Data science project

Topic: Social Media Sentiment Analysis

Name: Ahmed Mohsen Ahmed

ID: 202200061

TKH, Coventry University

School of Computing

Submission date: 27/4/2025

Table of Contents

[Problem Statement (Research Questions): 4](#_Toc196506862)

[Data Description: 5](#_Toc196506863)

[Train.csv, test.csv: 5](#_Toc196506864)

[Data Preprocessing: 6](#_Toc196506865)

[Data Cleaning: 6](#_Toc196506866)

[Handle missing value: 6](#_Toc196506867)

[Handle duplicated comments: 6](#_Toc196506868)

[Removing Outliers: 7](#_Toc196506869)

[Feature Engineering: 7](#_Toc196506870)

[Create Comment length column: 7](#_Toc196506871)

[Scale comment length: 7](#_Toc196506872)

[Categorize the comment length: 8](#_Toc196506873)

[Transforming the sentimental values: 8](#_Toc196506874)

[Analytics approaches: 9](#_Toc196506875)

[Machine Learning Model: 9](#_Toc196506876)

[Hyper Tuning: 10](#_Toc196506877)

Table Figure:

# Problem Statement (Research Questions):

Our aim for this project is to solve some Research Question that will answer some question that about elections of 2024 that requires us to collect a number of data that can answer such questions. First question asks the question if we can create a model that can predict comment sentimental value then we want know if comment length has any effect on the sentimental values and for last question.

So research Questions are:

1. Can we create an ML model to predict sentimental values of comments?
2. Is there any correlation between comment length and sentiment values?

# Data Description:

The dataset was collected from kaggle and further data was scraped from reddit comments from the r/Election2024 by using the reddit API.

For our data we have two files Train.csv and Test.csv which are used for our model that is originally split from the original dataset 75% train and 25% test.

## Train.csv, test.csv:

Both Train and Test .csv both have same columns:

Tweet\_id: is id for the every tweet and its auto-increment.

User\_handle: username of user who send the comment.

Comment: The comments itself.

Sentiment (Target): is value of sentimental value of comment ranging from -1 to 1 (-1= Negative, 0= Neutral, 1= Positive).

Comment length: is the length of comment in words.

Scaled\_length: is a Scaled version of the comments to be easier on the model.

Length Category: is Category of length that less than 50 is short and between 50 and 150 is medium and anything above 150 is long.

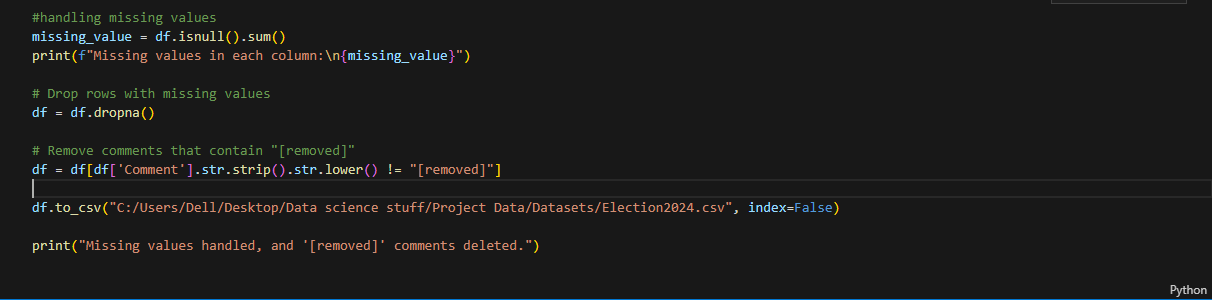
# Data Preprocessing:

Couple steps were taken to prepare data before using it in the model:

## Data Cleaning:

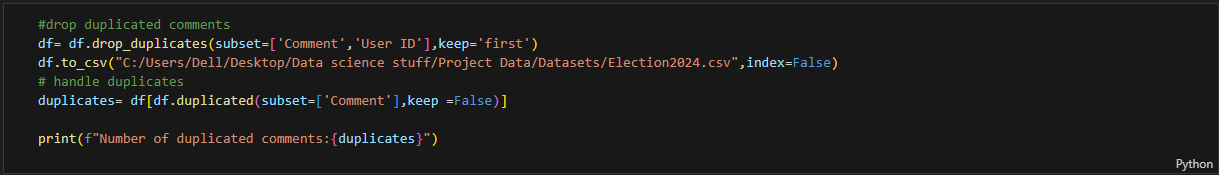
### Handle missing value:

We handle missing value by drops all the missing rows and also removing any comment that was removed by reddit



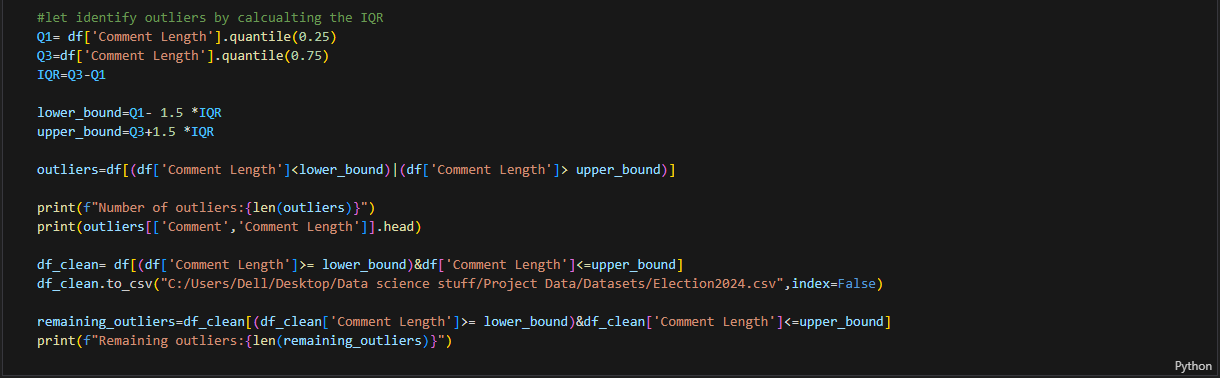
### Handle duplicated comments:

By using pandas we just use drop\_duplicates



### Removing Outliers:

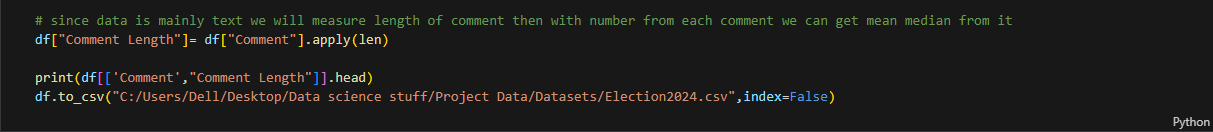
We calculate IQR to find values that are outside of normal range to remove



## Feature Engineering:

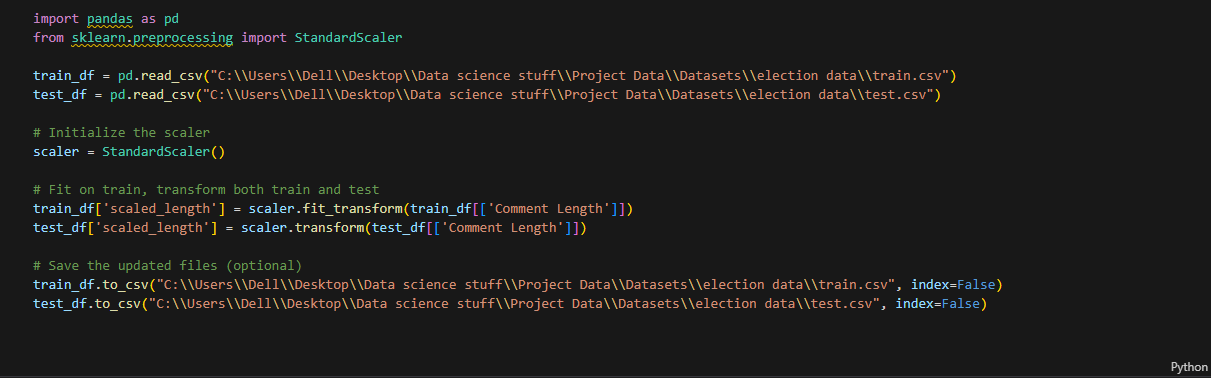
### Create Comment length column:

We wanted to count the length of each comment so we can use it in answering one of our research questions



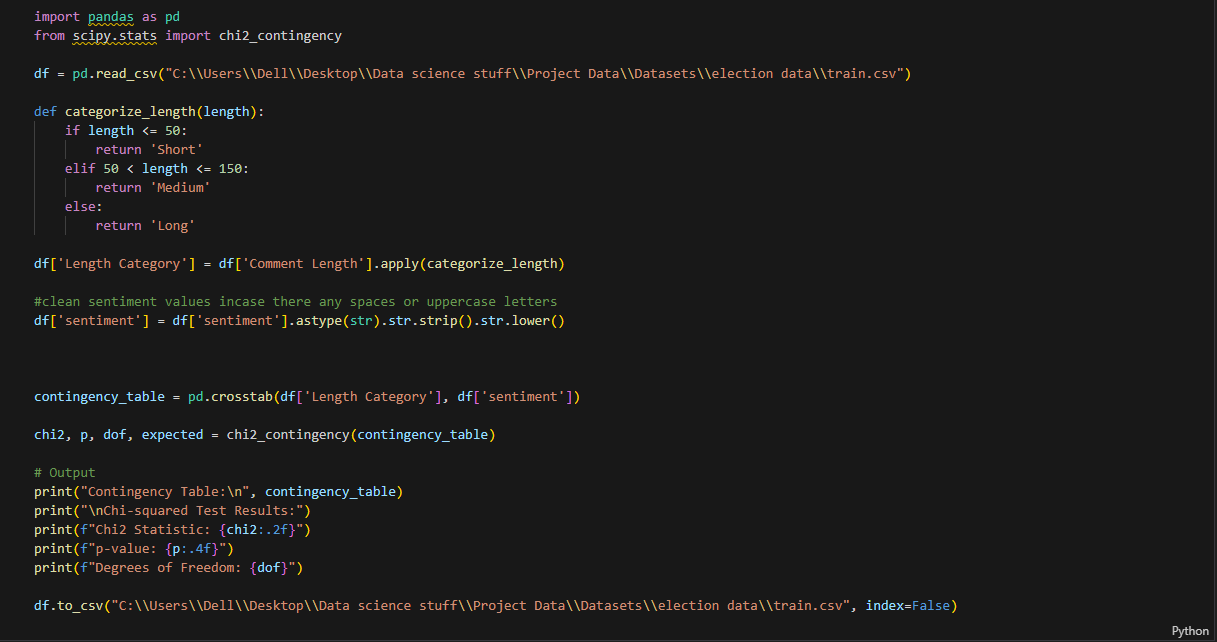
### Scale comment length:

We scale the comment length so we want use it in our model



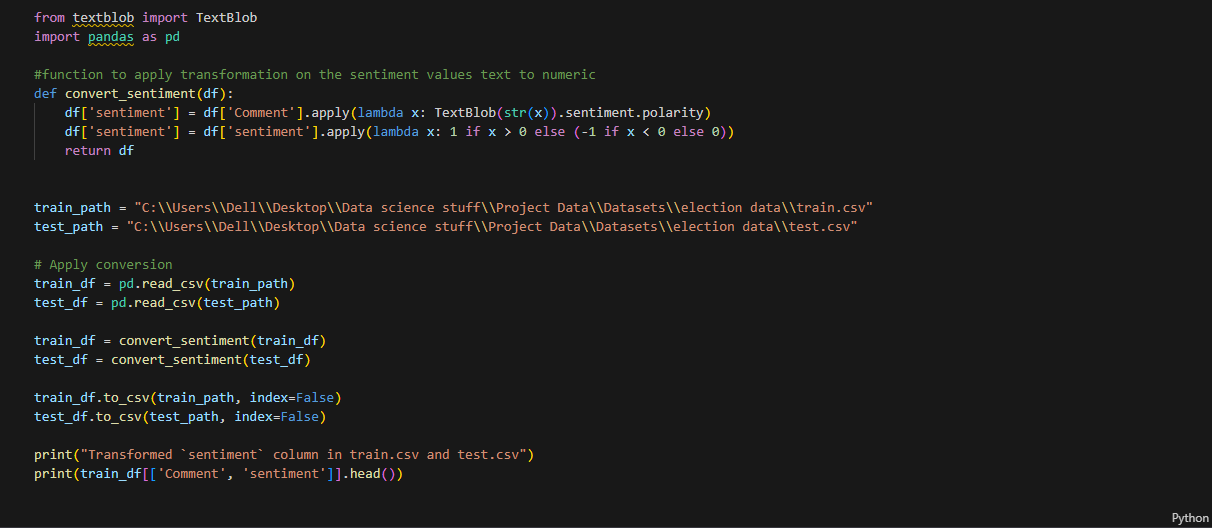
### Categorize the comment length:

While we are doing the Chi squared test we categorize the comment length to make it easier when we answer our question:



### Transforming the sentimental values:

We transform the sentimental value from text to numeric



# Analytics approaches:

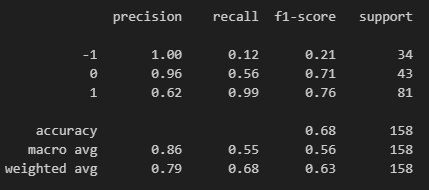
## Machine Learning Model:

For our model we decided to go for Logistic Regression classifier + TF-IDF (Term frequency-Inverse Document Frequency) which is a great model for classification for this problem.

TF-IDF: converts each comment to a numeric vector for how often each word is repeated and common words are given lower weight while the more unique are given a bigger weight

Logistic Regression: takes the converted numeric vector it classifies them into our sentimental categories (Positive, Neutral, and Negative).

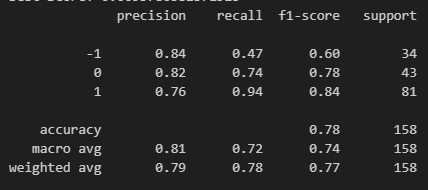
We fit model with the train data and then we give test data x field only so it can predict the target here was result after predictions:



We can see that all predicted negative from the model was correct but he did not get all the negative comments right due to low recall meaning it missed some negative comments but that is due cause most of the data is positive I also predicted most neutral values but didn’t get them all and positive guess some and it catch almost all the positive comment so model is really good at finding positive comments.

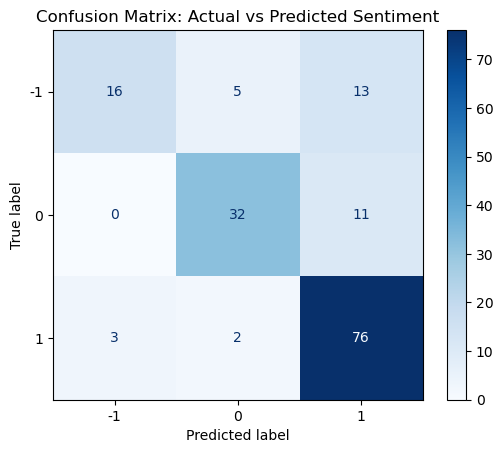
## Hyper Tuning:

After hyper tuning our model using the Grid search method:

Here result after hyper tuning:  


The most noticeable change is the accuracy has increased by 10% while precision for negative comments got was less the recall got higher which is better and same for neutral comment the recall on positive got little worst but the precision got a lot better.

So we were able to create a ML model that can predict sentiment value of comment given and here a visualization of the performance of the model:



This a confusion matrix that shows what the true sentimental value vs what was predicted and tells you how many was right.