NLP LAB-1

In [1]:

```
# !pip install spacy
```

In [2]:

```
 pip install gensim
```

```
Requirement already satisfied: gensim in c:\users\bhavy\anaconda3\lib\si
te-packages (4.1.2)
Requirement already satisfied: scipy>=0.18.1 in c:\users\bhavy\anaconda3
\lib\site-packages (from gensim) (1.9.3)
Requirement already satisfied: numpy>=1.17.0 in c:\users\bhavy\anaconda3
\lib\site-packages (from gensim) (1.23.5)
Requirement already satisfied: smart-open>=1.8.1 in c:\users\bhavy\anaco
nda3\lib\site-packages (from gensim) (6.3.0)
Note: you may need to restart the kernel to use updated packages.
```

```
WARNING: Ignoring invalid distribution -rotobuf (c:\users\bhavy\anaconda
3\lib\site-packages)
WARNING: Ignoring invalid distribution -rotobuf (c:\users\bhavy\anaconda
3\lib\site-packages)
WARNING: Ignoring invalid distribution -rotobuf (c:\users\bhavy\anaconda
3\lib\site-packages)
WARNING: Ignoring invalid distribution -rotobuf (c:\users\bhavy\anaconda
3\lib\site-packages)
WARNING: Ignoring invalid distribution -rotobuf (c:\users\bhavy\anaconda
3\lib\site-packages)
WARNING: Ignoring invalid distribution -rotobuf (c:\users\bhavy\anaconda
3\lib\site-packages)
```

In [3]:

```
# pip install transformers
```

In [4]:

```
text = """AN INTRODUCTION TO LOGISTIC REGRESSION
Purpose: the purpose of this document is Twofold
1.  Introducing logistic regression
2.  connecting linear regression with logistic regression.
Before we dive into the new concept of LOGISTIC REGRESSION, Let’s take a sense of what
REGRESSION ANALYSIS
As the “Hello World” of machine learning algorithms, regression analysis is a simple s
The first regression analysis technique that we will examine is LINEAR REGRESSION, whi

Imagine you’re back in high school and it's the year 2015. During your senior year, a
But before you have a chance to bid for your first Bitcoin on Coinbase, your father in
You also keep a tally of how many days have passed since you first started paper tradi
But instead, the value of Bitcoin exploded in the opposite direction . Nonetheless, yo
You first reach into your toolbox for a scatterplot. With the blank scatterplot in you

After plotting the x and y values on the scatterplot, you can immediately see a trend
Based on the upward trajectory of the curve, it might be time to quit hoping for a dro
In order to assess whether it’s worth borrowing from your friend, you will need to fir
Then you need to figure out whether the return on investment will be adequate to pay b
It’s now time to reach out for an algorithm. One of the simplest algorithms in machine
Linear regression
comprises a straight line that splits your data points on a scatterplot. The goal of l
This means that if you were to draw a vertical line from the regression line to each d

Another important feature of regression is slope. For example, if you wish to estimate

The hyperplane reveals that you stand to make a loss at the day 800 if you plan to buy

Logistic regression :
A large part of data analysis boils down to a simple question: is something “A” or “B?
Machine learning accommodates such questions through logistic equations, and specifica
Logistic regression adopts the sigmoid function to analyse data and predict discrete c
Logistic regression is typically used for binary classification to predict two disc re

To do this, the sigmoid function is added to compute the result and convert numerical
Logistic regression with more than two outcome values is known as multinomial logistic
A sigmoid curve looks as follows

AN INTRODUCTION TO GRADIENT DESCENT
This is an introduction to gradient descent . Let us explore  how does it work  and wh
Before we get into the technicality let us first understand the basic terms.

•   What is Gradient ?
In simple words “Gradient” measures how much the output of a function will change if y
•   What is Descent ?
Descent is and act of moving downwards.
•   But now What is Gradient Descent ???
Gradient descent is an optimization algorithm which is commonly used to train machine
•   From this we can assume that Gradient Ascent would mean the exact opposite but is
So well, When you are maximizing a differentiable function you can use gradient ascent
•   How does gradient descent work:
1.  Consider a large bowl with which you would normally keep fruits or eat cereal. Thi

2.  Now, a random point on any part of the surface of the bowl can be assumed as the b
The bottom of the bowl will have the lowest set of squared residuals (error) and hence

3.  Goal is to reach the bottom ( or the nearest )
4.  Here the Gradient Descent comes into play, It identifies the optimal value by taki
"""
```

Count

In [5]:

```
count = len(text.split())
print("Count of Words in the given Sentence:", count)
```

```
Count of Words in the given Sentence: 1260
```

In [6]:

```
text.count('regression')
```

Out[6]:

```
23
```

In [7]:

```
def word_count(str):
    counts = {}
    words = str.split()
    for i in words:
        if i in counts:
            counts[i] += 1
        else:
            counts[i] = 1
    return counts
print( word_count(text))
```

```
{'AN': 2, 'INTRODUCTION': 2, 'TO': 2, 'LOGISTIC': 2, 'REGRESSION':
2, 'Purpose:': 1, 'the': 68, 'purpose': 1, 'of': 30, 'this': 4, 'do
cument': 1, 'is': 28, 'Twofold': 1, '1.': 4, 'Introducing': 1, 'log
istic': 6, 'regression': 20, '2.': 2, 'connecting': 1, 'linear': 5,
'with': 4, 'regression.': 1, 'Before': 2, 'we': 4, 'dive': 1, 'int
o': 7, 'new': 1, 'concept': 1, 'REGRESSION,': 2, 'Let’s': 1, 'tak
e': 2, 'a': 40, 'sense': 1, 'what': 2, 'Linear': 2, 'Regression':
1, ':-': 1, 'ANALYSIS': 1, 'As': 2, '“Hello': 1, 'World”': 1, 'mach
ine': 3, 'learning': 5, 'algorithms,': 1, 'analysis': 3, 'simple':
4, 'supervised': 1, 'technique': 2, 'used': 4, 'to': 53, 'find': 2,
'best': 1, 'trendline': 2, 'describe': 2, 'dataset.': 3, 'The': 5,
'first': 6, 'that': 12, 'will': 6, 'examine': 1, 'LINEAR': 1, 'whic
h': 4, 'uses': 1, 'straight': 2, 'line': 7, 'To': 3, 'unpack': 1,
'technique,': 1, 'let’s': 1, 'an': 10, 'example.': 1, 'Imagine': 1,
'you’re': 1, 'back': 3, 'in': 12, 'high': 1, 'school': 1, 'and': 2
4, "it's": 1, 'year': 1, '2015.': 1, 'During': 1, 'your': 22, 'seni
or': 1, 'year,': 1, 'news': 1, 'headline': 1, 'piques': 1, 'interes
t': 1, 'Bitcoin.': 2, 'With': 2, 'natural': 1, 'tendency': 1, 'chas
e': 1, 'next': 2, 'shiny': 1, 'object,': 1, 'you': 34, 'tell': 1,
```

In [8]:

```
sample_text = """ Hi My name is Bhavya and my userid is bhavya123 274. Neural networks

CNNs are designed to analyze visual imagery, such as photographs or videos. They use a

One of the key advantages of CNNs is that they can automatically learn to recognize co

To train a CNN, a large dataset of labeled images is typically required. The network i

While CNNs have been highly successful in many applications, they are not without limi

Despite these challenges, CNNs have shown great promise in a wide range of application

In addition to image analysis, CNNs are also being used in a growing number of applica

One advantage of using CNNs for natural language processing is that they can automatic

Another area where CNNs are being used is in the field of robotics, where they are bei

In the healthcare industry, CNNs are being used to analyze 888 medical images and diag

In the financial industry, CNNs are being used to 234 analyze patterns in stock prices
```

In [9]:

```
def count_words(sample_text):
    cnt = []
    wrds = sample_text.split()
    for i in wrds:
        num_cnt = 0
        for j in i:
            if j.isalpha() ==True:
                num_cnt += 1
        cnt.append(num_cnt)
    return cnt
```

In [10]:

```
x = count_words(sample_text)
print(x)
```

```
[2, 2, 4, 2, 6, 3, 2, 6, 2, 6, 0, 6, 8, 3, 1, 5, 2, 7, 8, 10, 4, 3, 7,
7, 2, 3, 9, 3, 8, 2, 3, 5, 5, 4, 7, 2, 14, 5, 2, 7, 4, 3, 5, 2, 9, 8, 3,
4, 11, 5, 2, 5, 4, 3, 4, 2, 6, 7, 4, 3, 6, 12, 7, 2, 6, 5, 2, 3, 13, 6,
7, 3, 4, 3, 8, 2, 7, 6, 7, 4, 2, 11, 2, 6, 4, 3, 1, 6, 2, 13, 6, 2, 7,
8, 4, 3, 5, 5, 9, 8, 2, 1, 4, 8, 14, 2, 3, 7, 4, 5, 8, 2, 1, 3, 2, 7, 2,
7, 4, 4, 4, 3, 5, 2, 8, 8, 8, 4, 2, 5, 7, 2, 8, 3, 6, 2, 4, 5, 2, 4, 3,
4, 3, 4, 5, 5, 7, 7, 3, 7, 14, 3, 2, 3, 3, 10, 2, 4, 2, 4, 4, 3, 13, 5,
2, 9, 7, 6, 8, 7, 9, 8, 7, 11, 4, 5, 4, 10, 3, 5, 4, 2, 6, 11, 5, 12, 3,
5, 14, 2, 6, 5, 4, 4, 4, 4, 2, 1, 4, 5, 2, 12, 4, 11, 4, 2, 7, 7, 2, 7,
7, 2, 5, 1, 3, 1, 5, 7, 2, 7, 6, 2, 9, 8, 3, 7, 2, 7, 2, 9, 3, 7, 2, 3,
10, 7, 2, 4, 3, 9, 7, 5, 3, 4, 6, 2, 7, 2, 8, 4, 7, 2, 7, 4, 5, 1, 7, 2,
10, 8, 7, 2, 5, 3, 7, 3, 8, 11, 5, 2, 3, 5, 7, 3, 9, 3, 6, 7, 5, 4, 4,
4, 6, 10, 2, 4, 12, 4, 3, 3, 7, 11, 3, 9, 2, 4, 4, 3, 2, 15, 9, 2, 5, 9,
11, 8, 4, 2, 8, 10, 5, 4, 2, 6, 10, 5, 4, 7, 9, 2, 4, 4, 3, 2, 9, 2, 10,
2, 3, 5, 4, 4, 2, 7, 2, 8, 2, 11, 7, 5, 10, 4, 4, 5, 5, 7, 2, 1, 4, 5,
2, 12, 3, 3, 6, 2, 4, 2, 12, 9, 4, 2, 3, 6, 2, 7, 8, 3, 10, 12, 2, 11,
8, 2, 7, 3, 10, 3, 8, 3, 10, 4, 2, 3, 6, 2, 3, 4, 4, 10, 5, 2, 6, 12, 2,
3, 5, 2, 4, 2, 8, 2, 5, 8, 4, 3, 4, 5, 4, 2, 1, 7, 6, 2, 12, 2, 7, 8, 1
0, 3, 6, 11, 3, 7, 11, 4, 4, 4, 2, 8, 9, 2, 4, 8, 5, 2, 6, 3, 4, 8, 7,
8, 4, 4, 7, 3, 9, 2, 5, 4, 3, 7, 8, 10, 2, 4, 4, 3, 13, 5, 8, 4, 3, 3,
4, 4, 2, 4, 10, 2, 8, 9, 7, 9, 8, 7, 11, 4, 5, 4, 10, 3, 5, 4, 2, 4, 14,
5, 6, 11, 3, 8, 11, 7, 4, 5, 4, 3, 5, 4, 2, 2, 3, 5, 2, 8, 5, 4, 3, 5,
4, 2, 4, 8, 8, 3, 8, 5, 11, 3, 7, 11, 4, 4, 4, 2, 4, 6, 6, 3, 5, 9, 2,
2, 4, 10, 8, 9, 3, 7, 2, 7, 7, 2, 3, 10, 8, 4, 3, 5, 4, 2, 7, 0, 7, 6,
3, 8, 8, 3, 7, 4, 4, 4, 4, 2, 6, 6, 6, 4, 10, 2, 2, 8, 4, 6, 4, 14, 6,
4, 3, 4, 5, 4, 2, 4, 11, 10, 3, 9, 3, 8, 2, 3, 5, 5, 2, 9, 4, 6, 3, 5, 1
2, 4, 2, 3, 9, 8, 4, 3, 5, 4, 2, 0, 7, 8, 2, 5, 6, 3, 5, 9, 4, 2, 5, 2,
4, 11, 5, 6]
```

In [11]:

```
unq = sample_text.split()
```

In [12]:

```
clean = [un.strip('.,!%[]()') for un in unq]
clean_again = [un.replace("'s",'') for un in clean]
clean_again
```

Out[12]:

```
['Hi',
 'My',
 'name',
 'is',
 'Bhavya',
 'and',
 'my',
 'userid',
 'is',
 'bhavya123',
 '274',
 'Neural',
 'networks',
 'are',
 'a',
 'class',
 'of',
 'machine',
```

In [13]:

```
numeric = 0
alphabet = 0
alphanum = 0
for i in clean_again:
    if i.isalpha():
        alphabet += 1
    elif i.isdigit():
        numeric += 1
    elif i.isalnum():
        alphanum += 1
```

In [14]:

```
print('ALPHABET' , alphabet)
print('NUMERIC' , numeric)
print('ALPHA-NUMERIC' , alphanum)
```

```
ALPHABET 621
NUMERIC 3
ALPHA-NUMERIC 1
```

In [15]:

```
count2 = len(clean_again)
print("Count of Words in the given Sentence:", count2)
```

```
Count of Words in the given Sentence: 630
```