## Artificial Intelligence and Machine Learning

Project Report

Semester-IV (Batch-2022)

**Case Study**: - String Methods

[**Url:-**](about:blank) <https://drive.google.com/file/d/1-bE1uC3qDbEx7hr9bswHbJc5-iO6qLm8/view?usp=drive_link>

A red and white sign

Description automatically generated with low confidence

**Supervised By: Submitted By:**

Rajeev Thakur Kartik Taneja

Roll Number - 2210931007

Group - 14

**Department of Computer Science and Engineering**

## Chitkara University Institute of Engineering & Technology,

## Chitkara University, Punjab

**Description about Case Study: -**

* Read the given Technologies Dataset.
* Convert lowercase column , use str.lower()
* Convert lowercase column, use apply()
* Use apply() & lambda function
* Convert pandas column to lowercase , use map()
* Use the str.strip() method.
* Use the str.split() method.
* Use the str.contains() method.
* Use the str.replace() method.
* Use the str.startswith() method
* Use the str.endswith() method
* Use the str.cat() method
* Use the str.get() method
* Use the str.slice() method
* Use the str.find() method.

**Library: -**

* Pandas

**Methods: -**

1. **read\_csv():**

**Description**: Reads a CSV file and converts it into a data frame.

1. **str.lower():**

**Description:** This method returns a copy of the string with all its characters converted to lowercase.

1. **map():**

**Description:** This is a higher-order function in Python that applies a given function to all items in an iterable (like a list) and returns an iterator that yields the results.

1. **map(lower):**

**Description:** This appears to be an incomplete expression. Typically, **map()** is used with a function and an iterable. **lower** might refer to the **str.lower()** method mentioned earlier.

1. **map(upper):**

**Description:** Similar to the previous point, **map(upper)** seems incomplete. It might be an attempt to map the **str.upper()** method over an iterable.

1. **str.len():**

**Description:** This is not a method of string objects in Python. Instead, you would use **len()** function to get the length of a string.

1. **str.strip():**

**Description :** This method returns a copy of the string with leading and trailing whitespace removed.

1. **str.split():**

**Description:** This method splits a string into a list of substrings based on a specified separator.

1. **str.contains():**

**Description:** This method is not a built-in method for Python strings. However, it is a method in the pandas library used for string matching operations on Series and Indexes.

1. **str.replace():**

**Description():**This method returns a copy of the string with all occurrences of a specified substring replaced with another substring.

1. **str.startswith():**

**Description:** This method returns **True** if the string starts with the specified prefix; otherwise, it returns **False**.

1. **str.endswith():**

**Description:** This method returns **True** if the string ends with the specified suffix; otherwise, it returns **False**.

1. **str.cat():**

**Description:** This method concatenates strings in a series with a given separator.

1. **str.get():**

**Description:** This method is not a standard method for Python strings. However, it might refer to the **.get()** method used with dictionaries to retrieve a value for a given key.

1. **str.slice():**

**Description:** This is not a method for Python strings. Instead, you would typically use slicing syntax like **str[start:end]** to extract a portion of the string.

1. **str.find():**

**Description:** This method returns the lowest index in the string where the specified substring is found. If the substring is not found, it returns -1.