**#Python Script for processing target labels**

def correct\_spelling\_errors(label):

spell\_checker = SpellChecker(language='en')

corrected\_label = spell\_checker.correction(label)

return corrected\_label.lower()

df['target\_labels2'] = df['target\_labels'].apply(correct\_spelling\_errors)

y = df["target\_labels2"].tolist()

y = [label.lower() for label in y]

# Create a histogram

plt.figure(figsize=(10, 6))

plt.hist(y, bins='auto', color='teal', edgecolor='black')

# Add axis labels

plt.xlabel('Sentiment')

plt.ylabel('Count')

plt.gca().spines['top'].set\_visible(False)

plt.gca().spines['right'].set\_visible(False)

plt.grid(False)

plt.show()

target\_labels = [label.capitalize() for label in df["target\_labels2"].tolist()]

label\_mapping = {"Negative": 1, "Neutral": 0, "Positive": 2}

pred\_labels\_numeric = [label\_mapping[label] for label in pred\_labels\_tb]

target\_numeric = [label\_mapping[label] for label in target\_labels] show()

**#Python script for computing the chi-square test**

from scipy.stats import chi2\_contingency

# Example counts of occurrence for each sentiment class

class\_negative = 1094

class\_positive = 251

class\_neutral = 174

# Create a contingency table

observed\_counts = [[class\_negative, class\_positive, class\_neutral],

[class\_negative, class\_positive, class\_neutral],

[class\_negative,class\_positive, class\_neutral]]

# Perform chi-square test

chi2, p\_value, \_, \_ = chi2\_contingency(observed\_counts)

# Print the results

print("Chi-square statistic:", chi2**)**

print("P-value:", p\_value)

**Python code for data splitting**

from sklearn.model\_selection import train\_test\_split

#assigning text and labels to var

x = df["clean\_tokens2"]

y = df["target\_labels"]

x\_temp, x\_test, y\_temp, y\_test = train\_test\_split(x,

y,

test\_size = 0.2,

train\_size = 0.8,

stratify = y,

random\_state = 50)

x\_train, x\_val, y\_train, y\_val = train\_test\_split(x\_temp,

y\_temp,

test\_size = 0.5,

train\_size = 0.5,

stratify = y\_temp,

random\_state = 50)

# Convert labels to numerical forms

label\_mapping = {'negative': 1, 'neutral': 0, 'positive': 2}

y\_train\_mapped = [label\_mapping[label] for label in y\_train]

y\_val\_mapped = [label\_mapping[label] for label in y\_val]

y\_test\_mapped = [label\_mapping[label] for label in y\_test]

**#Python script for computing the chi-square test**

from scipy.stats import chi2\_contingency

# Example counts of occurrence for each sentiment class

class\_negative = 1094

class\_positive = 251

class\_neutral = 174

# Create a contingency table

observed\_counts = [[class\_negative, class\_positive, class\_neutral],

[class\_negative, class\_positive, class\_neutral],

[class\_negative,class\_positive, class\_neutral]]

# Perform chi-square test

chi2, p\_value, \_, \_ = chi2\_contingency(observed\_counts)

# Print the results

print("Chi-square statistic:", chi2**)**

print("P-value:", p\_value)