
“Techno – Social Excellence”

Marathwada Mitramandal’s

**INSTITUTE OF TECHNOLOGY (MMIT)**

Lohgaon, Pune-411047

“Excellence in the field of AI & DS”

# Department of

# Artificial Intelligence & Data Science

# LABORATORY MANUAL

## 317529: Software Laboratory III

TE AI & DS (2020 Course)

Semester - II

Prepared By

**Ms. R. A. Agrawal**

# VISION & MISSION OF THE INSTITUTE

## VISION

Techno-Social Excellence

## MISSION

- Enhance technology transfer
- Implement entrepreneurship
- Promote global competency
- Integrate innovative pedagogy
- Create excellent human resource

# VISION & MISSION OF AI & DS DEPARTMENT

## VISION

Excellence in the field of Artificial Intelligence and Data Science

## MISSION

- To encourage the students for learning various AI & DS based tool and techniques
- To groom students technologically superior
- To train students in a way to meet the needs of society

## PROGRAM OUTCOMES

**PO1. Engineering knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

**PO2. Problem analysis:** Identify, formulate, research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

**PO3. Design/development of solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

**PO4. Conduct investigations of complex problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

**PO5. Modern tool usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.

**PO6. The engineer and society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

**PO7. Environment and sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts and demonstrate the knowledge of, and need for sustainable development.

**PO8. Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

**PO9. Individual and team work:** Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

**PO10. Communication:** Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

**PO11. Project management and finance:** Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

**PO12. Life-long learning:** Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

# Instructions to students

(DO'S AND DON'TS)

## DO'S:

- Conduct yourself in a responsible manner at all the times in laboratory
- Keep your belongings in the rack.
- Wear your College ID card
- Do entry in the lab register
- Switch OFF your mobile phone during practical session
- Know the theory behind the practical before coming to the lab.
- Shut down the machines properly after completing your practical session
- Arrange your chairs properly before leaving the lab
- Maintain discipline and silence in the lab

## DON'TS:

- Avoid unnecessary talking while doing the practical
- Do not unplug any connections without permission
- Do not upload, delete or alter any documents/files or software installed on machine
- Do not turn off the PCs directly
- Don't talk aloud in lab

# INDEX

| Sr. No. | Title of Assignment |
|---|---|
| | **Group A- Data Science** |
| 1. | **Data Wrangling I**<br>Perform the following operations using Python on any open source dataset (e.g., data.csv)<br>• Import all the required Python Libraries.<br>• Locate an open source data from the web (e.g. https://www.kaggle.com). Provide a cleardescription of the data and its source (i.e., URL of the web site).<br>• Load the Dataset into pandas data frame.<br>• Data Preprocessing: check for missing values in the data using pandas insult(), describe()function to get some initial statistics. Provide variable descriptions. Types of variables etc. Check the dimensions of the data frame.<br>• Data Formatting and Data Normalization: Summarize the types of variables by checking the data types (i.e., character, numeric, integer, factor, and logical) of the variables in the data set. If variables are not in the correct data type, apply proper type conversions.<br>• Turn categorical variables into quantitative variables in Python.<br>• In addition to the codes and outputs, explain every operation that you do in the above steps and explain everything that you do to import/read/scrape the data set. |
| 2. | **Data Wrangling II**<br>Create an "Academic performance" dataset of students and perform the following operations using Python.<br>1. Scan all variables for missing values and inconsistencies. If there are missing values and/or inconsistencies, use any of the suitable techniques to deal with them.<br>2. Scan all numeric variables for outliers. If there are outliers, use any of the suitable techniques to deal with them.<br>3. Apply data transformations on at least one of the variables. The purpose of this transformation should be one of the following reasons: to change the scale for better understanding of the variable, to convert a non-linear relation into a linear one, or to decrease the skewness and convert the distribution into a normal distribution. Reason and document your approach properly. |
| 3. | **Descriptive Statistics - Measures of Central Tendency and variability**<br>Perform the following operations on any open source dataset (e.g., data.csv)<br>1. Provide summary statistics (mean, median, minimum, maximum, standard deviation) for a dataset (age, income etc.) with numeric variables grouped by one of the qualitative (categorical) variable. For example, if your categorical variable is age groups and quantitative variable is income, then provide summary statistics of income grouped bythe age groups. Create a list that contains a numeric value for each response to the categorical variable. |

| | |
|---|---|
| | 2. Write a Python program to display some basic statistical details like percentile, mean, standard deviation etc. of the species of 'Iris-sentosa', 'Iris-versicolor' and 'Iris- versicolor' of iris.csv dataset.<br>Provide the codes with outputs and explain everything that you do in this step. |
| **4** | **Data Analytics I**<br>Create a Linear Regression Model using Python/R to predict home prices using Boston Housing Dataset (https://www.kaggle.com/c/boston-housing). The Boston Housing dataset contains information about various houses in Boston through different parameters. There are 506 samplesand 14 feature variables in this dataset.<br>The objective is to predict the value of prices of the house using the given features. |
| **5** | **Data Analytics II**<br>1. Implement logistic regression using Python/R to perform classification onSocial_Network_Ads.csv dataset.<br>2. Compute Confusion matrix to find TP, FP, TN, FN, Accuracy, Error rate, Precision,<br>Recall on the given dataset. |
| **6** | **Data Analytics III**<br>1. Implement Simple Naïve Bayes classification algorithm using Python/R on iris.csvdataset.<br>2. Compute Confusion matrix to find TP, FP, TN, FN, Accuracy, Error rate, Precision,<br>Recall on the given dataset. |
| **7** | **Text Analytics**<br>1. Extract Sample document and apply following document preprocessing methods: Tokenization, POS Tagging, stop words removal, Stemming and Lemmatization.<br>2. Create representation of document by calculating Term Frequency and Inverse Document<br>Frequency. |
| **8** | **Data Visualization I**<br>1. Use the inbuilt dataset 'titanic'. The dataset contains 891 rows and contains information about the passengers who boarded the unfortunate Titanic ship. Use the Seaborn libraryto see if we can find any patterns in the data.<br>2. Write a code to check how the price of the ticket (column name: 'fare') for each<br>passenger is distributed by plotting a histogram. |
| **9** | **Data Visualization II**<br>1. Use the inbuilt dataset 'titanic' as used in the above problem. Plot a box plot for distribution of age with respect to each gender along with the information about whether they survived or not. (Column names : 'sex' and 'age')<br>Write observations on the inference from the above statistics. |
| **10** | **Data Visualization III**<br>Download the Iris flower dataset or any other dataset into a DataFrame. (e.g., https://archive.ics.uci.edu/ml/datasets/Iris ).<br>Scan the dataset and give the inference as:<br>1. List down the features and their types (e.g., numeric, nominal) available in the dataset. |

| | |
|---|---|
| | 2. Create a histogram for each feature in the dataset to illustrate the feature distributions.<br>3. Create a box plot for each feature in the dataset.<br>Compare distributions and identify outliers. |
| **Group B- Data Analytics – JAVA/SCALA** | |
| **1.** | Create databases and tables, insert small amounts of data, and run simple queries using Impala. |
| **2.** | Design a distributed application using Map-Reduce which processes a log file of a system.. |
| **3.** | Locate dataset (e.g., sample_weather.txt) for working on weather data which reads the textinput files and finds average for temperature, dew point and wind speed. |
| **Group C- Mini Projects/ Case Study – PYTHON/R** | |
| **1.** | Use the following dataset and classify tweets into positive and negative tweets. https://www.kaggle.com/ruchi798/data-science-tweets |
| **2.** | Develop a movie recommendation model using the scikit-learn library in python.<br>Refer dataset<br>https://github.com/rashida048/Some-NLP-Projects/blob/master/movie_dataset.csv |

# Lab Assignment 1

# Title: Data Wrangling I

**PROBLEM STATEMENT:**

Perform the following operations using Python on any open source dataset (e.g., data.csv)
1. Import all the required Python Libraries.
2. Locate an open source data from the web (e.g., https://www.kaggle.com). Provide a clear description of the data and its source (i.e., URL of the web site).
3. Load the Dataset into pandas dataframe.
4. Data Preprocessing: check for missing values in the data using pandas isnull(), describe() function to get some initial statistics. Provide variable descriptions. Types of variables etc. Check the dimensions of the data frame.
5. Data Formatting and Data Normalization: Summarize the types of variables by checking the data types (i.e., character, numeric, integer, factor, and logical) of the variables in the data set. If variables are not in the correct data type, apply proper type conversions.
6. Turn categorical variables into quantitative variables in Python.

## THEORY:

## What is Data Wrangling?

Data Munging, commonly referred to as Data Wrangling, is the cleaning and transforming of one type of data to another type to make it more appropriate into a processed format. Data wrangling involves processing the data in various formats and analyzes and get them to be used with another set of data and bringing them together into valuable insights. It further includes data aggregation, data visualization, and training statistical models for prediction. data wrangling is one of the most important steps of the data science process. The quality of data analysis is only as good as the quality of data itself, so it is very important to maintain data quality.

## NEED FOR WRANGLING:

Wrangling the data is crucial, yet it is considered as a backbone to the entire analysis part. The main purpose of data wrangling is to make raw data usable. In other words, getting data into a shape. 0n average, data scientists spend 75% of their time wrangling the data, which is not a surprise at all. The important needs of data wrangling include,

- The quality of the data is ensured.
- Supports timely decision-making and fastens data insights.
- Noisy, flawed, and missing data are cleaned.
- It makes sense to the resultant dataset, as it gathers data that acts as a preparation stage for the data mining process.

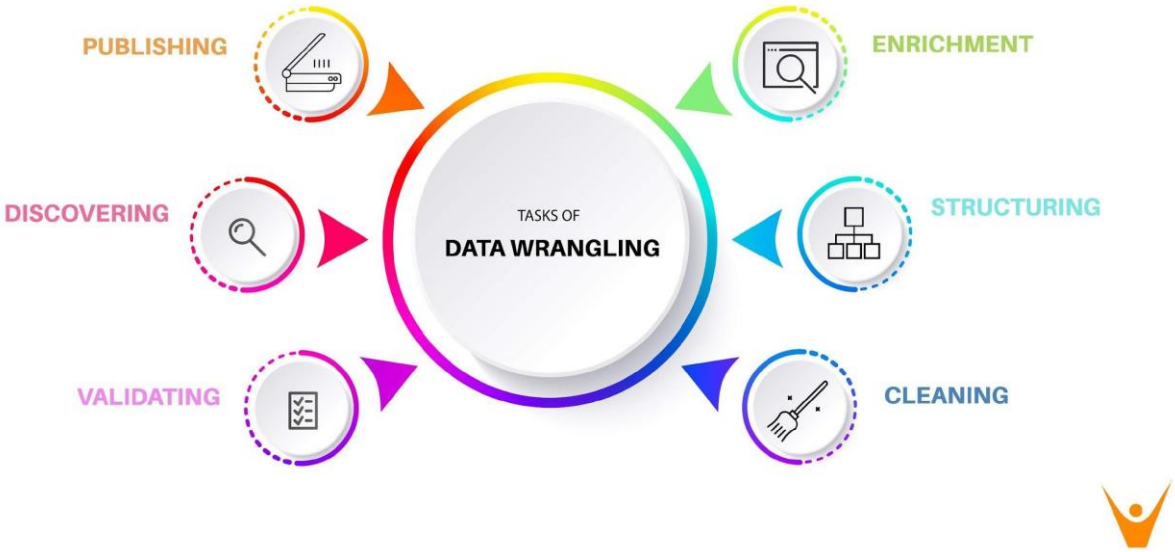
- Helps to make concrete and take a decision by cleaning and structuring raw data into the required format.

- Raw data are pieced together to the required format.

- To create a transparent and efficient system for data management, the best solution is to have all data in a centralized location so it can be used in improving compliance.

- Wrangling the data helps make decisions promptly and helps the wrangler clean, enrich, and transform the data into a perfect picture.

**DATA WRANGLING STEPS:**




**1. DISCOVERING:**

Discovering is a term for an entire analytic process, and it's a good way to learn how to use the data to explore and it brings out the best approach for analytics explorations. It is a step in which the data is to be understood more deeply.

**2. STRUCTURING:**

Raw data is given randomly. There will not be any structure to it in most cases because raw data comes from many formats of different shapes and sizes. The data must be organized in such a manner where the analytics attempt to use it in his analysis part.

**3. CLEANING:**

High-quality analysis happens here where every piece of data is checked carefully and redundancies are removed that don't fit the data for analysis. Data containing the Null values have to be changed either to an empty string or zero and the formatting will be standardized to make the data of higher quality. The goal of data cleaning or remediation is to ensure that there are no possible ways that the final data could be influenced that is to be taken for final analysis.

**4. ENRICHING:**

Enriching is like adding some sense to the data. In this step, the data is derived into new kinds of data from the data which already exits from cleaning into the formatted manner. This is where the data need to strategize that you have in your hand and to make sure that you have is the best-enriched data. The best way to get the refined data is to down sample, upscale it, and finally augur the data.

**5. VALIDATING:**

For analysis and evaluation of the quality of specific data set data quality rules are used. After processing the data, the quality and consistency are verified which establish a strong surface to the security issues. These are to be conducted along multiple dimensions and to adhere to syntactic constraints.

**6. PUBLISHING:**

The final part of the data wrangling is Publishing which gives the sole purpose of the entire wrangling process. Analysts prepare the wrangled data that use further down the line that is its purpose after all. The finalized data must match its format for the eventual data's target. Now the cooked data can be used for analytics.

**DATA WRANGLING IN PYTHON:**

Pandas are an open-source mainly used for Data Analysis. Data wrangling deals with the following functionalities.

- **Data exploration:** Visualization of data is made to analyze and understand the data.

- **Dealing with missing values:** Having Missing values in the data set has been a common issue when dealing with large data set and care must be taken to replace them. It can be replaced either by mean, mode or just labelling them as NaN value.

- **Reshaping data:** Here the data is either modified from the addressing of pre-existing data or the data is modified and manipulated according to the requirements.

- **Filtering data:** The unwanted rows and columns are filtered and removed which makes the data into a compressed format.

- **Others:** After making the raw data into an efficient dataset, it is bought into useful for data visualization, data analyzing, training the model, etc.

**How is Data Preprocessing performed?**

Data Preprocessing is carried out to remove the cause of unformatted real-world data which we discussed above. First of all, let's explain how missing data can be handled during Data Preparation. Three different steps can be executed which are given below -

- **Ignoring the missing record -** It is the simplest and efficient method for handling the missing data. But, this method should not be performed at the time when the number of missing values is immense or when the pattern of data is related to the unrecognized primary root of the cause of the statement problem.

- **Filling the missing values manually -** This is one of the best-chosen methods of Data Preparation process. But there is one limitation that when there are large data set, and missing values are significant then, this approach is not efficient as it becomes a time-consuming task.

- **Filling using computed values -** The missing values can also be occupied by computing mean, mode or median of the observed given values. Another method could be the predictive values in Data Preprocessing are that are computed by using any Machine Learning or Deep Learning tools and algorithms. But one drawback of this approach is that it can generate bias within the data as the calculated values are not accurate concerning the observed values.

## Data Formatting

- **Incorrect data types**

We should make sure that every column is assigned to the correct data type. This can be checked through the property dtypes.

## Data Normalization with Pandas

Data Normalization could also be a typical practice in machine learning which consists of transforming numeric columns to a standard scale. In machine learning, some feature values differ from others multiple times. The features with higher values will dominate the learning process.

Data Normalization involves adjusting values measured on different scales to a common scale.

Normalization applies only to columns containing numeric values. Normalization methods are:

- Simple feature scaling
- min max
- z-score

### Min-Max scaling

$$X_{norm} = \frac{X - X_{min}}{X_{max} - X_{min}}$$

### Z-score normalization

$Z=(x-\mu)/\sigma$

### Simple feature scaling

$$x_{new} = \frac{x_{old}}{x_{max}}$$

## Convert Categorical Variable to Numeric

When we look at the categorical data, the first question that arises to anyone is how to handle those data, because machine learning is always good at dealing with numeric values. We could make machine learning models by using text data. So, to make predictive models we have to convert categorical data into numeric form.

**Method 1: Using replace() method**

Replacing is one of the methods to convert categorical terms into numeric. For example, We will take a dataset of people's salaries based on their level of education. This is an ordinal type of categorical variable. We will convert their education levels into numeric terms.

***Syntax:***

***replace(to_replace=None, value=None, inplace=False, limit=None, regex=False, method='pad')***

**Method 2: Using get_dummies() / One Hot Encoding**

Replacing the values is not the most efficient way to convert them. Pandas provide a method called get_dummies which will return the dummy variable columns.

***Syntax: pandas.get_dummies(data, prefix=None, prefix_sep='_', dummy_na=False, columns=None, sparse=False, drop_first=False, dtype=None)***

**One-Hot Encoding: The Standard Approach for Categorical Data**

One hot encoding is the most widespread approach, and it works very well unless your categorical variable takes on a large number of values One hot encoding creates new (binary) columns, indicating the presence of each possible value from the original data.**It uses get_dummies() Method**

**Method 3:**

**Label Encoding** refers to converting the labels into a numeric form so as to convert them into the machine-readable form. Machine learning algorithms can then decide in a better way how those labels must be operated. It is an important pre-processing step for the structured dataset in supervised learning.

## CONCLUSION:

They will understand how important data wrangling is for data and using different techniques optimized results can be obtained. Hence wrangle the data, before processing it for analysis.

# Lab Assignment 2

# Title: Data Wrangling II

**PROBLEM STATEMENT:**

Create an "Academic performance" dataset of students and perform the following operations using Python.

1. Scan all variables for missing values and inconsistencies. If there are missing values and/or inconsistencies, use any of the suitable techniques to deal with them.

2. Scan all numeric variables for outliers. If there are outliers, use any of the suitable techniques to deal with them.

3. Apply data transformations on at least one of the variables. The purpose of this transformation should be one of the following reasons: to change the scale for better understanding of the variable, to convert a non- linear relation into a linear one, or to decrease the skewness and convert the distribution into a normal distribution.

**THEORY:**

**Working with Missing Data-**

Missing Data can occur when no information is provided for one or more items or for a whole unit. Missing Data is a very big problem in a real-life scenarios. Missing Data can also refer to as NA(Not Available) values in pandas. In DataFrame sometimes many datasets simply arrive with missing data, either because it exists and was not collected or it never existed.

Pandas treat None and NaN as essentially interchangeable for indicating missing or null values. To facilitate this convention, there are several useful functions for detecting, removing, and replacing null values in Pandas DataFrame :

- isnull()
- notnull()
- dropna()
- fillna()
- replace()

**Checking for missing values using isnull() and notnull():-**

In order to check missing values in Pandas DataFrame, a function isnull() and notnull(). Both function help in checking whether a value is NaN or not. These function can also be used in Pandas Series in order to find null values in a series.

1. Checking for missing values using isnull()

In order to check null values in Pandas DataFrame, we use isnull() function this function return dataframe of Boolean values which are True for NaN values.

Dataframe.isnull():-

Syntax: Pandas.isnull("DataFrame Name") or DataFrame.isnull() Parameters: Object to check null values for
Return Type: Dataframe of Boolean values which are True for NaN values

2. Checking for missing values using notnull()

In order to check null values in Pandas Dataframe, we use notnull() function this function return dataframe of Boolean values which are False for NaN values.

Dataframe.notnull():-

Syntax: Pandas.notnull("DataFrame Name") or DataFrame.notnull()

Parameters: Object to check null values for

Return Type: Dataframe of Boolean values which are False for NaN values

3. Filling missing values using fillna(), replace() and interpolate()

In order to fill null values in a datasets, we use fillna(), replace() and interpolate() function these function replace NaN values with some value of their own. All these function help in filling a null values in datasets of a DataFrame.

1. fillna() manages and let the user replace NaN values with some value of their own.

Syntax:

DataFrame.fillna(value=None, method=None, axis=None, inplace=False, limit=None, downcast=None,

Parameters:

value : Static, dictionary, array, series or dataframe to fill instead of NaN.

method : Method is used if user doesn't pass any value. Pandas has different methods

like bfill, backfill or ffill which fills the place with value in the Forward index or Previous/Back respectively. axis: axis takes int or string value for rows/columns. Input can be 0 or 1 for Integer and 'index' or 'columns' for String

inplace: It is a boolean which makes the changes in data frame itself if True.

limit : This is an integer value which specifies maximum number of consequetive forward/backward NaN value fills.

downcast : It takes a dict which specifies what dtype to downcast to which one. Like Float64 to int64.

2. dataframe.replace() function is used to replace a string, regex, list, dictionary, series, number etc. from a dataframe. This is a very rich function as it has many variations.The most powerful thing about this function is that it can work with Python regex (regular expressions).

Syntax: DataFrame.replace(to_replace=None, value=None, inplace=False, limit=None, regex=False, method='pad', axis=None)

Parameters:

to_replace : [str, regex, list, dict, Series, numeric, or None] pattern that we are trying to replace in dataframe.

value : Value to use to fill holes (e.g. 0), alternately a dict of values specifying which value to use for each column (columns not in the dict will not be filled). Regular expressions, strings and lists or dicts of such objects are also allowed.

inplace : If True, in place. Note: this will modify any other views on this object (e.g. a column from a DataFrame). Returns the caller if this is True.

limit : Maximum size gap to forward or backward fill

regex : Whether to interpret to_replace and/or value as regular expressions. If this is True then to_replace must be a string. Otherwise, to_replace must be None because this parameter will be interpreted as a regular expression or a list, dict, or array of regular expressions.

method : Method to use when for replacement, when to_replace is a list.

Returns: filled : NDFrame

4. Dropping missing values using dropna()

Pandas dropna() method allows the user to analyze and drop Rows/Columns with Null values in different ways.

Syntax:

DataFrameName.dropna(axis=0, how='any', thresh=None, subset=None, inplace=False)

Parameters:

axis: axis takes int or string value for rows/columns. Input can be 0 or 1 for Integer and 'index' or 'columns' for String.

how: how takes string value of two kinds only ('any' or 'all'). 'any' drops the row/column if ANY value is Null and 'all' drops only if ALL values are null.

thresh: thresh takes integer value which tells minimum amount of na values to drop.

subset: It's an array which limits the dropping process to passed rows/columns through list.

inplace: It is a boolean which makes the changes in data frame itself if True

**Detect and Remove the Outliers**

An Outlier is a data-item/object that deviates significantly from the rest of the (so-called normal)objects. They can be caused by measurement or execution errors.

Detecting the outliers

Outliers can be detected using visualization, implementing mathematical formulas on the dataset, or using the statistical approach.

**Using Boxplot**

It captures the summary of the data effectively and efficiently with only a simple box and whiskers. Boxplot summarizes sample data using 25th, 50th, and 75th percentiles. One can get insights(quartiles, median, and outliers) into the dataset by just looking at its boxplot.

**Using ScatterPlot**

It is used when you have paired numerical data, or when your dependent variable has multiple values for each reading independent variable, or when trying to determine the relationship between the two variables. In the process of utilizing the scatter plot, one can also use it for outlier detection.

**1. Z-score:** Z- Score is also called a standard score. This value/score helps to understand that how far is the data point from the mean. And after setting up a threshold value one can utilize z score values of data points to define the outliers.

Zscore = (data_point -mean) / std. deviation

**2. IQR (Inter Quartile Range)**

IQR (Inter Quartile Range) Inter Quartile Range approach to finding the outliers is the most commonly used and most trusted approach used in the research field.

IQR = Quartile3 – Quartile1

**What is Interquartile Range IQR?**

IQR is used to measure variability by dividing a data set into quartiles. The data is sorted in ascending order and split into 4 equal parts. Q1, Q2, Q3 called first, second and third quartiles are the values which separate the 4 equal parts.

- Q1 represents the 25th percentile of the data.

- Q2 represents the 50th percentile of the data.
- Q3 represents the 75th percentile of the data. If a dataset has 2n / 2n+1 data points, then

Q1 = median of the dataset.

Q2 = median of n smallest data points. Q3 = median of n highest data points.

IQR is the range between the first and the third quartiles namely Q1 and Q3: IQR = Q3 – Q1. The data points which fall below Q1 – 1.5 IQR or above Q3 + 1.5 IQR are outliers.

Removing the outliers

For removing the outlier, one must follow the same process of removing an entry from the dataset using its exact position in the dataset because in all the above methods of detecting the outliers end result is the list of all those data items that satisfy the outlier definition according to the method used.

How to delete exactly one row in python?

dataframe.drop( row_index, inplace = True)

**Data transformation:-**

Data transformation is the process of converting raw data into a format or structure that would be more suitable for model building and also data discovery in general. Data transformation predominantly deals with normalizing also known as scaling data , handling skewness and aggregation of attributes

**Min Max Scaler - normalization**

MinMaxScaler() is applied when the dataset is not distorted. It normalizes the data into a range between 0 and 1 based on the formula:

**Standard Scaler - standardization**

We use standardization when the dataset conforms to normal distribution. StandardScaler() converts the numbers into the standard form of mean = 0 and variance = 1 based on z-score formula:

x' = (x – mean) / standard deviation.

Robust Scaling- RobustScaler() is more suitable for dataset with skewed distributions and outliers because it transforms the data based on median and quantile, specifically

x' = (x – median) / inter-quartile range.

**Z score normalization:**

Z score normalization is- In Z score normalization, we perform following mathematical transformation.

**Skewness of data:**

skewness() :

Skewness basically gives the shape of normal distribution of values.

If skewness value lies above +1 or below -1, data is highly skewed. If it lies between +0.5 to -0.5, it is moderately skewed. If the value is 0, then the data is symmetric the skewness level, we should know whether it is positively skewed or negatively skewed.

**Positively skewed data:**

If tail is on the right as that of the second image in the figure, it is right skewed data. It is also called positive skewed data.Common transformations of this data include square root, cube root, and log.

a. Cube root transformation:

The cube root transformation involves converting x to x^(1/3). This is a fairly strong transformation with a substantial effect on distribution shape: but is weaker than the logarithm. It can be applied to negative and zero values too. Negatively skewed data.

b. Square root transformation:

Applied to positive values only. Hence, observe the values of column before applying.

c. Logarithm transformation:

The logarithm, x to log base 10 of x, or x to log base e of x (ln x), or x to log base 2 of x, is a strong transformation and can be used to reduce right skewness.

**Negatively skewed data:**

If the tail is to the left of data, then it is called left skewed data. It is also called negatively skewed data.

Common transformations include square , cube root and logarithmic.

a. Square transformation:

The square, x to $x^2$, has a moderate effect on distribution shape and it could be used to reduce left skewness. Another method of handling skewness is finding outliers and possibly removing them.

How to transform features into Normal/Gaussian Distribution:- How to check if a variable is following Normal Distribution

There are various ways in which we can check the distribution of the variables. Some of them are:

- Histogram
- Q-Q plot
- KDE plot
- Skewness

Checking the distribution with Skewness dataframe.skew()

- The variables with skewness > 1 are highly positively skewed.

- The variables with skewness < -1 are highly negatively skewed.
- The variables with 0.5 < skewness < 1 are moderately positively skewed.
- The variables with -0.5 < skewness < -1 are moderately negatively skewed.
- And, the variables with -0.5 < skewness < 0.5 are symmetric i.e normally distributed

**CONCLUSION:**

Students will learn about data transformation techniques and outliers. Techniques to detect & remove outliers. Normal Distribution, Scaling and techniques to transform data.

# Lab Assignment 3

## Title: Descriptive Statistics

**PROBLEM STATEMENT:**

Descriptive Statistics - Measures of Central Tendency and variability perform the following operations on anyopen source dataset (e.g., data.csv)

1. Provide summary statistics (mean, median, minimum, maximum, standard deviation) for a dataset (age, income etc.) with numeric variables grouped by one of the qualitative (categorical) variables. For example,if your categorical variable is age groups and quantitative variable is income, then provide summary statistics of income grouped by the age groups. Create a list that contains a numeric value for each response to the categorical variable.

2. Write a Python program to display some basic statistical details like percentile, mean, standard deviationetc. of the species of 'Iris-setosa', 'Iris-versicolor' and 'Iris-versicolor' of iris.csv dataset.

## THEORY:

**What is Statistics?**

Statistics is the science of collecting data and analyzing them to infer proportions (sample) that are representative of the population. In other words, statistics is interpreting data in order to make predictions for the population.

**There are two branches of Statistics.**

- DESCRIPTIVE STATISTICS: Descriptive Statistics is a statistics or a measure that describes the data.
- INFERENTIAL STATISTICS: Using a random sample of data taken from a population to describeand make inferences about the population is called Inferential Statistics.

**Descriptive Statistics**

Descriptive Statistics is summarizing the data at hand through certain numbers like mean, median etc. so as to make the understanding of the data easier. It does not involve any generalization or inference beyond what is available. This means that the descriptive statistics are just the representation of the data (sample) available and not based on any theory of probability.

**Commonly Used Measures**

**1. Measures of Central Tendency**

**2. Measures of Dispersion (or Variability)**

**Measures of Central Tendency**

A Measure of Central Tendency is a one number summary of the data that typically describes the center of thedata. These one number summary is of three types.

1. Mean: Mean is defined as the ratio of the sum of all the observations in the data to the total number of observations. This is also known as Average. Thus mean is a number around which the entire data set is spread.

2. Median: Median is the point which divides the entire data into two equal halves. One-half of the data is less than the median, and the other half is greater than the same. Median is calculated by first arranging the data in either ascending or descending order.

- If the number of observations is odd, median is given by the middle observation in the sorted form.

- If the number of observations is even, median is given by the mean of the two middle observations in the sorted form.

An important point to note that the order of the data (ascending or descending) does not affect the median

3. Mode: Mode is the number which has the maximum frequency in the entire data set, or in other words,mode is the number that appears the maximum number of times. A data can have one or more than one mode.

## How to calculate summary statistics?

A large number of methods collectively compute descriptive statistics and other related operations on DataFrame. Most of these are aggregations like sum(), mean() etc.

**Functions & Description: To calculate Mean, Standard Deviation, Median, Max, and Min we can applythese functions.**

| Sr.No. | Function | Description |
|---|---|---|
| 1 | count() | Number of non-null observations |
| 2 | sum() | Sum of values |
| 3 | mean() | Mean of Values |
| 4 | median() | Median of Values |
| 5 | mode() | Mode of values |

| | | |
|---|---|---|
| 6 | std() | Standard Deviation of the Values |
| 7 | min() | Minimum Value |
| 8 | max() | Maximum Value |
| 9 | abs() | Absolute Value |
| 10 | prod() | Product of Values |
| 11 | cumsum() | Cumulative Sum |
| 12 | cumprod() | Cumulative Product |

**Using the 'describe()' Method:-**

We can use the describe function to generate the statistics above and apply it to multiple columns simultaneously. It also provides the lower, median and upper percentiles.

## Aggregating statistics grouped by category

**Using 'groupby()' to Aggregate**

Suppose we wanted to know the average runtime for each genre. We can use the 'groupby()' method tocalculate these statistics:

The group by method is used to support this type of operations. More general, this fits in the moregeneral split-apply-combine pattern:

- Split the data into groups
- Apply a function to each group independently
- Combine the results into a data structure

The apply and combine steps are typically done together in pandas.

Grouping can be done by multiple columns at the same time. Provide the column names as a list tothe **groupby()** method.

## Count number of records by category-

- The value_counts() method counts the number of records for each category in a column.
- value_counts is a convenient shortcut to count the number of entries in each category of a variable

## Procedure:

1. Import required libraries
2. Read csv file
3. Provide summary statistics using predefined function like mean(),median(),mode(), describe() etc.
4. Categorize data using groupby() method and provide statistics.

## CONCLUSION:

To summarize, here we discussed how to generate summary statistics using the Pandas library. Here, we discussed how to use pandas methods to generate mean, median, max, min and standard deviation. We also saw describe () method which allows us to generate percentiles, in addition to the mean, median, max, min and standard deviation, for any numerical column. Finally, we showed how to generate aggregate statistics for categorical columns.

# Lab Assignment 4

# Title: Data Analytics I

**PROBLEM STATEMENT:**

Create a Linear Regression Model using Python/R to predict home prices using Boston Housing Dataset (https://www.kaggle.com/c/boston-housing). The Boston Housing dataset contains information about various houses in Boston through different parameters. There are 506 samples and 14 feature variables in this dataset.

**THEORY:**

Machine Learning is a part of Artificial Intelligence (AI), where the model will learn from the data and can predict the outcome. Machine Learning is a study of statistical computer algorithm that improves automatically from the data. Unlike computer algorithms, rely on human beings.

Types of Machine Learning Algorithms

- Supervised Machine Learning

In Supervised Learning, we will have both the independent variable (predictors) and the dependent variable (response). Our model will be trained using both independent and dependent variables. So we can predict the outcome when the test data is given to the model. Here, using the output our model can measure its accuracy and can learn over time. In supervised learning, we will solve both Regression and Classification problems.

- Unsupervised Machine Learning

In Unsupervised Learning, our model will won't be provided an output variable to train. So we can't use the model to predict the outcome like Supervised Learning. These algorithms will be used to analyze the data and find the hidden pattern in it. Clustering and Association Algorithms are part of unsupervised learning.

- Reinforcement Learning

Reinforcement learning is the training of machine learning models which make a decision sequentially. In simple words, the output of the model will depend on the present input, and the next input will depend on the previous output of the model.

**What is Regression?**

Regression analysis is a statistical method that helps us to understand the relationship between dependent and one or more independent variables,

- Dependent Variable

This is the Main Factor that we are trying to predict.

- Independent Variable

These are the variables that have a relationship with the dependent variable.

**What is Linear Regression?**

In Machine Learning lingo, Linear Regression (LR) means simply finding the best fitting line that explains the variability between the dependent and independent features very well or we can say it describes the linear relationship between independent and dependent features, and in linear regression, the algorithm predicts the

continuous features (e.g. Salary, Price), rather than deal with the categorical features (e.g. cat, dog).

**Simple Linear Regression**

Simple Linear Regression uses the slope-intercept (weight-bias) form, where our model needs to find the optimal value for both slope and intercept. So with the optimal values, the model can find the variability between the independent and dependent features and produce accurate results. In simple linear regression, the model takes a single independent and dependent variable. There are many equations to represent a straight line, we will stick with the common equation,

Here, y and x are the dependent variables, and independent variables respectively. b1(m) and b0(c) are slope and y-intercept respectively.

Slope(m) tells, for one unit of increase in x, How many units does it increase in y. When the line is steep, the slope will be higher, the slope will be lower for the less steep line.

Constant(c) means, What is the value of y when the x is zero. How the Model will Select the Best Fit Line?

First, our model will try a bunch of different straight lines from that it finds the optimal line that predicts our data points. For finding the best fit line our model uses the cost function. In machine learning, every algorithm has a cost function, and in simple linear regression, the goal of our algorithm is to find a minimal value for the cost function. And in linear regression (LR), we have many cost functions, but mostly used cost function is MSE(Mean Squared Error). It is also known as a Least Squared Method.

Yi – Actual value, Y^i – Predicted value, n – number of records.

( yi – yi_hat ) is a Loss Function. And you can find in most times people will interchangeably use the word loss and cost function. But they are different, and we are squaring the terms to neglect the negative value.

**Loss Function**

It is a calculation of loss for single training data.

**Cost Function**

It is a calculation of average loss over the entire dataset.

**Steps**

1. Our model will fit all possible lines and find an overall average error between the actual and predicted values for each line respectively.
2. Selects the line which has the lowest overall error. And that will be the best fit line.

## CONCLUSION:

We studied & applied the concepts of linear regression on the Boston housing dataset. Also we calculated the accuracy of the model.

# Lab Assignment 5

# Title: Data Analytics II

**PROBLEM STATEMENT:**

1. Implement logistic regression using Python/R to perform classification on Social_Network_Ads.csv dataset.
2. Compute Confusion matrix to find TP, FP, TN, FN, Accuracy, Error rate, Precision, Recall on the given dataset.

**THEORY:**

**What is Logistic Regression?**

- Logistic Regression: Classification techniques are an essential part of machine learning and data mining applications. Approximately 70% of problems in Data Science are classification problems. There are lots of classification problems that are available, but logistic regression is common and is a useful regression method for solving the binary classification problem.
- Another category of classification is Multinomial classification, which handles the issues where multiple classes are present in the target variable. For example, the IRIS dataset is a very famous example of multi-class classification. Other examples are classifying article/blog/document categories.
- Logistic Regression can be used for various classification problems such as spam detection. Diabetes prediction, if a given customer will purchase a particular product or will they churn another competitor, whether the user will click on a given advertisement link or not, and many more examples are in the bucket.
- Logistic Regression is one of the most simple and commonly used Machine Learning algorithms for two-class classification. It is easy to implement and can be used as the baseline for any binary classification problem. Its basic fundamental concepts are also constructive in deep learning.
- Logistic regression describes and estimates the relationship between one dependent binary variable and independent variables. Logistic regression is a statistical method for predicting binary classes. The outcome or target variable is dichotomous in nature. Dichotomous means there are only two possible classes. For example, it can be used for cancer detection problems. It computes the probability of an event occurring.
- It is a special case of linear regression where the target variable is categorical in nature. It uses a log of odds as the dependent variable. Logistic Regression predicts the probability of occurrence of a binary event utilising a logit function.
- Linear Regression Equation:

Where, y is a dependent variable and x1, x2 ... and Xn are explanatory variables.

- Sigmoid Function:

Apply Sigmoid function on linear regression:

- Differentiate between Linear and Logistic Regression is Linear regression gives you a continuous output, but logistic regression provides a constant output. An example of the continuous output is house price and stock

price. Examples of the discrete output is predicting whether a patient has cancer or not, predicting whether the customer will churn. Linear regression is estimated using Ordinary Least Squares (OLS) while logistic regression is estimated using Maximum Likelihood Estimation (MLE) approach.

- Sigmoid Function The sigmoid function, also called logistic function, gives an 'S' shaped curve that can take any real-valued number and map it into a value between 0 and 1. If the curve goes to positive infinity, y predicted will become 1, and if the curve goes to negative infinity, y predicted will become

0. If the output of the sigmoid function is more than 0.5, we can classify the outcome as 1 or YES, and if it is less than 0.5, we can classify it as 0 or NO. The outputcannotFor example: If the output is 0.75, we can say in terms of probability as: There is a 75 percent chance that a patient will suffer from cancer.

## Types of Logistic Regression

- Binary Logistic Regression: The target variable has only two possible outcomes such as Spam or Not Spam, Cancer or No Cancer.
- Multinomial Logistic Regression: The target variable has three or more nominal categories such as predicting the type of Wine.
- Ordinal Logistic Regression: the target variable has three or more ordinal categories such as restaurant or product rating from 1 to 5.

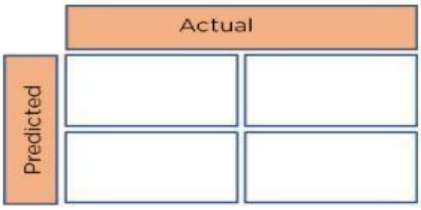
The two limitations of using a linear regression model for classification problems are:

- the predicted value may exceed the range (0,1)
- error rate increases if the data has outliers

## Confusion Matrix Evaluation Metrics

A confusion matrix presents a table layout of the different outcomes of the prediction and results of a classification problem and helps visualize its outcomes.

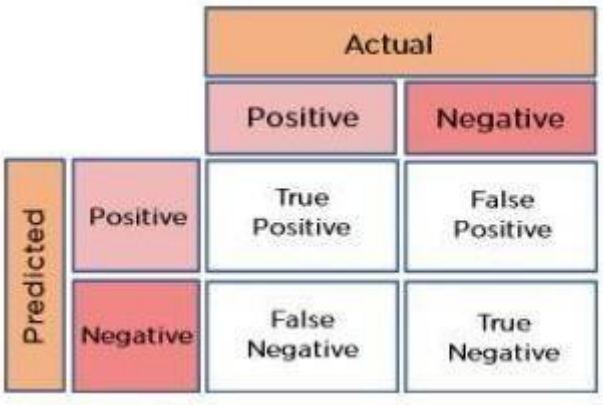
It plots a table of all the predicted and actual values of a classifier




Basic layout of a Confusion Matrix

How to Create a 2x2 Confusion Matrix?

We can obtain four different combinations from the predicted and actual values of a classifier:




Confusion Matrix

- True Positive: The number of times our actual positive values are equal to the predicted positive. You predicted a positive value, and it is correct.
- False Positive: The number of times our model wrongly predicts negative values as positives. You predicted a negative value, and it is actually positive.
- True Negative: The number of times our actual negative values are equal to predicted negative values. You predicted a negative value, and it is actually negative.
- False Negative: The number of times our model wrongly predicts negative values as positives. You predicted a negative value, and it is actually positive.
- Accuracy: The accuracy is used to find the portion of correctly classified values. It tells us how often our classifier is right. It is the sum of all true values divided by total values

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

- Precision: Precision is used to calculate the model's ability to classify positive values correctly. It is the true positives divided by the total number of predicted positive values.

$$\text{Precision} = \frac{TP}{TP + FP}$$

- Recall: It is used to calculate the model's ability to predict positive values. "How often does the model predict the correct positive values?". It is the true positives divided by the total number of actual positive values.

$$\text{Recall} = \frac{TP}{TP + FN}$$

- F1-Score: It is the harmonic mean of Recall and Precision. It is useful when you need to take both

Precision and Recall into account.

$$\text{F1-Score} = \frac{2 * \text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}$$

## CONCLUSION:

In this way we have done data analysis using logistic regression for Social Media Adv. and evaluate the performance of model.

# Lab Assignment 6

# Title: Data Analytics III

**PROBLEM STATEMENT:**

1. Implement Simple Naïve Bayes classification algorithm using Python/R on iris.csv dataset.

2. Compute Confusion matrix to find TP, FP, TN, FN, Accuracy, Error rate, Precision, Recall on the given dataset.

## THEORY:

### Naive Bayes algorithm

In machine learning, Naïve Bayes classification is a straightforward and powerful algorithm for the classification task. Naïve Bayes classification is based on applying Bayes' theorem with strong independence assumption between the features. Naïve Bayes classification produces good results when we use it for textual data analysis such as Natural Language Processing.

Naïve Bayes models are also known as simple Bayes or independent Bayes. All these names refer to the application of Bayes' theorem in the classifier's decision rule. Naïve Bayes classifier applies the Bayes' theorem in practice. This classifier brings the power of Bayes' theorem to machine learning.

2. Naive Bayes algorithm intuition

Naïve Bayes Classifier uses the Bayes' theorem to predict membership probabilities for each class such as the probability that given record or data point belongs to a particular class. The class with the highest probability is considered as the most likely class. This is also known as the Maximum A Posteriori (MAP).

The MAP for a hypothesis with 2 events A and B is MAP (A)

= max (P (A | B))

= max (P (B | A) * P (A))/P (B)

= max (P (B | A) * P (A))

Here, P (B) is evidence probability. It is used to normalize the result. It remains the same, So, removing it would not affect the result.

Naïve Bayes Classifier assumes that all the features are unrelated to each other. Presence or absence of a feature does not influence the presence or absence of any other feature.

## Types of Naive Bayes algorithm

There are 3 types of Naïve Bayes algorithm. The 3 types are listed below:-

1. Gaussian Naïve Bayes

2. Multinomial Naïve Bayes

3. Bernoulli Naïve Bayes

### Gaussian Naïve Bayes algorithm

When we have continuous attribute values, we made an assumption that the values associated with each class are distributed according to Gaussian or Normal distribution. For example, suppose the training data contains a continuous attribute x. We first segment the data by the class, and then compute the mean and variance of x in each class. Let μi be the mean of the values and let σi be the variance of the values associated with the ith class. Suppose we have some observation value xi . Then, the probability distribution of xi given a class can be computed by the following equation –

### Multinomial Naïve Bayes algorithm

With a Multinomial Naïve Bayes model, samples (feature vectors) represent the frequencies with which certain events have been generated by a multinomial (p1, . . . ,pn) where pi is the probability that event i occurs. Multinomial Naïve Bayes algorithm is preferred to use on data that is multinomially distributed. It is one of the standard algorithms which is used in text categorization classification.

### Bernoulli Naïve Bayes algorithm

In the multivariate Bernoulli event model, features are independent boolean variables (binary variables) describing inputs. Just like the multinomial model, this model is also popular for document classification tasks where binary term occurrence features are used rather than term frequencies.

### Applications of Naive Bayes algorithm

Naïve Bayes is one of the most straightforward and fast classification algorithm. It is very well suited for large volume of data. It is successfully used in various applications such as :

1. Spam filtering
2. Text classification

3. Sentiment analysis
4. Recommender systems

It uses Bayes theorem of probability for prediction of unknown class.

### Confusion Matrix

- True Positive: The number of times our actual positive values are equal to the predicted positive. You predicted a positive value, and it is correct.
- False Positive: The number of times our model wrongly predicts negative values as positives. You

predicted a negative value, and it is actually positive.

- True Negative: The number of times our actual negative values are equal to predicted negative values. You predicted a negative value, and it is actually negative.
- False Negative: The number of times our model wrongly predicts negative values as positives. You predicted a negative value, and it is actually positive.
- Accuracy: The accuracy is used to find the portion of correctly classified values. It tells us how often our classifier is right. It is the sum of all true values divided by total values.
- Precision: Precision is used to calculate the model's ability to classify positive values correctly. It is the true positives divided by the total number of predicted positive values.
- Recall: It is used to calculate the model's ability to predict positive values. "How often does the model predict the correct positive values?". It is the true positives divided by the total number of actual positive values.
- F1-Score: It is the harmonic mean of Recall and Precision. It is useful when you need to take both Precision and Recall into account.

## CONCLUSION:

In this way we have learned and performed data analysis using Naive Bayes Algorithm for Iris dataset and evaluated the performance of the model.

# Lab Assignment 7

# Title: Text Analytics

**PROBLEM STATEMENT:**

1. Extract Sample document and apply following document preprocessing methods: Tokenization, POS Tagging, stop words removal, Stemming and Lemmatization.
2. Create representation of documents by calculating Term Frequency and Inverse Document Frequency.

**THEORY:**

## Basic concepts of Text Analytics

- One of the most frequent types of day-to-day conversion is text communication. In our everyday routine, we chat, message, tweet, share status, email, create blogs, and offer opinions and criticism. All of these actions lead to a substantial amount of unstructured text being produced. It is critical to examine huge amounts of data in this sector of the online world and social media to determine people's opinions.
- Text mining is also referred to as text analytics. Text mining is a process of exploring sizable textual data and finding patterns. Text Mining processes the text itself, while NLP processes with the underlying metadata. Finding frequency counts of words, length of the sentence, presence/absence of specific words is known as text mining. Natural language processing is one of the components of text mining. NLP helps identify sentiment, finding entities in the sentence, and category of blog/article. Text mining is preprocessed data for text analytics. In Text Analytics, statistical and machine learning algorithms are used to classify information.

### Text Analysis Operations using natural language toolkit

NLTK(natural language toolkit) is a leading platform for building Python programs to work with human language data. It provides easy-to-use interfaces and lexical resources such as WordNet, along with a suite of text processing libraries for classification, tokenization, stemming, tagging, parsing, and semantic reasoning and many more. Analysing movie reviews is one of the classic examples to demonstrate a simple NLP Bag-of-words model, on movie reviews.

## Tokenization:

- Tokenization is the first step in text analytics. The process of breaking down a text paragraph into smaller chunks such as words or sentences is called Tokenization. Token is a single entity that is the building blocks for a sentence or paragraph.
- Sentence tokenization : split a paragraph into list of sentences using sent_tokenize() method

## Stop words removal:

- Stopwords considered as noise in the text. Text may contain stop words such as is, am, are, this, a, an, the, etc. In NLTK for removing stopwords, you need to create a list of stopwords and filter out your list of tokens from these words.

## Stemming and Lemmatization

- Stemming is a normalization technique where lists of tokenized words are converted into shortened root words to remove redundancy. Stemming is the process of reducing inflected (or sometimes derived) words to their word stem, base or root form. A computer program that stems word may be called a stemmer. E.g. A stemmer reduces the words like fishing, fished, and fisher to the stem fish. The stem need not be a word, for example the Porter algorithm reduces, argue, argued, argues, arguing, and argus to the stem argu .
- Lemmatization in NLTK is the algorithmic process of finding the lemma of a word depending on its meaning and context. Lemmatization usually refers to the morphological analysis of words, which aims to remove inflectional endings. It helps in returning the base or dictionary form of a word known as the lemma. Eg. Lemma for studies is study

## Lemmatization Vs Stemming

Stemming algorithm works by cutting the suffix from the word. In a broader sense cuts either the beginning or end of the word. On the contrary, Lemmatization is a more powerful operation, and it takes into consideration morphological analysis of the words. It returns the lemma which is the base form of all its inflectional forms. In-depth linguistic knowledge is required to create dictionaries and look for the proper form of the word. Stemming is a general operation while lemmatization is an intelligent operation where the proper form will be looked in the dictionary. Hence, lemmatization helps in forming better machine learning features.

## POS Tagging

POS (Parts of Speech) tell us about grammatical information of words of the sentence by assigning specific token (Determiner, noun, adjective , adverb , verb,Personal Pronoun etc.) as tag (DT,NN ,JJ,RB,VB,PRP etc) to each words. Word can have more than one POS depending upon the context where it is used. We can use POS tags as statistical NLP tasks. It distinguishes a sense of word which is very helpful in text realization and infer semantic information from text for sentiment analysis.

## Text Analysis Model using TF-IDF

Term frequency–inverse document frequency(TFIDF) , is a numerical statistic that is intended to reflect how important a word is to a document in a collection or corpus.

**Term Frequency (TF)**

It is a measure of the frequency of a word (w) in a document (d). TF is defined as the ratio of a word's occurrence in a document to the total number of words in a document. The denominator term in the formula is to normalize since all the corpus documents are of different lengths.

**Inverse Document Frequency (IDF)**

It is the measure of the importance of a word. Term frequency (TF) does not consider the importance of words. Some words such as' of', 'and', etc. can be most frequently present but are of little significance. IDF provides weightage to each word based on its frequency in the corpus D.

**Term Frequency — Inverse Document Frequency (TFIDF)**

It is the product of TF and IDF. TFIDF gives more weight-age to the word that is rare in the corpus (all the

documents). TFIDF provides more importance to the word that is more frequent in the document.

**CONCLUSION:**

We have performed Text Analysis experiment using TF-IDF algorithm

# Lab Assignment 8

## Title: Data Visualization I

**PROBLEM STATEMENT:**

1. Use the inbuilt dataset 'titanic'. The dataset contains 891 rows and contains information about the passengers who boarded the unfortunate Titanic ship. Use the Seaborn library to see if we can find any patterns in the data.

2. Write a code to check how the price of the ticket (column name: 'fare') for each passenger is distributed by plotting a histogram.

**THEORY:**

Data visualization is the graphical representation of information and data. By using visual elements like charts, graphs, and maps, data visualization tools provide an accessible way to see and understand trends, outliers, and patterns in data.

In the world of Big Data, data visualization tools and technologies are essential to analyze massive amounts of information and make data-driven decisions.

Common general types of data visualization:

- Charts
- Tables
- Graphs
- Maps
- Infographics
- Dashboards

More specific examples of methods to visualize data:

- Area Chart
- Bar Chart
- Cartogram
- Gantt Chart
- Heat Map
- Highlight Table Histogram
- Scatter Plot (2D or 3D)

Seaborn is a Python data visualization library based on matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphics. Seaborn helps you explore and understand your data. Its plotting functions operate on dataframes and arrays containing whole datasets and internally perform the necessary semantic mapping and statistical aggregation to produce informative plots. Its dataset-oriented, declarative API lets you focus on what the different elements of your plots mean, rather than on the details of how to draw them.

By convention, it is imported with the shorthand sns.

```
# Import seaborn

import seaborn as sns
```

Behind the scenes, seaborn uses matplotlib to draw its plots. For interactive work, it’s recommended to use a Jupyter/IPython interface in matplotlib mode, or else you’ll have to call matplotlib.pyplot.show() when you want to see the plot.

**Bar Chart:**

Bar charts represent categorical data with rectangular bars whose lengths are proportional to the values they represent. They are used to compare the magnitudes of different categories.

Bar charts are effective for displaying discrete data categories and comparing their frequencies or values.

**Histogram:**

Histograms are similar to bar charts but are used to represent the distribution of continuous data by dividing the data into intervals or bins and plotting the frequencies of observations within each interval.

Histograms are useful for visualizing the distribution of numerical data and identifying patterns such as skewness, central tendency, and variability.

**Line Chart:**

Line charts connect data points with straight lines, typically used to represent trends or changes over time.

Line charts are effective for visualizing time-series data and showing trends, patterns, and fluctuations over time.

**Scatter Plot:**

Scatter plots display individual data points as dots on a two-dimensional plane, with one variable plotted on the x-axis and another variable plotted on the y-axis. They are used to visualize relationships and correlations between two continuous variables.

Scatter plots are useful for identifying patterns, trends, and relationships between variables and for detecting outliers.

**Pie Chart:**

Pie charts represent parts of a whole by dividing a circle into sectors, with each sector's angle proportional to the percentage or proportion of the total it represents.

Pie charts are effective for displaying the composition or distribution of categorical data and comparing the relative sizes of different categories.

**Heatmap:**

Heatmaps visualize data using colors to represent the magnitude of values in a matrix or grid. Darker colors typically represent higher values, while lighter colors represent lower values.

Heatmaps are useful for visualizing large datasets, identifying patterns, and exploring relationships between variables, especially in correlation matrices or spatial data.

**Box Plot (Box-and-Whisker Plot):**

Box plots display the distribution of numerical data and identify outliers, quartiles, and the median. They consist of a box representing the interquartile range (IQR) and "whiskers" extending to the minimum and maximum values within a specified range.

Box plots are useful for comparing distributions, detecting outliers, and summarizing the spread and central tendency of numerical data.

Now, let’s perform the operations in the problem statement on our data set.

- Loading the dataset and libraries -
- Some patterns can be seen by performing various operations like-
- Assign a variable to x to plot a univariate distribution along the x axis:
- Check how well the histogram represents the data by specifying a different bin width

**CONCLUSION:**

We have successfully implemented operations of the ‘seaborn’ library on the ‘titanic’ dataset, and explored some patterns in the data. We have also successfully plotted a histogram to see the ticket price distribution.

# Lab Assignment 9

## Title: Data Visualization II

**PROBLEM STATEMENT:**

1. Use the inbuilt dataset 'titanic' as used in the above problem. Plot a box plot for distribution of age with respect to each gender along with the information about whether they survived or not. (Column names : 'sex' and 'age')

2. Write observations on the inference from the above statistics.

**THEORY:**

Data visualization is the graphical representation of information and data. By using visual elements like charts, graphs, and maps, data visualization tools provide an accessible way to see and understand trends, outliers, and patterns in data.

In the world of Big Data, data visualization tools and technologies are essential to analyze massive amounts of information and make data-driven decisions.

Common general types of data visualization:

- Charts
- Tables
- Graphs
- Maps
- Infographics
- Dashboards

More specific examples of methods to visualize data:

- Area Chart
- Bar Chart
- Cartogram
- Gantt Chart
- Heat Map
- Highlight
- Table
- Histogram
- Scatter Plot (2D or 3D)

Seaborn is a Python data visualization library based on matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphics. Seaborn helps you explore and understand your data. Its plotting functions operate on dataframes and arrays containing whole datasets and internally perform the necessary semantic mapping and statistical aggregation to produce informative plots. Its dataset-oriented, declarative API lets you focus on what the different elements of your plots mean, rather than on the details of how to draw them.

By convention, it is imported with the shorthand sns.

# Import seaborn

import seaborn as sns

Behind the scenes, seaborn uses matplotlib to draw its plots. For interactive work, it’s recommended to use a Jupyter/IPython interface in matplotlib mode, or else you’ll have to call matplotlib.pyplot.show() when you want to see the plot.

To plot a box plot for the distribution of age with respect to each gender and survival status on the Titanic dataset, you can use Python libraries such as Pandas and Matplotlib. Here's a step-by-step guide to do that:

- First, load the Titanic dataset into a Pandas DataFrame. You can use the seaborn library to load the dataset directly.
- Filter the dataset to include only the columns "Age," "Sex," and "Survived."
- Use the seaborn library to create a box plot with "Sex" on the x-axis, "Age" on the y-axis, and different colors for the "Survived" categories.

**Inferences -**

Let's try to understand the box plot for female. The first quartile starts at around 5 and ends at 22 which means that 25% of the passengers are aged between 5 and 25. The second quartile starts at around 23 and ends at around 32 which means that 25% of the passengers are aged between 23 and 32. Similarly, the third quartile starts and ends between 34 and 42, hence 25% passengers are aged within this range and finally the fourth or last quartile starts at 43 and ends around 65.

If there are any outliers or the passengers that do not belong to any of the quartiles, they are called outliers and are represented by dots on the box plot.

Now in addition to the information about the age of each gender, you can also see the distribution of the passengers who survived. For instance, you can see that among the male passengers, on average more younger people survived as compared to the older ones. Similarly, you can see that the variation among the age of female passengers who did not survive is much greater than the age of the surviving female passengers.

**CONCLUSION:**

We have successfully implemented operations of the ‘seaborn’ library on the ‘titanic’ dataset, and explored some patterns in the data. We have also successfully plotted a histogram to see the ticket price distribution

# Lab Assignment 10

## Title: Data Visualization III

**PROBLEM STATEMENT:**

Download the Iris flower dataset or any other dataset into a DataFrame. Scan the dataset and give the inference as:

1. List down the features and their types (e.g., numeric, nominal) available in the dataset.
2. Create a histogram for each feature in the dataset to illustrate the feature distributions.
3. Create a boxplot for each feature in the dataset.
4. Compare distributions and identify outliers.

THEORY:

**Histogram:-**

Pandas.DataFrame.hist() function is useful in understanding the distribution of numeric variables. This function splits up the values into the numeric variables. Its main functionality is to make the Histogram of a given Data frame.

The distribution of data is represented by Histogram. When Function Pandas DataFrame.hist() is used, it automatically calls the function matplotlib.pyplot.hist() on each series in the DataFrame

Syntax: DataFrame.hist(data, column=None, by=None, grid=True, xlabelsize=None, xrot=None, ylabelsize=None, yrot=None, ax=None, sharex=False, sharey=False, figsize=None, layout=None, bins=10, backend=None, legend=False, **kwargs)

Parameters:

data: DataFrame column: str or sequence

xlabelsize: int, default None ylabelsize: int, default None

ax: Matplotlib axes object, default None

**kwargs

All other plotting keyword arguments to be passed to matplotlib.pyplot.hist().

Return:

matplotlib.AxesSubplot or numpy.ndarray

**Box Plot :-**

Box Plot is the visual representation of the depicting groups of numerical data through their quartiles. Boxplot is also used for detect the outlier in data set. It captures the summary of the data efficiently with a simple box and whiskers and allows us to compare easily across groups. Boxplot summarizes a sample data using 25th, 50th and 75th percentiles. These percentiles are also known as the lower quartile, median and upper quartile.

A box plot consist of 5 things.

- Minimum
- First Quartile or 25%

- Median (Second Quartile) or 50%
- Third Quartile or 75%
- Maximum

Draw the boxplot using seaborn library:

Syntax :

seaborn.boxplot(x=None, y=None, hue=None, data=None, order=None, hue_order=None, orient=None, color=None, palette=None, saturation=0.75, width=0.8, dodge=True, fliersize=5, linewidth=None, whis=1.5, notch=False, ax=None, **kwargs)

Parameters:

x = feature of dataset y = feature of dataset

hue = feature of dataset

data = dataframe or full dataset color = color name

Identify outliers:-

**Detect and Remove the Outliers using Python**

An Outlier is a data-item/object that deviates significantly from the rest of the (so-called normal)objects. They can be caused by measurement or execution errors. The analysis for outlier detection is referred to as outlier mining. There are many ways to detect the outliers, and the removal process is the data frame same as removing a data item from the panda's data frame.

Here pandas data frame is used for a more realistic approach as in real-world project need to detect the outliers arouse during the data analysis step, the same approach can be used on lists and series-type objects.

**Detecting the outliers**

Outliers can be detected using visualization, implementing mathematical formulas on the dataset, or using the statistical approach. All of these are discussed below.

**Visual Inspection:**

Visualizing the data using scatter plots, histograms, box plots, or QQ plots can help identify outliers. Data points that fall far from the main cluster or distribution are potential outliers.

**Statistical Methods:**

Z-Score: Calculate the z-score for each data point, representing the number of standard deviations away from the mean. Data points with z-scores above a certain threshold (e.g., 3) are considered outliers.

IQR (Interquartile Range): Calculate the IQR, the difference between the 75th and 25th percentiles. Data points outside a certain range of the IQR (e.g., 1.5 times the IQR) are considered outliers.

**Machine Learning Models:**

Train a regression, clustering, or density-based model and identify data points with high residuals, anomalies, or low densities as outliers.

**Domain Knowledge:**

Use domain-specific knowledge or business rules to identify outliers. For example, in a temperature dataset, values outside a certain range might be considered outliers based on knowledge of normal temperature ranges.

**Removal Techniques:**

**Deletion:**

Delete outliers from the dataset. This method can be used when outliers are due to errors or data entry mistakes. However, it may lead to loss of information and reduced sample size.

**Transformation:**

Transform the data using mathematical functions such as logarithmic, square root, or Box-Cox transformations to reduce the impact of outliers while preserving the overall distribution.

**Winsorization:**

Winsorize the data by replacing outliers with the nearest values within a specified percentile range (e.g., replacing values above the 95th percentile with the 95th percentile value).

**Imputation:**

Replace outliers with estimated values using interpolation, mean, median, or regression imputation methods. This method is useful when outliers cannot be deleted, and missing values need to be replaced.

**Clustering:**

Assign outliers to separate clusters or label them as noise using clustering algorithms such as DBSCAN or isolation forest.

**Robust Models:**

Use robust statistical methods or machine learning models that are less sensitive to outliers, such as robust regression, random forests, or support vector machines.

**CONCLUSION:**

We have successfully implemented operations on the ‘iris’ dataset, also we have successfully plotted a histogram, boxplot and identified outliers.

**Group B**

**Lab Assignment 1**

**Title : Database Querying using Impala**

## Problem Statement

Create databases and tables, insert small amounts of data, and run simple queries using Impala.

## Theory

Impala is an open-source, massively parallel processing SQL query engine for data stored in Hadoop clusters. It is designed to provide high-performance, interactive SQL queries directly on data stored in Hadoop Distributed File System (HDFS) or HBase without requiring data movement or transformation.

Here are some key points about Impala:

Real-Time Querying: Impala enables real-time querying of data stored in Hadoop, providing users with fast response times for their SQL queries. This is achieved through its massively parallel processing architecture.

SQL Compatibility: Impala is compatible with the SQL-92 standard, making it easy for users who are familiar with SQL to query data stored in Hadoop without needing to learn new query languages or tools.

Integration with Hadoop Ecosystem: Impala integrates seamlessly with other components of the Hadoop ecosystem, such as HDFS for storage, HBase for NoSQL data storage, and Hive for metadata management. It can also work alongside tools like Apache Spark, Apache Kafka, and Apache Flume.

Interactive Analytics: Impala is optimized for interactive analytics workloads, allowing users to explore and analyze large datasets interactively using familiar SQL syntax. This makes it suitable for use cases such as ad-hoc analysis, business intelligence reporting, and data exploration.

Massively Parallel Processing (MPP): Impala employs a massively parallel processing architecture to distribute query execution across multiple nodes in a Hadoop cluster. This parallelism enables Impala to process large volumes of data in parallel, resulting in fast query execution times.

User-Friendly Interfaces: Impala provides various interfaces for interacting with the query engine, including a command-line shell (impala-shell), JDBC/ODBC drivers for integration with third-party

tools and applications, and web-based interfaces such as Hue.

Overall, Impala is a powerful tool for performing real-time SQL queries on data stored in Hadoop clusters, enabling users to derive insights and make data-driven decisions efficiently.

## Steps for Installation of Impala

Installing Impala typically involves several steps, including setting up prerequisites, downloading the Impala software, configuring the environment, and starting the necessary services. Below are general steps for installing Impala, but keep in mind that specific instructions may vary depending on your environment and the distribution of Hadoop you are using (e.g., Cloudera, Hortonworks, Apache Hadoop).

Ensure Prerequisites:

Check compatibility with your Hadoop distribution and version.
Ensure that your cluster meets the hardware and software requirements specified by Impala.

Download Impala:

Download the Impala parcel or package suitable for your Hadoop distribution from the official Impala website or repository.

Install Impala Services:

If you're using Cloudera Manager, you can install Impala services through the Cloudera Manager interface.
If you're installing manually, follow the installation instructions provided by your Hadoop distribution or Impala documentation.

Configure Impala:

Configure Impala settings such as memory limits, CPU resources, and other parameters based on your cluster requirements. This may involve editing configuration files such as impala-conf.xml.
Ensure that Impala daemons are configured to communicate with other Hadoop services in your cluster, such as HDFS, Hive Metastore, and HBase.
Start Impala Services:

If you're using Cloudera Manager, start Impala services through the Cloudera Manager interface.

If you're managing services manually, start the necessary Impala daemons (impalad, statestored, catalogd) on each node in your cluster.

Verify Installation:

Check the status of Impala services to ensure they are running without errors.
Use command-line tools like impala-shell to connect to Impala and run basic queries to verify that it's functioning correctly.
You can also use web-based interfaces like Hue to interact with Impala and run queries.

Testing and Troubleshooting:

Perform thorough testing of Impala functionality, including running queries on sample datasets and verifying performance.
Monitor system logs and metrics to identify any issues or performance bottlenecks, and troubleshoot as needed.

Security Configuration (Optional):

Configure security settings such as authentication, authorization, and encryption based on your organization's requirements and best practices.
Integrate Impala with Kerberos for authentication and Sentry for fine-grained access control if necessary.

## Conclusion

Thus we have studied Creating databases and tables, inserting small amounts of data, and running simple queries using Impala.

## Group B

## Lab Assignment 2

## Title : SCALA program using Apache Spark Framework

# Problem Statement

Write a simple program in SCALA using Apache Spark framework

# Theory

Apache Spark is an open-source distributed computing framework designed for processing large-scale data analytics workloads. It provides an easy-to-use interface, supports various programming languages such as Java, Scala, Python, and R, and offers a wide range of libraries for diverse data processing tasks. Spark is known for its speed, scalability, and fault tolerance, making it suitable for a wide range of use cases, including batch processing, real-time stream processing, machine learning, and graph processing.
Here's an overview of Apache Spark along with installation steps:

**Components of Apache Spark:**

1. **Spark Core**: This is the foundational component of Spark that provides distributed task scheduling, fault recovery, memory management, and basic I/O functionalities. It also contains the resilient distributed dataset (RDD) API, which is the primary data abstraction in Spark.
2. **Spark SQL**: Spark SQL provides a high-level API for interacting with structured data, enabling users to run SQL queries on Spark RDDs and DataFrame/Dataset APIs. It supports various data formats and integrates with external data sources like Hive, Parquet, JSON, and JDBC.
3. **Spark Streaming**: Spark Streaming enables real-time stream processing and analysis of data streams. It ingests data from sources like Kafka, Flume, Kinesis, and performs transformations and computations using micro-batch processing.
4. **Spark MLlib**: MLlib is Spark's machine learning library that provides scalable implementations of machine learning algorithms for classification, regression, clustering, collaborative filtering, and more. It's built on top of Spark Core and supports both batch and streaming data processing.
5. **Spark GraphX**: GraphX is Spark's API for graph processing and analytics. It provides optimized

graph computation primitives and algorithms for processing large-scale graph data.

**Installation Steps:**

**Install Scala**

Step 1) java -version

Step 2) Install Scala from the apt repository by running the following commands to search for scala and install it.

sudo apt search scala ⇒ Search for the package

sudo apt install scala ⇒ Install the package

Step 3) To verify the installation of Scala, run the following command.

scala -version

**Apache Spark Framework Installation**

Apache Spark is an open-source, distributed processing system used for big data workloads. It utilizes in-memory caching, and optimized query execution for fast analytic queries against data of any size.

Step 1) Now go to the official Apache Spark download page and grab the latest version (i.e. 3.2.1) at the time of writing this article. Alternatively, you can use the wget command to download the file directly in the terminal.

wget https://apachemirror.wuchna.com/spark/spark-3.2.1/spark-3.2.1-bin-hadoop2.7.tgz

Step 2) Extract the Apache Spark tar file.

tar -xvzf spark-3.1.1-bin-hadoop2.7.tgz

Step 3) Move the extracted Spark directory to /opt directory.

sudo mv spark-3.1.1-bin-hadoop2.7 /opt/spark

Configure Environmental Variables for Spark

Step 4) Now you have to set a few environmental variables in .profile file before starting up the spark.

echo "export SPARK_HOME=/opt/spark" >> ~/.profile

echo "export PATH=$PATH:/opt/spark/bin:/opt/spark/sbin" >> ~/.profile

echo "export PYSPARK_PYTHON=/usr/bin/python3" >> ~/.profile

Step 5) To make sure that these new environment variables are reachable within the shell and available to Apache Spark, it is also mandatory to run the following command to take recent changes into effect.

source ~/.profile

Step 6) ls -l /opt/spark

Start Apache Spark in Ubuntu

Step 7) Run the following command to start the Spark master service and slave service.

start-master.sh

start-workers.sh spark://localhost:7077

(if workers not starting then remove and install openssh:

sudo apt-get remove openssh-client openssh-server

sudo apt-get install openssh-client openssh-server)

Step 8) Once the service is started go to the browser and type the following URL access spark page. From the page, you can see my master and slave service is started.

http://localhost:8080/

Step 9) You can also check if spark-shell works fine by launching the spark-shell command.

Spark-shell

## Conclusion

Thus we have studied installation of Apache Spark Framework and a simple SCALA program using it.