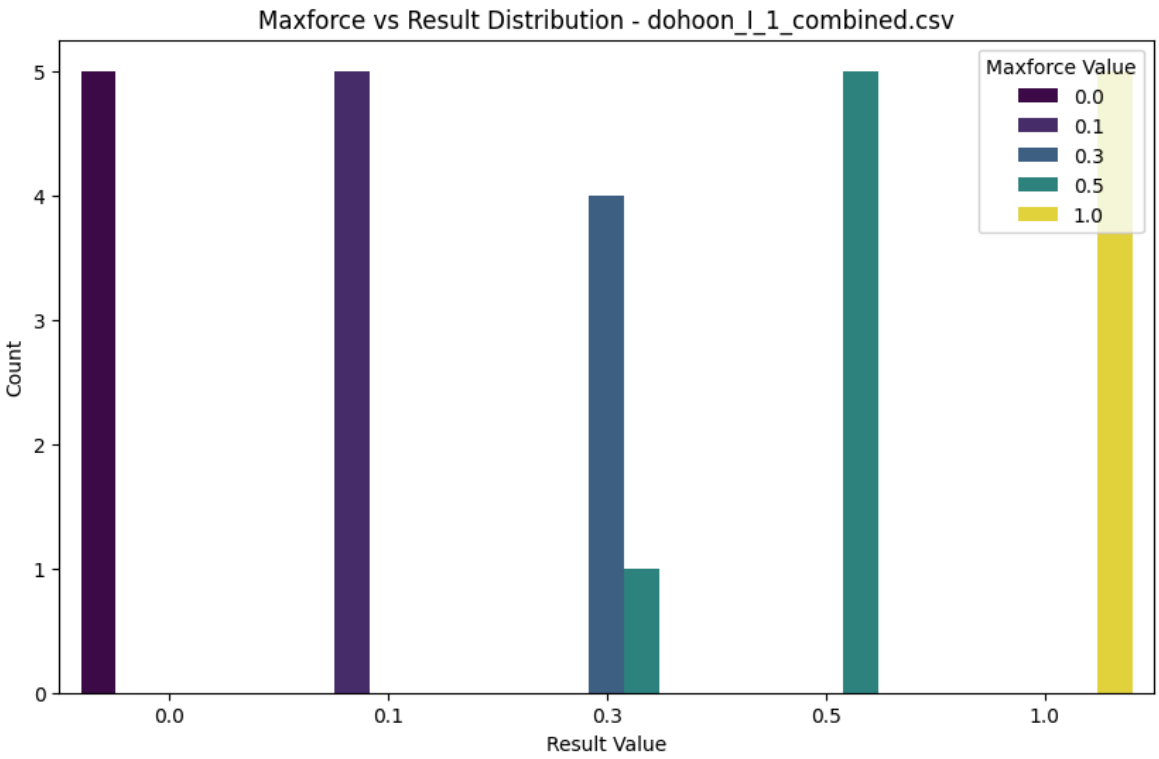
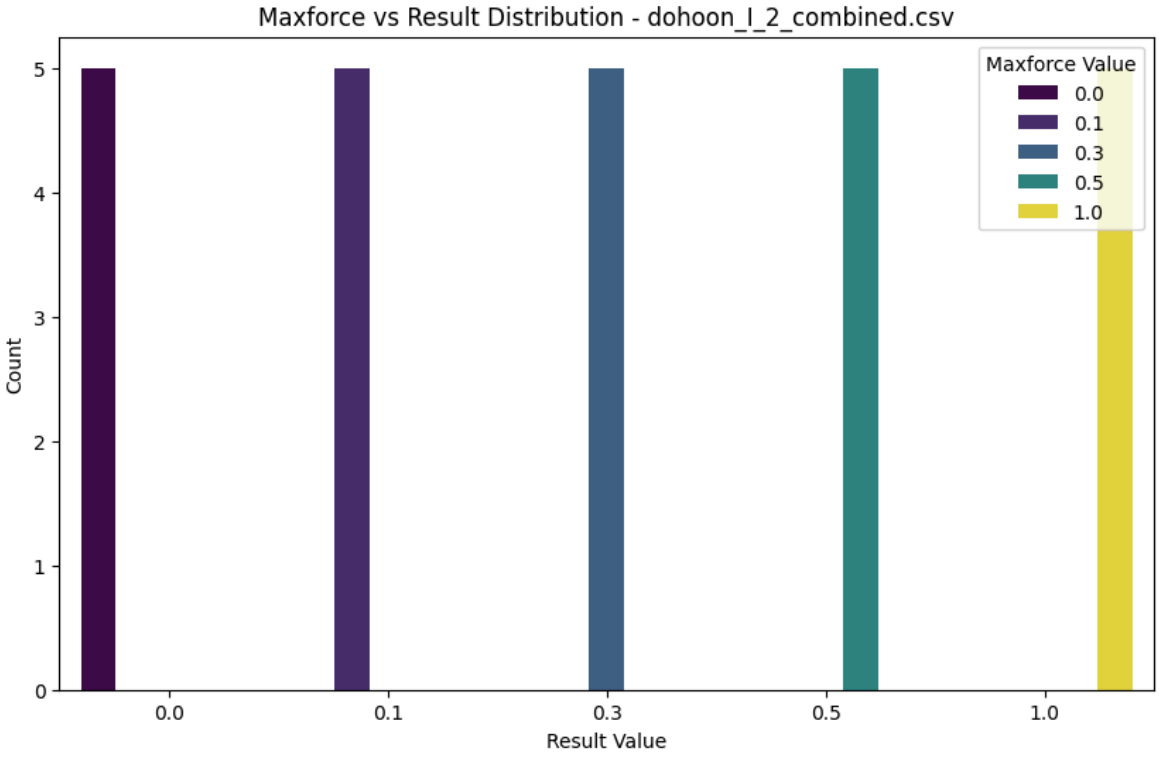
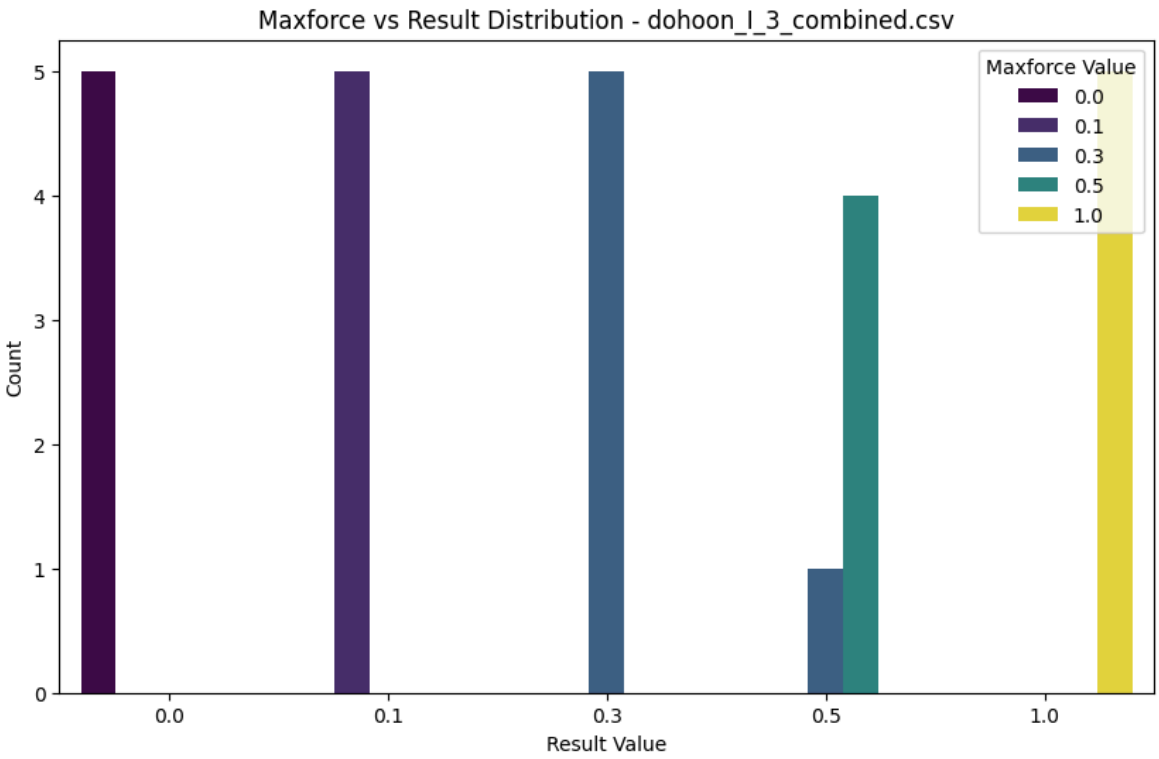
🔗 각 데이터 별 결과 시각화

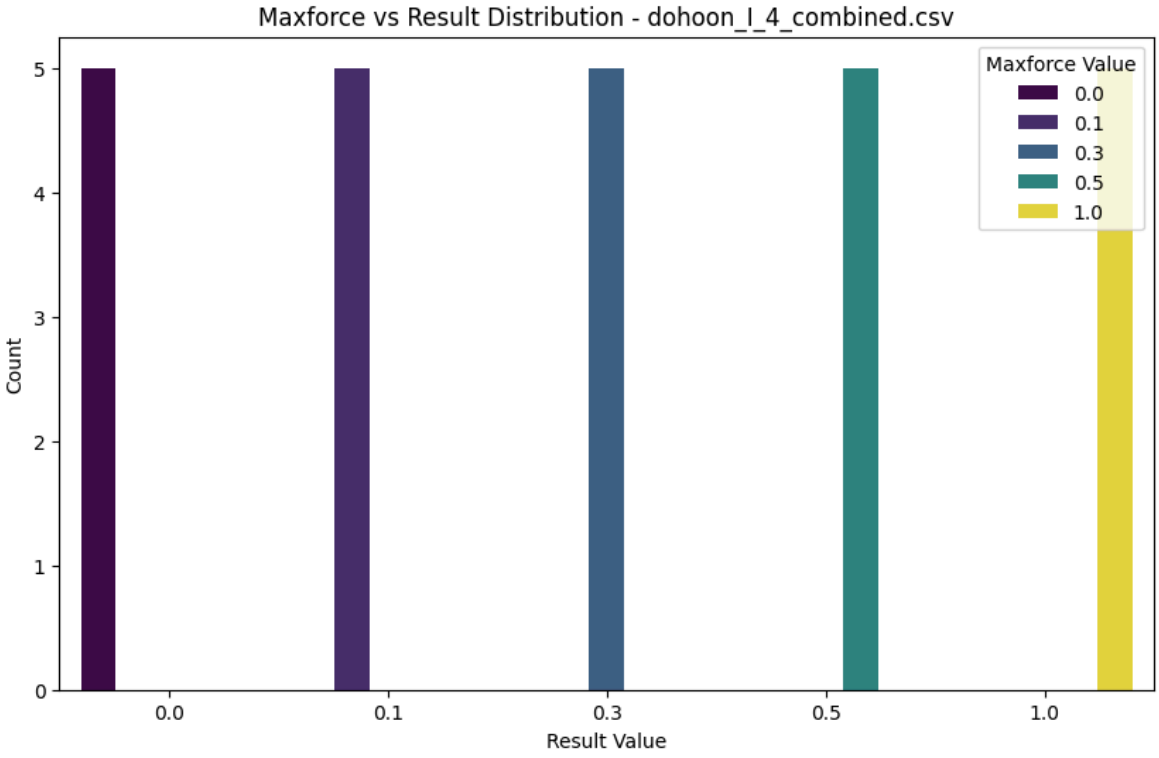
In [16]:
```
import pandas as pd
import os
import matplotlib.pyplot as plt
import seaborn as sns

# 파일이 저장된 디렉토리 경로 (예시)
directory_path = './combined_data'

# 디렉토리에서 모든 CSV 파일 리스트 가져오기
csv_files = [f for f in os.listdir(directory_path) if f.endswith('.csv')]
```

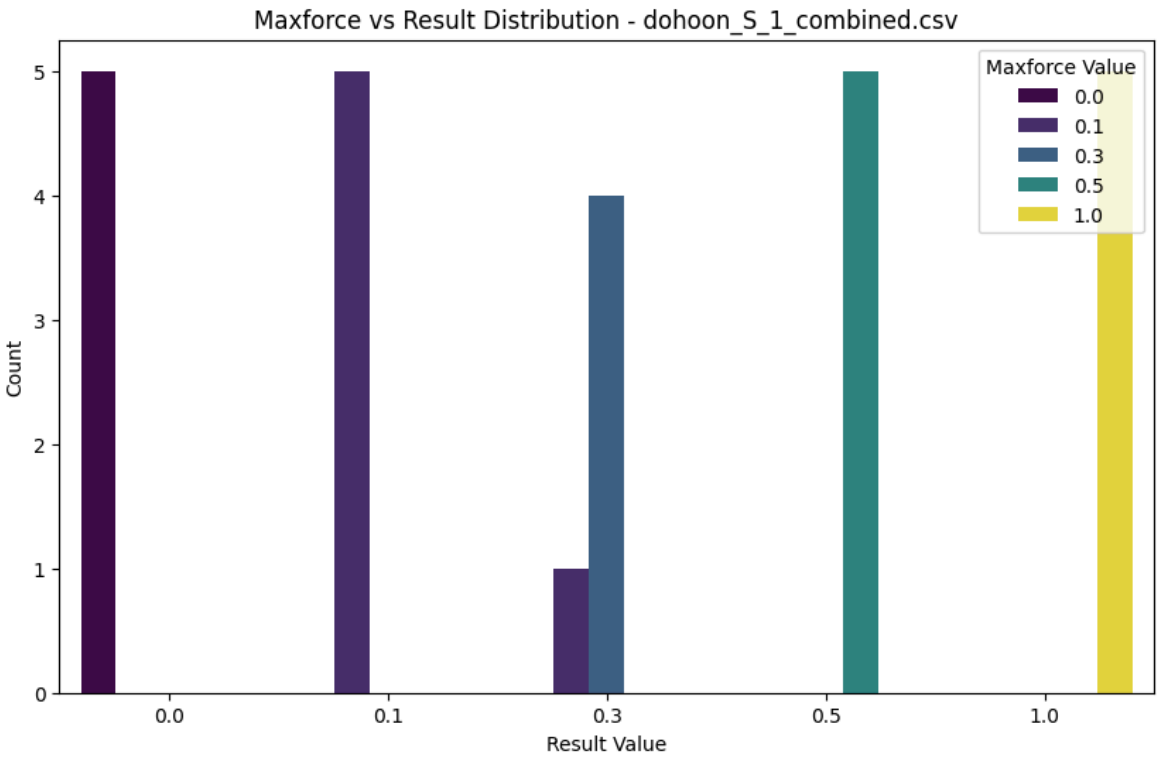
In [18]:
```
# 각 CSV 파일에 대해 작업 수행
for csv_file in csv_files:
    file_path = os.path.join(directory_path, csv_file)

    # CSV 파일 읽기
    df = pd.read_csv(file_path)

    # maxforce와 result 열이 있는지 확인
    if 'maxforce' in df.columns and 'result' in df.columns:
        maxforce_values = [0.0, 0.1, 0.3, 0.5, 1.0]
        result_values = [0.0, 0.1, 0.3, 0.5, 1.0]

        distribution = []

        for maxforce_value in maxforce_values:
            for result_value in result_values:
                count = len(df[(df['maxforce'] == maxforce_value) & (df['result'
                distribution.append({
                    'maxforce_value': maxforce_value,
                    'result_value': result_value,
                    'count': count
                })

        distribution_df = pd.DataFrame(distribution)

        # 막대그래프 생성
        plt.figure(figsize=(10, 6))
        sns.barplot(x='result_value', y='count', hue='maxforce_value', data=dist

        # 그래프 세부 설정
        plt.title(f'Maxforce vs Result Distribution - {csv_file}')
        plt.xlabel('Result Value')
        plt.ylabel('Count')
        plt.legend(title='Maxforce Value')

        # 그래프 출력
        plt.show()

    else:
        print(f"'maxforce' or 'result' column missing in {csv_file}")
```

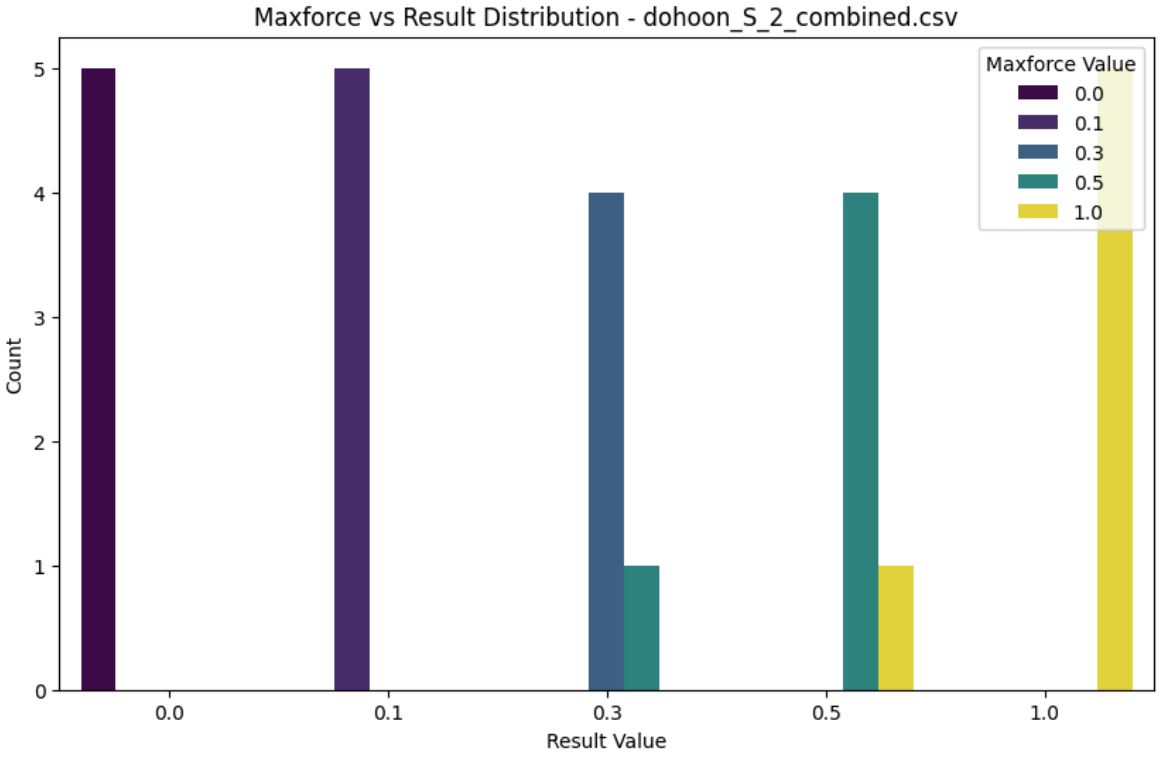

Maxforce vs Result Distribution - dohoon_I_1_combined.csv
Maxforce Value
0.0
0.1
0.3
0.5
1.0
Count
0
1
2
3
4
5
0.0
0.1
0.3
0.5
1.0
Result Value

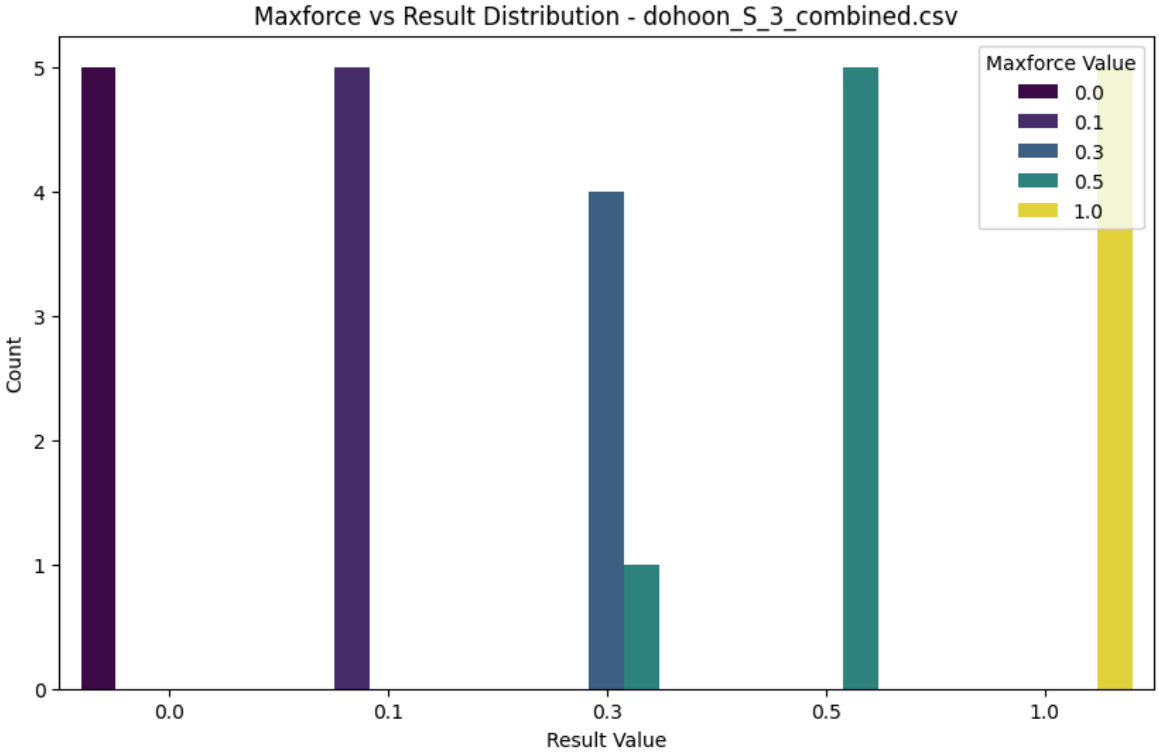

Maxforce vs Result Distribution - dohoon_I_2_combined.csv
Maxforce Value
0.0
0.1
0.3
0.5
1.0
Count
0
1
2
3
4
5
0.0
0.1
0.3
0.5
1.0
Result Value

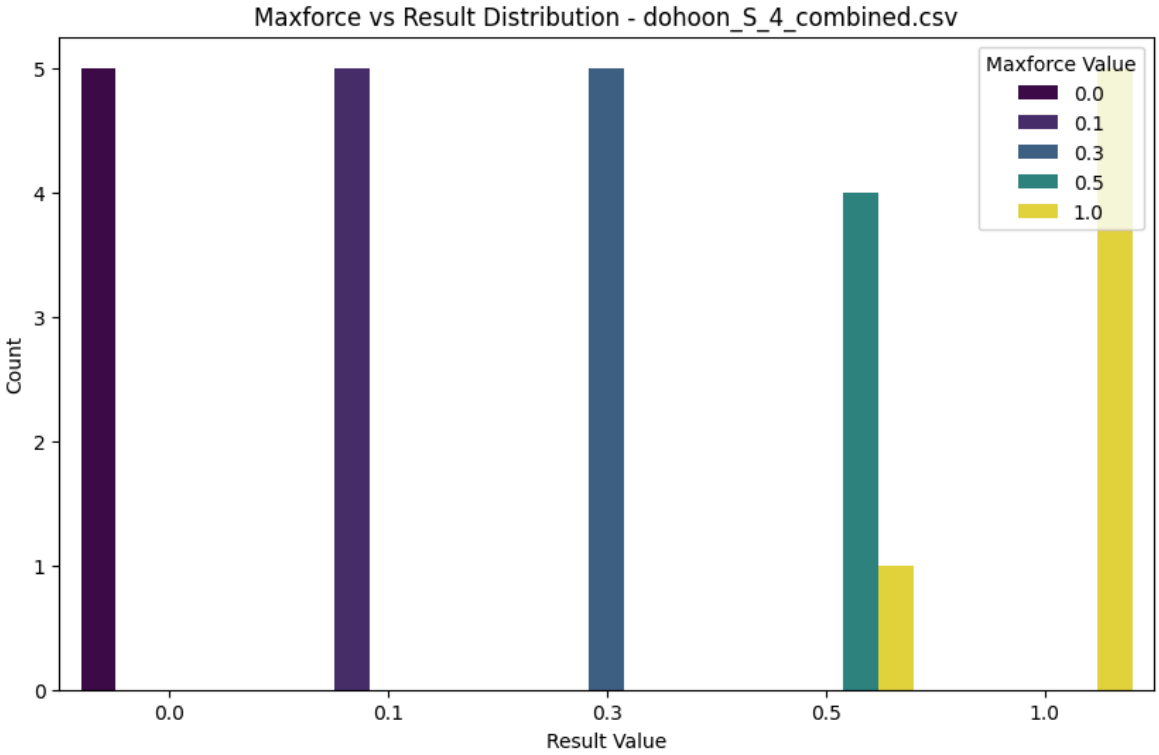

Maxforce vs Result Distribution - dohoon_I_3_combined.csv
Maxforce Value
0.0
0.1
0.3
0.5
1.0
Count
0
1
2
3
4
5
0.0
0.1
0.3
0.5
1.0
Result Value

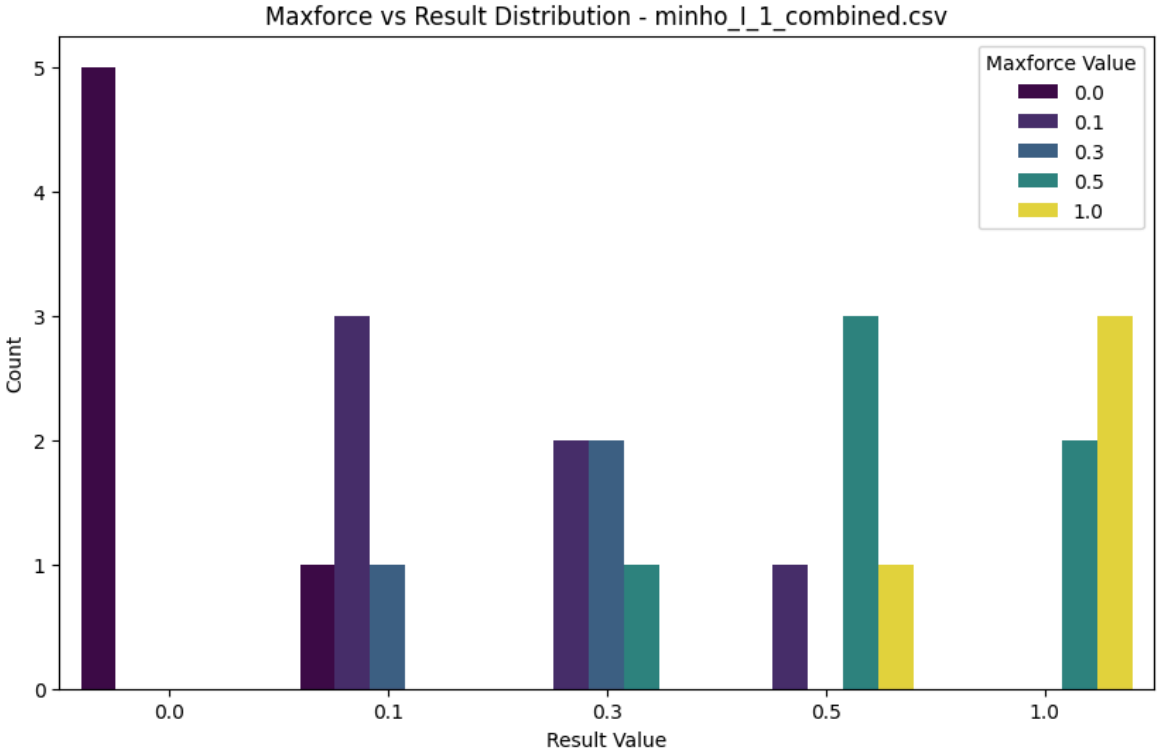

Maxforce vs Result Distribution - dohoon_I_4_combined.csv
Maxforce Value
0.0
0.1
0.3
0.5
1.0
Count
0
1
2
3
4
5
0.0
0.1
0.3
0.5
1.0
Result Value

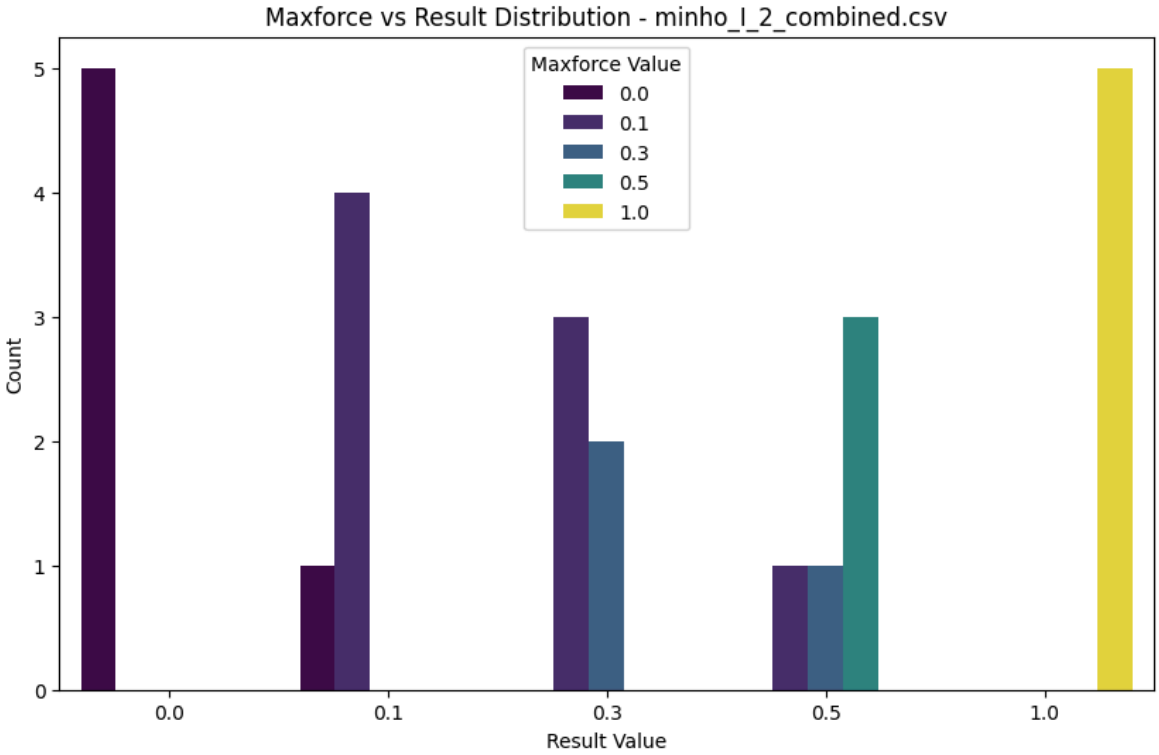

Maxforce vs Result Distribution - dohoon_S_1_combined.csv
Maxforce Value
0.0
0.1
0.3
0.5
1.0
Count
0
1
2
3
4
5
0.0
0.1
0.3
0.5
1.0
Result Value

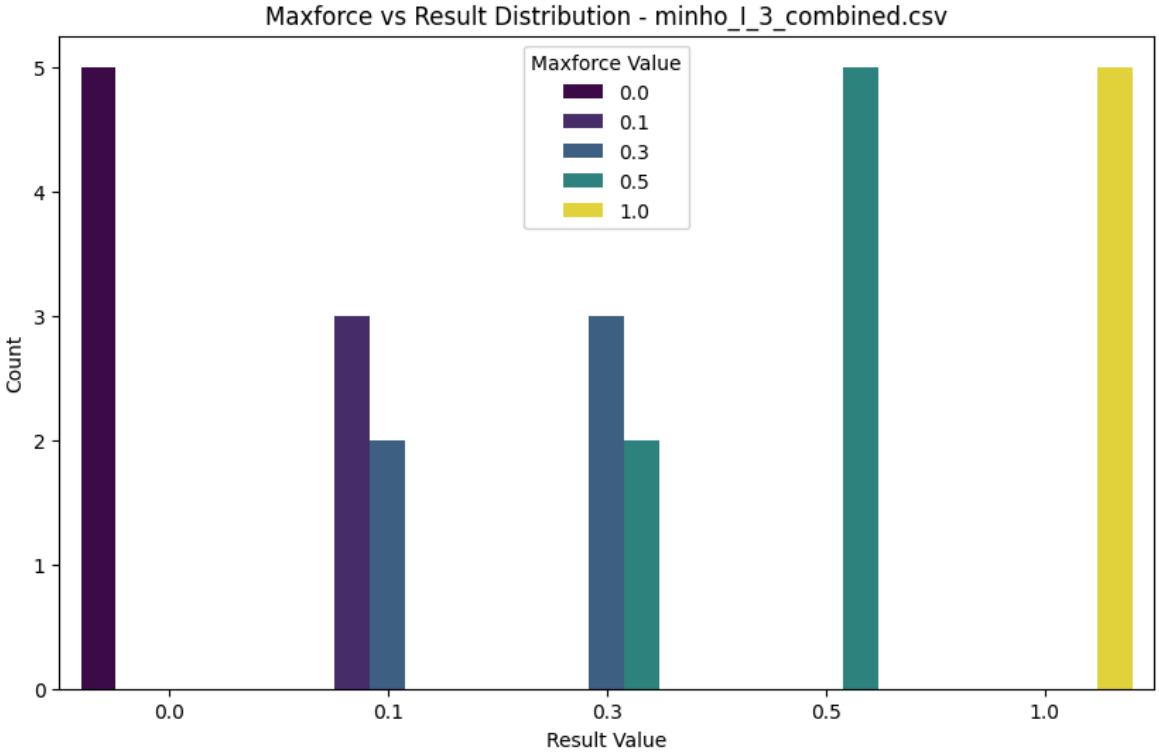

Maxforce vs Result Distribution - dohoon_S_2_combined.csv
Maxforce Value
0.0
0.1
0.3
0.5
1.0
Count
0
1
2
3
4
5
0.0
0.1
0.3
0.5
1.0
Result Value

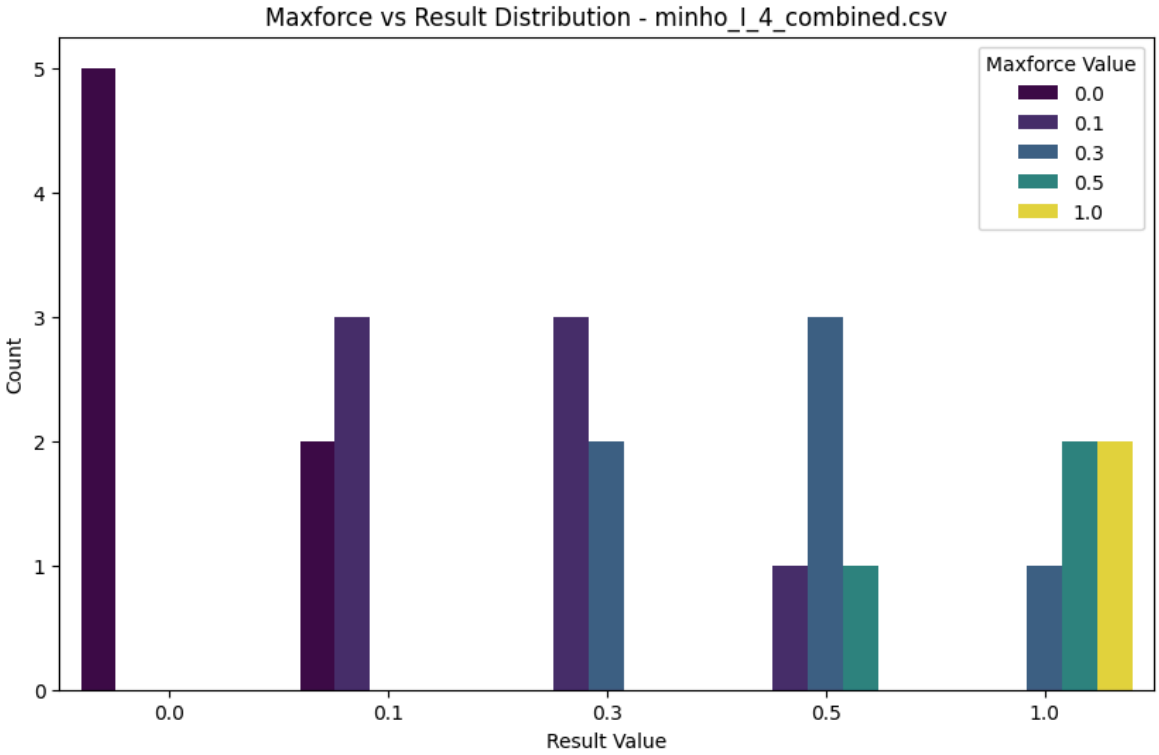

Maxforce vs Result Distribution - dohoon_S_3_combined.csv
Maxforce Value
0.0
0.1
0.3
0.5
1.0
Count
0
1
2
3
4
5
0.0
0.1
0.3
0.5
1.0
Result Value

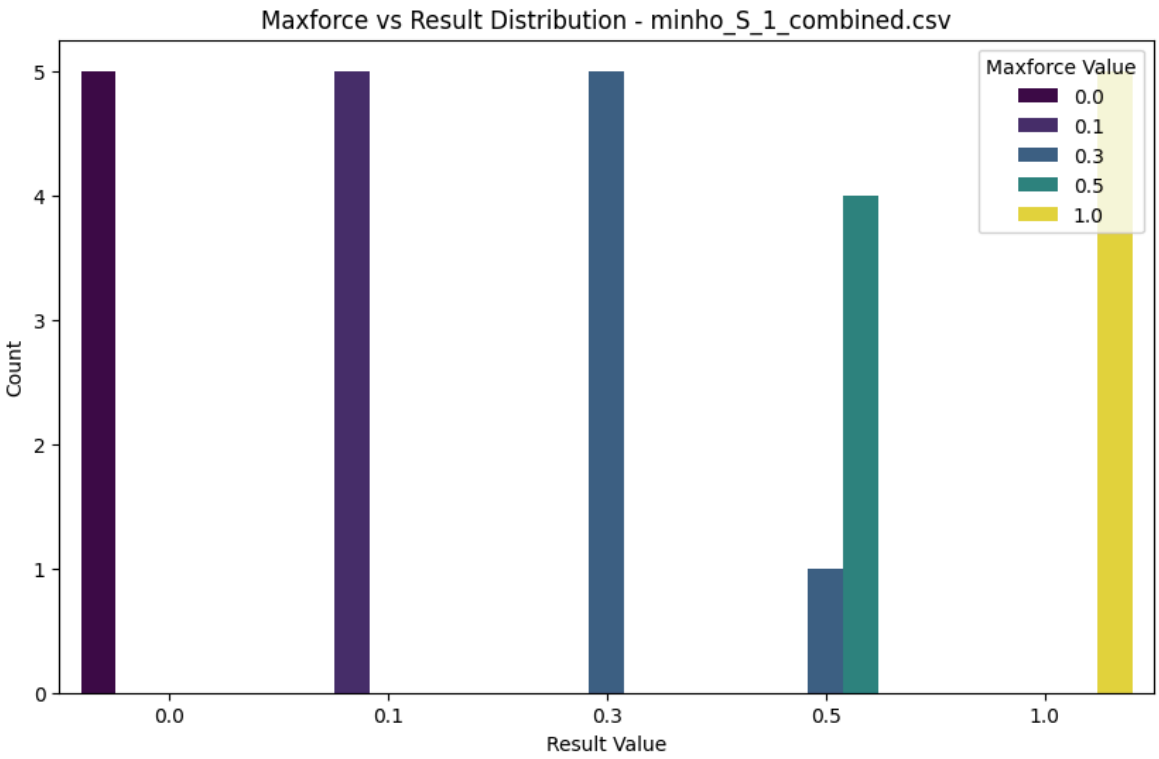

Maxforce vs Result Distribution - dohoon_S_4_combined.csv
Maxforce Value
0.0
0.1
0.3
0.5
1.0
Count
0
1
2
3
4
5
0.0
0.1
0.3
0.5
1.0
Result Value

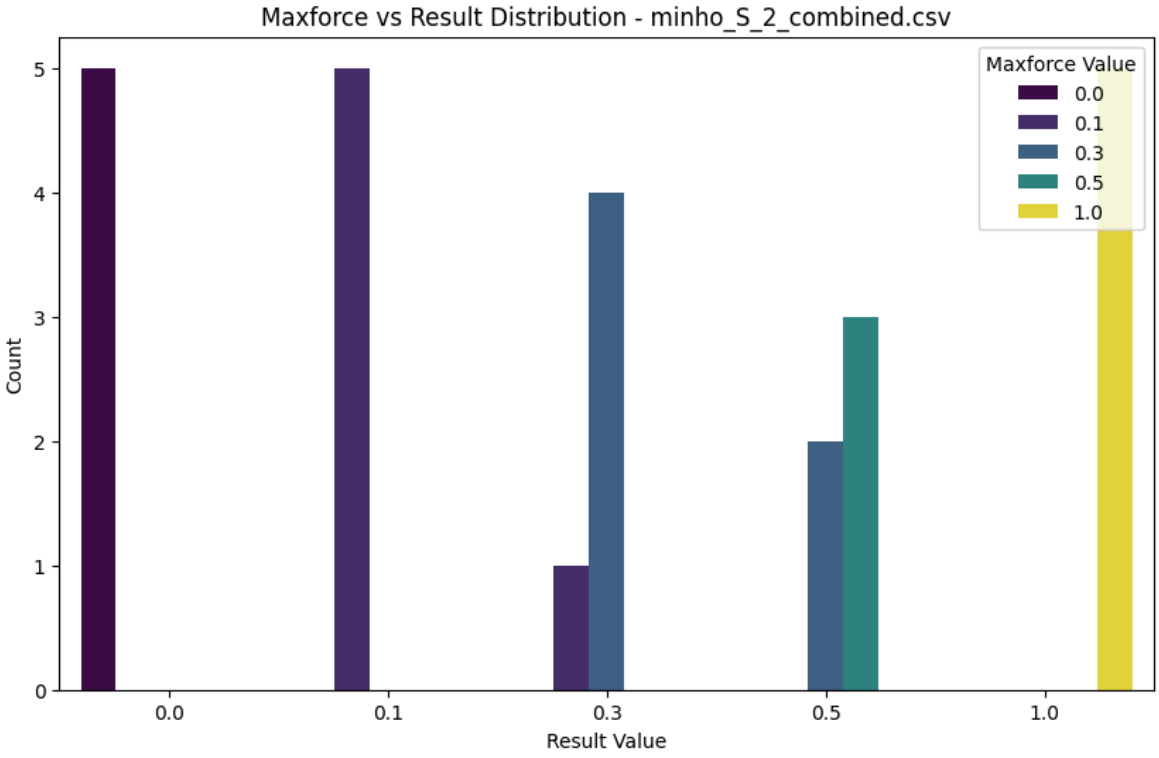

Maxforce vs Result Distribution - minho_I_1_combined.csv
Maxforce Value
0.0
0.1
0.3
0.5
1.0
Count
0
1
2
3
4
5
0.0
0.1
0.3
0.5
1.0
Result Value

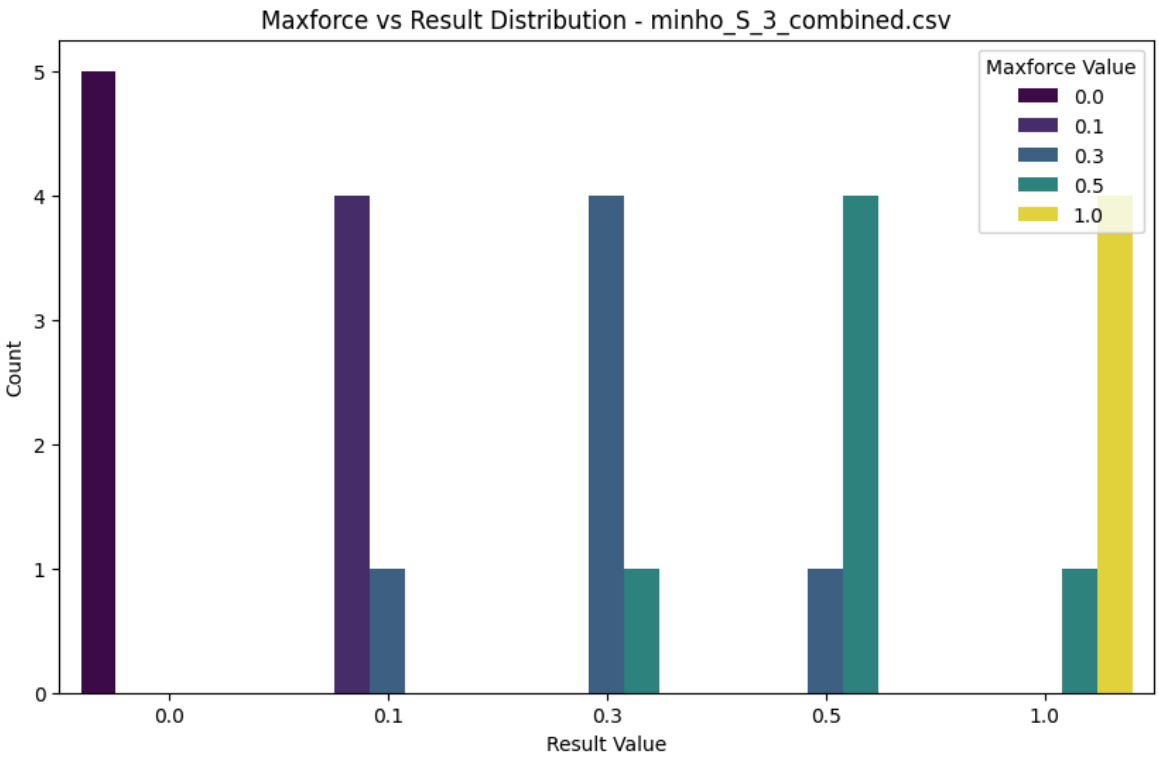

Maxforce vs Result Distribution - minho_I_2_combined.csv
Maxforce Value
0.0
0.1
0.3
0.5
1.0
Count
0
1
2
3
4
5
0.0
0.1
0.3
0.5
1.0
Result Value

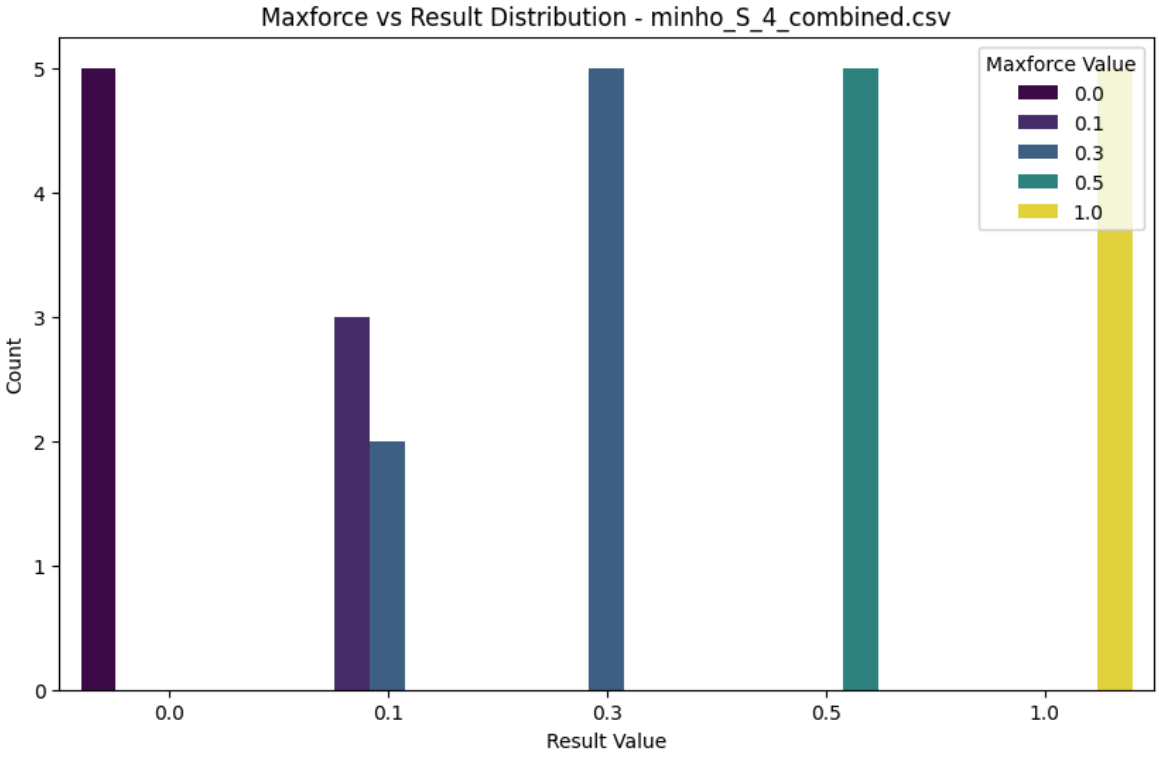

Maxforce vs Result Distribution - minho_I_3_combined.csv
Maxforce Value
0.0
0.1
0.3
0.5
1.0
Count
Result Value
0
1
2
3
4
5
0.0
0.1
0.3
0.5
1.0

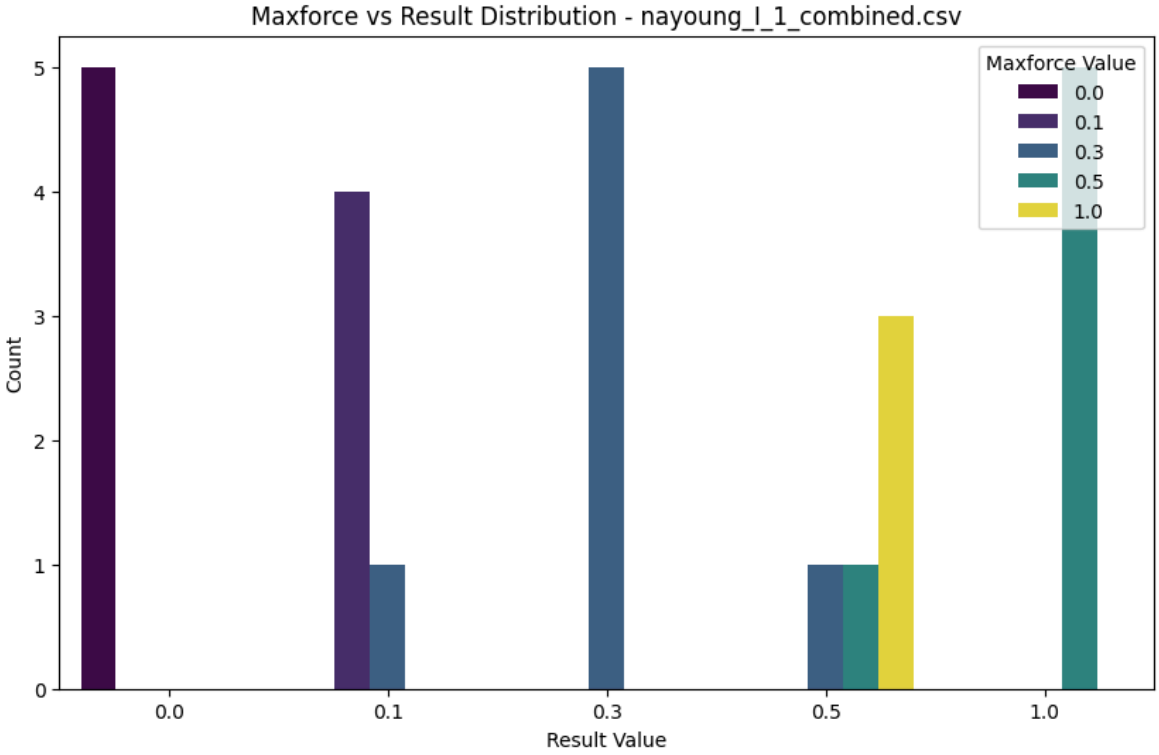

Maxforce vs Result Distribution - minho_I_4_combined.csv
Maxforce Value
0.0
0.1
0.3
0.5
1.0
Count
Result Value
0
1
2
3
4
5
0.0
0.1
0.3
0.5
1.0

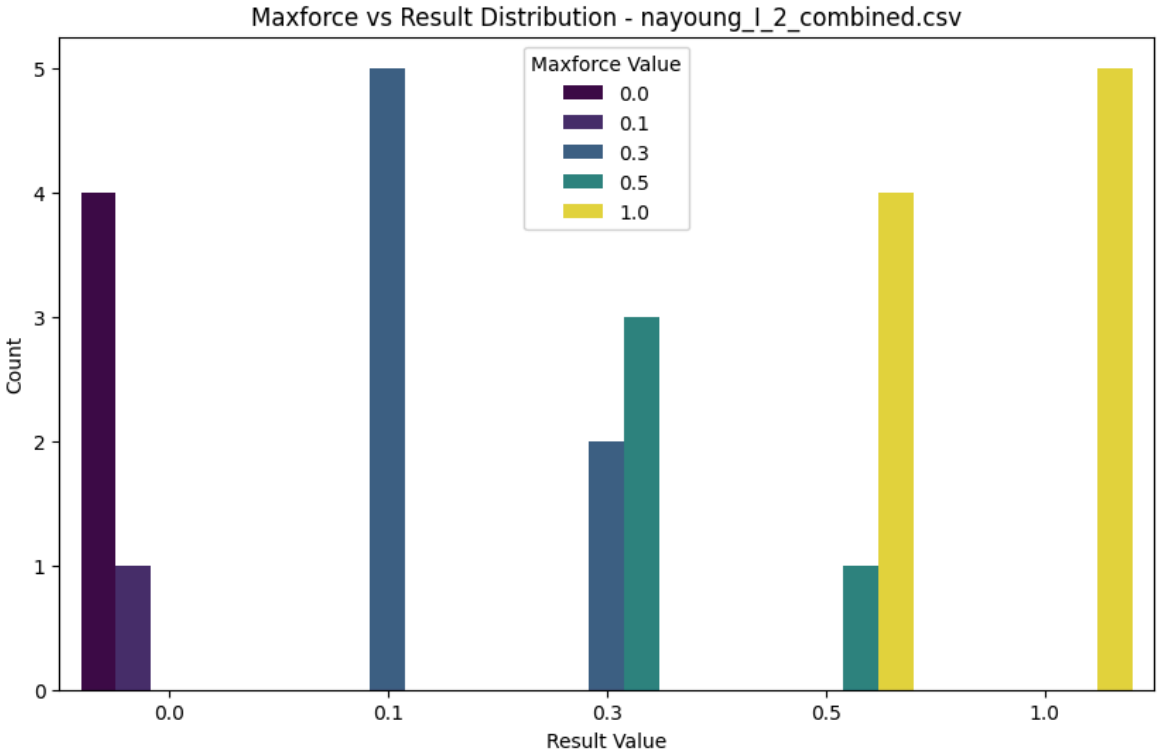

Maxforce vs Result Distribution - minho_S_1_combined.csv
Maxforce Value
0.0
0.1
0.3
0.5
1.0
Count
0
1
2
3
4
5
0.0
0.1
0.3
0.5
1.0
Result Value

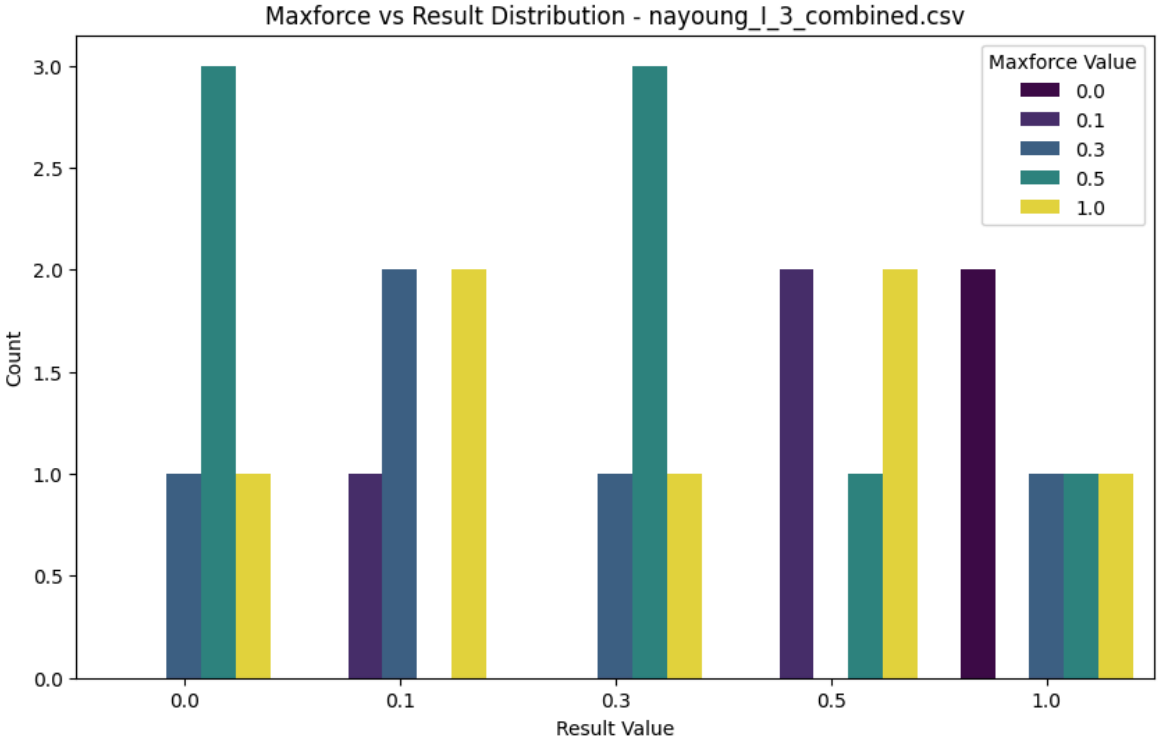

Maxforce vs Result Distribution - minho_S_2_combined.csv
Maxforce Value
0.0
0.1
0.3
0.5
1.0
Count
0
1
2
3
4
5
0.0
0.1
0.3
0.5
1.0
Result Value

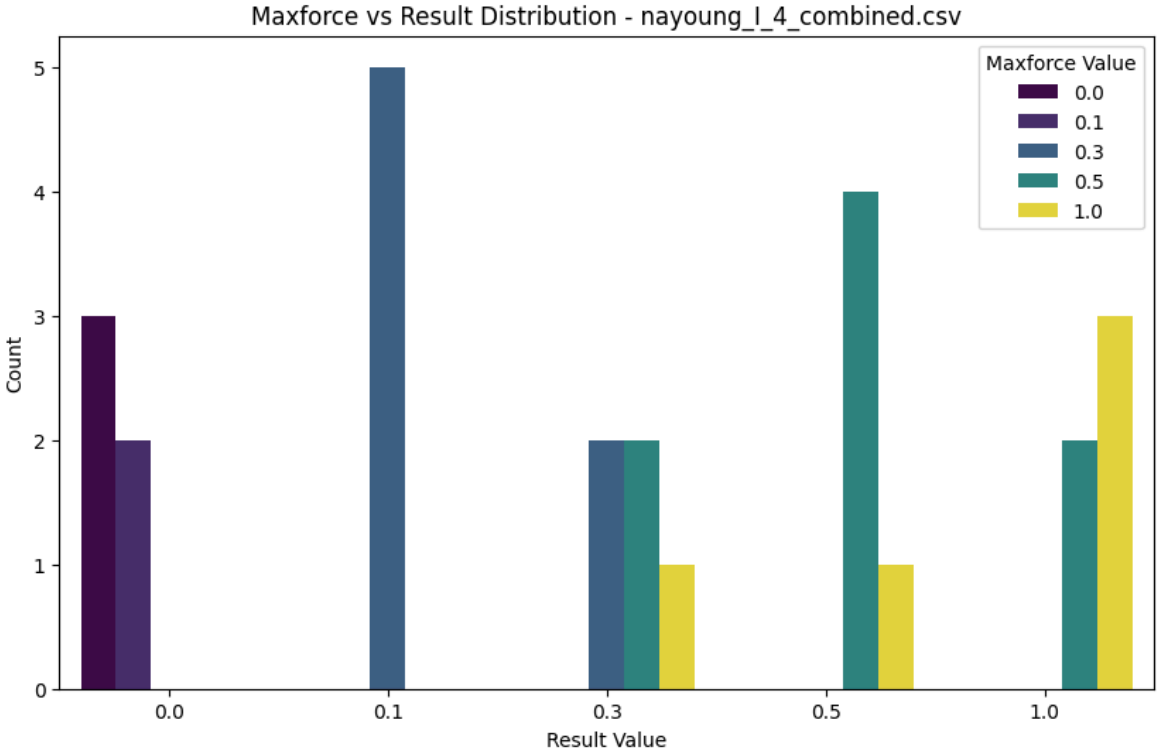

Maxforce vs Result Distribution - minho_S_3_combined.csv
Maxforce Value
0.0
0.1
0.3
0.5
1.0
Count
0
1
2
3
4
5
0.0
0.1
0.3
0.5
1.0
Result Value

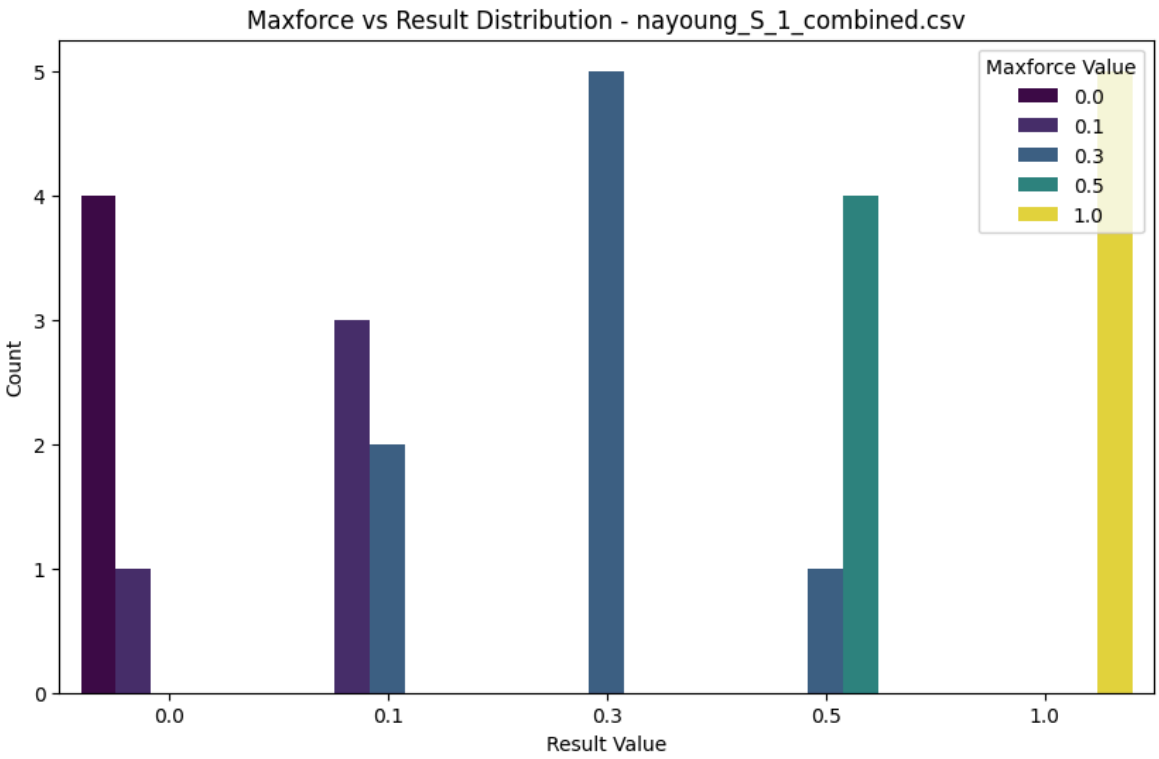

Maxforce vs Result Distribution - minho_S_4_combined.csv
Maxforce Value
0.0
0.1
0.3
0.5
1.0
Count
0
1
2
3
4
5
0.0
0.1
0.3
0.5
1.0
Result Value

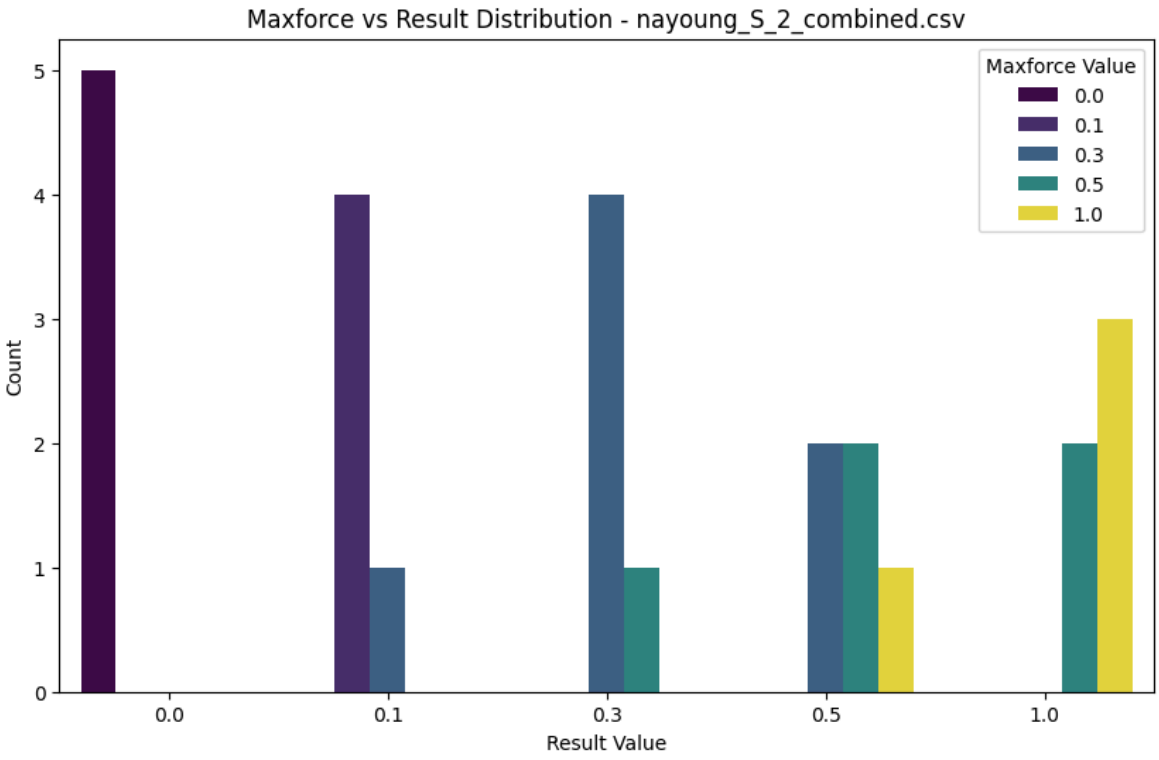

Maxforce vs Result Distribution - nayoung_I_1_combined.csv
Maxforce Value
0.0
0.1
0.3
0.5
1.0
Count
0
1
2
3
4
5
0.0
0.1
0.3
0.5
1.0
Result Value

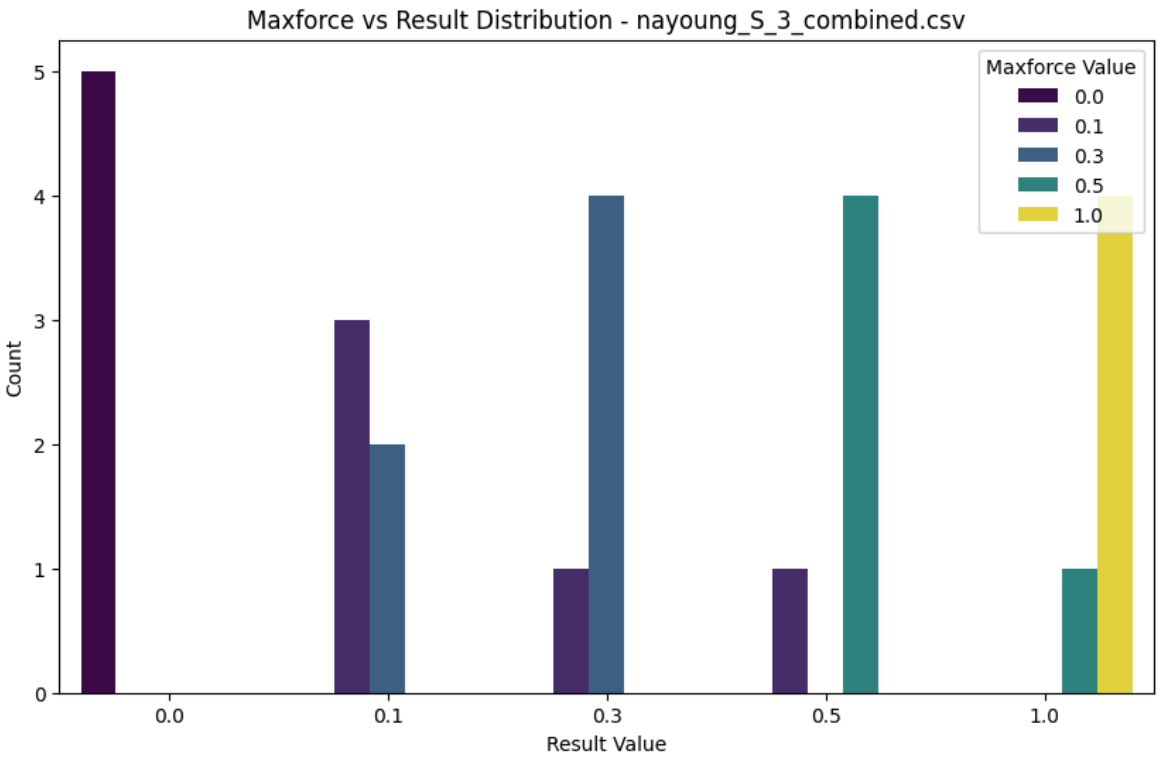

Maxforce vs Result Distribution - nayoung_I_2_combined.csv
Maxforce Value
0.0
0.1
0.3
0.5
1.0
Count
0
1
2
3
4
5
0.0
0.1
0.3
0.5
1.0
Result Value

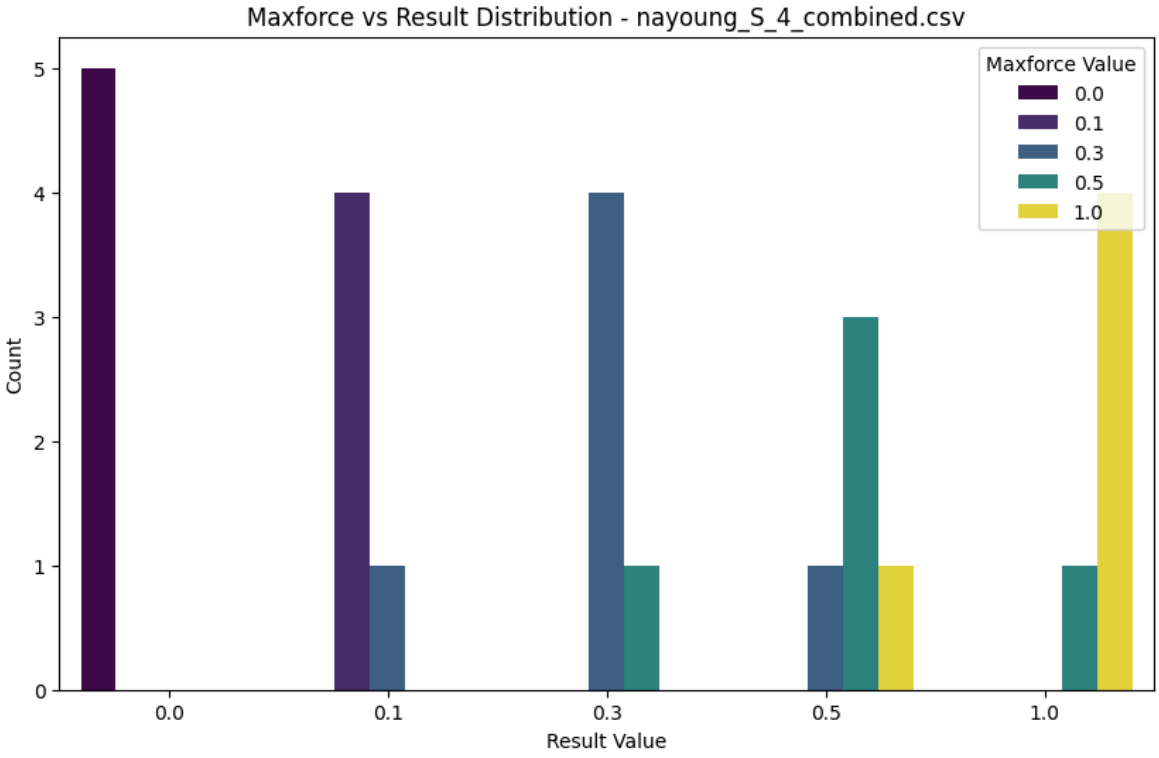

Maxforce vs Result Distribution - nayoung_l_3_combined.csv
Maxforce Value
0.0
0.1
0.3
0.5
1.0
Count
Result Value
0.0
0.1
0.3
0.5
1.0

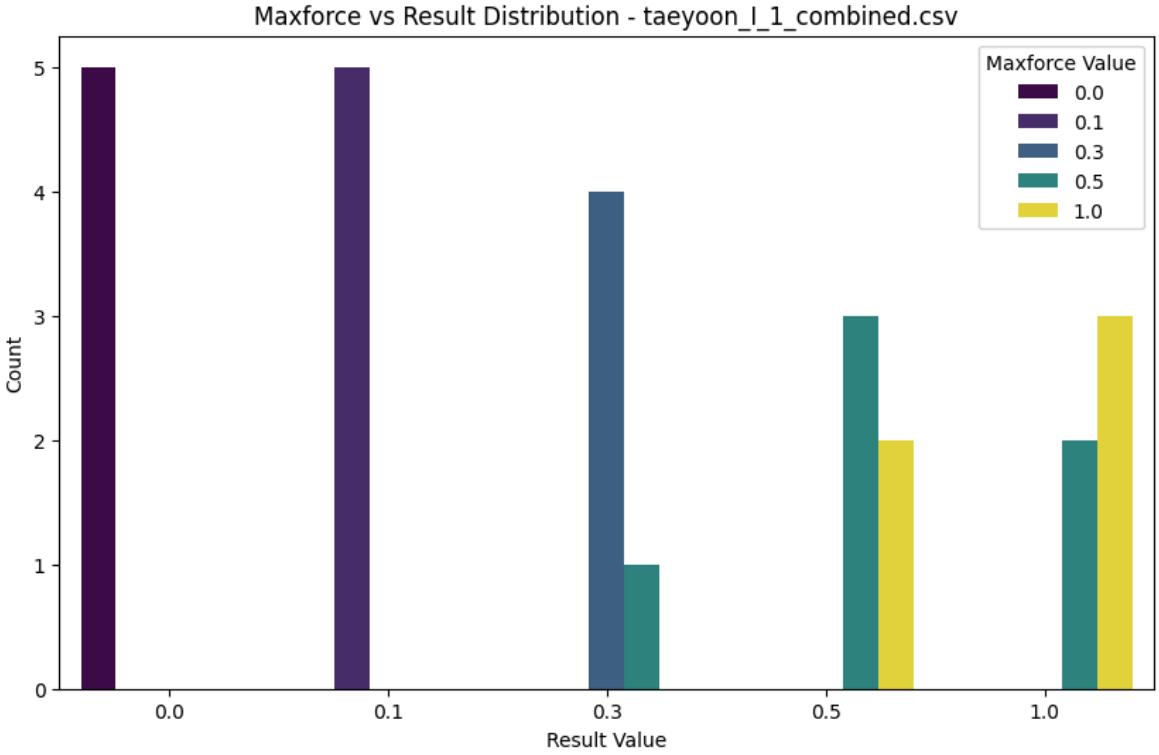

Maxforce vs Result Distribution - nayoung_l_4_combined.csv
Maxforce Value
0.0
0.1
0.3
0.5
1.0
Count
Result Value
0.0
0.1
0.3
0.5
1.0

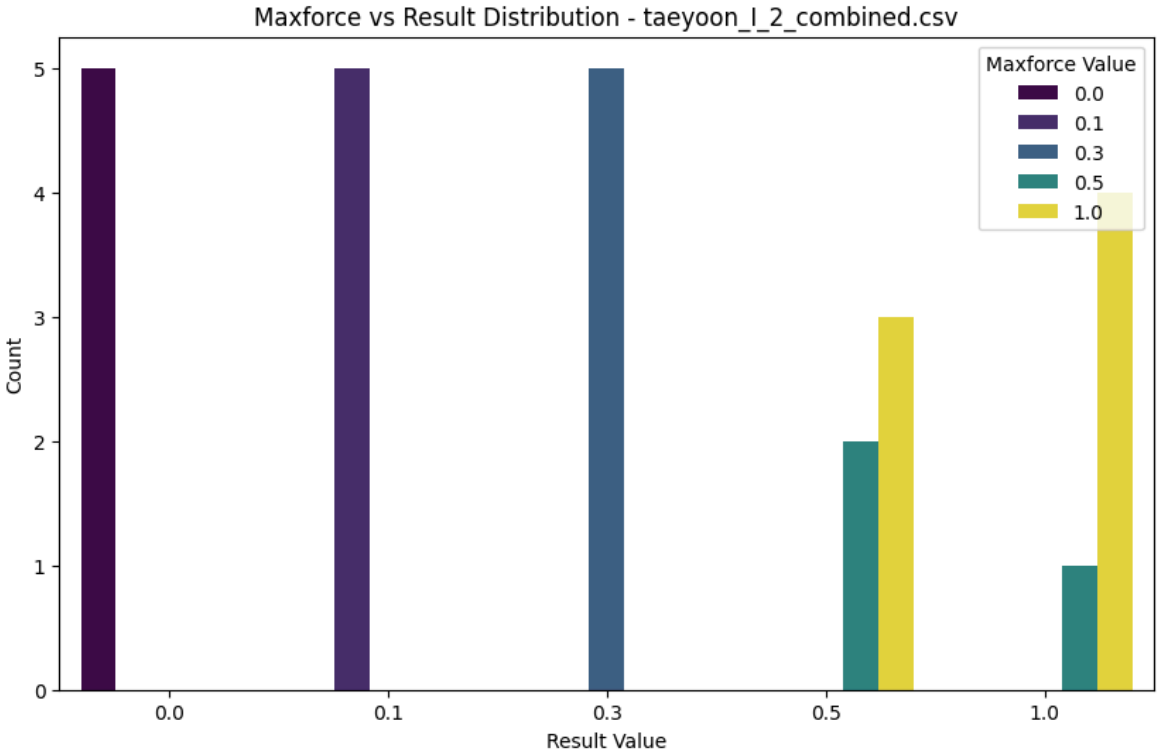

Maxforce vs Result Distribution - nayoung_S_1_combined.csv
Maxforce Value
0.0
0.1
0.3
0.5
1.0
Count
0
1
2
3
4
5
0.0
0.1
0.3
0.5
1.0
Result Value

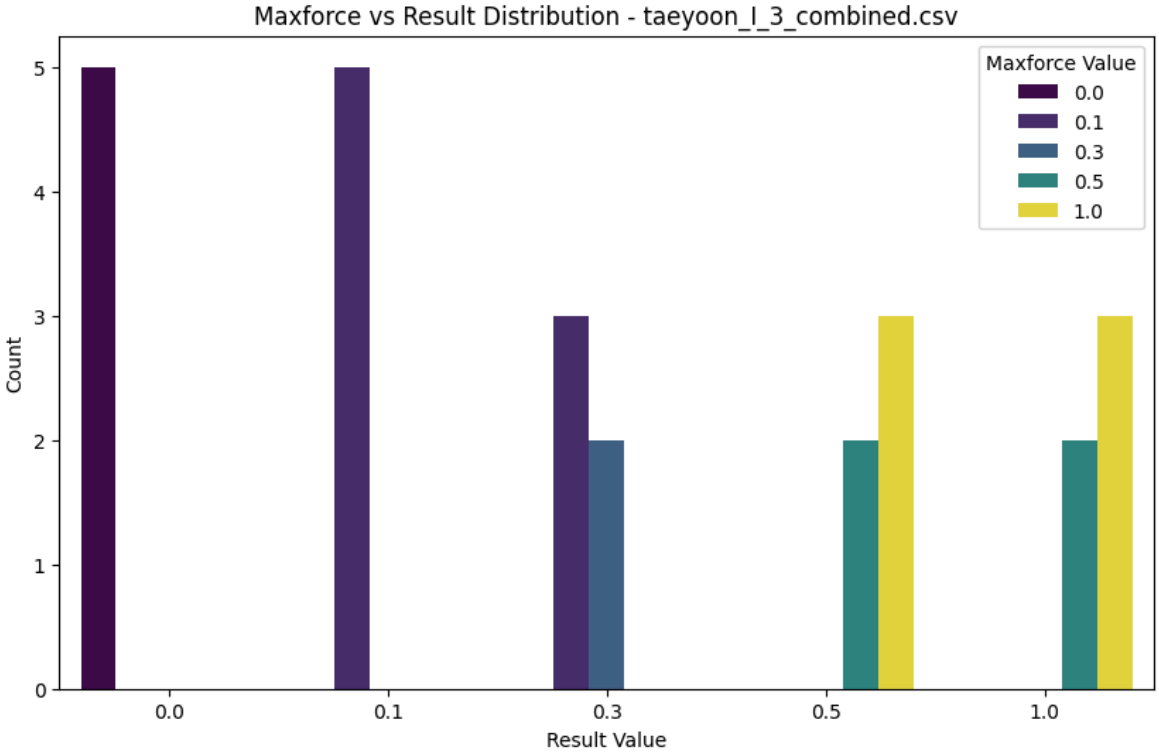

Maxforce vs Result Distribution - nayoung_S_2_combined.csv
Maxforce Value
0.0
0.1
0.3
0.5
1.0
Count
0
1
2
3
4
5
0.0
0.1
0.3
0.5
1.0
Result Value

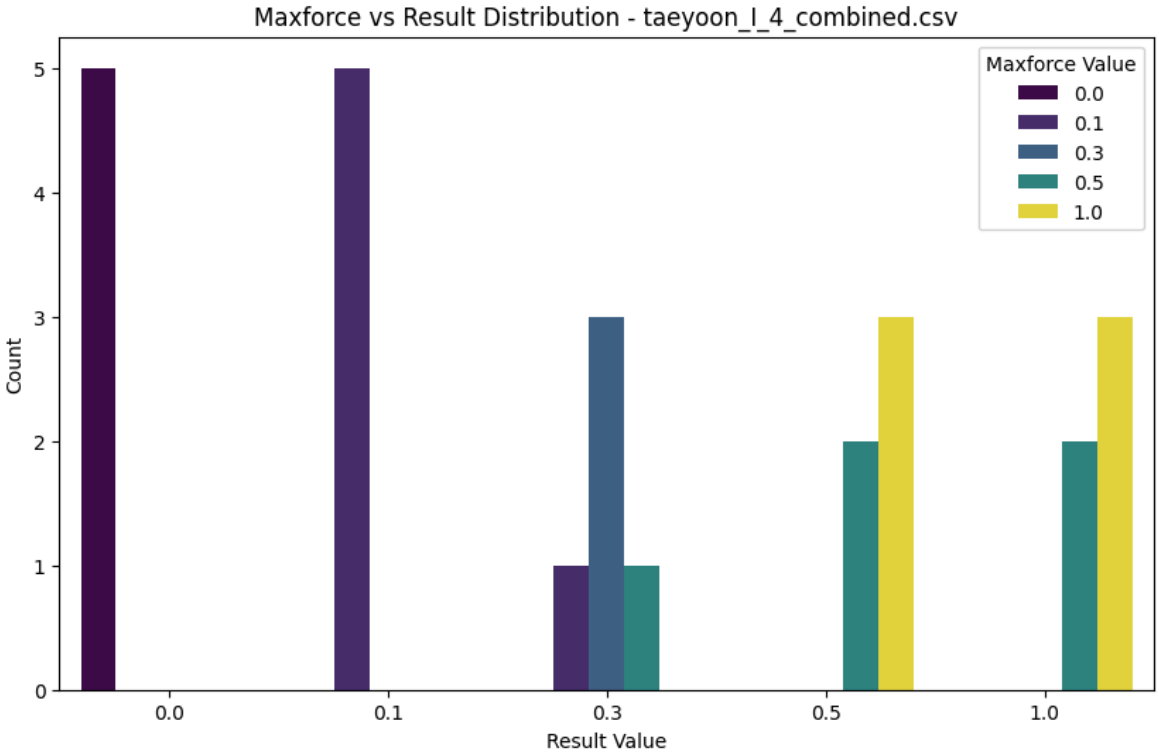

Maxforce vs Result Distribution - nayoung_S_3_combined.csv
Maxforce Value
0.0
0.1
0.3
0.5
1.0
Count
0
1
2
3
4
5
0.0
0.1
0.3
0.5
1.0
Result Value

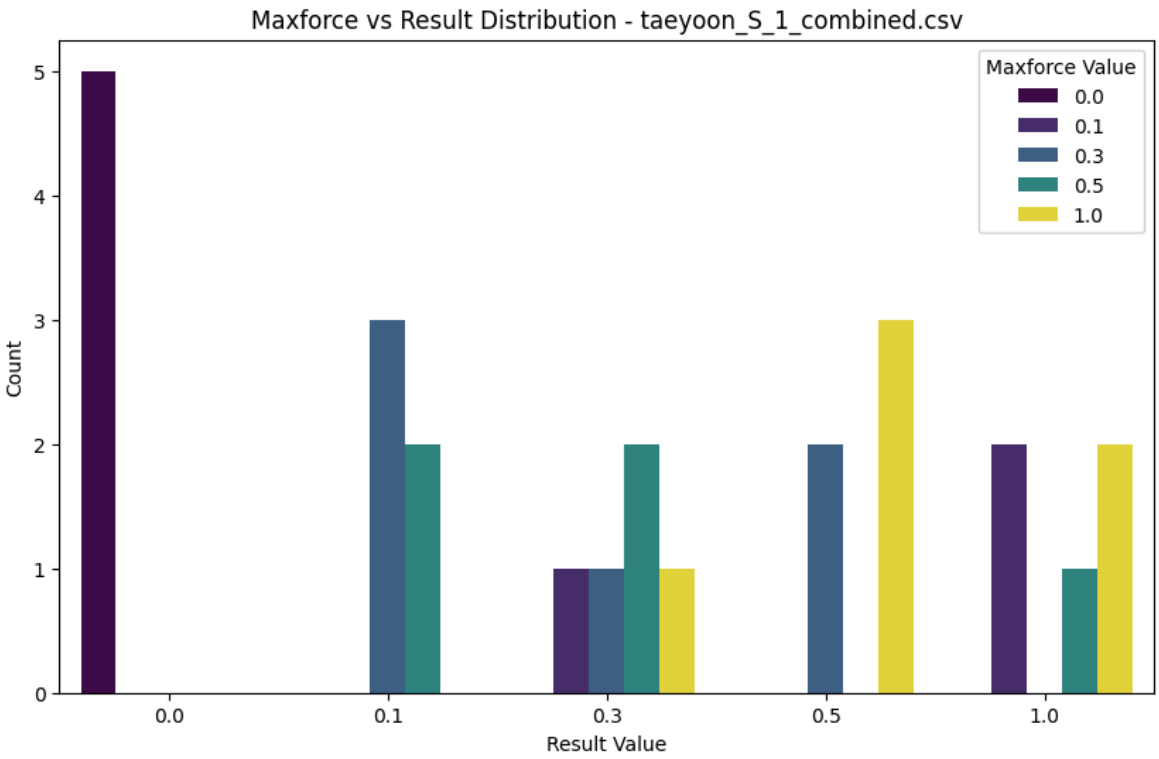

Maxforce vs Result Distribution - nayoung_S_4_combined.csv
Maxforce Value
0.0
0.1
0.3
0.5
1.0
Count
0
1
2
3
4
5
0.0
0.1
0.3
0.5
1.0
Result Value

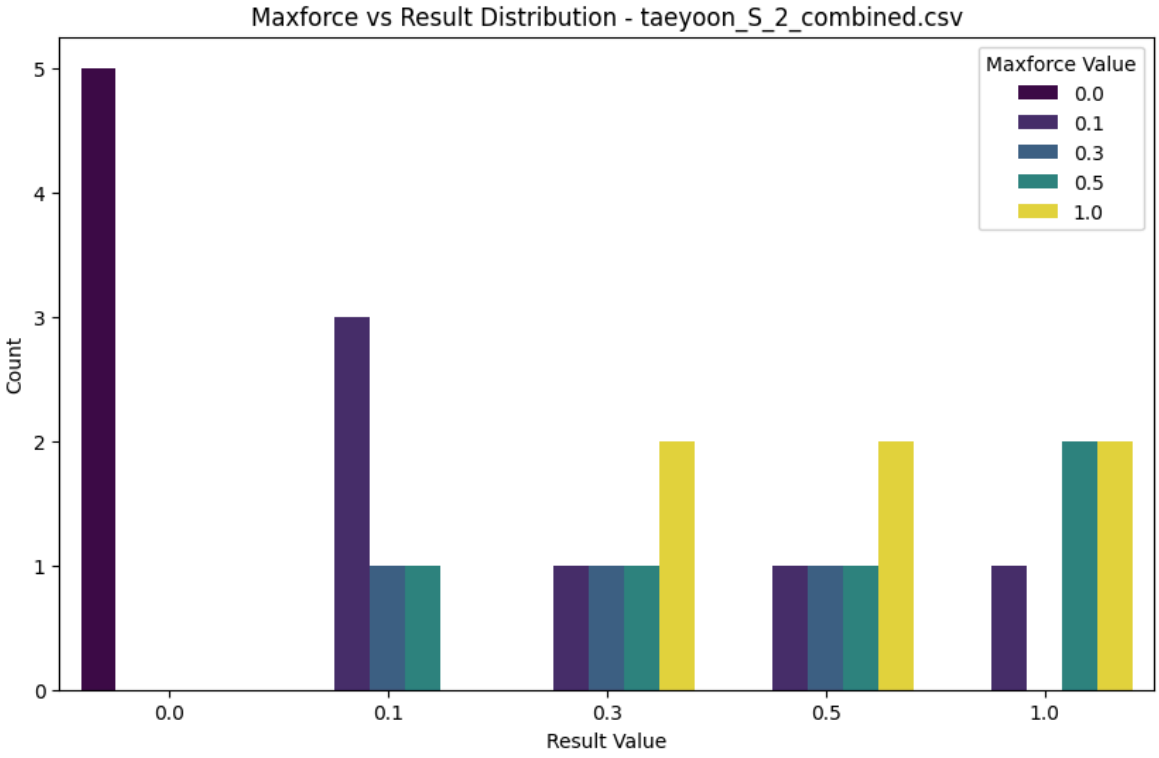

Maxforce vs Result Distribution - taeyoon_l_1_combined.csv
Maxforce Value
0.0
0.1
0.3
0.5
1.0
Count
0
1
2
3
4
5
0.0
0.1
0.3
0.5
1.0
Result Value

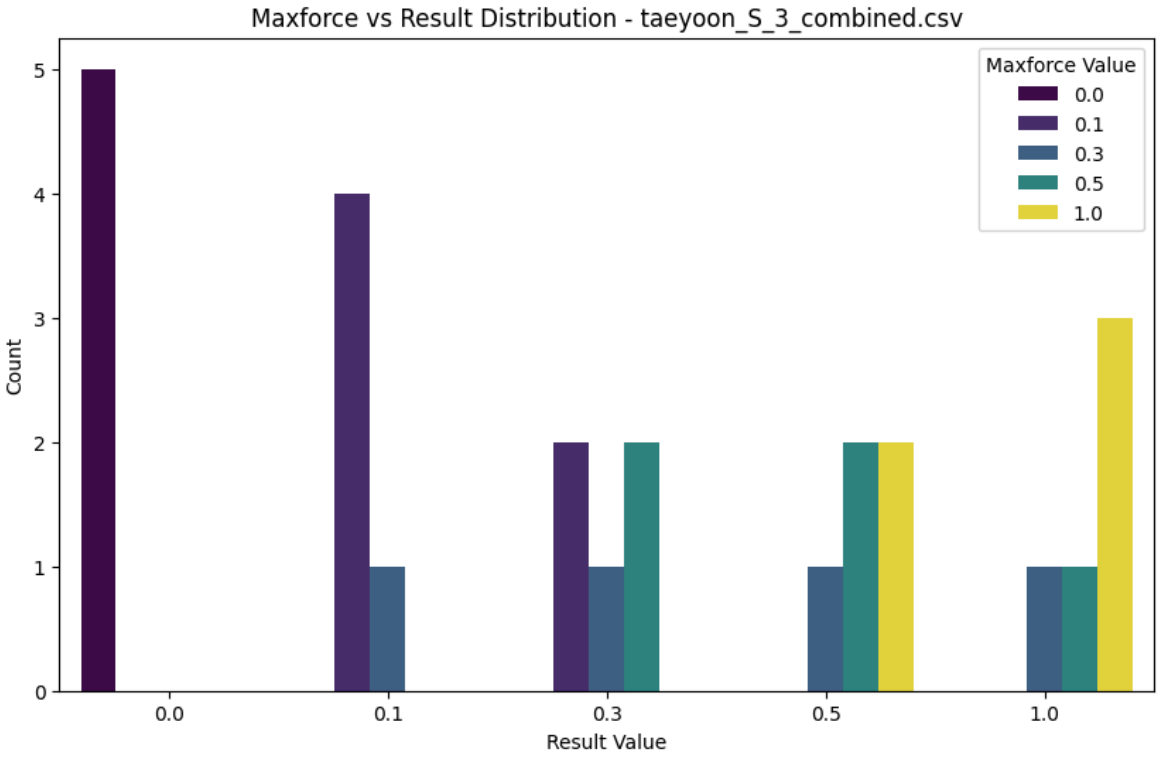

Maxforce vs Result Distribution - taeyoon_l_2_combined.csv
Maxforce Value
0.0
0.1
0.3
0.5
1.0
Count
0
1
2
3
4
5
0.0
0.1
0.3
0.5
1.0
Result Value

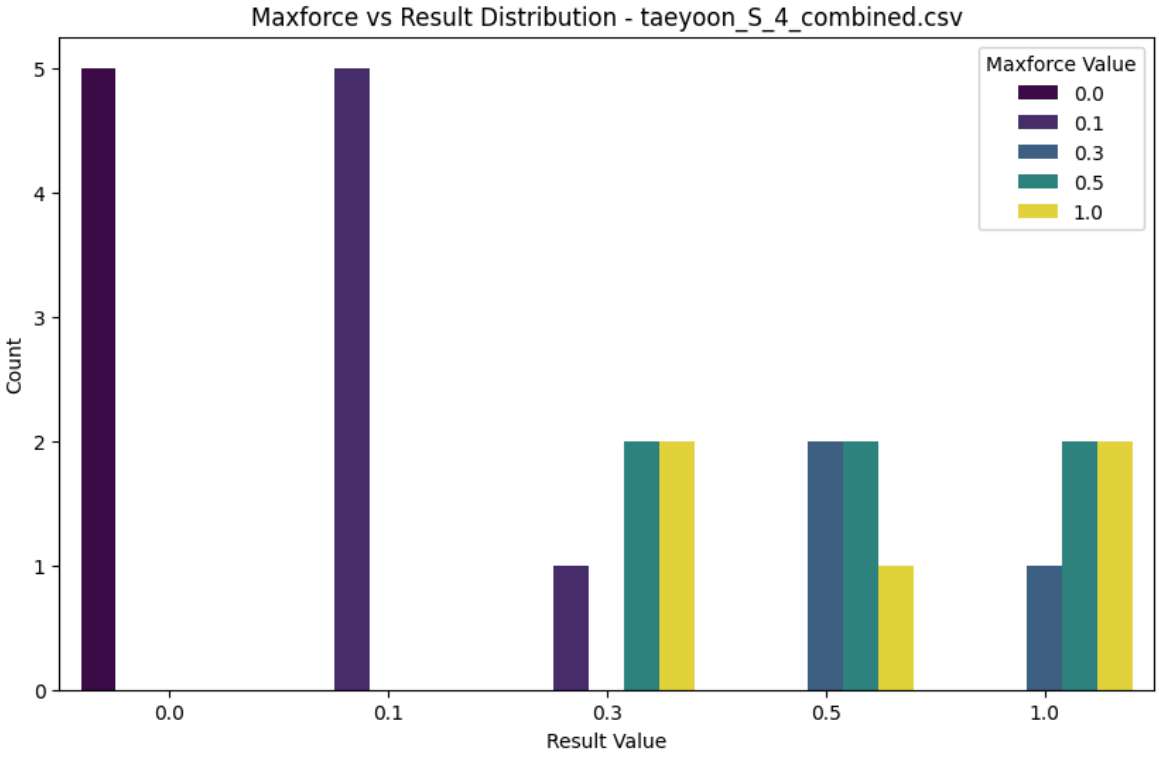

Maxforce vs Result Distribution - taeyoon_I_3_combined.csv
Maxforce Value
0.0
0.1
0.3
0.5
1.0
Count
0
1
2
3
4
5
0.0
0.1
0.3
0.5
1.0
Result Value


Maxforce vs Result Distribution - taeyoon_I_4_combined.csv
Maxforce Value
0.0
0.1
0.3
0.5
1.0
Count
0
1
2
3
4
5
0.0
0.1
0.3
0.5
1.0
Result Value


Maxforce vs Result Distribution - taeyoon_S_1_combined.csv
Maxforce Value
0.0
0.1
0.3
0.5
1.0
Count
0
1
2
3
4
5
0.0
0.1
0.3
0.5
1.0
Result Value

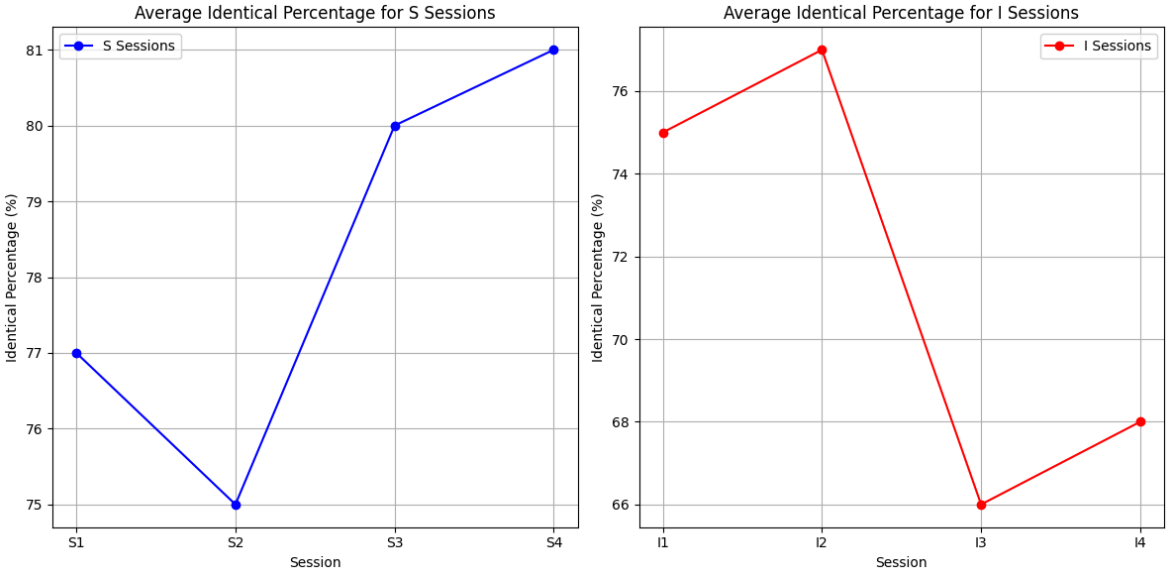

Maxforce vs Result Distribution - taeyoon_S_2_combined.csv
Maxforce Value
0.0
0.1
0.3
0.5
1.0
Count
0
1
2
3
4
5
0.0
0.1
0.3
0.5
1.0
Result Value





In [19]:
```python
# S와 I에 해당하는 identical_percentage 저장 딕셔너리
identical_percentages = {'S1': [], 'S2': [], 'S3': [], 'S4': [],
                         'I1': [], 'I2': [], 'I3': [], 'I4': []}

# 각 CSV 파일에 대해 작업 수행
for csv_file in csv_files:
    file_path = os.path.join(directory_path, csv_file)

    # CSV 파일 읽기
    df = pd.read_csv(file_path)

    # maxforce와 result 열이 있는지 확인
    if 'maxforce' in df.columns and 'result' in df.columns:
```

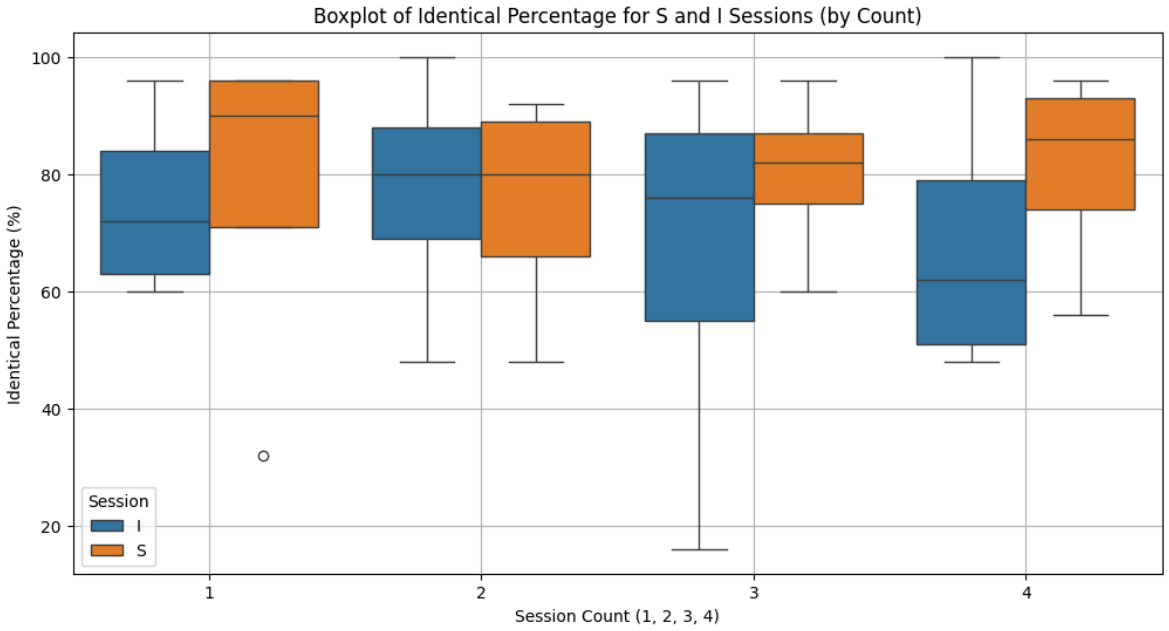
```
        # maxforce와 result가 동일한 값의 비율 계산
        identical_percentage = (df['maxforce'] == df['result']).mean() * 100

        # 파일 이름에서 S1, S2, S3, S4 또는 I1, I2, I3, I4를 추출하여 해당 리스트
        if '_S_1_' in csv_file:
            identical_percentages['S1'].append(identical_percentage)
        elif '_S_2_' in csv_file:
            identical_percentages['S2'].append(identical_percentage)
        elif '_S_3_' in csv_file:
            identical_percentages['S3'].append(identical_percentage)
        elif '_S_4_' in csv_file:
            identical_percentages['S4'].append(identical_percentage)
        elif '_I_1_' in csv_file:
            identical_percentages['I1'].append(identical_percentage)
        elif '_I_2_' in csv_file:
            identical_percentages['I2'].append(identical_percentage)
        elif '_I_3_' in csv_file:
            identical_percentages['I3'].append(identical_percentage)
        elif '_I_4_' in csv_file:
            identical_percentages['I4'].append(identical_percentage)
    else:
        print(f"'maxforce' or 'result' column missing in {csv_file}")

# S1, S2, S3, S4, I1, I2, I3, I4의 평균 계산
average_S1 = sum(identical_percentages['S1']) / len(identical_percentages['S1'])
average_S2 = sum(identical_percentages['S2']) / len(identical_percentages['S2'])
average_S3 = sum(identical_percentages['S3']) / len(identical_percentages['S3'])
average_S4 = sum(identical_percentages['S4']) / len(identical_percentages['S4'])

average_I1 = sum(identical_percentages['I1']) / len(identical_percentages['I1'])
average_I2 = sum(identical_percentages['I2']) / len(identical_percentages['I2'])
average_I3 = sum(identical_percentages['I3']) / len(identical_percentages['I3'])
average_I4 = sum(identical_percentages['I4']) / len(identical_percentages['I4'])

# S와 I의 평균을 리스트로 저장
averages_S = [average_S1, average_S2, average_S3, average_S4]
averages_I = [average_I1, average_I2, average_I3, average_I4]

# 두 개의 그래프 생성
fig, axs = plt.subplots(1, 2, figsize=(12, 6))

# S 세션 그래프 (왼쪽) - 파란색
axs[0].plot(['S1', 'S2', 'S3', 'S4'], averages_S, marker='o', color='blue', labe
axs[0].set_title('Average Identical Percentage for S Sessions')
axs[0].set_xlabel('Session')
axs[0].set_ylabel('Identical Percentage (%)')
axs[0].grid(True)
axs[0].legend()

# I 세션 그래프 (오른쪽) - 빨간색
axs[1].plot(['I1', 'I2', 'I3', 'I4'], averages_I, marker='o', color='red', label
axs[1].set_title('Average Identical Percentage for I Sessions')
axs[1].set_xlabel('Session')
axs[1].set_ylabel('Identical Percentage (%)')
axs[1].grid(True)
axs[1].legend()

# 그래프 출력
plt.tight_layout()
plt.show()
```



In [21]:

```python
# 각 횟수별로 identical_percentage 저장 딕셔너리
identical_percentages = {'Session': [], 'Percentage': [], 'Count': []}

# 각 CSV 파일에 대해 작업 수행
for csv_file in csv_files:
    file_path = os.path.join(directory_path, csv_file)

    # CSV 파일 읽기
    df = pd.read_csv(file_path)

    # maxforce와 result 열이 있는지 확인
    if 'maxforce' in df.columns and 'result' in df.columns:
        # maxforce와 result가 동일한 값의 비율 계산
        identical_percentage = (df['maxforce'] == df['result']).mean() * 100

        # 파일 이름에서 S1, S2, S3, S4 또는 I1, I2, I3, I4를 추출하여 데이터 추가
        if '_S_1_' in csv_file:
            identical_percentages['Session'].append('S')
            identical_percentages['Count'].append('1')
        elif '_S_2_' in csv_file:
            identical_percentages['Session'].append('S')
            identical_percentages['Count'].append('2')
        elif '_S_3_' in csv_file:
            identical_percentages['Session'].append('S')
            identical_percentages['Count'].append('3')
        elif '_S_4_' in csv_file:
            identical_percentages['Session'].append('S')
            identical_percentages['Count'].append('4')
        elif '_I_1_' in csv_file:
            identical_percentages['Session'].append('I')
            identical_percentages['Count'].append('1')
        elif '_I_2_' in csv_file:
            identical_percentages['Session'].append('I')
            identical_percentages['Count'].append('2')
        elif '_I_3_' in csv_file:
            identical_percentages['Session'].append('I')
            identical_percentages['Count'].append('3')
        elif '_I_4_' in csv_file:
            identical_percentages['Session'].append('I')
            identical_percentages['Count'].append('4')

        identical_percentages['Percentage'].append(identical_percentage)
```

```
# 데이터프레임으로 변환
df_percentages = pd.DataFrame(identical_percentages)

# Boxplot 생성
plt.figure(figsize=(12, 6))
sns.boxplot(x='Count', y='Percentage', hue='Session', data=df_percentages)

# 그래프 세부 설정
plt.title('Boxplot of Identical Percentage for S and I Sessions (by Count)')
plt.xlabel('Session Count (1, 2, 3, 4)')
plt.ylabel('Identical Percentage (%)')
plt.grid(True)

# 그래프 출력
plt.show()
```

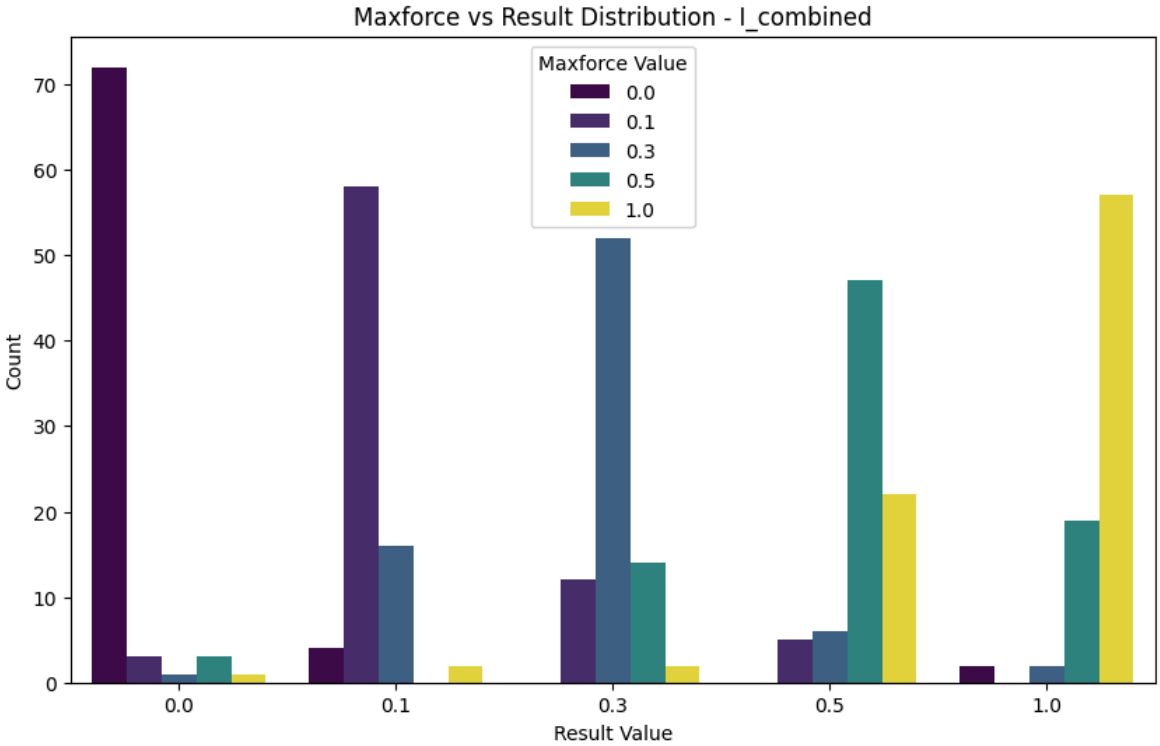


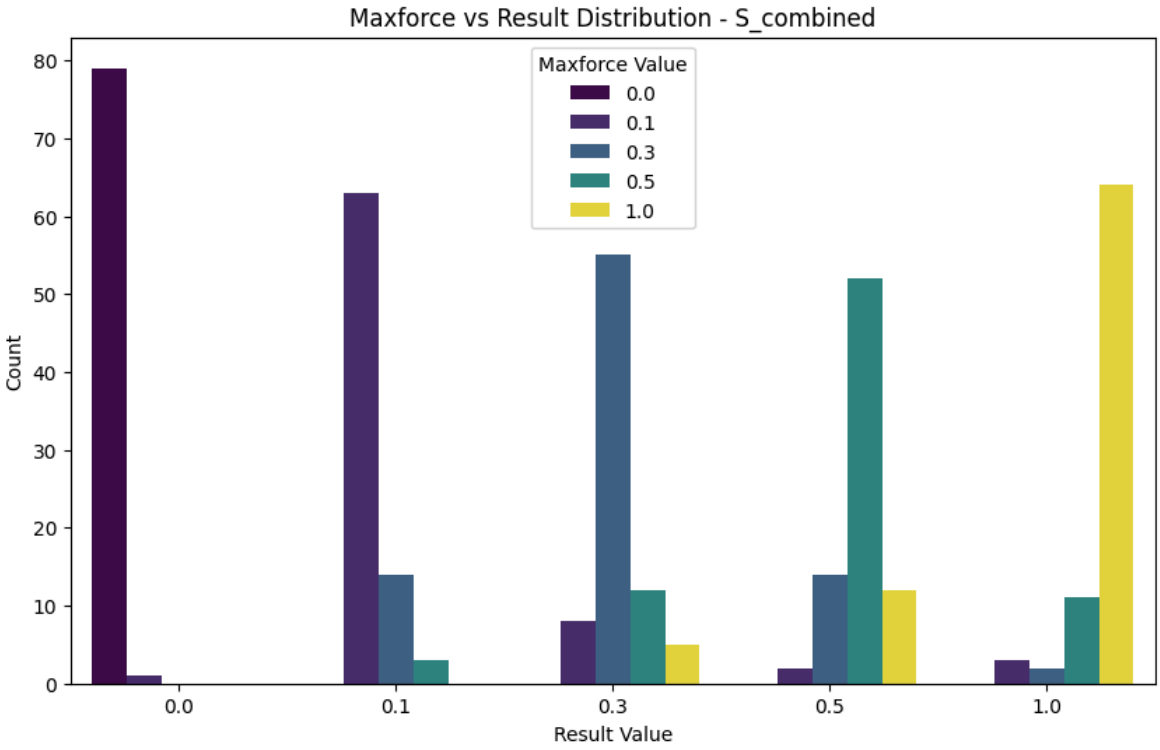
In [ ]:

🔗렌더링별 각 데이터 합산 결과

In [6]:
```
import re

# E_combined와 S_combined 파일 리스트 필터링
i_combined_files = [f for f in os.listdir(directory_path) if re.search(r'I_\d+_c
s_combined_files = [f for f in os.listdir(directory_path) if re.search(r'S_\d+_c

# E_combined와 S_combined 데이터프레임 각각 합치기
df_i_combined = pd.concat([pd.read_csv(os.path.join(directory_path, file)) for f
df_s_combined = pd.concat([pd.read_csv(os.path.join(directory_path, file)) for f

# 데이터프레임을 합친 후 각각 동일한 작업 수행
for df_combined, combined_name in [(df_i_combined, 'I_combined'), (df_s_combined

    # maxforce와 result 열이 있는지 확인
    if 'maxforce' in df_combined.columns and 'result' in df_combined.columns:
        maxforce_values = [0.0, 0.1, 0.3, 0.5, 1.0]
        result_values = [0.0, 0.1, 0.3, 0.5, 1.0]
```

```
        distribution = []

        for maxforce_value in maxforce_values:
            for result_value in result_values:
                count = len(df_combined[(df_combined['maxforce'] == maxforce_val
                distribution.append({
                    'maxforce_value': maxforce_value,
                    'result_value': result_value,
                    'count': count
                })

        distribution_df = pd.DataFrame(distribution)

        # 막대그래프 생성
        plt.figure(figsize=(10, 6))
        sns.barplot(x='result_value', y='count', hue='maxforce_value', data=dist

        # 그래프 세부 설정
        plt.title(f'Maxforce vs Result Distribution - {combined_name}')
        plt.xlabel('Result Value')
        plt.ylabel('Count')
        plt.legend(title='Maxforce Value')

        # 그래프 출력
        plt.show()

    else:
        print(f"'maxforce' or 'result' column missing in {combined_name}")
```





In [7]:
```python
# 결과를 저장할 리스트 초기화
comparison_results = []

# 각 데이터프레임에 대해 작업 수행
for df_combined, combined_name in [(df_i_combined, 'I_combined'), (df_s_combined

    # maxforce와 result 열이 있는지 확인
    if 'maxforce' in df_combined.columns and 'result' in df_combined.columns:
        # 상관계수 계산
        correlation = df_combined['maxforce'].corr(df_combined['result'])

        # 차이의 절대값 평균 계산
        mean_absolute_difference = (df_combined['maxforce'] - df_combined['resul

        # maxforce와 result가 동일한 값의 비율 계산
        identical_percentage = (df_combined['maxforce'] == df_combined['result']

        # 결과 저장
        comparison_results.append({
            'filename': combined_name,
            'correlation': correlation,
            'mean_absolute_difference': mean_absolute_difference,
            'identical_percentage': identical_percentage
        })

    else:
        print(f"'maxforce' or 'result' column missing in {combined_name}")

# 결과를 데이터프레임으로 변환
results_df = pd.DataFrame(comparison_results)

# 결과 출력
print(results_df)
```

	filename	correlation	mean_absolute_difference	identical_percentage
0	I_combined	0.799039	0.1055	71.50
1	S_combined	0.857327	0.0770	78.25

In [ ]: