {missing_sentiment}")
```

https://colab.research.google.com/drive/1vorrSk\_Mw42L3ZVzpxDW8AskGJMkzZDM#scrollTo=R2PDr7sK2gf\_

2/5

↻ Rows with missing sentiment: 6

```
# Drop rows with missing sentiment
merged_df = merged_df.dropna(subset=['market_sentiment'])
```

```
# Confirm it's clean now
print("✅ Final rows:", len(merged_df))
```

↻ ✅ Final rows: 211218

```
# Create profit flag: 1 if profit, 0 if loss
merged_df['is_profit'] = merged_df['closed_pnl'].apply(lambda x: 1 if x > 0 else 0)
```

```
# Rename for consistency
merged_df.rename(columns={'size_usd': 'trade_value_usd'}, inplace=True)
```

```
# Quick summary by sentiment
summary = merged_df.groupby('market_sentiment')[['is_profit', 'trade_value_usd']].mean()
print("📊 Summary:")
print(summary)
```

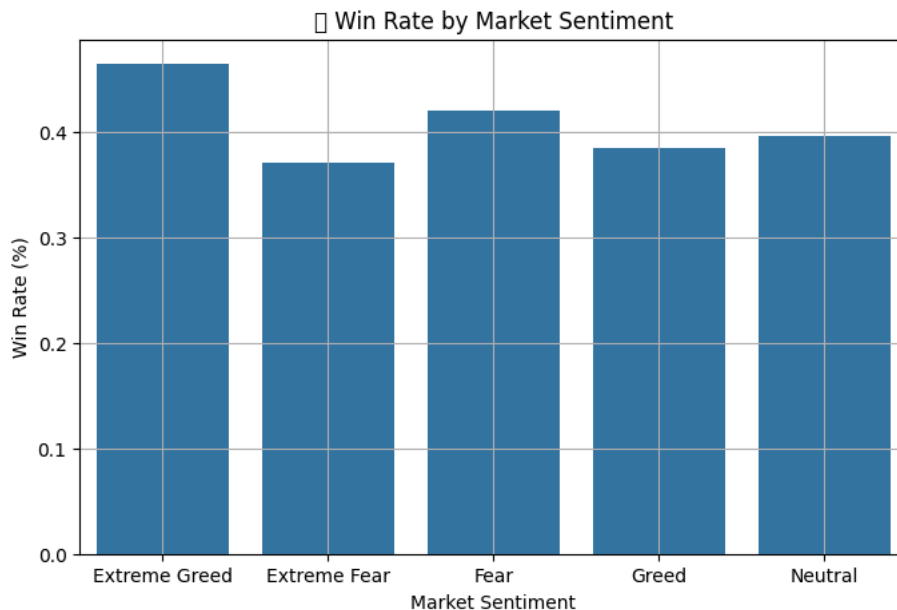
↻ 📊 Summary:

	is_profit	trade_value_usd
market_sentiment		
Extreme Fear	0.370607	5349.731843
Extreme Greed	0.464943	3112.251565
Fear	0.420768	7816.109931
Greed	0.384828	5736.884375
Neutral	0.396991	4782.732661

```
import seaborn as sns
import matplotlib.pyplot as plt
```



```
# Plot: Win rate (is_profit) across sentiment
plt.figure(figsize=(8, 5))
sns.barplot(data=merged_df, x='market_sentiment', y='is_profit', estimator='mean', errorbar=None)
plt.title('📊 Win Rate by Market Sentiment')
plt.ylabel('Win Rate (%)')
plt.xlabel('Market Sentiment')
plt.grid(True)
plt.show()
```

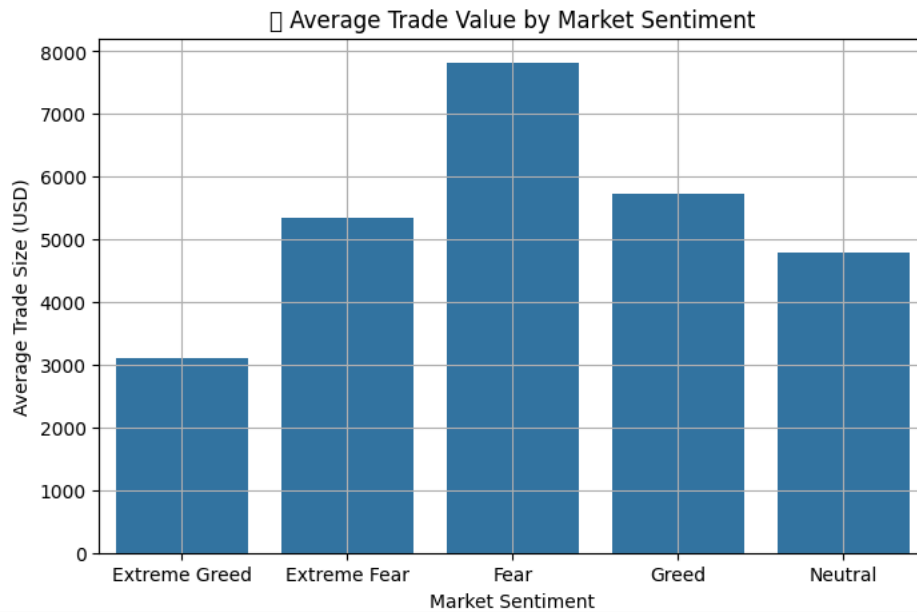
↻ /usr/local/lib/python3.11/dist-packages/IPython/core/pylabtools.py:151: UserWarning: Glyph 128200 (\N{CHART WITH UPWARDS TREND}) missing  
fig.canvas.print\_figure(bytes\_io, \*\*kw)





```
plt.figure(figsize=(8, 5))
```

```
sns.barplot(data=merged_df, x='market_sentiment', y='trade_value_usd', estimator='mean', errorbar=None)
plt.title('👛 Average Trade Value by Market Sentiment')
plt.ylabel('Average Trade Size (USD)')
plt.xlabel('Market Sentiment')
plt.grid(True)
plt.show()
```

 /usr/local/lib/python3.11/dist-packages/IPython/core/pylabtools.py:151: UserWarning: Glyph 128176 (\N{MONEY BAG}) missing from font(s)   
fig.canvas.print\_figure(bytes\_io, \*\*kw)



```
plt.figure(figsize=(10, 5))
sns.boxplot(data=merged_df, x='market_sentiment', y='closed_pnl')
plt.title('▼ Distribution of Closed PnL by Market Sentiment')
plt.xlabel('Market Sentiment')
plt.ylabel('Closed PnL')
plt.grid(True)
plt.show()
```

 /usr/local/lib/python3.11/dist-packages/IPython/core/pylabtools.py:151: UserWarning: Glyph 128315 (\N{DOWN-POINTING RED TRIANGLE}) missing from font(s)   
fig.canvas.print\_figure(bytes\_io, \*\*kw)

