

Name:- Ujwal Sahu

Branch:- B.VOC(AI&DS)

Division: B

EXPERIMENT NO 2

Title: Demonstrate handling of inconsistent text using python in Machine Learning.

Tools: Anaconda

Theory:

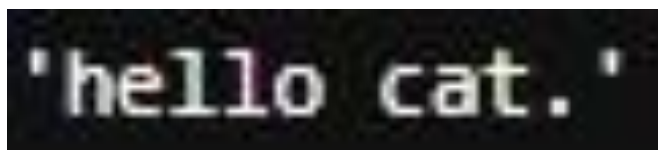
1.Definition: Inconsistent data is data that is inconsistent, conflicted, or incompatible within a dataset or across many datasets. Data inconsistencies can occur for a variety of reasons, including mistakes in data entry, data processing, or data integration. These discrepancies might show as disagreements in data element values, formats, or interpretations. Inconsistent data can lead to faulty analysis, untrustworthy outcomes, and data management challenges. In Python Handling inconsistent text typically involves tasks like correcting inconsistent casing, removing unwanted characters, dealing with extra whitespace, or fixing other formatting issues.

2. Code:

1] Handling of inconsistent text.

```
import re
def clean_text(text):
    text=text.lower()
    text=re.sub(r'\s+', ' ', text)
    text=re.sub(r'([!?,])\1+', r'\1',text)
    text=re.sub(r'\s*([!?,])\s*', r'\1', text)
    text=text.strip()
    return text
text="Hello CAT."
cleaned_text=clean_text(text)
cleaned_text
```

OUTPUT:



```
'hello cat.'
```

2] Example of data with inconsistent column names. import pandas as pd

```
data={'Name':['aman','mansi','shiksha'], 'Age (years)':[18,23,18]}
```

```
df=pd.DataFrame(data)
```

```
df.columns=df.columns.str.strip().str.lower().str.replace(' ','_') df
```

OUTPUT:

```
import pandas as pd
data={'Name':['aman','mansi','shiksha'], 'Age (years)':[18,23,18]}
df=pd.DataFrame(data)
df.columns=df.columns.str.strip().str.lower().str.replace(' ','_')
df
```

	name	age_years
0	aman	18
1	mansi	23
2	shiksha	18

3] Example of data with an outlier. import pandas as pd

```
data={'Name':['shiksha','goldi','sonam','mansi','chandan'],
```

```
'Age':[18,20,18,110,75]} df=pd.DataFrame(data)
```

```
df_no_outliers=df[df['Age']<100] df_no_outliers
```

OUTPUT:

```
import pandas as pd
data={'Name':['shiksha','goldi','sonam','mansi','chandan'], 'Age':[18,20,18,110,75]}
df=pd.DataFrame(data)
df_no_outliers=df[df['Age']<100]
df_no_outliers
```

	Name	Age
0	shiksha	18
1	goldi	20
2	sonam	18
4	chandan	75

4. Conclusion: When dealing with inconsistent or missing data in a dataset, one common approach is to use imputation techniques to fill in those missing values. A simple but useful imputation technique is mean substitution, where the mean value of a variable is used to replace any missing values for that variable. Data cleaning is the process of fixing or removing incorrect, corrupted, incorrectly formatted, duplicate, or incomplete data within a dataset.

For Faculty Use

Correction Parameters	Formative Assessment [40%]	Timely completion of Practical [40%]	Attendance / Learning Attitude [20%]	
Marks Obtained				