TITLE	Data Wrangling I
PROBLEM STATEMENT/ DEFINITION	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' data frame. 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. 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.
OBJECTIVE	1. To do pre-processing on the given dataset. 2. To provide initial statistics 3. Data formatting and type conversions on the dataset columns.
S/W PACKAGES AND HARDWARE APPARATUS USED	S/W- Jupyter Notebook/ Weka/ Python OS-LINUX 64 bit OS H/W: Core 2 DUO/i3/i5/i7 64-bit processor
REFERENCES	1. CHIRAG SHAH, "A HANDS-ON INTRODUCTION TO DATA SCIENCE",ISBN 978-1-108-47244-9 2. Wes McKinney and the Pandas Development Team, "Pandas: powerful Python data analysis toolkit" 3. https://pandas.pydata.org/
STEPS	**Refer to student activity flow chart if found necessary by subject teacher and relevant to the subject manual. Describe steps only.**
INSTRUCTIONS FOR WRITING JOURNAL	1. Title 2. Problem statement 3. Learning objective 4. Learning outcome 5. Theory (includes methods, libraries and functions, 6. Analysis (as per assignment), 7. Conclusion.

1. Title: Data Wrangling I

2. 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' data frame.
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.

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.

3. Learning objective:

1. To do pre-processing on the given dataset.
2. To provide initial statistics
3. Data formatting and type conversions on the dataset columns.

4. Learning outcome:

After performing this assignment students will be able to:

- Do preprocessing on the dataset.
- Provide initial statistics of all columns in the dataset.
- Apply data formatting and type conversions on the dataset columns.

5. Theory:

Libraries: Pythhon libraries Pandas

Pandas: **Pandas** is a Python library. **Pandas** is used to analyze data. Pandas allows importing data from various file formats such as comma-separated values, JSON, SQL database tables or queries, and Microsoft Excel.

Methods and functions:

Pandas.read_csv(): to load csv file into Pandas dataframe.

Shape(): Pandas shape function provides the shape of the dataset in terms of number of rows and columns.

dtypes: Pandas dtype attribute provides datatype of each column in the dataset.

Isnull(): to check presence of null values in the dataset.

describe(): The describe() method is **used for calculating some statistical data like percentile, mean and std of the numerical values of the Series or DataFrame**. It analyzes both numeric and object series and also the DataFrame column sets of mixed data types.

Converting column from object type to numerical data type: **pandas.to_numeric() function is used to convert object data type to numeric data type**. This function will try to change non-numeric objects (such as strings) into integers or floating-point numbers as appropriate.
Converting object data type to datetime data type:
Pandas to_datetime()method is used to convert string with date and time to datetime type.

Conversion of categorical feature/column to numeric column:
1. **Label Encoder:** It is used to transform non-numerical labels to numerical labels (or nominal categorical variables). Numerical labels are always between 0 and n_classes-1. This approach is more flexible because it allows encoding as many category columns as you would like and choose how to label the columns using a prefix. Proper naming will make the rest of the analysis just a little bit easier.
2. **Dummy Coding:** Dummy coding is a commonly used method for converting a categorical input variable into continuous variable. 'Dummy', as the name suggests is a duplicate variable which represents one level of a categorical variable. Presence of a level is represent by 1 and absence is represented by 0. For every level present, one dummy variable will be created.
3. **One-Hot Encoder:** Though label encoding is straight but it has the disadvantage that the numeric values can be misinterpreted by algorithms as having some sort of hierarchy/order in them. This ordering issue is addressed in another common alternative approach called 'One-Hot Encoding'. In this strategy, each category value is converted into a new column and assigned a 1 or 0 (notation for true/false) value to the column.

6. Analysis:

Observed missing values in the dataset and deal with them. Observed descriptive statistics of numerical columns in the dataset and the categorical features in the dataset and do conversion.

7. Conclusion:

Using python libraries Pandas dataset shape, datatypes are observed. Dataset is cleaned after finding missing values, descriptive cs are observed, columns data type is changed as per the requirement, categorical columns are converted to numeric type.

TITLE	Data Wrangling II
PROBLEM STATEMENT/ DEFINITION	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. Reason and document your approach properly.
OBJECTIVE	1. To do data cleansing on the given dataset. 2. To find outliers and deal with them 3. Data transformations for scaling/removing nonlinear relationship to linear/ to decrease skewness.
S/W PACKAGES AND HARDWARE APPARATUS USED	S/W- Jupyter Notebook/ Weka/ Python OS-LINUX 64 bit OS H/W: Core 2 DUO/i3/i5/i7 64-bit processor
REFERENCES	1. CHIRAG SHAH, “A HANDS-ON INTRODUCTION TO DATA SCIENCE”,ISBN 978-1-108-47244-9 2. Wes McKinney and the Pandas Development Team, “Pandas: powerful Python data analysis toolkit” 3. https://pandas.pydata.org/
STEPS	**Refer to student activity flow chart if found necessary by subject teacher and relevant to the subject manual. Describe steps only.**
INSTRUCTIONS FOR WRITING JOURNAL	1. Title 2. Problem statement 3. Learning objective 4. Learning outcome 5. Theory (includes methods, libraries and functions, 6. Analysis (as per assignment), 7. Conclusion.

1. Title: Data Wrangling II

2. 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.
Reason and document your approach properly.

3. Learning objective:
1. To do data cleansing on the given dataset.
2. To find outliers and deal with them
3. To apply transformations for scaling/removing nonlinear relationship to linear/ to decrease skewness.

4. Learning outcome:
After performing this assignment students will be able to:

- Find missing values and inconsistencies in the dataset and apply data cleansing on it.
- Find outliers in the dataset and deal with them.
- Apply data transformations such as scaling, removing nonlinear relationships, decrease skewness and convert to normal distribution.

5. Theory:
Libraries: Pythhon libraries Pandas, matplotlib, sklearn,seaborn
Pandas: **Pandas** is a Python library. **Pandas** is used to analyze data. Pandas allows importing data from various file formats such as comma-separated values, JSON, SQL database tables or queries, and Microsoft Excel.

Matplotlib: **Matplotlib** is a comprehensive library for creating static, animated, and interactive visualizations in Python.

Sklearn: Scikit-learn (Sklearn) is the most**useful and robust library for machine learning in Python**. It provides a selection of efficient tools for machine learning, preprocessing and statistical modeling including classification, regression, clustering and dimensionality reduction via a consistence interface in Python.

Seaborn: Seaborn is an open-source Python library built on top of matplotlib. It is used **for data visualization and exploratory data analysis**. Seaborn works easily with dataframes and the Pandas library. Graphs can help us find data trends that are useful in any machine learning or forecasting project.

Methods and functions:
Pandas.read_csv(): to load csv file into Pandas dataframe.
Isnull(): to check presence of null values in the dataset.
skew(): The DataFrame class of pandas has a **method skew() that computes the skewness of the data present in a given axis of the DataFrame object**. Skewness is computed for each row or each column of the data present in the DataFrame object.

distplot(): The seaborn. distplot() function is used to plot the distplot. The distplot represents the **univariate distribution of data** i.e. data distribution of a variable against the density distribution.

Dealing with Outliers:

Outlier is an observation in a given dataset that lies far from the rest of the observations.

Techniques of detecting outliers are:

- Boxplots
- Z-score
- Inter Quantile Range(IQR)

Methods of treating the outliers:

1. 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. Now to define an outlier threshold value is chosen which is generally 3.0. As 99.7% of the data points lie between +/- 3 standard deviation (using Gaussian Distribution approach). Data values above the threshold value are to be dealt with.

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

```
Q1 = np.percentile(df[column_name], 25, interpolation = 'midpoint')

Q3 = np.percentile(df[column_name], 75, interpolation = 'midpoint')

IQR = Q3 - Q1
```

To define the outlier base value is defined above and below datasets normal range namely Upper and Lower bounds, define the upper and the lower bound (1.5*IQR value is considered) :

*upper = Q3 +1.5*IQR*

*lower = Q1 – 1.5*IQR*

Deal with the values below lower and above upper.

Data Scaling:

MinMaxScaler(feature_range = (0, 1)) will transform each value in the column proportionally within the range [0,1] . Use this as the first scaler choice to transform a feature, as it will preserve the shape of the dataset (no distortion).

StandardScaler() will transform each value in the column to range about the mean 0 and standard deviation 1, ie, each value will be normalised by subtracting the mean and dividing by standard deviation. Use StandardScaler if you know the data distribution is normal.

If there are outliers, use **RobustScaler()**. Alternatively you could remove the outliers and use either of the above 2 scalers (choice depends on whether data is normally distributed)

Using The min-max feature scaling:

The min-max approach (often called normalization) rescales the feature to a hard and fast range of [0,1] by subtracting the minimum value of the feature then dividing by the range.

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

Standardization doesn’t have any fixed minimum or maximum value. Here, the values of all the columns are scaled in such a way that they all have a mean equal to 0 and standard deviation equal to 1. This scaling technique works well with outliers. Thus, this technique is preferred if outliers are present in the dataset.

6. Analysis:

Observe % of missing values and outliers in the dataset. Observe skewness or plot density distribution of each column and see if it normal or skewed. After data cleansing and applying data transformation plot density distribution to see the change.

7. Conclusion:

Using python libraries Pandas, matplotlib, sklearn data is cleaned after finding missing values and inconsistancies in the dataset, outliers are dealt with and data transformations for scaling are applied, skewed columns are converted to normal distribution.

PUNE INSTITUTE OF COMPUTER TECHNOLOGY, PUNE

ACADEMIC YEAR: 2024-25
DEPARTMENT of COMPUTER ENGINEERING DEPARTMENT
CLASS: T.E. **SEMESTER: II**
SUBJECT: DSBDAL

ASSINGMENT NO.	3
TITLE	Descriptive Statistics - Measures of Central Tendency and variability
PROBLEM STATEMENT /DEFINITION	Perform the following operations on any open-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) variable. 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 deviation etc. of the species of 'Iris-setosa', 'Iris-versicolor' and 'Iris- verginica' of iris.csv dataset. Provide the codes with outputs and explain everything that you do in this step.
OBJECTIVE	1. To provide summary statistics for a sample dataset 2. To identify some basic statistical details of specific columns of a dataset
OUTCOME	After completion of this assignment the students will be able to: 1. calculate mean, median, minimum, maximum, standard deviation for a dataset 2. identify some basic statistical details of specific columns of a dataset
S/W PACKAGES AND HARDWARE APPARATUS USED	Jupyter notebook, Pandas libraries, Windows/ Linux Operating System, I5 machines/ Laptops
REFERENCES	www.kaggle.com
STEPS	PartI: This is one sample set of steps, students may try any other way 1. Open Jupyter notebook file 2. Import Numpy 3. Import Pandas 4. Read "nba.csv" sample dataset in pandas data frame with the help of read_csv

	function, dataset available in the following link https://drive.google.com/file/d/1kH6mN8y0Eo6u7zlYvcEqP9uIaIPFdYMQ/view?usp=sharing 5. Use shape, head, dtypes, isnull to get familiar with the dataset 6. Calculate mode using mode() function on a sample column, eg. college 7. Calculate mean using mean() function on a sample column, eg. Salary 8. Use value_counts on sample column, eg. Age 9. Use groupby on a sample column, eg. Age and Salary PartI: This is one sample set of steps, students may try any other way 1. Open Jupyter notebook file 2. Import Numpy 3. Import Pandas 4. Read “Iris.csv” sample dataset in pandas data frame with the help of read_csv function, dataset available in the following link https://drive.google.com/file/d/1U8ks4kGRSMKSFkU1lwpf1SncwXaX29IZ/view?usp=sharing 5. Use shape, head, dtypes, isnull to get familiar with the dataset 6. Use value_counts on Species column to get count of ‘Iris-setosa’, ‘Iris-versicolor’ and ‘Iris- verginica’ of iris.csv dataset 7. Write a function to check for each species and use describe().transpose() function to display some basic statistical details like percentile, mean, standard deviation etc. of all species, sepal length, sepal width, petal length and petal width
INSTRUCTIONS FOR WRITING JOURNAL	1. Date 2. Assignment no. 3. Problem definition 4. Learning objective 5. Learning Outcome 6. Concepts related Theory 7. Algorithm 8. Test cases 10. Conclusion/Analysis

Prerequisites: Basic knowledge of DBMS

Concepts related Theory:

Descriptive statistics: It summarizes and organizes characteristics of a data set.

The **central tendency** concerns the averages of the values.

The **variability** or dispersion concerns how spread out the values are.

Measures of central tendency

It estimates the center, or average, of a data set. The mean, median, mode are 3 ways of finding the average.

Mean: The mean, or *M*, is the most commonly used method for finding the average.

To find the mean, simply add up all response values and divide the sum by the total number of responses. The total number of responses or observations is called *N*.

Example:

Mean number of library visits

Data set: 15, 3, 12, 0, 24, 3

Sum of all values: 15 + 3 + 12 + 0 + 24 + 3 = 57

Total number of responses *N* = 6

Mean: Divide the sum of values by *N* to find *M*: 57/6 = **9.5**

Median: The median is the value that's exactly in the middle of a data set.

To find the median, order each response value from the smallest to the biggest. Then, the median is the number in the middle. If there are two numbers in the middle, find their mean.

Example:

Median number of library visits

Ordered data set: 0, 3, 3, 12, 15, 24

Middle numbers: 3, 12

Median: Find the mean of the two middle numbers: (3 + 12)/2 = **7.5**

Mode: The mode is the simply the most popular or most frequent response value. A data set can have no mode, one mode, or more than one mode.

To find the mode, order your data set from lowest to highest and find the response that occurs most frequently.

Example:

Mode number of library visits

Ordered data set	: 0, 3, 3, 12, 15, 24
Mode	Find the most frequently occurring response: **3**

Measures of variability

It gives you a sense of how spread out the response values are. The range, standard deviation and variance each reflect different aspects of spread.

Range

The range gives you an idea of how far apart the most extreme response scores are. To find the range, simply subtract the lowest value from the highest value.

Example:

Range of visits to the library in the past year
Ordered data set: 0, 3, 3, 12, 15, 24
Range: 24 – 0 = **24**

Standard deviation

It is the average amount of variability in your dataset. It tells you, on average, how far each score lies from the mean. The larger the standard deviation, the more variable the data set is.

There are six steps for finding the standard deviation:

1. List each score and find their mean.
2. Subtract the mean from each score to get the deviation from the mean.
3. Square each of these deviations.
4. Add up all of the squared deviations.
5. Divide the sum of the squared deviations by $N – 1$.
6. Find the square root of the number you found.

Example:

Standard deviations of visits to the library in the past year

In the table below, you complete **Steps 1 through 4**.

Raw data	Deviation from mean	Squared deviation
15	15 – 9.5 = 5.5	30.25
3	3 – 9.5 = -6.5	42.25
12	12 – 9.5 = 2.5	6.25
0	0 – 9.5 = -9.5	90.25
24	24 – 9.5 = 14.5	210.25
3	3 – 9.5 = -6.5	42.25
M = 9.5	Sum = 0	Sum of squares = 421.5

Step 5: 421.5/5 = 84.3

Step 6: √84.3 = 9.18

From learning that ***s* = 9.18**, you can say that on average, each score deviates from the mean by 9.18 points.

Conclusion: Students learned descriptive Statistics - Measures of Central Tendency and variability by performing various operations on given dataset using Pandas library.

Review Questions:

1. What is mean?
2. What is mode?
3. What is median?
4. What do you mean by central tendency?
5. What do you mean by variability?
6. Explain groupby function
7. Explain transpose function
8. Explain value_count function
9. What are measures of central tendency?
10. What are measures of variability?

PUNE INSTITUTE OF COMPUTER TECHNOLOGY, PUNE

ACADEMIC YEAR: 2024-25
DEPARTMENT of COMPUTER ENGINEERING DEPARTMENT
CLASS: T.E. SEMESTER: II
SUBJECT: DSBDAL

ASSINGMENT NO.	4
TITLE	Data Analytics I
PROBLEM STATEMENT /DEFINITION	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. The objective is to predict the value of prices of the house using the given features.
OBJECTIVE	To create a linear regression model using a sample dataset
OUTCOME	After completion of this assignment the students will be able to: create a linear regression model using a sample dataset
S/W PACKAGES AND HARDWARE APPARATUS USED	Jupyter notebook, Pandas libraries, Windows/ Linux Operating System, I5 machines/ Laptops
REFERENCES	www.kaggle.com https://www.geeksforgeeks.org/python-linear-regression-using-sklearn/
STEPS	1. Importing all the required libraries 2. Reading the dataset 3. Exploring the data scatter 4. Data cleaning 5. Training our model 6. Exploring our results 7. Working with a smaller dataset
INSTRUCTIONS FOR WRITING JOURNAL	1. Date 2. Assignment no. 3. Problem definition 4. Learning objective 5. Learning Outcome

	6. Concepts related Theory 7. Algorithm 8. Test cases 10. Conclusion/Analysis

Prerequisites: Basic knowledge of DBMS

Concepts related Theory and design of assignment using a sample dataset available here:

https://www.kaggle.com/sohier/calcofi?select=bottle.csv

(Students are expected to go through the steps of linear regression applied to this sample dataset and then apply the same steps to the dataset given in the problem statement)

Linear Regression is a machine learning algorithm based on supervised learning. It performs a regression task. Regression models a target prediction value based on independent variables. It is mostly used for finding out the relationship between variables and forecasting. Different regression models differ based on – the kind of relationship between dependent and independent variables, they are considering and the number of independent variables being used.

Step 1: Importing all the required libraries

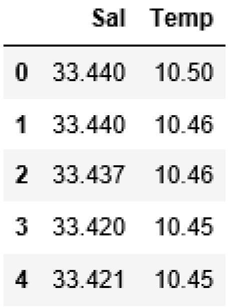
```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn import preprocessing, svm
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
```

Step 2: Reading the dataset

You can download the dataset here.

```
cd C:\Users\Dev\Desktop\Kaggle\Salinity

# Changing the file read location to the location of the dataset
df = pd.read_csv('bottle.csv')
df_binary = df[['Salnty', 'T_degC']]

# Taking only the selected two attributes from the dataset
df_binary.columns = ['Sal', 'Temp']
```

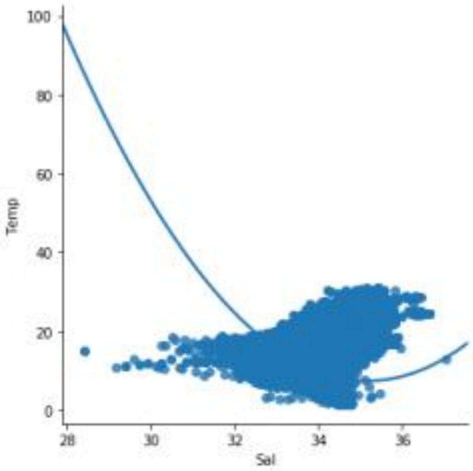
```
# Renaming the columns for easier writing of the code
df_binary.head()

# Displaying only the 1st  rows along with the column names
```



	Sal	Temp
0	33.440	10.50
1	33.440	10.46
2	33.437	10.46
3	33.420	10.45
4	33.421	10.45

Step 3: Exploring the data scatter

```
sns.lmplot(x ="Sal", y ="Temp", data = df_binary, order = 2, ci = None)

# Plotting the data scatter
```



Step 4: Data cleaning

```
# Eliminating NaN or missing input numbers
df_binary.fillna(method ='ffill', inplace = True)
```

Step 5: Training our model

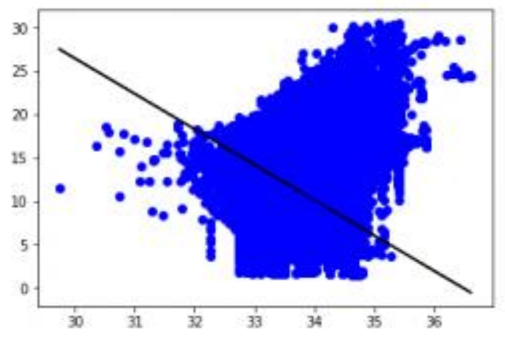
```
X = np.array(df_binary['Sal']).reshape(-1, 1)
y = np.array(df_binary['Temp']).reshape(-1, 1)
```

```
# Separating the data into independent and dependent variables
# Converting each dataframe into a numpy array
# since each dataframe contains only one column
df_binary.dropna(inplace = True)

# Dropping any rows with Nan values
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.25)

# Splitting the data into training and testing data
regr = LinearRegression()

regr.fit(X_train, y_train)
print(regr.score(X_test, y_test))
```

```
0.20780376990868232
```

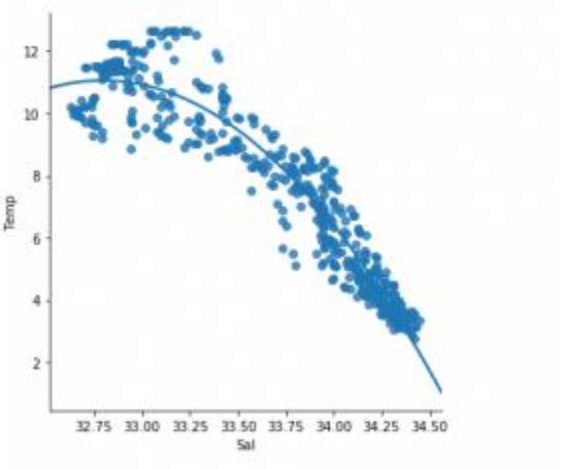
Step 6: Exploring our results

```
y_pred = regr.predict(X_test)
plt.scatter(X_test, y_test, color ='b')
plt.plot(X_test, y_pred, color ='k')

plt.show()
# Data scatter of predicted value
```



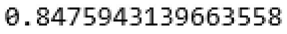
Step 7: Working with a smaller dataset

```
df_binary500 = df_binary[:][:500]

# Selecting the 1st 500 rows of the data
sns.lmplot(x ="Sal", y ="Temp", data = df_binary500,
                    order = 2, ci = None)
```



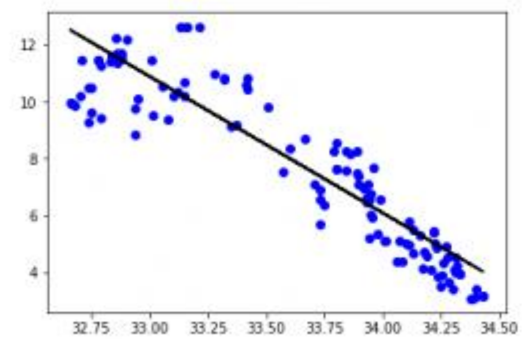
We can already see that the first 500 rows follow a linear model. Continuing with the same steps as before.

```
df_binary500.fillna(method ='ffill', inplace = True)

X = np.array(df_binary500['Sal']).reshape(-1, 1)
y = np.array(df_binary500['Temp']).reshape(-1, 1)

df_binary500.dropna(inplace = True)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.25)

regr = LinearRegression()
regr.fit(X_train, y_train)
print(regr.score(X_test, y_test))
```



```
0.8475943139663558
```

```
y_pred = regr.predict(X_test)
plt.scatter(X_test, y_test, color ='b')
plt.plot(X_test, y_pred, color ='k')

plt.show()
```


12
10
8
6
4
32.75
33.00
33.25
33.50
33.75
34.00
34.25
34.50

ASSIGNMENT-NO-5

TITLE	**Data Analytics II**
PROBLEM STATEMENT/ DEFINITION	1. Implement logistic regression using Python /R to perform classification on Social_Network_Ads.csv dataset. 2. ComputeConfusionmatrixtofindTP,FP,TN,FN,Accuracy, Errorrate, Precision,Recall on the given dataset.
OBJECTIVE	To understand how logistic regression works on the given dataset.
OUTCOME	To find the best scenario for the result to be achieved for a given data set using logistic regression.
S/W PACKAGES AND HARDWARE APPARATUS USED	Core 2 DUO/i3/i5/i7 64-bit processor OS-LINUX 64 bit OS Editor-gedit/Eclipse S/w- Jupyter Notebook/ Weka/ Python
REFERENCES	1. Chirag Shah, "A Hands-On Introduction To Data Science", Cambridge University Press, (2020), ISBN : ISBN 978-1-108-47244-9. Curriculum for Third Year of Computer Engineering (2019 Course), Savitribai Phule Pune University http://collegecirculars.unipune.ac.in/sites/documents/ Syllabus2020/Forms/AllItems.aspx #57/87 2. Giuseppe Bonaccorso, " Machine Learning Algorithms", Packt Publishing Limited, ISBN-10: 1785889621, ISBN-13: 978-1785889622
STEPS	**Refer to student activity flow chart if found necessary by subject teacher and relevant to the subjectmanual. Describe steps only.**
INSTRUCTIONS FOR WRITING JOURNAL	1. title 2. Problem statement 3. Learning objective 4. Learning outcome 5. Theory (includes methods, libraries and functions, 6. Analysis (as per assignment), 7. conclusion.

TITLE- Data Analytics II

PROBLEM STATEMENT/ DEFINITION-Implement logisticregression using Python /R to perform classification on Social_Network_Ads.csv dataset. ComputeConfusionmatrixtofindTP,FP,TN,FN,Accuracy,Errorrate,Precision, Recall on the given dataset.

LEARNING OBJECTIVE-

To understand how logistic regression works on the given dataset.

LEARNING OUTCOME

To find the best scenario for the result to be achieved for a given data set using logistic regression

THEORY- Logistic regression is a classification method which is based on the probability for a sample to belong to a class. As our probabilities must be continuous in R and bounded between (0, 1), it's necessary to introduce a threshold function to filter the term z. The name logistic comes from the decision to use the sigmoid (or logistic) function:

$$\sigma(z) = \frac{1}{1+e^{-z}} \; which\ becomes\ \sigma(\bar{x};\bar{w}) = \frac{1}{1+e^{-\bar{x}\cdot\bar{w}}}$$

A solution for classification is logistic regression. Instead of fitting a straight line or hyperplane, the logistic regression model uses the logistic function to squeeze the output of a linear equation between 0 and 1.

Methods-

```
from sklearn.model_selection import train_test_split
>>> X_train, X_test, Y_train, Y_test = train_test_split(X, Y,test_size=0.25)
```

Now we can train the model using the default parameters:

```
from sklearn.linear_model import LogisticRegression
>>> lr = LogisticRegression()
>>> lr.fit(X_train, Y_train)

LogisticRegression(C=1.0, class_weight=None, dual=False, fit_intercept=True,
intercept_scaling=1, max_iter=100, multi_class='ovr', n_jobs=1,
penalty='l2', random_state=None, solver='liblinear', tol=0.0001,verbose=0, warm_start=False)

>>> lr.score(X_test, Y_test)
0.95199999999999996
```

It's also possible to check the quality through a cross-validation (like for linear regression):

```
from sklearn.model_selection import cross_val_score


>>> cross_val_score(lr, X, Y, scoring='accuracy', cv=10)
array([ 0.96078431, 0.92156863, 0.96 , 0.98 , 0.96 ,

0.98 , 0.96 , 0.96 , 0.91836735, 0.97959184])
```

Classification metrics

A classification task can be evaluated in many different ways to achieve specific objectives. Of course, the most important metric is the accuracy, often expressed as:

$$Generic\ accuracy = 1 - \frac{Number\ of\ misclassified\ samples}{Total\ number\ of\ samples}$$

In scikit-learn, it can be assessed using the built-in accuracy_score() function:

```
from sklearn.metrics import accuracy_score

>>> accuracy_score(Y_test, lr.predict(X_test))
```

Let us understand the confusion matrix. In many cases, it's necessary to be able to differentiate between different kinds of misclassifications (we're considering the binary case with the conventional notation: 0-negative, 1-positive), because the relative weight is quite different. For this reason, we introduce the following definitions:

True positive: A positive sample correctly classified

False positive: A negative sample classified as positive

True negative: A negative sample correctly classified

False negative: A positive sample classified as negative

Scikit learn supports the following method to compute the confusion matrix.

```
from sklearn.metrics import confusion_matrix

>>> cm = confusion_matrix(y_true=Y_test, y_pred=lr.predict(X_test))
cm[::-1, ::-1]
```

CONCLUSION- Thus, logistic regression model on the given data set is applied .The results shows of fitting a logistic regression model on the given dataset and shown the features used in the model, their estimated weights the standard errors of the estimated weights.

ASSIGNMENT-NO-6

TITLE	**Data Analytics III**
PROBLEM STATEMENT/ DEFINITION	Implement Simple Naïve Bayes classification algorithm using Python/R on iris.csv dataset. Compute Confusion matrix to find TP, FP, TN, FN, Accuracy, Error rate, Precision, Recall on the given dataset.
OBJECTIVE	To understand how Naïve Bayes classification algorithm works on the given dataset
OUTCOME	To find the best scenario for the result to be achieved for a given data set using logistic regression.
S/W PACKAGES AND HARDWARE APPARATUS USED	Core 2 DUO/i3/i5/i7 64-bit processor OS-LINUX 64 bit OS Editor-gedit/Eclipse S/w- Jupyter Notebook/ Weka/ Python
REFERENCES	3. Chirag Shah, "A Hands-On Introduction To Data Science", Cambridge University Press, (2020), ISBN : ISBN 978-1-108-47244-9. Curriculum for Third Year of Computer Engineering (2019 Course), Savitribai Phule Pune University http://collegecirculars.unipune.ac.in/sites/documents/ Syllabus2020/Forms/AllItems.aspx #57/87 4. Giuseppe Bonaccorso, " Machine Learning Algorithms", Packt Publishing Limited, ISBN-10: 1785889621, ISBN-13: 978-1785889622
STEPS	**Refer to student activity flow chart if found necessary by subject teacher and relevant to the subjectmanual. Describe steps only.**
INSTRUCTIONS FOR WRITING JOURNAL	1. title 2. Problem statement 3. Learning objective 4. Learning outcome 5. Theory (includes methods, libraries and functions, 6. Analysis (as per assignment), 7. conclusion.

ASSIGNMENT-NO-6

TITLE- Data Analytics III

PROBLEM STATEMENT/ DEFINITION-Implement logisticregression using Python /R to perform classification on Social_Network_Ads.csv dataset. ComputeConfusionmatrixtofindTP,FP,TN,FN,Accuracy,Errorrate,Precision, Recall on the given dataset.

LEARNING OBJECTIVE-

To understand how logistic regression works on the given dataset.

LEARNING OUTCOME

To find the best scenario for the result to be achieved for a given data set using logistic regression

THEORY-
(includes methods, libraries and functions,

A naive Bayes classifier is called so because it's based on a naive condition, which implies the conditional independence of causes. This can seem very difficult to accept in many contexts where the probability of a particular feature is strictly correlated to another one.

For example, in spam filtering, a text shorter than 50 characters can increase the probability of the presence of an image, or if the domain has been already blacklisted for sending the same spam emails to million users, it's likely to find particular keywords.

Following three classification methods can be applied to the data set.

1. BernoulliNB()
2. GaussianNB()
3. MultinomialNB()

Bernoulli naive Bayes expects binary feature vectors; however, the class BernoulliNB has a binarize parameter, which allows us to specify a threshold that will be used internally to transform the features:

```
from sklearn.datasets import make_classification
>>> nb_samples = 300
>>> X, Y = make_classification(n_samples=nb_samples, n_features=2,
n_informative=2, n_redundant=0)
```

```
 from sklearn.naive_bayes import BernoulliNB
from sklearn.model_selection import train_test_split
>>> X_train, X_test, Y_train, Y_test = train_test_split(X, Y,
test_size=0.25)
```

```
>>> bnb = BernoulliNB(binarize=0.0)
>>> bnb.fit(X_train, Y_train)
>>> bnb.score(X_test, Y_test)
```

```
from sklearn.datasets import load_digits
from sklearn.model_selection import cross_val_score
>>> digits = load_digits()
>>> gnb = GaussianNB()
>>> mnb = MultinomialNB()
```

Analysis (as per assignment)

Classification metrics

A classification task can be evaluated in many different ways to achieve specific objectives. Of course, the most important metric is the accuracy, often expressed as:

$$Generic\ accuracy = 1 - \frac{Number\ of\ misclassified\ samples}{Total\ number\ of\ samples}$$

In scikit-learn, it can be assessed using the built-in accuracy_score() function:

```
from sklearn.metrics import accuracy_score

>>> accuracy_score(Y_test, lr.predict(X_test))
```

Let us understand the confusion matrix. In many cases, it's necessary to be able to differentiate between different kinds of misclassifications (we're considering the binary case with the conventional notation: 0-negative, 1-positive), because the relative weight is quite different. For this reason, we introduce the following definitions:

True positive: A positive sample correctly classified
False positive: A negative sample classified as positive
True negative: A negative sample correctly classified
False negative: A positive sample classified as negative

Scikit learn supports the following method to compute the confusion matrix and calculating the precision and recall

```
from sklearn.metrics import confusion_matrix

>>> cm = confusion_matrix(y_true=Y_test, y_pred=lr.predict(X_test))
cm[::-1, ::-1]

from sklearn.metrics import precision_score

>>> precision_score(Y_test, lr.predict(X_test))

from sklearn.metrics import recall_score

>>> recall_score(Y_test, lr.predict(X_test))
```

CONCLUSION-Thus, Naïve Bay’s Classifier model on the given data set is applied .The results shows the classification using various methods with precision and recall.

TITLE	Text Analytics
PROBLEM STATEMENT/ DEFINITION	1. Extract Sample document and apply following document preprocessing methods: Tokenization, POS Tagging, stop words removal, Stemming and Lemmatization. 2. Create representation of document by calculating Term Frequency and Inverse Document Frequency.
OBJECTIVE	1. To apply preprocessing steps like tokenization, POS tagging, stop word removal, stemming and Lemmatization on a sample document and bring the text into a form that is predictable and analyzable for a specific task 2. Calculate Term Frequency and Inverse Document Frequency which is used by search engines as a central tool in scoring and ranking a document's relevance given a user query.
S/W PACKAGES AND HARDWARE APPARATUS USED	**Operating System recommended** :- 64-bit Open source Linux or its derivative **Programming tools recommended**: - JAVA/Python/R/Scala
REFERENCES	1)Chirag Shah, “A Hands-On Introduction To Data Science”, Cambridge University Press,(2020), ISBN : ISBN 978-1-108-47244-9. 2. 2)Wes McKinney, “Python for Data Analysis”, O' Reilly media, ISBN : 978-1-449-31979-3.
STEPS	**Refer to student activity flow chart if found necessary by subject teacher and relevant to the subjectmanual. Describe steps only.**
INSTRUCTIONS FOR WRITING JOURNAL	1. title 2. Problem statement 3. Learning objective 4. Learning outcome 5. Theory (includes methods, libraries and functions, 6. Analysis (as per assignment), 7. conclusion.

HeadofDepartment
(Dr. M.S.Takalikar)

SubjectCo-ordinator
(Dr. S.S.Sonawane)

Theory:

Text Analytics:

Text analytics is the process of extracting meaning out of text. For example, this can be analyzing text written by customers in a customer survey, with the focus on finding common themes and trends. The idea is to be able to examine the customer feedback to inform the business on taking strategic action, in order to improve customer experience.

Tokenization

One of the very basic things to do is dividing a body of text into words or sentences, splitting a phrase, sentence, paragraph, or an entire text document into smaller units, such as individual words or terms.The tokens could be words, numbers or punctuation marks. In tokenization, smaller units are created by locating word boundariesThis is called tokenization.

Word Tokenization

Word Tokenization is the most commonly used tokenization algorithm. It splits a piece of text into individual words based on a certain delimiter. Depending upon delimiters, different word-level tokens are formed.

Drawbacks of Word Tokenization

One of the major issues with word tokens is dealing with Out Of Vocabulary (OOV) words**.** OOV words refer to the new words which are encountered at testing. These new words do not exist in the vocabulary. Hence, these methods fail in handling OOV words.

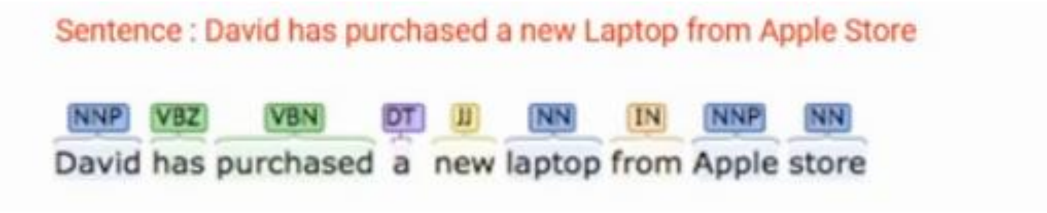
Character Tokenization

Character Tokenization splits apiece of text into a set of characters. It overcomes the drawbacks we saw above about Word Tokenization.Character Tokenizers handles OOV words coherently by preserving the information of the word. It breaks down the OOV word into characters and represents the word in terms of these character.

POS Tagging

- Classifying words in their part of speech and providing them labels according to their part of speech is called part of speech tagging or POS tagging OR POST. Hence the set of labels/tags is called a tagset Parts of speech tags are the properties of the words, which define their main context, functions, and usage in a sentence. Some of the commonly used parts of speech tags are
- **Nouns**: Which defines any object or entity
- **Verbs**: That defines some action.
- **Adjectives and Adverbs**: This acts as a modifier, quantifier, or intensifier in any sentence.
- In a sentence, every word will be associated with a proper part of the speech tag. For example**,** consider the sentence below

Example: David has purchased a new laptop from Apple store


Sentence : David has purchased a new Laptop from Apple Store
NNP
VBZ
VBN
DT
JJ
NN
IN
NNP
NN
David has purchased a new laptop from Apple store

In this sentence, every word is associated with a part of the speech tag which defines their functions. Here, David has an NNP tag which means it is a proper noun. Further, Has and purchased belong to the verb indicating that they are the actions. The Laptop and Apple store are the nouns. New is the adjective whose role is to modify the context of the laptop.
Parts of speech tags are defined by the relationship of words with the other words in the sentence. We can apply machine learning models and rule-based models to obtain the parts of speech tags of a word.

Stop Word Removal

Stop words:The words which are generally filtered out before processing a natural language are called **stop words**. These are actually the most common words in any language (like articles, prepositions, pronouns, conjunctions, etc) and does not add much information to the text. Examples of a few stop words in English are "the", "a", "an", "so", "what"

Stemming :

Stemming is an important part of the pipelining process in Natural language processing. The input to the stemmer is tokenized words.

Errors in Stemming:

There are mainly two errors in stemming –

- over-stemming
- under-stemming

Over-stemming occurs when two words are stemmed from the same root that are of different stems. Over-stemming can also be regarded as false-positives.

Under-stemming occurs when two words are stemmed from the same root that are not of different stems. Under-stemming can be interpreted as false-negatives.

Applications of stemming :

1. Stemming is used in information retrieval systems like search engines.
2. It is used to determine domain vocabularies in domain analysis.

Lemmatization :

Lemmatization is the process of converting a word to its base form. The difference between stemming and lemmatization is, lemmatization considers the context and converts the word to its meaningful base form, whereas stemming just removes the last few characters, often leading to incorrect meanings and spelling errors.
Examples of implementing this comes in the following sections. we will see how to implement lemmatization using the following python packages.

1. Wordnet Lemmatizer
2. Spacy Lemmatizer
3. TextBlob

4. CLiPS Pattern
5. Stanford CoreNLP
6. Gensim Lemmatizer
7. TreeTagger

TF-IDF

tf-idf, short for term frequency–inverse document frequency, is a numeric measure that is use to score the importance of a word in a document based on how often did it appear in that document and a given collection of documents. The intuition for this measure is : If a word appears frequently in a document, then it should be important and we should give that word a high score. But if a word appears in too many other documents, it's probably not a unique identifier, therefore we should assign a lower score to that word

TF Term Frequency

The first part of the formula tf(t,d)tf(t,d) is simply to calculate the number of times each word appeared in each document. Of course, as with common text mining methods: stop words like "a", "the", punctuation marks will be removed beforehand and words will all be converted to lower cases.

Inverse document frequency

The **inverse document frequency** is a measure of how much information the word provides, i.e., if it's common or rare across all documents. It is the logarithmically scaled inverse fraction of the documents that contain the word (obtained by dividing the total number of documents by the number of documents containing the term, and then taking the logarithm of that quotient)

TITLE	Data Visualization I
PROBLEM STATEMENT/ DEFINITION	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.
OBJECTIVE	to make it easier to identify patterns, trends and outliers in large data sets.
S/W PACKAGES AND HARDWARE APPARATUS USED	**Operating System recommended** :- 64-bit Open source Linux or its derivative **Programming tools recommended**: - JAVA/Python/R/Scala
REFERENCES	1)Chirag Shah, “A Hands-On Introduction To Data Science”, Cambridge University Press,(2020), ISBN : ISBN 978-1-108-47244-9. 2. 2)Wes McKinney, “Python for Data Analysis”, O' Reilly media, ISBN : 978-1-449-31979-3.
STEPS	**Refer to student activity flow chart if found necessary by subject teacher and relevant to the subjectmanual. Describe steps only.**
INSTRUCTIONS FOR WRITING JOURNAL	1. title 2. Problem statement 3. Learning objective 4. Learning outcome 5. Theory (includes methods, libraries and functions, 6. Analysis (as per assignment), 7. conclusion.

Theory:

Data Visualization

Data Visualization is the presentation of data in pictorial format. It is extremely important for Data Analysis, primarily because of the fantastic ecosystem of data-centric Python packages. And it helps to understand the data, however, complex it is, the significance of data by summarizing and presenting a huge amount of data in a simple and easy-to-understand format and helps communicate information clearly and effectively

Pandas and Seaborn is one of those packages and makes importing and analyzing data much easier. In this article, we will use Pandas and Seaborn to analyze data.

Pandas

Pandas offer tools for cleaning and process your data. It is the most popular Python library that is used for data analysis. In pandas, a data table is called a dataframe.

Seaborn

Seaborn is an amazing visualization library for statistical graphics plotting in Python. It is built on the top of matplotlib library and also closely integrated into the data structures from pandas.

Installation

For python environment :

```
pip install seaborn
```

Seaborn: statistical data visualization

Seaborn helps to visualize the statistical relationships, To understand how variables in a dataset are related to one another and how that relationship is dependent on other variables, we perform statistical analysis. This Statistical analysis helps to visualize the trends and identify various patterns in the dataset.

These are the plot will help to visualize:

- Line Plot
- Scatter Plot
- Box plot
- Point plot
- Count plot
- Violin plot
- Swarm plot
- Bar plot
- KDE Plot

TITLE	Data Visualization II
PROBLEM STATEMENT/ DEFINITION	1. Use the inbuilt dataset 'titanic' as used in the assignment#7. 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.
OBJECTIVE	To implement the data visualization techniques
S/W PACKAGES AND HARDWARE APPARATUS USED	1. Operating System : 64-bit Open source Linux or its derivative 2. Programming Languages: PYTHON/R
REFERENCES	• Mark Gardner, “Beginning R: The Statistical Programming Language”, Wrox Publication, ISBN: 978-1-118-16430-3 • David Dietrich, Barry Hiller, “Data Science and Big Data Analytics”, EMC education services, Wiley publications, 2012, ISBN0-07-120413-X Luis Torgo, “Data Mining with R, Learning with Case Studies”, CRC Press, Talay and Francis Group, ISBN9781482234893
STEPS	**Refer to student activity flow chart if found necessary by subject teacher and relevant to the subject manual. Describe steps only.**
INSTRUCTIONS FOR WRITING JOURNAL	1. Title 2. Problem statement 3. Learning objective 4. Learning outcome 5. Theory (includes methods, libraries and functions, 6. Analysis (as per assignment), 7. conclusion.

Head of Department
(Dr. M.S.Takalikar)

Subject Co-ordinator
(Dr. S.S.Sonawane)

Assignment No. 9

- **Aim:**

  Summary statistics, data visualization, boxplot for the features on the 'titanic' dataset or any other dataset.

- **Problem Statement / Definition:**

  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')

  - Write observations on the inference from the above statistics.

- **Prerequisites**
  - Database management system, Python/R programming

- **Learning Objectives**
  - Learn to use dataset, dataframes, features of dataset in an application
  - Learn to compute summary statistics for the features.
  - Learn to use visualization techniques.

- **Learning Outcome:**
  - Students will be able to compute statistics on the features of the dataset, use histograms and boxplot on the features of the dataset.

- **Theory:**

  Data analysis is a process of inspecting, cleansing, transforming, and modelling data with the goal of discovering useful information, informing conclusions, and supporting decision-making. Data analysis has multiple facets and approaches, encompassing diverse techniques under a variety of names, while being

used in different business, science, and social science domains.
A data set (or dataset) is a collection of data. Most commonly a data set corresponds to the contents of a single database table, or a single statistical data matrix, where every column of the table represents a particular variable, and each row corresponds to a given member of the data set in question.

A boxplot shows the distribution of the data with more detailed information. It shows the outliers more clearly, maximum, minimum, quartile(Q1), third quartile(Q3), interquartile range(IQR), and median. You can calculate the middle 50% from the IQR.

Boxplot is a very interesting plot that basically plots a 5 number summary. to get 5 number summary some terms we need to describe.

Median – Middle value in series after sorting

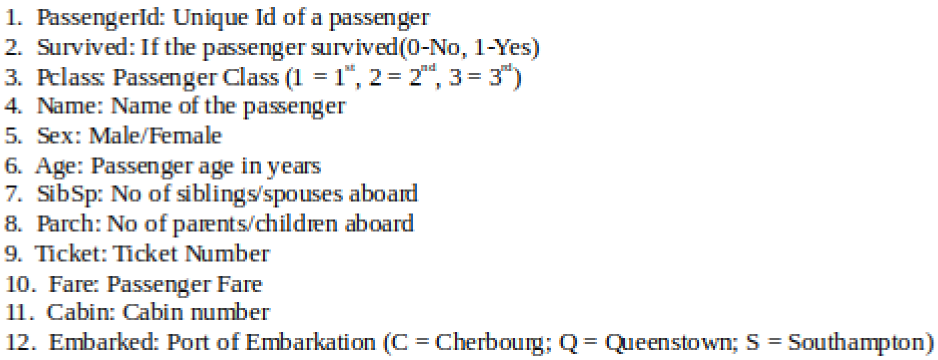
Percentile – Gives any number which is number of values present before this percentile like for example 50 under 25th percentile so it explains total of 50 values are there below 25th percentile

Minimum and Maximum – These are not minimum and maximum values, rather they describe the lower and upper boundary of standard deviation which is calculated using Interquartile range(IQR).

Titanic dataset:

It is one of the most popular datasets used for understanding machine learning basics. It contains information of all the passengers aboard the RMS Titanic, which unfortunately was shipwrecked. This dataset can be used to predict whether a given passenger survived or not. The csv file can be downloaded from Kaggle.

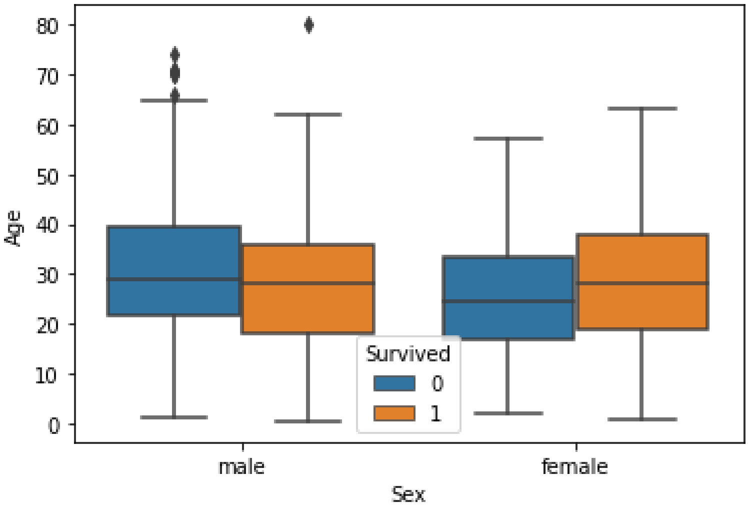
Description:



1. PassengerId: Unique Id of a passenger
2. Survived: If the passenger survived(0-No, 1-Yes)
3. Pclass: Passenger Class (1 = 1st, 2 = 2nd, 3 = 3rd)
4. Name: Name of the passenger
5. Sex: Male/Female
6. Age: Passenger age in years
7. SibSp: No of siblings/spouses aboard
8. Parch: No of parents/children aboard
9. Ticket: Ticket Number
10. Fare: Passenger Fare
11. Cabin: Cabin number
12. Embarked: Port of Embarkation (C = Cherbourg; Q = Queenstown; S = Southampton)

Following is a box plot for distribution of age with respect to each gender along with the information about whether they survived or not.

```
import numpy as np
import pandas pd
import matplotlib.pyplot as plt
import seaborn as sns
from seaborn import load_dataset
#titanic dataset
data = pd.read_csv("titanic_train.csv")

sns.boxplot(data['Sex'], data["Age"], data["Survived"])
plt.show()
```



TITLE	Data Visualization III
PROBLEM STATEMENT/ DEFINITION	Download the Iris flower dataset or any other dataset into a DataFrame. (e.g., https://archive.ics.uci.edu/ml/datasets/Iris). 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 box plot for each feature in the dataset. 4. Compare distributions and identify outliers.
OBJECTIVE	To implement the data visualization techniques
S/W PACKAGES AND HARDWARE APPARATUS USED	1. Operating System : 64-bit Open source Linux or its derivative 2. Programming Languages: PYTHON/R
REFERENCES	- Mark Gardner, “Beginning R: The Statistical Programming Language”, Wrox Publication, ISBN: 978-1-118-16430-3 - David Dietrich, Barry Hiller, “Data Science and Big Data Analytics”, EMC education services, Wiley publications, 2012, ISBN0-07-120413-X - Luis Torgo, “Data Mining with R, Learning with Case Studies”, CRC Press, Talay and Francis Group, ISBN9781482234893
STEPS	**Refer to student activity flow chart if found necessary by subject teacher and relevant to the subject manual. Describe steps only.**
INSTRUCTIONS FOR WRITING JOURNAL	1. Title 2. Problem statement 3. Learning objective 4. Learning outcome 5. Theory (includes methods, libraries and functions, 6. Analysis (as per assignment), 7. conclusion.

Head of Department
(Dr. M.S.Takalikar)

Subject Co-ordinator
(Dr. S.S.Sonawane)

P:F:-LTL-UG / 03 / R1

Assignment No. 10

- **Aim:**

  Summary statistics, data visualization, histogram and boxplot for the features on the Iris dataset or any other dataset.

- **Problem Statement / Definition:**

  - Download the Iris flower dataset or any other dataset into a DataFrame. (e.g., https://archive.ics.uci.edu/ml/datasets/Iris). Scan the dataset and give the inference as:
    - List down the features and their types (e.g., numeric, nominal) available in the dataset.
    - Create a histogram for each feature in the dataset to illustrate the feature distributions.
    - Create a box plot for each feature in the dataset.

- **Prerequisites**

  - Database management system, Python/R programming

- **Learning Objectives**

  - Learn to use dataset, dataframes, features of dataset in an application
  - Learn to compute summary statistics for the features.
  - Learn to use visualization techniques.

- **Learning Outcome:**

  - Students will be able to compute statistics on the features of the dataset, use histograms and boxplot on the features of the dataset.

- **Theory:**

Data analysis is a process of inspecting, cleansing, transforming, and modelling data with the goal of discovering useful information, informing conclusions, and supporting decision-making. Data analysis has multiple facets and approaches, encompassing diverse techniques under a variety of names, while being used in different business, science, and social science domains.

A data set (or dataset) is a collection of data. Most commonly a data set corresponds to the contents of a single database table, or a single statistical data matrix, where every column of the table represents a particular variable, and each row corresponds to a given member of the data set in question.

Iris flower dataset:

The Iris Dataset contains four features (length and width of sepals and petals) of 50 samples of three species of Iris (Iris setosa, Iris virginica and Iris versicolor). These measures were used to create a linear discriminant model to classify the species. The dataset is often used in data mining, classification and clustering examples and to test algorithms.

```
Attribute Information:
         -> sepal length in cm
         -> sepal width in cm
         -> petal length in cm
         -> petal width in cm
         -> class:
                  Iris Setosa
                  Iris Versicolour
                  Iris Virginica

        Number of Instances: 150
```

Summary statistic:

Mean, standard deviation, regression, sample size determination and hypothesis testing are the fundamental data analytics methods.

Mean: The sum of all the data entries divided by the number of entries.

$$\text{Population Mean: } \mu = \frac{\Sigma x}{N}$$

$$\text{Sample Mean: } \overline{x} = \frac{\Sigma x}{n}$$

Range: The difference between the maximum and minimum data entries in the set.

Range = (Max. data entry) – (Min. data entry)

Standard deviation:

The standard deviation measure variability and consistency of the sample or population. In most real-world applications, consistency is a great advantage. In statistical data analysis, less variation is often better.

$$\text{Population Standard Deviation} = \sigma = \sqrt{\frac{\sum(x-\mu)^2}{N}}$$

$$\text{Sample Standard Deviation } = s = \sqrt{\frac{\sum(x-\overline{x})^2}{n-1}}$$

Variance: The average squared deviation from the mean is also known as the variance.

Percentile: Let p be any integer between 0 and 100. The pth percentile of data set is the data value at which p percent of the value in the data set are less than or equal to this value.

- How to calculate percentiles: Use the following steps for calculating percentiles for small data sets.
- Step 1: Sort the data in ascending order (from smallest to largest)

$\left(\frac{p}{100}\right)n,$ • Step Step 3: 2: Calculate ith = the 100 where p is the percentile and n is the sample size.

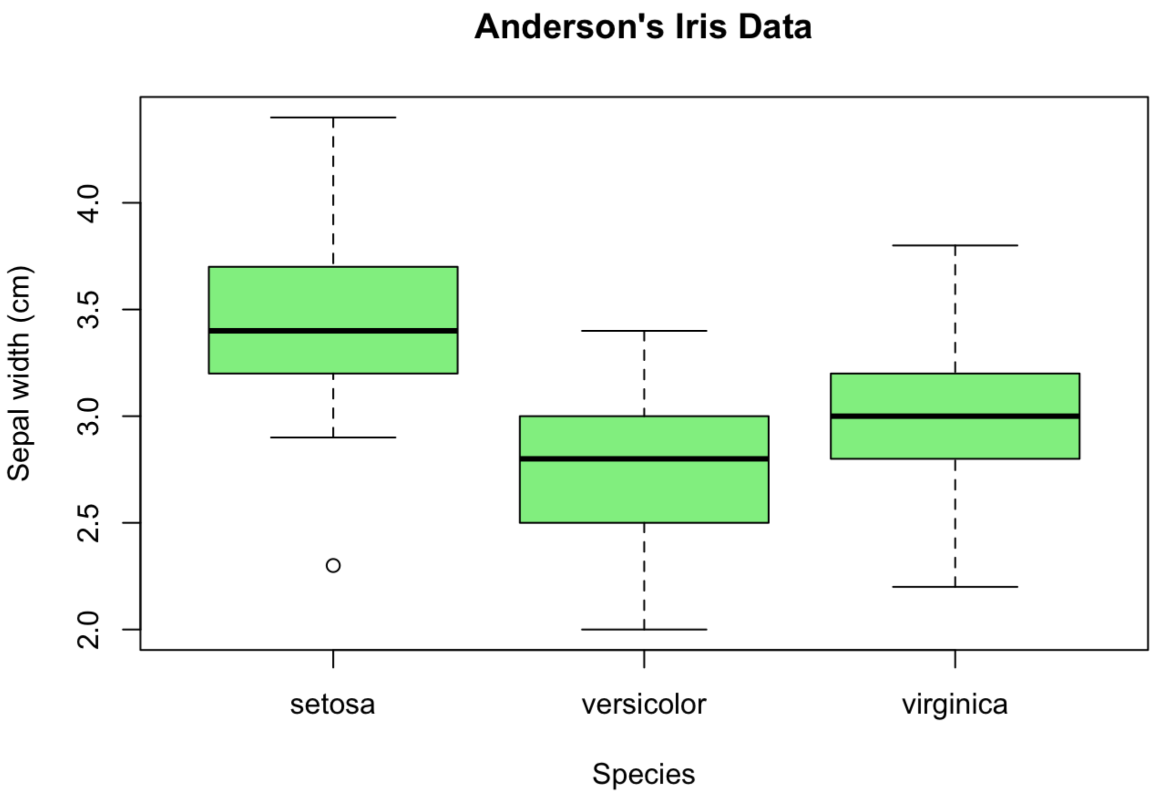
Step 3: If i is an integer the pth percentile is the mean of the data values in position i and i+1.If i is not an integer then round up to the next integer and use the value in this position.

Summary statistic on Iris dataset:

```
 Summary Statistics:
                Min  Max   Mean    SD   Class Correlation
 sepal length: 4.3  7.9   5.84  0.83    0.7826
 sepal width: 2.0  4.4   3.05  0.43   -0.4194
 petal length: 1.0  6.9   3.76  1.76    0.9490  (high!)
 petal width: 0.1  2.5   1.20  0.76    0.9565  (high!)

Class Distribution: 33.3% for each of 3 classes.
```

Box Plot:

A boxplot shows the distribution of the data with more detailed information. It shows the outliers more clearly, maximum, minimum, quartile(Q1), third quartile(Q3), interquartile range(IQR), and median. You can calculate the middle 50% from the IQR.

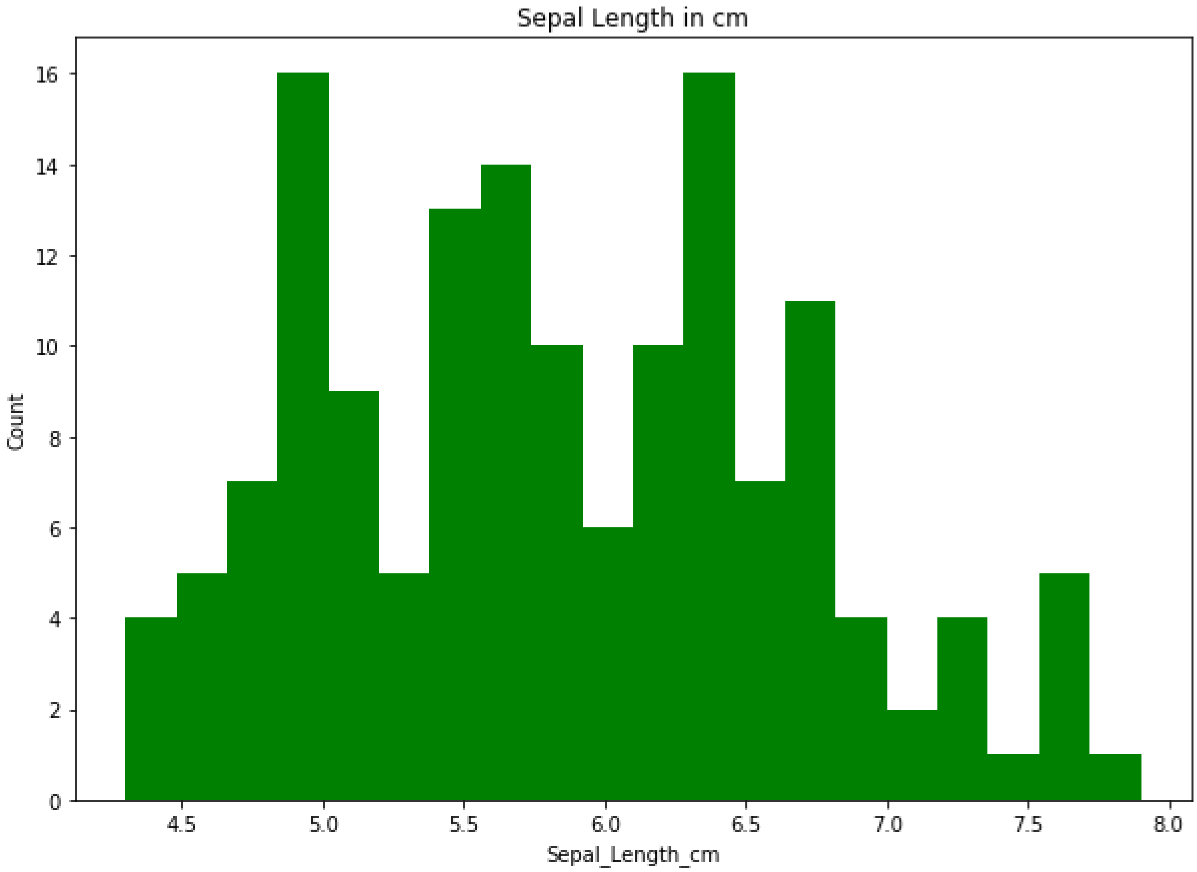


Histogram:

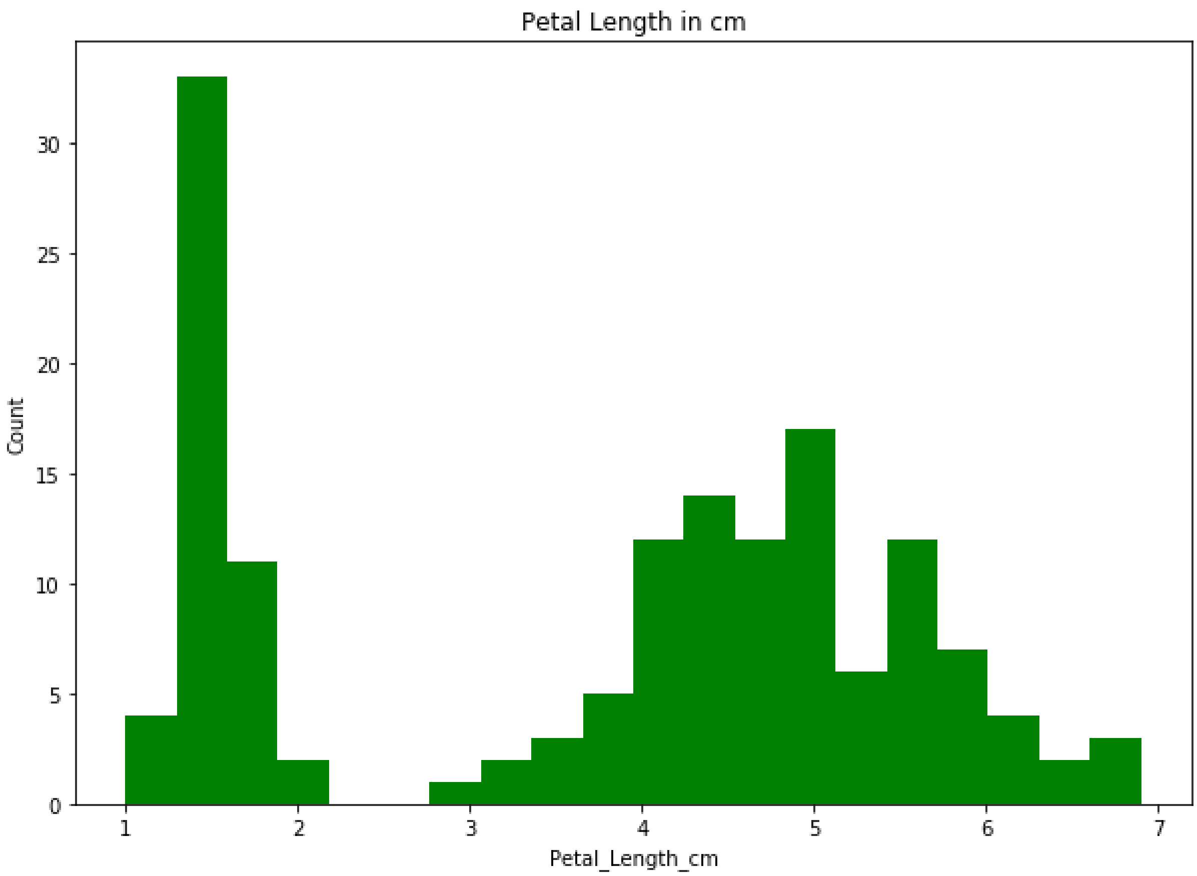
Both histograms and box plots are used to explore and present the data in an easy and understandable manner. Histograms are preferred to determine the underlying probability distribution of a data. Box plots on the other hand are more useful when comparing between several data sets. They are less detailed than histograms and take up less space.

A histogram is a value distribution plot of numerical columns. It basically creates bins in various ranges in values and plots it where we can visualize how values are distributed. We can have a look where more values lie like in positive, negative, or at the center(mean)

Histograms and box plots are very similar in that they both help to visualize and describe numeric data. Although histograms are better in determining the underlying distribution of the data, box plots allow you to compare multiple data sets better than histograms as they are less detailed and take up less space. It is recommended that you plot your data graphically before proceeding with further statistical analysis.



Histogram for Sepal Length



Histogram for Petal Length

TITLE	Hadoop code to count number of occurrences
PROBLEM STATEMENT/ DEFINITION	Write a code in JAVA for a simple Word Count application that counts the number of occurrences of each word in a given input set using the Hadoop Map-Reduce framework on local-standalone set-up.
OBJECTIVE	● Learn Map reduce for counting occurrences using Hadoop ● Learn to setup Hadoop environment
S/W PACKAGES AND HARDWARE APPARATUS USED	1. Operating System : 64-bit Open source Linux or its derivative 2. Programming Language: JAVA 3. Hadoop Environment
REFERENCES	● Tom White, “HADOOP The Definitive Guide”, O’REILLY ● Donald Miner & Adam Shook, “MapReduce Design Patterns”, O’REILLY
STEPS	**Refer to theory, algorithm, test input, test output.**
INSTRUCTIONS FOR WRITING JOURNAL	1. Date 2. Assignment no. 3. Problem definition 4. Learning objective 5. Learning outcome 6. Related Mathematics 7. Concepts related Theory 8. Test cases 9. Program code with proper documentation. 10. Output of program. 11. Conclusion and applications (the verification and testing of outcomes)

Head of Department
(Dr. M.S.Takalikar)

Subject Co-ordinator
(Dr. S.S.Sonawane)

Assignment No. 11

● **Aim**: Hadoop code to count number of occurrences.

● **Problem Statement / Definition**: Write a code in JAVA for a simple Word Count application that counts the number of occurrences of each word in a given input set using the Hadoop Map-Reduce framework on local-standalone set-up.

● **Prerequisites** : JAVA Programming

● **Learning Objectives**

- Learn MapReduce using Hadoop
- Learn to setup Hadoop environment

● Learning Outcome:

Students will be able to decompose problem into subproblems and to learn how to implement counting using Hadoop.

● **Related Mathematics :**

Mathematical Model

Let S be the system set:

S = {s; e; X; Y; Fme;DD;NDD; Fc; Sc}

s=start state ,e=end state

X=set of inputs X = {X1,X2,X3,X4}

where

X1 = Word count

X2 = U

X3 = Occurrences

X4 = Timestamp

Y= set of outputs Y = {Y1,Y2}

Y1 = Lines

Y2 = Average rating

Fme is the set of main functions

Fme = {f1,f2}

where

DD= Deterministic Data Text Data

NDD=Non-deterministic data No non deterministic data

Fc =failure case: No failure case identified for this application

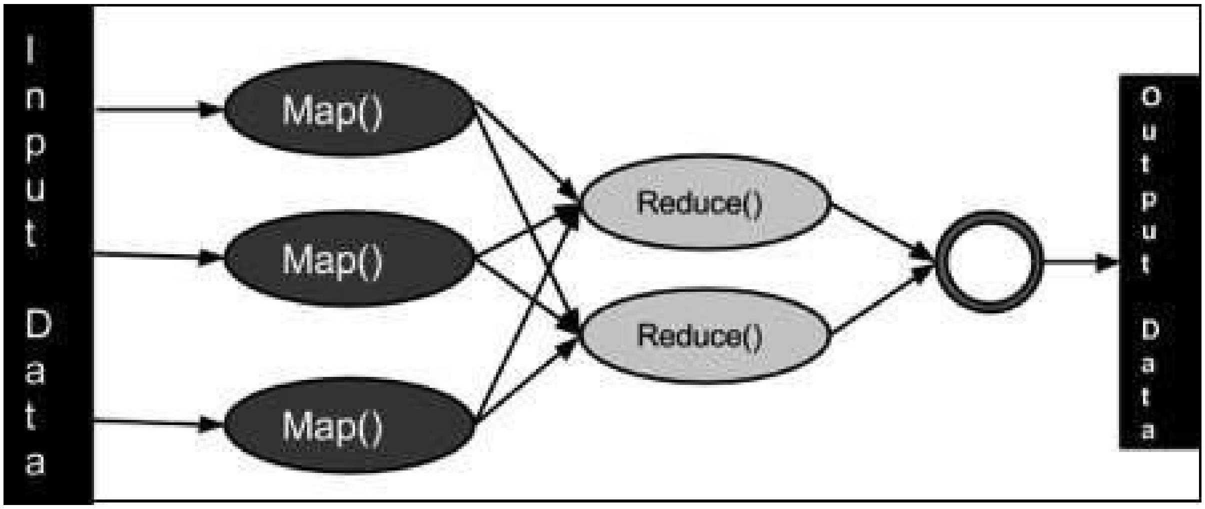
● **Theory**

Hadoop

Hadoop is an open source distributed processing framework that manages data processing and storage for big data applications running in clustered systems. It is at the center of a growing ecosystem of big data technologies that are primarily used to support advanced analytics initiatives, including predictive analysis, data mining and machine learning applications. Hadoop can handle various forms of structured and unstructured data, giving users more flexibility for collecting, processing and analyzing data than relational databases and data warehouse provide.

MapReduce

MapReduce is a processing technique and a program model for distributed computing based on java. The MapReduce algorithm contains two important tasks, namely Map and Reduce. Map takes a set of data and converts it into another set of data, where individual elements are broken down into tuples (key/value pairs). Secondly, reduce task, which takes the output from a map as an input and combines those data tuples into a smaller set of tuples. As the sequence of the name MapReduce implies, the reduce task is always performed after the map job.



●During a MapReduce job, Hadoop sends the Map and Reduce tasks to the appropriate servers in the cluster.

● The framework manages all the details of data-passing such as issuing tasks, verifying task completion, and copying data around the cluster between the nodes.

● Most of the computing takes place on nodes with data on local disks that reduces the network traffic.

● After completion of the given tasks, the cluster collects and reduces the data to form an appropriate result, and sends it back to the Hadoop server.

Mapper Class

The first stage in Data Processing using MapReduce is the Mapper Class. Here, RecordReader processes each Input record and generates the respective key-value pair. Hadoop's Mapper store saves this intermediate data into the local disk.

- Input Split

It is the logical representation of data. It represents a block of work that contains a single map task in the MapReduce Program.

Record Reader

It interacts with the Input split and converts the obtained data in the form of Key-Value Pairs.

Reducer Class

The Intermediate output generated from the mapper is fed to the reducer which processes it and generates the final output which is then saved in the HDFS.

Driver Class

The major component in a MapReduce job is a Driver Class. It is responsible for setting up a MapReduce Job to run-in Hadoop. We specify the names of Mapper and Reducer Classes long with data types and their respective job names.

How to run Hadoop Program:

1.start hadoop. start-all.sh

2.Check all components of Hadoop whether it is ready or not jps

3.Assuming environment variables are set as follows:

```
export JAVA_HOME=/usr/java/default

export PATH=${JAVA_HOME}/bin:${PATH}

export HADOOP_CLASSPATH=${JAVA_HOME}/lib/tools.jar
```

4.copy the code of to the home directory

5.Compile code

```
javac -classpath <hadooop-core.jar file> -d <Your New Directory>/ <sourceCode.java>
```

6.Create JAR file for: a.Mapper Class b.Driver Class c.Reducer Class

```
jar -cvf <File you have to create> -C <Directory you have obtained in previous command>
```

7.Run code on Hadoop Framework hadoop fs -put <source file path> /input

8.Now run program using ur Jar file

```
hadoop jar <your jar file> <directory name without /> /input/<your file name> /output/<output
file name>
```

9.Read Output file

```
hadoop fs -cat /output/<your file>/part-r-00000
```

Link : http://www.pavanjaiswal.com/2015/07/hadoop-260-single-node-setup-on-fedora.html

- **Test data:**

Normal text file.

Code: Count.java

```
package demo;

import java.io.IOException;

import org.apache.hadoop.conf.Configuration;

import org.apache.hadoop.fs.Path;

import org.apache.hadoop.io.LongWritable;

import org.apache.hadoop.io.Text;

import org.apache.hadoop.mapreduce.Job;

import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;

import org.apache.hadoop.mapreduce.lib.input.TextInputFormat;

import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;

import org.apache.hadoop.mapreduce.lib.output.TextOutputFormat;

import org.apache.hadoop.util.GenericOptionsParser;

public class count

{
                public static void main(String[] args) throws IOException,
  ClassNotFoundException, InterruptedException
            {

                        Configuration c=new

                        Configuration(); String[] files=new
  GenericOptionsParser(c,args).getRemainingArgs();

                        Path input=new Path(files[0]);
                        Path output=new

                        Path(files[1]); Job j=new

                        Job(c,"wordcount");

                        j.setJarByClass(count.class);

                        j.setMapperClass(mapper.clas;

                        j.setReducerClass(reduce.clas);

                        j.setOutputKeyClass(Text.clas;

                        j.setOutputValueClass(LongWritable.class);

                        FileInputFormat.addInputPath(j, input);

                        FileOutputFormat.setOutputPath(j, output);

                        System.exit(j.waitForCompletion(true)?0:1);


            }

  }
```

Conclusion:

- Demonstrates how applications can access configuration parameters in the setup method of the Mapper (and Reducer) implementations.
- Demonstrates the utility of the GenericOptionsParser to handle generic Hadoop command-line options.
- Demonstrates how applications can use Counters and how they can set application-specific status information passed to the map (and reduce) method.

TITLE	Movie review using MapReduce
PROBLEM STATEMENT/ DEFINITION	Locate dataset (e.g., sample_weather.txt) for working on weather data which reads the text input files and finds average for temperature, dew point and wind speed using the Hadoop Map-Reduce framework on local-standalone set-up.
OBJECTIVE	● Learn Map reduce for counting occurrences using Hadoop ● Learn to setup Hadoop environment
S/W PACKAGES AND HARDWARE APPARATUS USED	1. Operating System : 64-bit Open source Linux or its derivative 2. Programming Language: JAVA 3. Hadoop Environment
REFERENCES	● Tom White, “HADOOP The Definitive Guide”, O’REILLY ● Donald Miner & Adam Shook, “MapReduce Design Patterns”, O’REILLY
STEPS	**Refer to theory, algorithm, test input, test output.**
INSTRUCTIONS FOR WRITING JOURNAL	1. Date 2. Assignment no. 3. Problem definition 4. Learning objective 5. Learning outcome 6. Related Mathematics 7. Concepts related Theory 8. Test cases 9. Program code with proper documentation. 10. Output of program. 11. Conclusion and applications (the verification and testing of outcomes)

Head of Department	**Subject Co-ordinator**
(Dr. M.S.Takalikar)	**(Dr. S.S.Sonawane)**

Assignment No. 12

● **Aim**: Movie review using MapReduce.

● **Problem Statement / Definition**: Locate dataset (e.g., sample_weather.txt) for working on weather data which reads the text input files and finds average for temperature, dew point and wind speed using the Hadoop Map-Reduce framework on local-standalone set-up.

● **Prerequisites** : JAVA Programming

● **Learning Objectives**

- Learn MapReduce using Hadoop
- Learn to setup Hadoop environment

● Learning Outcome:

Students will be able to decompose problem into subproblems and to learn how to implement counting using Hadoop.

● **Related Mathematics :**

Mathematical Model

Let S be the system set:

S = {s; e; X; Y; Fme;DD;NDD; Fc; Sc}

s=start state ,e=end state

X=set of inputs X = {X1,X2,X3,X4}

where

X1 = Word count

X2 = U

X3 = Occurrences

X4 = Timestamp

Y= set of outputs Y = {Y1,Y2}

Y1 = Lines

Y2 = Average rating

Fme is the set of main functions

Fme = {f1,f2}

where

DD= Deterministic Data Text Data

NDD=Non-deterministic data No non deterministic data

Fc =failure case: No failure case identified for this application

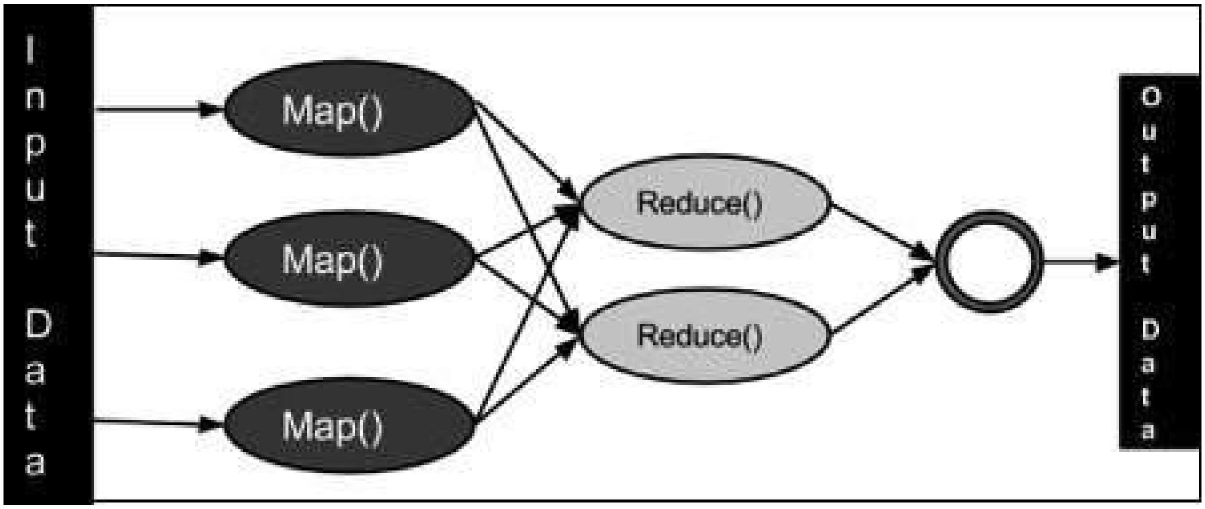
● **Theory**

Hadoop

Hadoop is an open source distributed processing framework that manages data processing and storage for big data applications running in clustered systems. It is at the center of a growing ecosystem of big data technologies that are primarily used to support advanced analytics initiatives, including predictive analysis, data mining and machine learning applications. Hadoop can handle various forms of structured and unstructured data, giving users more flexibility for collecting, processing and analyzing data than relational databases and data warehouse provide.

MapReduce

MapReduce is a processing technique and a program model for distributed computing based on java. The MapReduce algorithm contains two important tasks, namely Map and Reduce. Map takes a set of data and converts it into another set of data, where individual elements are broken down into tuples (key/value pairs). Secondly, reduce task, which takes the output from a map as an input and combines those data tuples into a smaller set of tuples. As the sequence of the name MapReduce implies, the reduce task is always performed after the map job.



●During a MapReduce job, Hadoop sends the Map and Reduce tasks to the appropriate servers in the cluster.

● The framework manages all the details of data-passing such as issuing tasks, verifying task completion, and copying data around the cluster between the nodes.

● Most of the computing takes place on nodes with data on local disks that reduces the network traffic.

● After completion of the given tasks, the cluster collects and reduces the data to form an appropriate result, and sends it back to the Hadoop server.

Mapper Class

The first stage in Data Processing using MapReduce is the Mapper Class. Here, RecordReader processes each Input record and generates the respective key-value pair. Hadoop's Mapper store saves this intermediate data into the local disk.

- Input Split

It is the logical representation of data. It represents a block of work that contains a single map task in the MapReduce Program.

Record Reader

It interacts with the Input split and converts the obtained data in the form of Key-Value Pairs.

Reducer Class

The Intermediate output generated from the mapper is fed to the reducer which processes it and generates the final output which is then saved in the HDFS.

Driver Class

The major component in a MapReduce job is a Driver Class. It is responsible for setting up a MapReduce Job to run-in Hadoop. We specify the names of Mapper and Reducer Classes long with data types and their respective job names.

How to run Hadoop Program:

1.start hadoop. start-all.sh

2.Check all components of Hadoop whether it is ready or not jps

3.Assuming environment variables are set as follows:

```
export JAVA_HOME=/usr/java/default
```

```
export PATH=${JAVA_HOME}/bin:${PATH}
```

```
export HADOOP_CLASSPATH=${JAVA_HOME}/lib/tools.jar
```

4.copy the code of to the home directory

5.Compile code

```
javac -classpath <hadooop-core.jar file> -d <Your New Directory>/ <sourceCode.java>
```

6.Create JAR file for: a.Mapper Class b.Driver Class c.Reducer Class

```
jar -cvf <File you have to create> -C <Directory you have obtained in previous command>
```

7.Run code on Hadoop Framework hadoop fs -put <source file path> /input

8.Now run program using ur Jar file

```
hadoop jar <your jar file> <directory name without /> /input/<your file name> /output/<output file name>
```

9.Read Output file

```
hadoop fs -cat /output/<your file>/part-r-00000
```

- **Test data:**

Use dataset sample_weather.txt contains temperature, dew point and wind speed.

Conclusion:

- Demonstrates how applications can access configuration parameters in the setup method of the Mapper (and Reducer) implementations.
- Demonstrates the utility of the GenericOptionsParser to handle generic Hadoop command-line options.
- Demonstrates how applications can use Counters and how they can set application-specific status information passed to the map (and reduce) method.

TITLE	Write a simple program in SCALA using Apache Spark framework.
PROBLEM STATEMENT/ DEFINITION	Implement bubble/quick sort Algorithm using scala programming
OBJECTIVE	1) Big Data Analysis with Apache Spark.
S/W PACKAGES AND HARDWARE APPARATUS USED	scala-2.11.7.deb oracle-java7-installer
REFERENCES	1) https://www.analyticsvidhya.com/blog/2017/01/scala/
STEPS	**Refer to student activity flow chart if found necessary by subject teacher and relevant to the subject manual.** **Describe steps only.**
INSTRUCTIONS FOR WRITING JOURNAL	1. title 2. Problem statement 3. Learning objective 4. Learning outcome 5. Theory (includes methods, libraries and functions, 6. Analysis (as per assignment), 7. conclusion.

______________________ ______________________________

HeadofDepartment **SubjectCo-ordinator**
(Dr. M.S.Takalikar) **(Dr. S.S.Sonawane)**

1. Title:-

Write a simple program in SCALA using Apache Spark framework.

2. Problem statement :

Implement Bubble/Quick sort Algorithm using scala programming

3. Learning objective :

Big Data Analysis with Apache Spark.

4. Learning outcome:

Students can able to write the scala program using object oriented features.

5. Theory:

1) Installing Scala

Scala can be installed in any Unix or windows based system. Below are the steps to install for Ubuntu (14.04) for scala version 2.11.7. I am showing the steps for installing Scala (2.11.7) with Java version 7. It is necessary to install Java before installing Scala. You can also install latest version of Scala(2.12.1) as well.

Step 0: Open the terminal

Step 1: Install Java

```
 $ sudo apt-add-repository ppa:webupd8team/java
 $ sudo apt-get update
 $ sudo apt-get install oracle-java7-installer
```

If you are asked to accept Java license terms, click on “Yes” and proceed. Once finished, let us check whether Java has installed successfully or not. To check the Java version and installation, you can type:

```
 $ java -version
```

Step 2: Once Java is installed, we need to install Scala

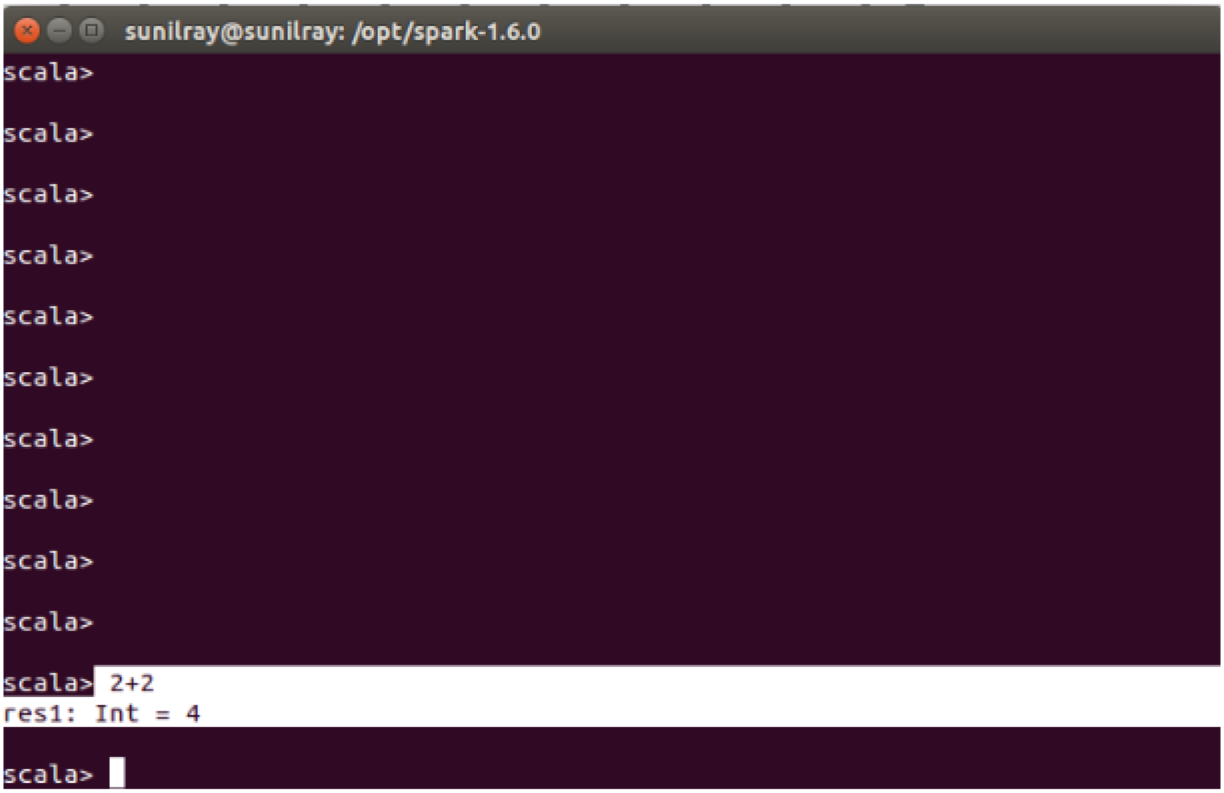
```
 $ cd ~/Downloads

 $ wget http://www.scala-lang.org/files/archive/scala-2.11.7.deb

 $ sudo dpkg -i scala-2.11.7.deb

 $ scala –version
```

This will show you the version of Scala installed

2. Prerequisites for Learning Scala

Scala being an easy to learn language has minimal prerequisites. If you are someone with basic knowledge of C/C++, then you will be easily able to get started with Scala. Since Scala is developed on top of Java. Basic programming function in Scala is similar to Java. So, if you have some basic knowledge of Java syntax and OOPs concept, it would be helpful for you to work in Scala.

Warming up: Running your first Scala program in Shell:
Let’s write a first program which adds two numbers.



```
sunilray@sunilray: /opt/spark-1.6.0
scala>

scala>

scala>

scala>

scala>

scala>

scala>

scala>

scala>

scala>

scala> 2+2
res1: Int = 4

scala> 
```

3. Things to note about Scala

- It is case sensitive
- If you are writing a program in Scala, you should save this program using “.scala”
- Scala execution starts from main() methods
- Any identifier name cannot begin with numbers. For example, variable name “123salary” is invalid.
- You can not use Scala reserved keywords for variable declarations or constant or any identifiers.

4. Variable declaration in Scala

In Scala, you can declare a variable using ‘var’ or ‘val’ keyword. The decision is based on whether it is a constant or a variable. If you use ‘var’ keyword, you define a variable as mutable variable. On the other hand, if you use ‘val’, you define it as immutable. Let’s first declare a variable using “var” and then using “val”.

4.1 Declare using var

```
var Var1 : String = "Ankit"
```

In the above Scala statement, you declare a mutable variable called “Var1” which takes a string value. You can also write the above statement without specifying the type of variable. Scala will automatically identify it. For example:

```
var Var1 = "Gupta"
```

4.2 Declare using val

```
val Var2 : String = "Ankit"
```

In the above Scala statement, we have declared an immutable variable “Var2” which takes a string “Ankit”. Try it for without specifying the type of variable. If you want to read about mutable and immutable please refer this link.

5. Operations on variables

You can perform various operations on variables. There are various kinds of operators defined in Scala. For example: Arithmetic Operators, Relational Operators, Logical Operators, Bitwise Operators, Assignment Operators.

Lets see “+” , “==” operators on two variables ‘Var4’, “Var5”. But, before that, let us first assign values to “Var4” and “Var5”.

```
scala> var Var4 = 2
Output: Var4: Int = 2
scala> var Var5 = 3
Output: Var5: Int = 3
```

Now, let us apply some operations using operators in Scala.

Apply ‘+’ operator

```
Var4+Var5
Output:
res1: Int = 5
```

Apply “==” operator

```
Var4==Var5
Output:
res2: Boolean = false
```

If you want to know complete list of operators in Scala refer this link:

6. The if-else expression in Scala

In Scala, if-else expression is used for conditional statements. You can write one or more conditions inside “if”. Let’s declare a variable called “Var3” with a value 1 and then compare “Var3” using if-else expression.

```
var Var3 =1
if (Var3 ==1){
 println("True")}else{
 println("False")}
Output: True
```

In the above snippet, the condition evaluates to True and hence True will be printed in the output.

7. Iteration in Scala

Like most languages, Scala also has a FOR-loop which is the most widely used method for iteration. It has a simple syntax too.

```
for( a <- 1 to 10){
 println( "Value of a: " + a );
 }
Output:
Value of a: 1
Value of a: 2
Value of a: 3
Value of a: 4
Value of a: 5
Value of a: 6
Value of a: 7
Value of a: 8
Value of a: 9
Value of a: 10
```

Scala also supports “while” and “do while” loops. If you want to know how both work, please refer this link.

8. Declare a simple function in Scala and call it by passing value

You can define a function in Scala using “def” keyword. Let’s define a function called “mul2” which will take a number and multiply it by 10. You need to define the return type of function, if a function not returning any value you should use the “Unit” keyword.

In the below example, the function returns an integer value. Let’s define the function “mul2”:

```
def mul2(m: Int): Int = m * 10
Output: mul2: (m: Int)Int
```

Now let’s pass a value 2 into mul2

```
mul2(2)
Output:
res9: Int = 20
```

6. Analysis (as per assignment),

7. conclusion:- Hence students can able to understand the basic concept of scala program using apache spark framework.