Indian Institute of Information Technology Surat

Lab Report on
Machine Learning (CS 601) Practical

Submitted by

[RAHUL KUMAR SINGH] (UI21CS44)

Course Faculty

Dr. Pradeep Kumar Roy

Dr. Rajesh K. Ahir

Department of Computer Science and Engineering

Indian Institute of Information Technology Surat

Gujarat-394190, India

Jan-2024

Table of Contents

Lab No: 1

Aim:
Data Collection from E-Commerce, Twitter and Similar Platforms

Description:
Write a Python script for:

(a) Collecting tweets that may incorporate owner, date of post, number of retweet, number of followers, no of followers , and other associated information from Twitter and store it into a .csv file. (The size of collected tweets >5000)

(b) To scrap users reviews from any E-commerce or similar portals (Ex- Amazon, Flipkart, Yelp) and store it into a csv file that may incorporate date of post, number of likes/dislikes, reviews, location, and other associated fields (The size of collected reviews>5000).

Source Code:

For Task (a):

```
# ---------------------------------------------------------------------------------
from selenium import webdriver
from selenium.webdriver.common.by import By
from fake_useragent import UserAgent
from webdriver_manager.firefox import GeckoDriverManager
import time
import json
import os
from selenium.webdriver.common.keys import Keys

MY_USERNAME_VAR = os.getenv('USERNAME')
MY_PASS_VAR = os.getenv('PASS')
def wait_for_window(self, timeout = 2):
    time.sleep(round(timeout / 1000))
    wh_now = self.driver.window_handles
    wh_then = self.vars["window_handles"]
    if len(wh_now) > len(wh_then):
        return set(wh_now).difference(set(wh_then)).pop()
keywords = ["WWE","Rock","RomanReigns"]
ulrs = []
options = webdriver.FirefoxOptions()
options.headless = False
ua = UserAgent()
userAgent = ua.random
```

```
options.add_argument(f'user-agent={userAgent}')

driver =
webdriver.Firefox(executable_path=GeckoDriverManager().install(),options=options)
driver.get("https://twitter.com/i/flow/login")
driver.maximize_window()
time.sleep(10)
try:
    input_element = driver.find_element(By.CSS_SELECTOR,
'.r-30o5oe.r-1niwhzg.r-17gur6a.r-1yadl64.r-deolkf.r-homxoj.r-poiln3')
    input_element.click()
    time.sleep(5)
    password_x = driver.find_element(By.CSS_SELECTOR,
'.r-30o5oe.r-1niwhzg.r-17gur6a.r-1yadl64.r-deolkf.r-homxoj.r-poiln3.r-7cikom.r-1ny4l
3l.r-t60dpp.r-1dz5y72.r-fdjqy7.r-13qz1uu')
    password_x.click()
    password_x.send_keys(MY_PASS_VAR)
    time.sleep(5)
    with open('keyword_numbers.json', 'w') as file:
        json.dump(keyword_numbers, file)

except Exception as e:
    print(ulrs)
    print("An error occurred:", str(e))
```

For Task (b):

```
import csv
from selenium import webdriver
from selenium.webdriver.common.by import By
import time

def extract_reviews(product_url, num_reviews_to_scrape=10):
    driver = webdriver.Chrome()
    driver.get(product_url)
    time.sleep(8)
    reviews = []
    review_elements = driver.find_elements(By.CSS_SELECTOR, '.a-section.review')
    temp_Date = ""
    for review_element in review_elements[:num_reviews_to_scrape]:
        time.sleep(1)
        review = {}
        review['author'] = review_element.find_element(By.CSS_SELECTOR,
```

```
'.a-profile-name').text.strip()
        temp_Date = review_element.find_element(By.CSS_SELECTOR,
'.review-date').text.strip()
        review['date'] = temp_Date[temp_Date.find('on')+3:]
        review['location'] = temp_Date[12:temp_Date.find('on')-1]
        review['text'] = review_element.find_element(By.CSS_SELECTOR,
'.review-text-content').text.strip()
        review['rating'] =
review_element.find_element_by_xpath('//i[@data-hook="review-star-rating"]').text.st
rip()
        review['title'] = review_element.find_element(By.CSS_SELECTOR,
'.review-title').text.strip()
        reviews.append(review)
        print(review)
    driver.quit()
    return reviews
product_url =
'https://www.amazon.in/ZAPCASE-Compatible-Xiaomi-Covers-Carbon/product-reviews/B07GQ
Y2RN2/ref=cm_cr_arp_d_paging_btm_next_2?ie=UTF8&reviewerType=all_reviews'
reviews_data = []
for i in range(1,4):
    reviews_data += extract_reviews(product_url+'&pageNumber='+str(i),
num_reviews_to_scrape=10)
def export_csv(reviews, csv_filename='reviews_data.csv'):
    with open(csv_filename, 'w', newline='', encoding='utf-8') as csv_file:
        fieldnames = ['date','names','location','reviewtitles','ratings','reviews']
        writer = csv.DictWriter(csv_file, fieldnames=fieldnames)

        writer.writeheader()
        for review in reviews:
            writer.writerow({'date': review['date'], 'names': review['author'],
'location': review['location'], 'reviewtitles': review['title'], 'ratings':
review['rating'], 'reviews': review['text']})

export_csv(reviews_data)
```

Output:

For Task (a):

	link	text	name	username	date	is_rt	n_comment	n_rt	n_quote	n_like
2	https://twitter.com/WWE/status/17420177	IF YA SMELL..... @TheRock has come back to #WWERaw!	WWE	@WWE	Jan 2, 2024 · 2:5	FALSE	1087	9916	1856	49991
3	https://twitter.com/WWE/status/17457398	Who had the best Instagram photo of the week?! https://www.wwe.co	WWE	@WWE	Jan 12, 2024 · 9	FALSE	42	29	2	252
4	https://twitter.com/WWE/status/17456794	These #RoyalRumble crashers were RUTHLESS! https://youtu.be/V\	WWE	@WWE	Jan 12, 2024 · 5	FALSE	45	65	2	566
5	https://twitter.com/WWE/status/17456419	An All Mighty moment in the 2023 Men's #RoyalRumble Match!	WWE	@WWE	Jan 12, 2024 · 3	FALSE	57	193	13	2233
6	https://twitter.com/WWE/status/17455966	Outta nowhere! 😲	WWE	@WWE	Jan 12, 2024 · 1	FALSE	69	340	10	3655
7	https://twitter.com/ShawnMichaels/status/	Thanks for stopping by the WWE Performance Center @dkelce1!	Shawn Michaels	@ShawnMichae	Jan 11, 2024 · 9	TRUE	49	209	7	2301
8	https://twitter.com/WWE/status/17455524	The Tribal Chief is unfazed 🤣 @WWERomanReigns @HeymanHus	WWE	@WWE	Jan 11, 2024 · 9	FALSE	119	502	26	4421
9	https://twitter.com/WWENXT/status/1745	It was great to have @dkelce1 come out and visit the WWE Performa	WWE NXT	@WWENXT	Jan 11, 2024 · 8	TRUE	165	360	276	3575
10	https://twitter.com/WWE/status/17455426	Attention @NHLFlyers fans! @GrittyNHL has the #WWEGoldenTitle /	WWE	@WWE	Jan 11, 2024 · 8	FALSE	50	138	12	991
11	https://twitter.com/WWENXT/status/1745	The #DustyClassic continues NEXT WEEK on #WWENXT with these	WWE NXT	@WWENXT	Jan 11, 2024 · 8	TRUE	36	150	5	857
12	https://twitter.com/WWE/status/17455222	Available NOW on @WWEShop! Get these all-new shirts featuring R	WWE	@WWE	Jan 11, 2024 · 7	FALSE	62	198	12	1613
13	https://twitter.com/WWENXT/status/1745	Take a look at @tiffstrattonwwe's eventful day as @FallonHenleyWWE	WWE NXT	@WWENXT	Jan 11, 2024 · 4	TRUE	41	180	21	1068
14	https://twitter.com/WWE/status/17454907	Enjoy a cold one on #NationalMilkDay just like @TheRock & @RealK	WWE	@WWE	Jan 11, 2024 · 5	FALSE	41	184	5	1341
15	https://twitter.com/WWE/status/17454819	Happy #NationalMilkDay!	WWE	@WWE	Jan 11, 2024 · 4	FALSE	105	524	103	4258
16	https://twitter.com/NXTLevelUp/status/17	Can Big Body Javi upset @JoeGacy tomorrow night on #NXTLevelUp	NXT Level Up	@NXTLevelUp	Jan 11, 2024 · 4	TRUE	40	66	5	384
17	https://twitter.com/WWETheBump/status/	Thanks for joining us on #WWETheBump, @MiaYim! 2024 is shaping	WWE's The Bun	@WWETheBum	Jan 11, 2024 · 3	TRUE	18	73	1	422
18	https://twitter.com/WWEShop/status/1745	Live, Laugh, Love! R-Truth & The Judgment Day have 2 new tees ava	WWEShop.com	@WWEShop	Jan 11, 2024 · 1	TRUE	141	541	224	4118
19	https://twitter.com/WWE/status/17454656	They weren't supposed to be in the #RoyalRumble match! What happ	WWE	@WWE	Jan 11, 2024 · 3	FALSE	28	102	2	691
20	https://twitter.com/WWE/status/17453775	👀	WWE	@WWE	Jan 11, 2024 · 9	FALSE	104	382	19	5927
21	https://twitter.com/WWE/status/17453171	Message sent. #SmackDown	WWE	@WWE	Jan 11, 2024 · 5	FALSE	70	321	7	5293
22	https://twitter.com/WWE/status/17452491	ICYMI: Watch R-Truth's lovely tribute to The Judgment Day - https:/	WWE	@WWE	Jan 11, 2024 · 1	FALSE	59	441	8	3880
23	https://twitter.com/DMcIntyreWWE/status	Say it to my chest	Drew	@DMcIntyreWV	Jan 10, 2024 · 1	TRUE	322	723	146	14094

Figure 1.1 Output for Twitter Data Collection

For Task (b):

	date	names	location	reviewtitles	ratings	reviews
2	14 November 20	Vishal S Prabhu	India	Good Deal	5.0 out of 5 stars	Good product. Fits well. Overall, a good deal.
3	18 July 2023	Kartik Jain	India	Nice cover	4.0 out of 5 stars	It's good you can try it.
4	22 October 2022	Kedar	India	Good cover and	5.0 out of 5 stars	Good fit. Effective price.
5	26 May 2023	Placeholder	India	Just okay....	3.0 out of 5 stars	The product is overall good at this price. It could have been better. The build quality - 3/5 stars. Handling and comfort - 4/5 stars. The fitting to the phone - 2/5 stars... Overall the product is good if you want the best cover in this budget. But if you want the perfect and the best cover for your Mi A2 you can opt to buy another cover at a higher price...
6	30 March 2023	Saleem Shah	India	Best quality	5.0 out of 5 stars	This is one of the best quality. Go for it
7	27 December 20	Yashwanth Redc	India	Good	4.0 out of 5 stars	Good
8	19 March 2023	Alfaz Jikani	India	Fit	5.0 out of 5 stars	Fit to Mi A2 as expected.
9	2 June 2022	joel	India	It doesn't comple	1.0 out of 5 stars	The product doesn't protect the phone completely the borders are not thick which protects the phone from a fall there is no point of taking this case if it doesn't provi
10	30 September 2	Placeholder	India	Money worthy	3.0 out of 5 stars	Not bad
11	7 January 2022	Keshav chandra	India	Ok for the price	4.0 out of 5 stars	Flexible silicon type rubber , does the job. Perfect fit for the Mi A2. Overall recommended. I dont think it will protect the phone from a fall but for sure adds grip
12	26 April 2022	Amazon Custom	India	Good Product	4.0 out of 5 stars	The product seems to be good.But the price is too much.It could have been a little less.
13	17 February 202	vijay madrecha	India	Looks decent, bu	3.0 out of 5 stars	Fitting and built quality is good, but I am skeptical if it is genuine Zapcase cover or not. The paper box has Zapcase mentioned on it, but the inner transparent cover
14	8 November 202	Amaresh Swain	India	Average	3.0 out of 5 stars	Finishing is average, material good
15	13 May 2021	nilotpal	India	Good not great.	4.0 out of 5 stars	Finish of the product is not great. It's 'okay'. I would have given 5star if this was below 100rs. But for 150rs, this is a good product. Should save the phone from minc
16	21 March 2021	Abhiram	India	Worth it.	4.0 out of 5 stars	I've liked this product very much. Its is completely same as given in image. It feels so good while holding the phone.
17	14 December 20	Atul Sharma	India	Exact fit..	4.0 out of 5 stars	Exact fit to phone..

Figure 1.2 Output for Amazon Review Scraping

Conclusion:

- Efficient and direct access to Twitter's data through the API.
- Provides real-time data retrieval, enabling instant updates.
- Offers structured data in JSON format for easy processing.
- Overcomes API limitations for certain tasks, such as scraping dynamic content using custom scraping.

Lab No: 2

Aim:
To perform exploratory data analysis on the attached dataset

Description:
Perform the Exploratory Data Analysis (EDA) by considering the following tasks. Use the attached dataset for the same.

1. Check for Duplication
2. Missing Values Calculation
3. Data Reduction (Some columns or variables can be dropped if they do not add value to our analysis.)
4. Feature Engineering
5. Creating Features
6. Data Cleaning/Wrangling
7. Statistics Summary (Count, Mean, Standard Deviation, median, mode, minimum value, maximum value, range, standard deviation)
8. Analyzing/visualizing the dataset by taking one variable at a time
9. Data Transformation

Source Code:

Import Libraries and Read Dataset

```
import pandas as pd
from sklearn.decomposition import PCA
from sklearn.preprocessing import StandardScaler, LabelEncoder
import matplotlib.pyplot as plt
import seaborn as sns
from datetime import datetime

df = pd.read_csv('cars_data.csv')
print(df.head())
```

1. Check for Duplication

```
duplicates = df.duplicated()
print(df[duplicates])
```

```
num_duplicates = df.duplicated().sum()
percentage_duplicates = (num_duplicates / len(df)) * 100

print(f"Number of duplicate rows: {num_duplicates}")
print(f"Percentage of duplicate rows: {percentage_duplicates:.2f}%")
```

```
df = df.drop_duplicates()
print(df.head())
```

2. Missing Values Calculation

```
total_missing = df.isnull().sum().sum()
print(total_missing)
```

```
missing_by_column = df.isnull().sum()
print(missing_by_column)
```

```
percentage_missing = (df.isnull().sum() / len(df)) * 100
print(percentage_missing)
```

3. Data Reduction (Some columns or variables can be dropped if they do not add value to our analysis.)

[ ]:
```
# Replace missing values
df['Price'].fillna(0, inplace=True)
df['New_Price'].fillna(0, inplace=True)
df.dropna(inplace=True) # Dropping few inconsequntial records
```

[ ]:
```
# Drop irrelevant columns for analysis
cols_to_drop = ['Name','Location','Fuel_Type','Transmission','Owner_Type','Mileage','Engine','Power','New_Price']
dropdf = df.drop(columns=cols_to_drop)

scaler = StandardScaler()
cars_data_scaled = scaler.fit_transform(dropdf)

# Apply Principal Component Analysis (PCA) for dimensionality reduction
pca = PCA(n_components=2)
cars_pca = pca.fit_transform(cars_data_scaled)

plt.figure(figsize=(10, 6))
plt.scatter(cars_pca[:, 0], cars_pca[:, 1])
plt.title('PCA: First Two Principal Components')
plt.xlabel('Principal Component 1')
plt.ylabel('Principal Component 2')
plt.show()
```

4. Feature Engineering

[ ]:
```
selected_features = df[['S.No.', 'Kilometers_Driven', 'Seats', 'Price']]

scaler = StandardScaler()
scaled_features = scaler.fit_transform(selected_features)

num_components = 2
pca = PCA(n_components=num_components)
reduced_features = pca.fit_transform(scaled_features)
reduced_features_df = pd.DataFrame(data=reduced_features, columns=['PC1', 'PC2'])
final_data = pd.concat([df, reduced_features_df], axis=1)
print(final_data.head())
```

5. Creating Features

[ ]:
```
cars_df = df.copy()
cars_df['Brand'] = cars_df['Name'].str.split().str[0]
cars_df['Mileage'] = cars_df['Mileage'].str.split().str[0]
cars_df['Mileage'] = pd.to_numeric(df['Mileage'], errors='coerce')
current_year = datetime.now().year
cars_df['Age'] = current_year - cars_df['Year']
cars_df['Price_per_Mile'] = cars_df['Price'] / cars_df['Mileage']

print("\nCars Dataset with New Features:")
print(cars_df)
```

[ ]:
```
# Visualization
plt.figure(figsize=(12, 6))

plt.subplot(1, 2, 1)
sns.scatterplot(x='Age', y='Price', data=cars_df, hue='Brand', palette='Set1')
plt.title('Age vs Price')

plt.subplot(1, 2, 2)
sns.scatterplot(x='Mileage', y='Price_per_Mile', data=cars_df, hue='Brand', palette='Set2')
plt.title('Mileage vs Price_per_Mile')

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

6. Data Cleaning/Wrangling

```
[ ]: clean_df = df.copy()

clean_df['Brand'] = clean_df['Name'].str.split().str[0]
clean_df['Engine'] = clean_df['Engine'].str.extract('(\d+)').astype(float)
clean_df['Mileage'] = clean_df['Mileage'].str.extract('(\d+)').astype(float)
clean_df['Power'] = clean_df['Power'].str.extract('(\d+)').astype(float)
clean_df['New_Price'] = clean_df['New_Price'].str.extract('(\d+)').astype(float)
clean_df['New_Price'].fillna(0, inplace=True)
current_year = datetime.now().year
clean_df['Mileage'][clean_df['Mileage']==0] = 1
clean_df['Age'] = current_year - clean_df['Year']
clean_df['Price_per_Mile'] = clean_df['Price'] / clean_df['Mileage']
clean_df = clean_df.drop(['Name', 'Year'], axis=1)

print("\nCleaned and Wrangled Dataset:")
print(clean_df)
```

7. Statistics Summary (Count, Mean, Standard Deviation, median, mode, minimum value, maximum value, range, standard deviation)

```
[ ]: print("Dataset Information:")
print(df.info())

print("\nSummary Statistics:")
print(df.describe())
```

```
[ ]: cars_data = dropdf.copy()
summary_stats = {
    'Count': cars_data.shape[0],
    'Mean': cars_data.mean(),
    'Standard Deviation': cars_data.std(),
    'Median': cars_data.median(),
    'Mode': cars_data.mode().iloc[0],
    'Minimum Value': cars_data.min(),
    'Maximum Value': cars_data.max(),
    'Range': cars_data.max() - cars_data.min(),
}
summary_df = pd.DataFrame(summary_stats)

print("\nStatistics Summary:")
print(summary_df)
```

```
[ ]: import seaborn as sns
import matplotlib.pyplot as plt

sns.heatmap(df.isnull(), cbar=False, cmap='viridis')
plt.show()
```

8. Analyzing/visualizing the dataset by taking one variable at a time

```
[ ]: cars_data = clean_df.copy()
     def visualize_variable(variable_name):
         plt.figure(figsize=(8, 6))
         plt.hist(cars_data[variable_name], bins=20, color='skyblue', edgecolor='black')
         plt.title(f'Distribution of {variable_name}')
         plt.xlabel(variable_name)
         plt.ylabel('Frequency')
         plt.show()

     numerical_variables = cars_data.select_dtypes(include='number').columns
     for variable in numerical_variables:
         visualize_variable(variable)
```

9. Data Transformation

```
[ ]: cars_data = clean_df.copy()
     # Encode Categorical Variables
     label_encoder = LabelEncoder()
     cars_data['Brand'] = label_encoder.fit_transform(cars_data['Brand'])
     cars_data['Fuel_Type'] = label_encoder.fit_transform(cars_data['Fuel_Type'])
     cars_data['Transmission'] = label_encoder.fit_transform(cars_data['Transmission'])

     # Scale Numerical Features
     numerical_features = ['Price', 'Mileage', 'Engine']
     scaler = StandardScaler()
     cars_data[numerical_features] = scaler.fit_transform(cars_data[numerical_features])

     print("\nTransformed Dataset:")
     print(cars_data.head())
```

Output:

1. Check for Duplication

1. Check for Duplication

```
[2]: duplicates = df.duplicated()
     print(df[duplicates])

     Empty DataFrame
     Columns: [S.No., Name, Location, Year, Kilometers_Driven, Fuel_Type, Transmission, Owner_Type, Mileage, Engine, Power, Seats, New_Price, Price]
     Index: []

[3]: num_duplicates = df.duplicated().sum()
     percentage_duplicates = (num_duplicates / len(df)) * 100

     print(f"Number of duplicate rows: {num_duplicates}")
     print(f"Percentage of duplicate rows: {percentage_duplicates:.2f}%")

     Number of duplicate rows: 0
     Percentage of duplicate rows: 0.00%
```

Figure 2.1 Output for task 1

2. Missing Values Calculation

```
2. Missing Values Calculation

[5]: total_missing = df.isnull().sum().sum()
     print(total_missing)

7636

[6]: missing_by_column = df.isnull().sum()
     print(missing_by_column)

S.No.                  0
Name                   0
Location               0
Year                   1
Kilometers_Driven      1
Fuel_Type              2
Transmission           1
Owner_Type             2
Mileage                3
Engine                46
Power                 46
Seats                 53
New_Price           6247
Price               1234
dtype: int64
```

Figure 2.2 Output for task 2

3. Data Reduction (Some columns or variables can be dropped if they do not add value to our analysis.)

```
      S.No.    Year  Kilometers_Driven  Seats  Price
0         0  2010.0            72000.0    5.0   1.75
1         1  2015.0            41000.0    5.0  12.50
2         2  2011.0            46000.0    5.0   4.50
3         3  2012.0            87000.0    7.0   6.00
4         4  2013.0            40670.0    5.0  17.74
...     ...     ...                ...    ...    ...
7248   7248  2011.0            89411.0    5.0   0.00
7249   7249  2015.0            59000.0    5.0   0.00
7250   7250  2012.0            28000.0    5.0   0.00
7251   7251  2013.0            52262.0    5.0   0.00
7252   7252  2014.0            72443.0    5.0   0.00
```

Figure 2.3 Output for task 3

4. Feature Engineering

```
   S.No.                              Name    Location    Year  \
0    0.0            Maruti Wagon R LXI CNG      Mumbai  2010.0
1    1.0  Hyundai Creta 1.6 CRDi SX Option        Pune  2015.0
2    2.0                      Honda Jazz V     Chennai  2011.0
3    3.0                 Maruti Ertiga VDI     Chennai  2012.0
4    4.0   Audi A4 New 2.0 TDI Multitronic  Coimbatore  2013.0

   Kilometers_Driven Fuel_Type Transmission Owner_Type     Mileage   Engine  \
0            72000.0       CNG       Manual      First  26.6 km/kg   998 CC
1            41000.0    Diesel       Manual      First  19.67 kmpl  1582 CC
2            46000.0    Petrol       Manual      First   18.2 kmpl  1199 CC
3            87000.0    Diesel       Manual      First  20.77 kmpl  1248 CC
4            40670.0    Diesel    Automatic     Second   15.2 kmpl  1968 CC

       Power  Seats  New_Price  Price       PC1       PC2
0  58.16 bhp    5.0          0   1.75  0.746503 -0.304137
1  126.2 bhp    5.0          0  12.50  1.421956 -0.656362
2   88.7 bhp    5.0  8.61 Lakh   4.50  0.906441 -0.545824
3  88.76 bhp    7.0          0   6.00  1.453131  1.470690
4  140.8 bhp    5.0          0  17.74  1.760415 -0.703806
```

Figure 2.4 Output for task 4

5. Creating Features

```
2       2                                 Honda Jazz V     Chennai
3       3                           Maruti Ertiga VDI     Chennai
4       4              Audi A4 New 2.0 TDI Multitronic  Coimbatore
...   ...                                         ...         ...
7248  7248             Volkswagen Vento Diesel Trendline   Hyderabad
7249  7249                        Volkswagen Polo GT TSI      Mumbai
7250  7250                         Nissan Micra Diesel XV     Kolkata
7251  7251                        Volkswagen Polo GT TSI        Pune
7252  7252  Mercedes-Benz E-Class 2009-2013 E 220 CDI Avan...       Kochi

        Year  Kilometers_Driven Fuel_Type Transmission Owner_Type  Mileage  \
0     2010.0            72000.0       CNG       Manual      First      NaN
1     2015.0            41000.0    Diesel       Manual      First      NaN
2     2011.0            46000.0    Petrol       Manual      First      NaN
3     2012.0            87000.0    Diesel       Manual      First      NaN
4     2013.0            40670.0    Diesel    Automatic     Second      NaN
...      ...                ...       ...          ...        ...      ...
7248  2011.0            89411.0    Diesel       Manual      First      NaN
7249  2015.0            59000.0    Petrol    Automatic      First      NaN
7250  2012.0            28000.0    Diesel       Manual      First      NaN
7251  2013.0            52262.0    Petrol    Automatic      Third      NaN
7252  2014.0            72443.0    Diesel    Automatic      First      NaN

       Engine      Power  Seats  New_Price  Price          Brand   Age  \
0      998 CC  58.16 bhp    5.0          0   1.75         Maruti  14.0
1     1582 CC  126.2 bhp    5.0          0  12.50        Hyundai   9.0
2     1199 CC   88.7 bhp    5.0  8.61 Lakh   4.50          Honda  13.0
3     1248 CC  88.76 bhp    7.0          0   6.00         Maruti  12.0
4     1968 CC  140.8 bhp    5.0          0  17.74           Audi  11.0
...       ...        ...    ...        ...    ...            ...   ...
7248  1598 CC  103.6 bhp    5.0          0   0.00     Volkswagen  13.0
7249  1197 CC  103.6 bhp    5.0          0   0.00     Volkswagen   9.0
7250  1461 CC   63.1 bhp    5.0          0   0.00         Nissan  12.0
7251  1197 CC  103.6 bhp    5.0          0   0.00     Volkswagen  11.0
7252  2148 CC    170 bhp    5.0          0   0.00  Mercedes-Benz  10.0
```

Figure 2.5.1 Tabular Representation

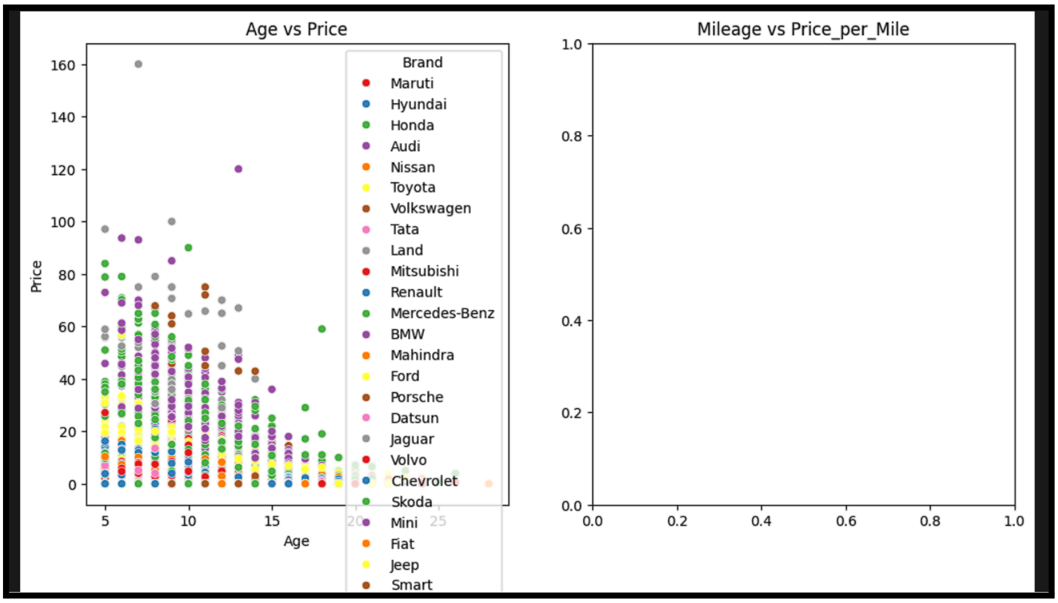

Figure 2.5.2 Graphical Representation

6. Data Cleaning/Wrangling

```
Cleaned and Wrangled Dataset:
      S.No.    Location  Kilometers_Driven Fuel_Type Transmission Owner_Type  \
0         0      Mumbai            72000.0       CNG       Manual      First
1         1        Pune            41000.0    Diesel       Manual      First
2         2     Chennai            46000.0    Petrol       Manual      First
3         3     Chennai            87000.0    Diesel       Manual      First
4         4  Coimbatore            40670.0    Diesel    Automatic     Second
...     ...         ...                ...       ...          ...        ...
7248   7248   Hyderabad            89411.0    Diesel       Manual      First
7249   7249      Mumbai            59000.0    Petrol    Automatic      First
7250   7250     Kolkata            28000.0    Diesel       Manual      First
7251   7251        Pune            52262.0    Petrol    Automatic      Third
7252   7252       Kochi            72443.0    Diesel    Automatic      First

      Mileage  Engine  Power  Seats  New_Price  Price          Brand   Age  \
0        26.0   998.0   58.0    5.0        0.0   1.75         Maruti  14.0
1        19.0  1582.0  126.0    5.0        0.0  12.50        Hyundai   9.0
2        18.0  1199.0   88.0    5.0        8.0   4.50          Honda  13.0
3        20.0  1248.0   88.0    7.0        0.0   6.00         Maruti  12.0
4        15.0  1968.0  140.0    5.0        0.0  17.74           Audi  11.0
...       ...     ...    ...    ...        ...    ...            ...   ...
7248     20.0  1598.0  103.0    5.0        0.0   0.00     Volkswagen  13.0
7249     17.0  1197.0  103.0    5.0        0.0   0.00     Volkswagen   9.0
7250     23.0  1461.0   63.0    5.0        0.0   0.00         Nissan  12.0
7251     17.0  1197.0  103.0    5.0        0.0   0.00     Volkswagen  11.0
7252     10.0  2148.0  170.0    5.0        0.0   0.00  Mercedes-Benz  10.0
```

Figure 2.6 Output for task 6

7. Statistics Summary (Count, Mean, Standard Deviation, median, mode, minimum value, maximum value, range, standard deviation)

```
Statistics Summary:
                   Count          Mean  Standard Deviation   Median     Mode  \
S.No.               7191   3627.190655         2094.568997   3629.0      0.0
Year                7191   2013.391322            3.235169   2014.0   2014.0
Kilometers_Driven   7191  58606.050897        84711.727076  53226.0  60000.0
Seats               7191      5.279516            0.811614      5.0      5.0
Price               7191      7.888618           10.819356      4.7      0.0

                   Minimum Value  Maximum Value      Range
S.No.                        0.0         7252.0     7252.0
Year                      1996.0         2019.0       23.0
Kilometers_Driven          171.0      6500000.0  6499829.0
Seats                        0.0           10.0       10.0
Price                        0.0          160.0      160.0
```

Figure 2.7 Output for task 7

8. Analyzing/visualizing the dataset by taking one variable at a time

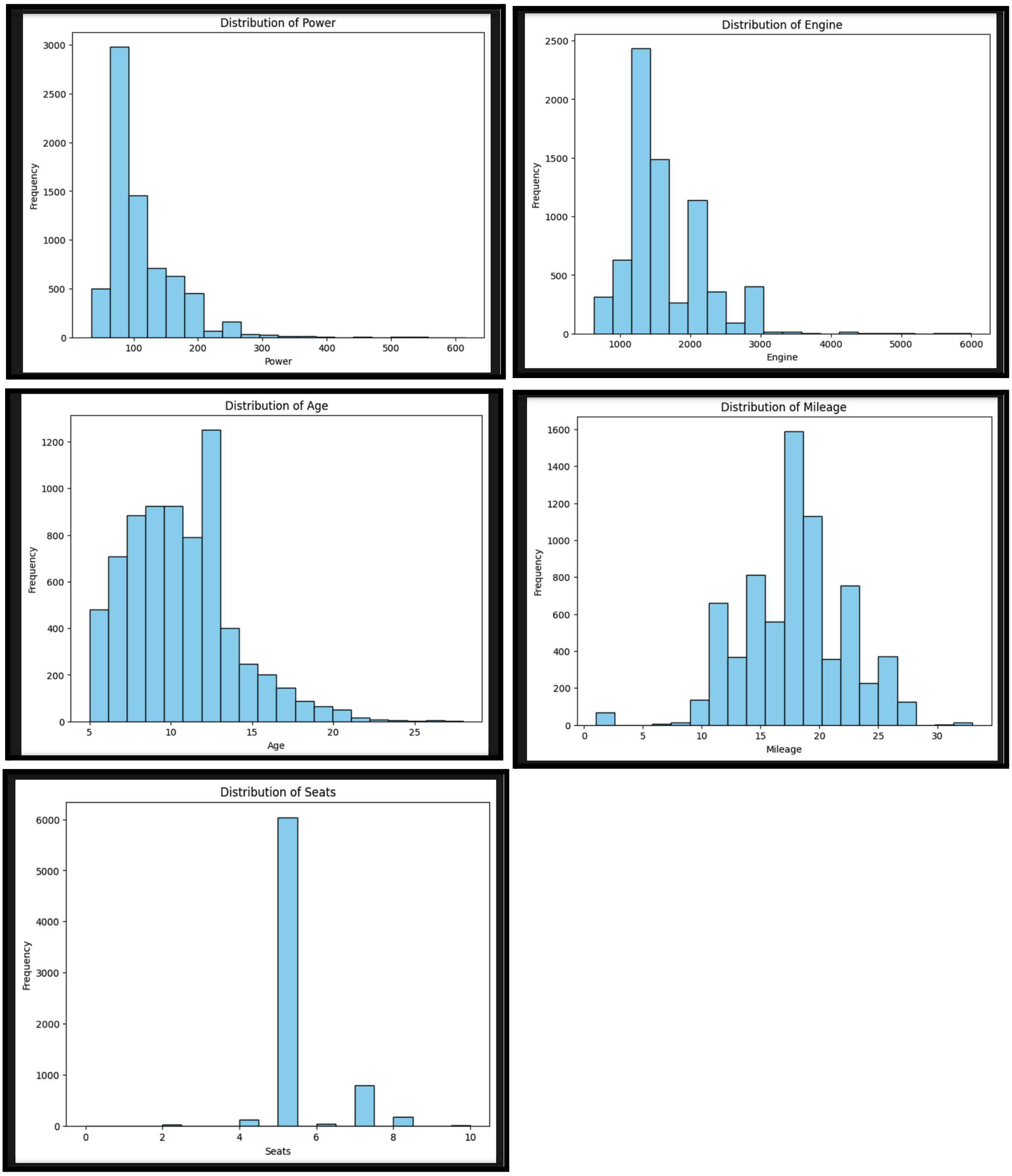

Figure 2.8 Output for task 8

9. Data Transformation

```
Transformed Dataset:
   S.No.    Location  Kilometers_Driven  Fuel_Type  Transmission Owner_Type  \
0      0      Mumbai            72000.0          0             1      First
1      1        Pune            41000.0          1             1      First
2      2     Chennai            46000.0          3             1      First
3      3     Chennai            87000.0          1             1      First
4      4  Coimbatore            40670.0          1             0     Second

    Mileage    Engine  Power  Seats  New_Price     Price  Brand   Age  \
0  1.842662 -1.039810   58.0    5.0        0.0 -0.567413     19  14.0
1  0.275923 -0.058350  126.0    5.0        0.0  0.426246     11   9.0
2  0.052103 -0.702013   88.0    5.0        8.0 -0.313221     10  13.0
3  0.499743 -0.619664   88.0    7.0        0.0 -0.174571     19  12.0
4 -0.619356  0.590354  140.0    5.0        0.0  0.910596      1  11.0

   Price_per_Mile
0        0.067308
1        0.657895
2        0.250000
3        0.300000
4        1.182667
```

Figure 2.9 Output for task 9

Conclusion:

- EDA provides a comprehensive overview of the cars dataset
- Identification and handling of missing values, outliers, and anomalies ensure data integrity and improve analysis accuracy.
- Descriptive statistics, including mean, median, and standard deviation, offer a summary of numerical attributes, aiding in understanding central tendencies and data dispersion.
- Visualization techniques, such as histograms and kernel density plots, reveal the distributions of key features, providing insights into the data's underlying patterns.
- Techniques like correlation, mutual information, or model-based feature importance assessments help prioritize variables based on their impact on the target variable.

Lab No: 3

Aim:

To perform linear regression and utilize Python libraries to plot attribute relations, design optimal line fitting, and analyze global minima for given data.

Description:

Perform the following task with using inbuilt Python Libraries.:

- Plot the input-output relation for given attributes.
- Design a mathematical function to find the best-fitted line for the given data (attached here).
- Plot Error vs. Slope graph and show the global minima for the sample data X={2, 4, 6, 8} and Y={3, 7, 5, 10} considering different learning rate values (alpha).

Source Code:

Task1: Plot the input-output relation for given attributes. ¶

```python
import pandas as pd
import matplotlib.pyplot as plt

# Load data
csv_file_path = 'Salary_Data.csv'
data = pd.read_csv(csv_file_path)

# Extracting input-output column
years_of_experience = data['YearsExperience']
salary = data['Salary']

# Plotting the input-output relationship
plt.figure(figsize=(10, 6))
plt.scatter(years_of_experience, salary, color='blue', marker='o')
plt.title('Input-Output Relationship: Years of Experience vs Salary')
plt.xlabel('Years of Experience')
plt.ylabel('Salary')
plt.grid(True)
plt.show()
```

Task2: Design a mathematical function to find the best-fitted line for the given data (attached here).

```python
import numpy as np
def linear_regression(x, y):
    n = len(x)
    mean_x, mean_y = np.mean(x), np.mean(y)
    m = np.sum((x - mean_x) * (y - mean_y)) / np.sum((x - mean_x) ** 2)
    b = mean_y - m * mean_x
    return m, b
years_of_experience = data['YearsExperience']
salary = data['Salary']
slope, intercept = linear_regression(years_of_experience, salary)
print(f"Best-fitted line: y = {slope:.2f}x + {intercept:.2f}")
```

```
Best-fitted line: y = 9449.96x + 25792.20
```

```python
best_fit_line = slope * years_of_experience + intercept

# Plotting the input-output relationship and the best-fitted line
plt.figure(figsize=(10, 6))
plt.scatter(years_of_experience, salary, color='blue', marker='o', label='Data points')
plt.plot(years_of_experience, best_fit_line, color='red', label='Best-fitted line')
plt.title('Input-Output Relationship with Best-Fitted Line')
plt.xlabel('Years of Experience')
plt.ylabel('Salary')
plt.legend()
plt.grid(True)
plt.show()
```

Task3: Plot Error vs. Slope graph and show the global minima for the sample data X={2, 4, 6, 8} and Y={3, 7, 5, 10} considering different learning rate values (alpha).

```
import numpy as np
import matplotlib.pyplot as plt
X = np.array([2, 4, 6, 8])
Y = np.array([3, 7, 5, 10])
def mean_squared_error(slope, X, Y):
    predictions = slope * X
    error = np.mean((predictions - Y) ** 2)
    return error
def gradient_descent(X, Y, alpha, iterations):
    slopes = []
    errors = []
    slope = 0
    for _ in range(iterations):
        slope = slope - alpha * (1/len(X)) * np.sum((slope * X - Y) * X)
        error = mean_squared_error(slope, X, Y)
        slopes.append(slope)
        errors.append(error)
    return slopes, errors
alpha_values = [0.01, 0.02, 0.03, 0.04]
plt.figure(figsize=(10, 6))
for alpha in alpha_values:
    slopes, errors = gradient_descent(X, Y, alpha, iterations=100)
    plt.plot(slopes, errors, label=f'Alpha = {alpha}')
plt.title('Error vs. Slope for Different Learning Rates')
plt.xlabel('Slope')
plt.ylabel('Mean Squared Error')
plt.legend()
plt.grid(True)
plt.show()
```

Output:

1. Plot the input-output relation for given attributes.

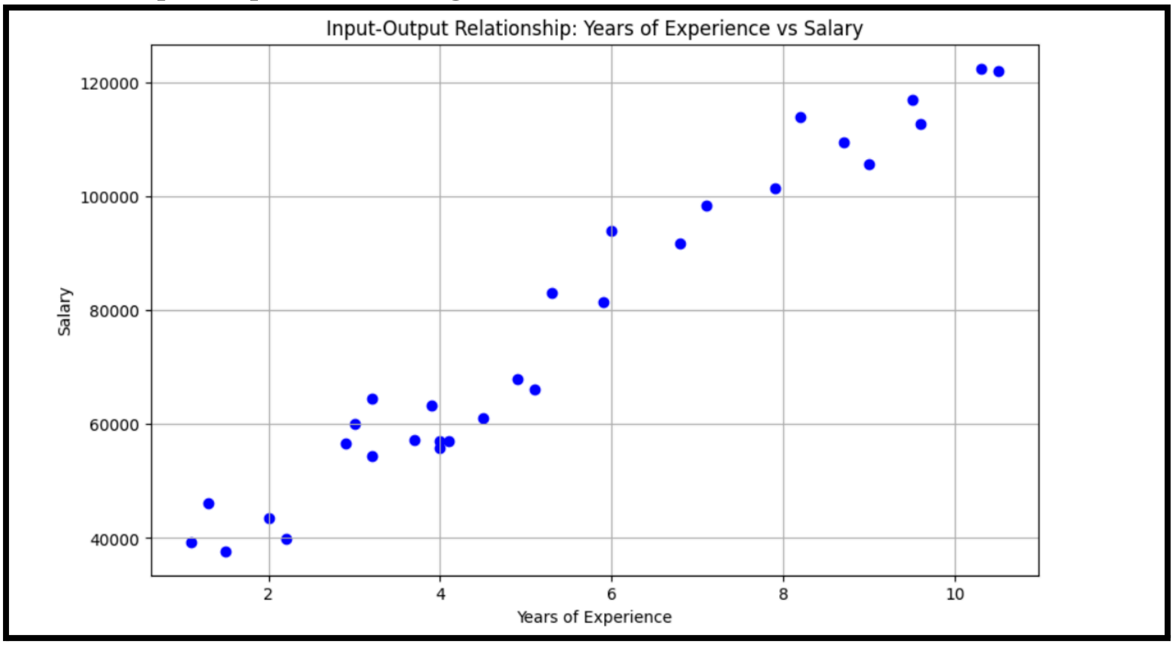

Figure 3.1 Output for Input-Output Relation

2. Design a mathematical function to find the best-fitted line for the given data

```
Best-fitted line: y = 9449.96x + 25792.20
```

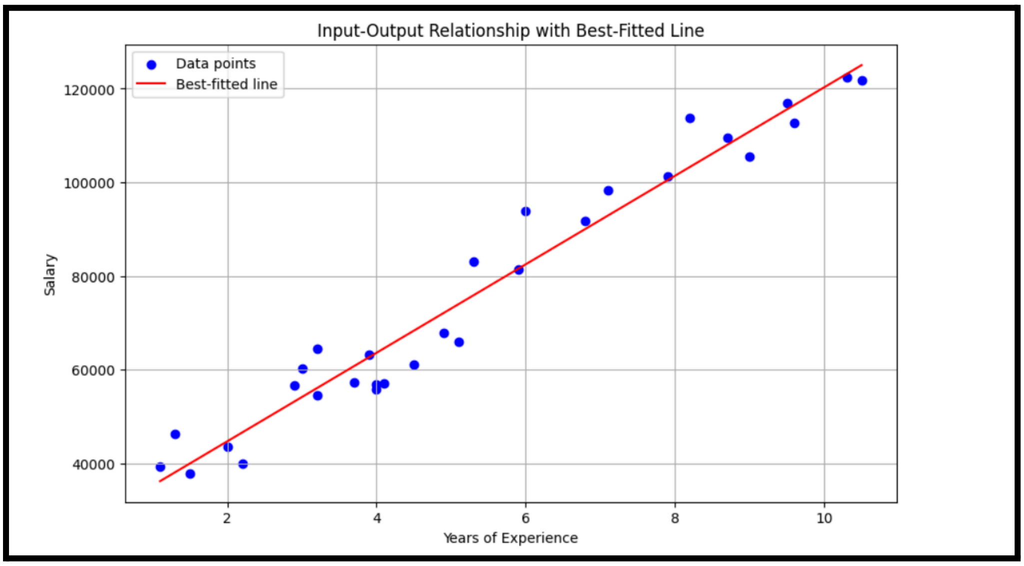

Figure 3.2 Output for Best-Fit Line

3. Plot Error vs. Slope graph and show the global minima for the sample data X={2, 4, 6, 8} and Y={3, 7, 5, 10} considering different learning rate values (alpha).

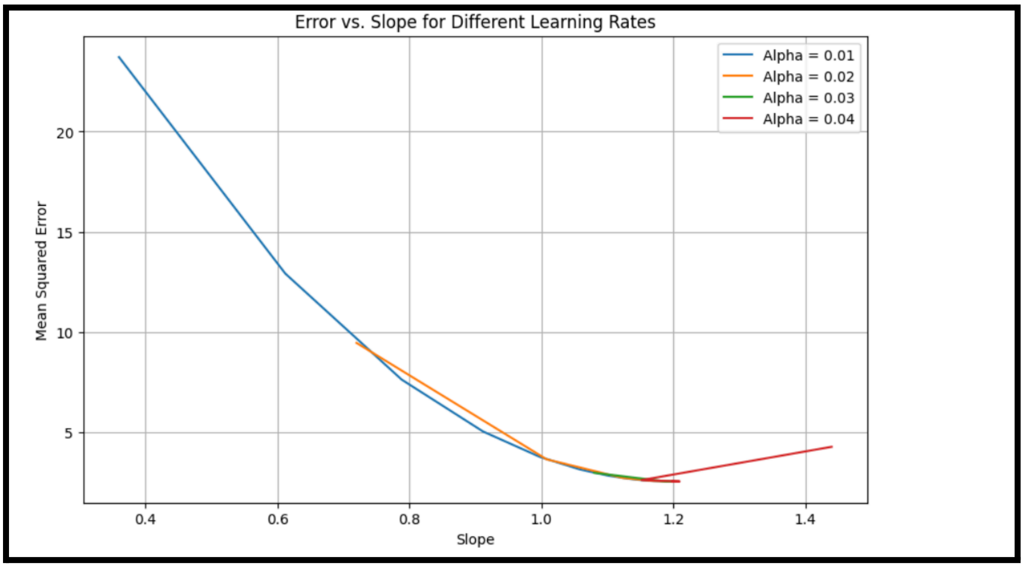

Figure 3.3 Error vs Slope Graph

Conclusion:

- A custom linear regression function was developed to find the best-fitted line for the given data.
- Utilized the gradient descent algorithm to minimize the cost function, aiming to find the optimal slope for the given linear regression problem.
- Plotted the Error vs. Slope graph for each learning rate, illustrating the convergence behavior over epochs.
- The impact of the learning rate on the convergence speed and the final error was observed through the plotted graphs.

Lab No: 4

Aim:

Study of essential text pre-processing techniques. Write python script for the essential text preprocessing techniques. Store the preprocessed data into a separate column of .CSV file. Compare the outcomes with and without using libraries for the same.

Description:

Perform the following task with using inbuilt Python Libraries:

- Lower Casing: Converts text into lower case text. It Helps ensure uniformity in text analysis and processing, as it treats uppercase and lowercase forms of words as the same.
- Tokenization: Break the text into individual words or tokens. It Facilitates analysis at the word level, making it easier to extract meaningful information and perform various natural language processing tasks.
- Punctuation Mark Removal: Eliminate punctuation marks from the text. Enhances the accuracy of text analysis by removing non-alphanumeric characters that don't contribute to the core meaning of the text.
- Stop Word Removal: Exclude common words (stop words) like "and," "the," and "is" that don't carry significant meaning. Improves the efficiency of text processing and analysis by focusing on content-bearing words.
- Stemming: Reduce words to their root or base form by removing suffixes. Aims to group variations of a word together, simplifying analysis and information retrieval. For example, "running" becomes "run."
- Lemmatization: Similar to Stemming but considers the word's context to reduce it to its base or dictionary form (lemma). Results in more accurate representation of the base form of a word, addressing potential ambiguities introduced by stemming.
- Translation: Convert text from one language to another. Facilitates cross-language communication and analysis, enabling understanding of content in different linguistic contexts.
- Emoji to Text: Translate emojis (emotion icons) into their corresponding textual representation. Helps in extracting meaning from textual data that includes emojis, making it easier for analysis and understanding sentiment.

Source Code:

```
# Study of essential text pre-processing techniques. Write python script for the essential text preprocessing techniques. Store the preprocessed data into a separate column of .CSV file. Compare the outcomes with and without using libraries for the same.
```

```
## Perform the following task with using inbuilt Python Libraries:
```

```python
import pandas as pd
import nltk
from nltk.tokenize import word_tokenize
from nltk.corpus import stopwords
from nltk.stem import PorterStemmer, WordNetLemmatizer
from deep_translator import GoogleTranslator
import emoji
import string
import re
```

```python
nltk.download('stopwords')
```

```python
nltk.download('punkt')
```

```python
nltk.download('wordnet')
```

```python
data = pd.read_csv("PTweet_WWE.csv")
data.head()
```

```python
df = pd.DataFrame(data['text'])
df.head()
```

```
### 1. Lower Casing
```

```python
# Task 1: Lowercasing
df['lowercased_text'] = df['text'].apply(lambda x: x.lower())
df.head()
```

```
### 2. Tokenization
```

```python
# Task 2: Tokenization
# df['tokens'] = df['lowercased_text'].apply(lambda x: re.findall(r'\b\w+\b', x))
df['tokens'] = df['lowercased_text'].apply(lambda x: word_tokenize(x))
df.head()
```

```
### 3. Punctuation Mark Removal
```

```python
# Task 3: Punctuation Mark Removal
df['cleaned_text'] = df['tokens'].apply(lambda x: ''.join(char for char in x if char not in string.punctuation))
df.head()
```

```
### 4. Stop Word Removal
```

```python
# Task 4: Stop Word Removal
stop_words = set(stopwords.words('english'))
df['filtered_text'] = df['tokens'].apply(lambda x: ' '.join(word for word in x if word not in stop_words))
df.head()
```

```
### 5. Stemming
```

```python
# Task 5: Stemming
stemmer = PorterStemmer()
df['stemmed_Text'] = df['tokens'].apply(lambda x: ' '.join(stemmer.stem(word) for word in x))
df.head()
```

```
### 6. Lemmatization
```

```python
# Task 6: Lemmatization
lemmatizer = WordNetLemmatizer()
df['lemmatized_text'] = df['tokens'].apply(lambda x: ' '.join(lemmatizer.lemmatize(word) for word in x))
df.head()
```

```
### 7. Translation
```

```python
# Task 7: Translation
# translator = google_translator()
df['translated_text'] = df['lowercased_text'].apply(lambda x: GoogleTranslator(source='auto', target='es').translate(x))  # Translate to Spanish
df.head()
```

```
### 8. Emoji to text
```

```python
# Task 8: Emoji to Text
df['emoji_to_text'] = df['text'].apply(lambda x: emoji.demojize(x))
df.head()
```

```
## Perform the following task without using inbuilt Python Libraries (The last two task (Translation and Emoji) are not possible without libraies):
```

```python
import re
import string

# Sample text data
text_data = data.head()['text']

# Task 1: Lowercasing
lowercased_texts = [text.lower() for text in text_data]

# Task 2: Tokenization
tokenized_texts = [re.findall(r'\b\w+\b', text) for text in text_data]

# Task 3: Punctuation Mark Removal
cleaned_texts = [''.join(char for char in text if char not in string.punctuation) for text in text_data]

# Task 4: Stop Word Removal
stop_words = set(["a", "an", "the", "is", "from", "this"])
filtered_texts = [' '.join(word for word in text.split() if word.lower() not in stop_words) for text in text_data]

# Task 5: Stemming
def simple_stemming(text):
    return ' '.join(word[:4] if len(word) > 4 else word for word in text.split())

stemmed_texts = [simple_stemming(text) for text in text_data]

# Task 6: Lemmatization
def simple_lemmatization(text):
    return ' '.join(word[:-2] if word.endswith("es") else word for word in text.split())

lemmatized_texts = [simple_lemmatization(text) for text in text_data]

# Display results
for i in range(len(text_data)):
    print(f"\nOriginal Text: {text_data[i]}")
    print(f"Lowercased Text: {lowercased_texts[i]}")
    print(f"Tokenized Text: {tokenized_texts[i]}")
    print(f"Cleaned Text: {cleaned_texts[i]}")
    print(f"Filtered Text: {filtered_texts[i]}")
    print(f"Stemmed Text: {stemmed_texts[i]}")
    print(f"Lemmatized Text: {lemmatized_texts[i]}")
```

Output:

Twitter Data:

	link	text	name	username	date	is_rt	n_comment	n_rt	n_quote	n_like
0	https://twitter.com/WWE/status/174201779437427...	IF YA SMELL..... @TheRock has come back to #W...	WWE	@WWE	Jan 2, 2024 · 2:59 AM UTC	False	1088	9916	1856	49998
1	https://twitter.com/WWE/status/174573989977118...	Who had the best Instagram photo of the week?!...	WWE	@WWE	Jan 12, 2024 · 9:30 AM UTC	False	47	39	3	342
2	https://twitter.com/WWE/status/174567949971749...	These #RoyalRumble crashers were RUTHLESS! ht...	WWE	@WWE	Jan 12, 2024 · 5:30 AM UTC	False	46	72	2	624
3	https://twitter.com/WWE/status/174564199274113...	An All Mighty moment in the 2023 Men's #RoyalR...	WWE	@WWE	Jan 12, 2024 · 3:00 AM UTC	False	58	213	17	2454
4	https://twitter.com/WWE/status/174559668762676...	Outta nowhere! 😳	WWE	@WWE	Jan 12, 2024 · 12:00 AM UTC	False	70	354	10	3853

Figure 4.0 Twitter Data

Perform the following task with using inbuilt Python Libraries:

1. Lower Casing

	text	lowercased_text
0	IF YA SMELL..... @TheRock has come back to #W...	if ya smell..... @therock has come back to #w...
1	Who had the best Instagram photo of the week?!...	who had the best instagram photo of the week?!...
2	These #RoyalRumble crashers were RUTHLESS! ht...	these #royalrumble crashers were ruthless! ht...
3	An All Mighty moment in the 2023 Men's #RoyalR...	an all mighty moment in the 2023 men's #royalr...
4	Outta nowhere! 😳	outta nowhere! 😳

Figure 4.1.1 Lower Casing

2. Tokenization

	text	lowercased_text	tokens
0	IF YA SMELL..... @TheRock has come back to #W...	if ya smell..... @therock has come back to #w...	[if, ya, smell, ..., @, therock, has, come, ...
1	Who had the best Instagram photo of the week?!...	who had the best instagram photo of the week?!...	[who, had, the, best, instagram, photo, of, th...
2	These #RoyalRumble crashers were RUTHLESS! ht...	these #royalrumble crashers were ruthless! ht...	[these, #, royalrumble, crashers, were, ruthle...
3	An All Mighty moment in the 2023 Men's #RoyalR...	an all mighty moment in the 2023 men's #royalr...	[an, all, mighty, moment, in, the, 2023, men, ...
4	Outta nowhere! 😳	outta nowhere! 😳	[outta, nowhere, !, 😳]

Figure 4.1.2 Tokenization

3. Punctuation Mark Removal

	text	lowercased_text	tokens	cleaned_text
0	IF YA SMELL..... @TheRock has come back to #W...	if ya smell..... @therock has come back to #w...	[if, ya, smell, ..., @, therock, has, come, ...	ifyasmell.....therockhascomebacktowweraw
1	Who had the best Instagram photo of the week?!...	who had the best instagram photo of the week?!...	[who, had, the, best, instagram, photo, of, th...	whohadthebestinstagramphotooftheweekhttps//www...
2	These #RoyalRumble crashers were RUTHLESS! ht...	these #royalrumble crashers were ruthless! ht...	[these, #, royalrumble, crashers, were, ruthle...	theseroyalrumblecrasherswereruthlesshttps//tub...
3	An All Mighty moment in the 2023 Men's #RoyalR...	an all mighty moment in the 2023 men's #royalr...	[an, all, mighty, moment, in, the, 2023, men, ...	anallmightymomentinthe2023men'sroyalrumblematch
4	Outta nowhere! 😳	outta nowhere! 😳	[outta, nowhere, !, 😳]	outtanowhere 😳

Figure 4.1.3 Punctuation Mark Removal

4. Stop Word Removal

	text	lowercased_text	tokens	cleaned_text	filtered_text
0	IF YA SMELL..... @TheRock has come back to #W...	if ya smell..... @therock has come back to #w...	[if, ya, smell, ..., @, therock, has, come, ...	ifyasmell.....therockhascomebacktowweraw	ya smell ..... @ therock come back # wweraw !
1	Who had the best Instagram photo of the week?!...	who had the best instagram photo of the week?!...	[who, had, the, best, instagram, photo, of, th...	whohadthebestinstagramphotooftheweekhttps//www...	best instagram photo week ? ! https : //www.ww...
2	These #RoyalRumble crashers were RUTHLESS! ht...	these #royalrumble crashers were ruthless! ht...	[these, #, royalrumble, crashers, were, ruthle...	theseroyalrumblecrasherswereruthlesshttps//tub...	# royalrumble crashers ruthless ! https : //tu...
3	An All Mighty moment in the 2023 Men's #RoyalR...	an all mighty moment in the 2023 men's #royalr...	[an, all, mighty, moment, in, the, 2023, men, ...	anallmightymomentinthe2023men'sroyalrumblematch	mighty moment 2023 men 's # royalrumble match !
4	Outta nowhere! 😳	outta nowhere! 😳	[outta, nowhere, !, 😳]	outtanowhere 😳	outta nowhere ! 😳

Figure 4.1.4 Stop Word Removal

5. Stemming

	text	lowercased_text	tokens	cleaned_text	filtered_text	stemmed_Text
0	IF YA SMELL..... @TheRock has come back to #W...	if ya smell..... @therock has come back to #w...	[if, ya, smell, ..., @, therock, has, come, ...	ifyasmell.....therockhascomebacktowweraw	ya smell ..... @ therock come back # wweraw !	if ya smell ..... @ therock ha come back to # ...
1	Who had the best Instagram photo of the week?!...	who had the best instagram photo of the week?!...	[who, had, the, best, instagram, photo, of, th...	whohadthebestinstagramphotooftheweekhttps//www...	best instagram photo week ? ! https : //www.ww...	who had the best instagram photo of the week ?...
2	These #RoyalRumble crashers were RUTHLESS! ht...	these #royalrumble crashers were ruthless! ht...	[these, #, royalrumble, crashers, were, ruthle...	theseroyalrumblecrasherswereruthlesshttps//tub...	# royalrumble crashers ruthless ! https : //tu...	these # royalrumbl crasher were ruthless ! htt...
3	An All Mighty moment in the 2023 Men's #RoyalR...	an all mighty moment in the 2023 men's #royalr...	[an, all, mighty, moment, in, the, 2023, men, ...	anallmightymomentinthe2023men'sroyalrumblematch	mighty moment 2023 men 's # royalrumble match !	an all mighti moment in the 2023 men 's # roya...
4	Outta nowhere! 😲	outta nowhere! 😲	[outta, nowhere, !, 😲]	outtanowhere😲	outta nowhere ! 😲	outta nowhe r ! 😲

Figure 4.1.5 Stemming

6. Lemmatization

	text	lowercased_text	tokens	cleaned_text	filtered_text	stemmed_Text	lemmatized_text
0	IF YA SMELL..... @TheRock has come back to #W...	if ya smell..... @therock has come back to #w...	[if, ya, smell, ..., @, therock, has, come, ...	ifyasmell.....therockhascomebacktowweraw	ya smell ..... @ therock come back # wweraw !	if ya smell ..... @ therock ha come back to # ...	if ya smell ..... @ therock ha come back to # ...
1	Who had the best Instagram photo of the week?!...	who had the best instagram photo of the week?!...	[who, had, the, best, instagram, photo, of, th...	whohadthebestinstagramphotooftheweekhttps//www...	best instagram photo week ? ! https : //www.ww...	who had the best instagram photo of the week ?...	who had the best instagram photo of the week ?...
2	These #RoyalRumble crashers were RUTHLESS! ht...	these #royalrumble crashers were ruthless! ht...	[these, #, royalrumble, crashers, were, ruthle...	theseroyalrumblecrasherswereruthlesshttps//tub...	# royalrumble crashers ruthless ! https : //tu...	these # royalrumbl crasher were ruthless ! htt...	these # royalrumble crasher were ruthless ! ht...
3	An All Mighty moment in the 2023 Men's #RoyalR...	an all mighty moment in the 2023 men's #royalr...	[an, all, mighty, moment, in, the, 2023, men, ...	anallmightymomentinthe2023men'sroyalrumblematch	mighty moment 2023 men 's # royalrumble match !	an all mighti moment in the 2023 men 's # roya...	an all mighty moment in the 2023 men 's # roya...
4	Outta nowhere! 😲	outta nowhere! 😲	[outta, nowhere, !, 😲]	outtanowhere😲	outta nowhere ! 😲	outta nowhe r ! 😲	outta nowhere ! 😲

Figure 4.1.6 Lemmatization

7. Translation

	text	lowercased_text	tokens	cleaned_text	filtered_text	stemmed_Text	lemmatized_text	translated_text
0	IF YA SMELL..... @TheRock has come back to #W...	if ya smell..... @therock has come back to #w...	[if, ya, smell, ..., @, therock, has, come, ...	ifyasmell.....therockhascomebacktowweraw	ya smell ..... @ therock come back # wweraw !	if ya smell ..... @ therock ha come back to # ...	if ya smell ..... @ therock ha come back to # ...	si hueles..... ¡@therock ha regresado a #wweraw!
1	Who had the best Instagram photo of the week?!...	who had the best instagram photo of the week?!...	[who, had, the, best, instagram, photo, of, th...	whohadthebestinstagramphotooftheweekhttps//www...	best instagram photo week ? ! https : //www.ww...	who had the best instagram photo of the week ?...	who had the best instagram photo of the week ?...	¿Quién tuvo la mejor foto de Instagram de la s...
2	These #RoyalRumble crashers were RUTHLESS! ht...	these #royalrumble crashers were ruthless! ht...	[these, #, royalrumble, crashers, were, ruthle...	theseroyalrumblecrasherswereruthlesshttps//tub...	# royalrumble crashers ruthless ! https : //tu...	these # royalrumbl crasher were ruthless ! htt...	these # royalrumble crasher were ruthless ! ht...	¡Estos intrusos del #royalrumble fueron despia...
3	An All Mighty moment in the 2023 Men's #RoyalR...	an all mighty moment in the 2023 men's #royalr...	[an, all, mighty, moment, in, the, 2023, men, ...	anallmightymomentinthe2023men'sroyalrumblematch	mighty moment 2023 men 's # royalrumble match !	an all mighti moment in the 2023 men 's # roya...	an all mighty moment in the 2023 men 's # roya...	¡Un momento poderoso en el combate #royalrumbl...
4	Outta nowhere! 😲	outta nowhere! 😲	[outta, nowhere, !, 😲]	outtanowhere😲	outta nowhere ! 😲	outta nowhe r ! 😲	outta nowhere ! 😲	¡de la nada! 😲

Figure 4.1.7 Translation

8. Emoji to text

text	lowercased_text	tokens	cleaned_text	filtered_text	stemmed_Text	lemmatized_text	translated_text	emoji_to_text
IF YA SMELL..... @TheRock has come ck to #W...	if ya smell..... @therock has come back to #w...	[if, ya, smell, ..., @, therock, has, come, ...	ifyasmell.....therockhascomebacktowweraw	ya smell ..... @ therock come back # wweraw !	if ya smell ..... @ therock ha come back to # ...	if ya smell ..... @ therock ha come back to # ...	si hueles..... ¡@therock ha regresado a #wweraw!	IF YA SMELL..... @TheRock has come back to #W...
o had the best Instagram oto of the week?!...	who had the best instagram photo of the week?!...	[who, had, the, best, instagram, photo, of, th...	whohadthebestinstagramphotooftheweekhttps//www...	best instagram photo week ? ! https : //www.ww...	who had the best instagram photo of the week ?...	who had the best instagram photo of the week ?...	¿Quién tuvo la mejor foto de Instagram de la s...	Who had the best Instagram photo of the week?!...
These alRumble hers were UTHLESS! ht...	these #royalrumble crashers were ruthless! ht...	[these, #, royalrumble, crashers, were, ruthle...	theseroyalrumblecrasherswereruthlesshttps//tub...	# royalrumble crashers ruthless ! https : //tu...	these # royalrumbl crasher were ruthless ! htt...	these # royalrumble crasher were ruthless ! ht...	¡Estos intrusos del #royalrumble fueron despia...	These #RoyalRumble crashers were RUTHLESS! ht...
All Mighty oment in the 2023 Men's #RoyalR...	an all mighty moment in the 2023 men's #royalr...	[an, all, mighty, moment, in, the, 2023, men, ...	anallmightymomentinthe2023men'sroyalrumblematch	mighty moment 2023 men 's # royalrumble match !	an all mighti moment in the 2023 men 's # roya...	an all mighty moment in the 2023 men 's # roya...	¡Un momento poderoso en el combate #royalrumbl...	An All Mighty moment in the 2023 Men's #RoyalR...
Outta vhere! 😲	outta nowhere! 😲	[outta, nowhere, !, 😲]	outtanowhere 😲	outta nowhere ! 😲	outta nowher ! 😲	outta nowhere ! 😲	¡de la nada! 😲	Outta nowhere! :astonished_face:

Figure 4.1.8 Emoji To Text

Perform the following task without using inbuilt Python Libraries (Wont't work for last two tasks):

```
Original Text: IF YA SMELL.....  @TheRock has come back to #WWERaw!
Lowercased Text: if ya smell.....  @therock has come back to #wweraw!
Tokenized Text: ['IF', 'YA', 'SMELL', 'TheRock', 'has', 'come', 'back', 'to', 'WWERaw']
Cleaned Text: IF YA SMELL  TheRock has come back to WWERaw
Filtered Text: IF YA SMELL..... @TheRock has come back to #WWERaw!
Stemmed Text: IF YA SMEL @The has come back to #WWE
Lemmatized Text: IF YA SMELL..... @TheRock has come back to #WWERaw!

Original Text: Who had the best Instagram photo of the week?!  https://www.wwe.com/gallery/the-25-best-instagram-photos-of-the-week-january-7-2024#fid-40
650941
Lowercased Text: who had the best instagram photo of the week?!  https://www.wwe.com/gallery/the-25-best-instagram-photos-of-the-week-january-7-2024#fid-
40650941
Tokenized Text: ['Who', 'had', 'the', 'best', 'Instagram', 'photo', 'of', 'the', 'week', 'https', 'www', 'wwe', 'com', 'gallery', 'the', '25', 'best', 'i
nstagram', 'photos', 'of', 'the', 'week', 'january', '7', '2024', 'fid', '40650941']
Cleaned Text: Who had the best Instagram photo of the week  httpswwwwwecomgallerythe25bestinstagramphotosoftheweekjanuary72024fid40650941
Filtered Text: Who had best Instagram photo of week?! https://www.wwe.com/gallery/the-25-best-instagram-photos-of-the-week-january-7-2024#fid-40650941
Stemmed Text: Who had the best Inst phot of the week http
Lemmatized Text: Who had the best Instagram photo of the week?! https://www.wwe.com/gallery/the-25-best-instagram-photos-of-the-week-january-7-2024#fid-4
0650941

Original Text: These #RoyalRumble crashers were RUTHLESS!  https://tube.mint.lgbt/VV5fxHfxCE4?si=naZCLWedRVreRISE
Lowercased Text: these #royalrumble crashers were ruthless!  https://tube.mint.lgbt/vv5fxhfxce4?si=nazclwedrvrerise
Tokenized Text: ['These', 'RoyalRumble', 'crashers', 'were', 'RUTHLESS', 'https', 'tube', 'mint', 'lgbt', 'VV5fxHfxCE4', 'si', 'naZCLWedRVreRISE']
Cleaned Text: These RoyalRumble crashers were RUTHLESS  httpstubemintlgbtVV5fxHfxCE4sinaZCLWedRVreRISE
Filtered Text: These #RoyalRumble crashers were RUTHLESS! https://tube.mint.lgbt/VV5fxHfxCE4?si=naZCLWedRVreRISE
Stemmed Text: Thes #Roy cras were RUTH http
Lemmatized Text: These #RoyalRumble crashers were RUTHLESS! https://tube.mint.lgbt/VV5fxHfxCE4?si=naZCLWedRVreRISE

Original Text: An All Mighty moment in the 2023 Men's #RoyalRumble Match!
Lowercased Text: an all mighty moment in the 2023 men's #royalrumble match!
Tokenized Text: ['An', 'All', 'Mighty', 'moment', 'in', 'the', '2023', 'Men', 's', 'RoyalRumble', 'Match']
Cleaned Text: An All Mighty moment in the 2023 Mens RoyalRumble Match
Filtered Text: All Mighty moment in 2023 Men's #RoyalRumble Match!
Stemmed Text: An All Migh mome in the 2023 Men' #Roy Matc
Lemmatized Text: An All Mighty moment in the 2023 Men's #RoyalRumble Match!

Original Text: Outta nowhere! 😲
Lowercased Text: outta nowhere! 😲
Tokenized Text: ['Outta', 'nowhere']
Cleaned Text: Outta nowhere 😲
Filtered Text: Outta nowhere! 😲
Stemmed Text: Outt nowh 😲
Lemmatized Text: Outta nowhere! 😲
```

Figure 4.2 Without Library

Conclusion:

- Lowercasing ensures uniformity, treating uppercase and lowercase forms equally, preventing discrepancies in analysis.
- Tokenization breaks down text into meaningful units, enabling granular analysis at the word level and facilitating various natural language processing tasks.
- Punctuation mark removal eliminates non-alphanumeric characters, reducing noise and focusing on the core meaning of the text.
- Stop word removal improves efficiency by excluding common words, allowing a focus on content-bearing words and enhancing the relevance of analysis.
- Stemming and lemmatization contribute to word form normalization, reducing words to their base form for better consistency and information retrieval.
- Translation enables the understanding of text in different languages, fostering cross-language communication and analysis.
- Emoji-to-text conversion aids in extracting emotional context from textual data, contributing to sentiment analysis and understanding user expressions.