**NIKITHA YELDI**

**700758003**

**Summer 2024: CS 5710 –Machine Learning**

**Assignment - 01**

* The above code snippet imports several essential libraries used for data manipulation, visualization, statistical analysis, and machine learning in Python.
* ‘numpy’, ‘pandas’ – Handle and manipulate data efficiently.
* ‘seaborn’, ‘matplotlib’ – Visualize data.
* ‘scipy.stats’ – Perform statistical analysis.
* ‘sklearn’ – Implement and evaluate machine learning models.

A screen shot of a computer code

Description automatically generated

GITHUB LINK:

**https://github.com/nyeldi/machinelearning**

A screenshot of a computer

Description automatically generated

A graph of different sizes and shapes

Description automatically generated with medium confidence

* To obtain and display the basic statistical description of the dataset using pandas, used ‘describe()’ method. It gives statistics for numerical columns for numerical columns, such as count, mean, standard deviation, minimum, and maximum values, along with the 25th, 50th (median), and 75th percentiles.
* The above code reads the CSV file data into pandas DataFrame called ‘df’.
* ‘df.head’ will display the first five rows of the DataFrame.

A screenshot of a computer

Description automatically generated

* To check dataset which contains any null (missing) values, used the isnull() method combined with sum() in pandas. So, this will give count the number of missing values in each column of DataFrame.
* Using df.fillna(df.mean(), inplace=True) is a straightforward and effective method to handle missing values, especially when they are relatively few. It ensures that your dataset remains balanced and ready for further processing or analysis. This technique is particularly useful when you do not want to drop rows or columns with missing values, which might otherwise lead to loss of important data.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

* To filter a DataFrame to select rows where the Calories column has values between 500 and 1000, used boolean indexing with the **.loc** accessor in pandas.
* Also to include the boundary values (i.e., 500 and 1000) in filter, we have to adjust the conditions to use >= and <=:

A screenshot of a data

Description automatically generated

* To compute aggregate statistics for selected columns in a DataFrame, used the agg() function in pandas. It allows to apply multiple aggregation operations like min, max, count, and mean on the specified columns.

A screenshot of a computer

Description automatically generated

* As previous question to filter a DataFrame to select rows where the Calories column has values between 500 and 100, used boolean indexing with the **.loc** accessor in pandas.
* Also to include the boundary values (i.e., 500 and 1000) in filter, we must adjust the conditions to use >= and <=:

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

* To delete the Maxpulse column from the main used df DataFrame and then check the structure and data types, we used the del.
* **print(df.head())**: Shows the first few rows of the modified DataFrame to verify that the Maxpulse column has been removed.
* **print(df.dtypes)**: Displays the data types of the columns in the modified DataFrame

A screenshot of a computer

Description automatically generated

* To create a new DataFrame df\_modified that contains all columns from the original DataFrame df except for the Maxpulse column we use **df.drop(columns=['Maxpulse'])**: which creates a new DataFrame by dropping the Maxpulse column.
* To convert the data type of the Calories column from float64 to int64 in a pandas DataFrame, we used the astype() method.
* This method changes the data type of a Series (column) to the specified type.
* Here **df['Calories'].astype(np.int64)**: Converts the Calories column to integer (int64).

**\*\*\*THANK YOU\*\*\***