SIES (NERUL) COLLEGE OF ARTS, SCIENCE AND COMMERCE

NAAC ACCREDITED 'A' GRADE COLLEGE

(ISO 9001:2015 CERTIFIED INSTITUTION)

NERUL, NAVI MUMBAI - 400706

Certificate

Seat No: 2630267

Certified that Vishal V. Varma

Of Class MSC.IT PART-1 has duly completed the practical course in the subject of Big Data Analystics during the academic year 2021-2022 as per the syllabus prescribed by the University of Mumbai.

Subject Teacher **External Examiner**

Head of Department **Principal**

INDEX

Practical No : 1

Install, configure and run Hadoop and HDFS ad explore HDFS

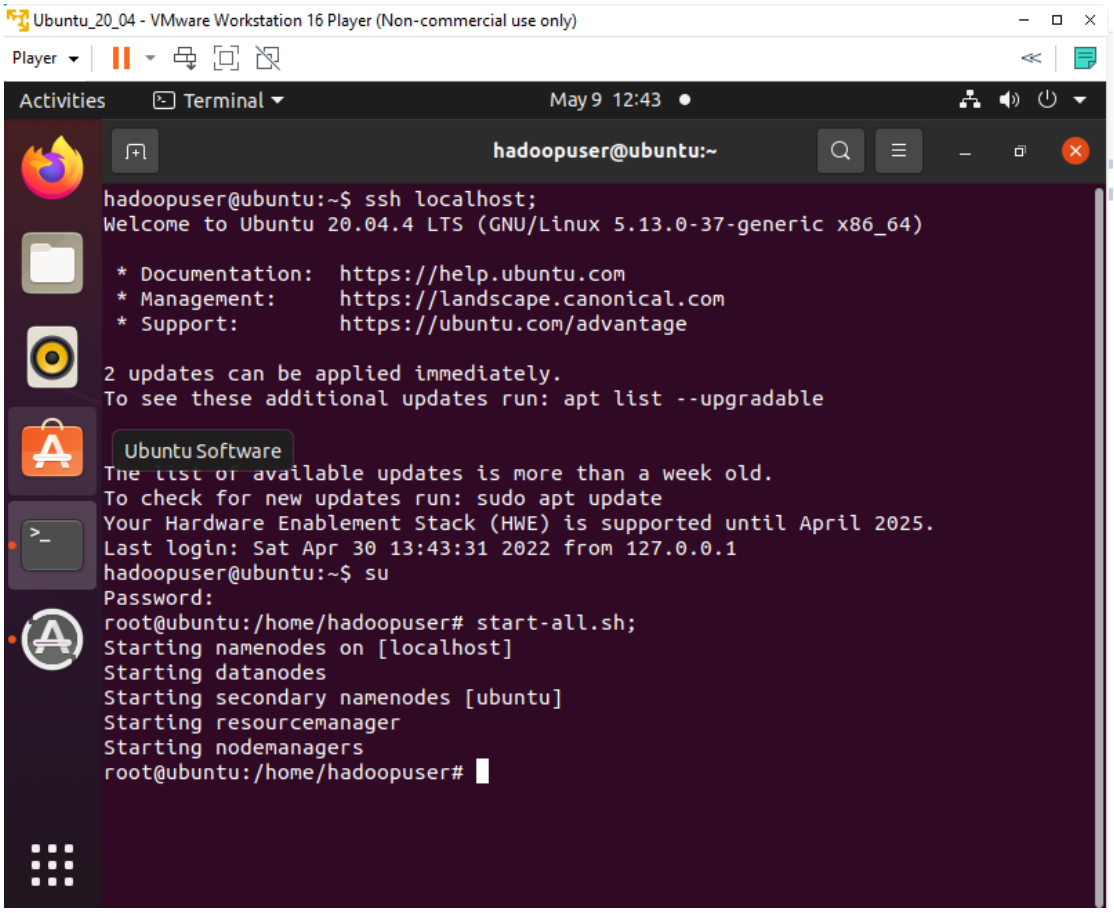
Step 1: Go to the terminal and start the Hadoop using following commands:

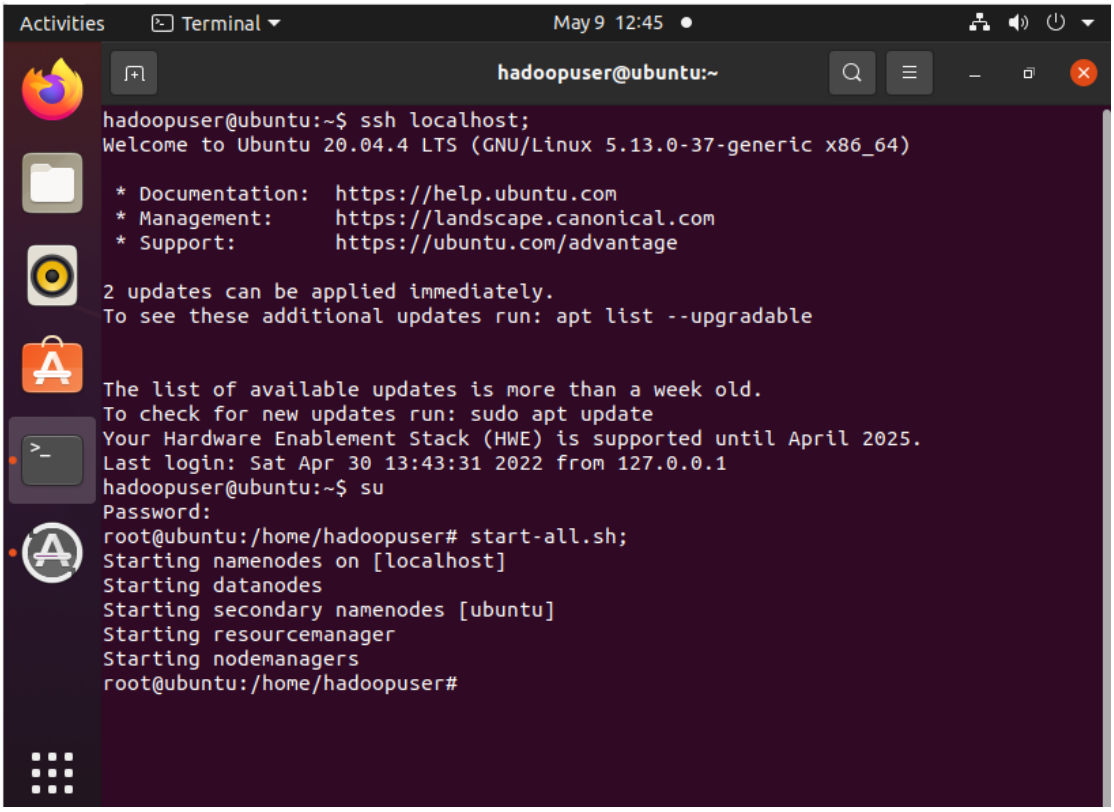
Ssh localhost;

Su

Password:

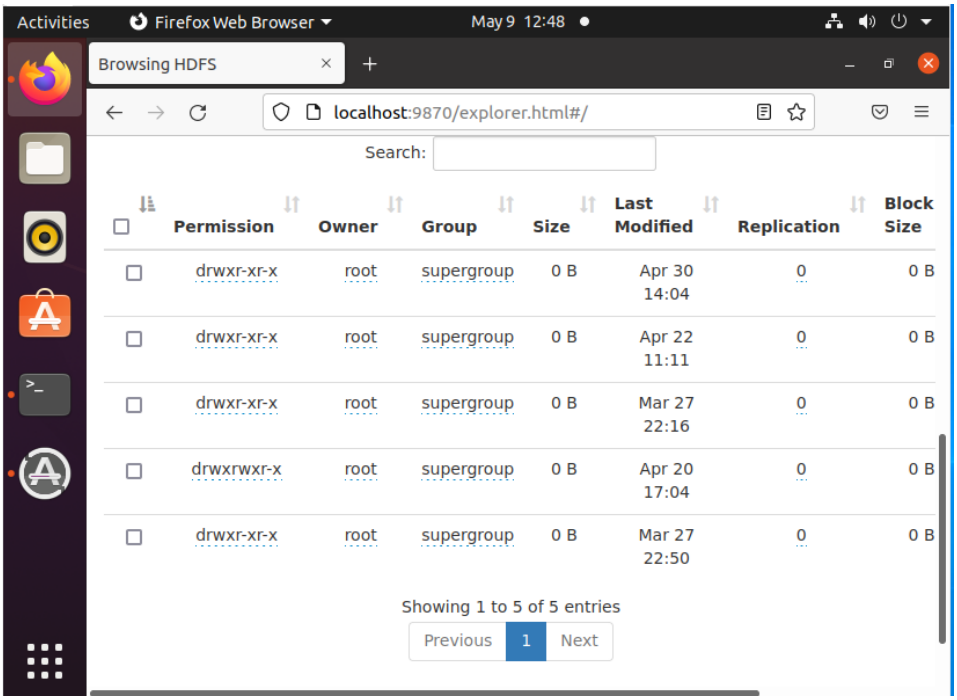
Start-all.sh;





Step 2: Browsing the HDFS directories through browser

Open the browser and type - localhost:9870



Step 3:

Browsing the HDFS directories through terminal

Command: hdfs dfs -ls /

```
root@ubuntu:/home/hadoopuser# hdfs dfs -ls /
Found 5 items
drwxr-xr-x   - root supergroup          0 2022-04-30 14:04 /Pig_Data
drwxr-xr-x   - root supergroup          0 2022-04-22 11:11 /WordC
drwxr-xr-x   - root supergroup          0 2022-03-27 22:16 /opt
drwxrwxr-x   - root supergroup          0 2022-04-20 17:04 /tmp
drwxr-xr-x   - root supergroup          0 2022-03-27 22:50 /user
root@ubuntu:/home/hadoopuser#
```

Step 4: To create a new directory enter the following command:

Hdfs dfs -mkdir /dir1

```
root@ubuntu:/home/hadoopuser# hdfs dfs -mkdir /dir1
root@ubuntu:/home/hadoopuser# hdfs dfs -ls /
Found 6 items
drwxr-xr-x   - root supergroup          0 2022-04-30 14:04 /Pig_Data
drwxr-xr-x   - root supergroup          0 2022-04-22 11:11 /WordC
drwxr-xr-x   - root supergroup          0 2022-05-09 12:53 /dir1
drwxr-xr-x   - root supergroup          0 2022-03-27 22:16 /opt
drwxrwxr-x   - root supergroup          0 2022-04-20 17:04 /tmp
drwxr-xr-x   - root supergroup          0 2022-03-27 22:50 /user
root@ubuntu:/home/hadoopuser#
```

Step 5: To remove a new directory enter the following command:

Hdfs dfs -rmdir /dir1

```
root@ubuntu:/home/hadoopuser# hdfs dfs -rmdir /dir1
root@ubuntu:/home/hadoopuser# hdfs dfs -ls /
Found 5 items
drwxr-xr-x   - root supergroup          0 2022-04-30 14:04 /Pig_Data
drwxr-xr-x   - root supergroup          0 2022-04-22 11:11 /WordC
drwxr-xr-x   - root supergroup          0 2022-03-27 22:16 /opt
drwxrwxr-x   - root supergroup          0 2022-04-20 17:04 /tmp
drwxr-xr-x   - root supergroup          0 2022-03-27 22:50 /user
root@ubuntu:/home/hadoopuser#
```

Step 6: To know the current directory pwd command is used

```
root@ubuntu:/home/hadoopuser# pwd
/home/hadoopuser
root@ubuntu:/home/hadoopuser#
```

Practical No : 2

Implement word count / frequency programs using MapReduce

```
hadoopuser@ubuntu:~$ sudo mkdir /home/hadoopuser/Downloads/wcpract
hadoopuser@ubuntu:~$ sudo mkdir /home/hadoopuser/Downloads/wcpract/input
hadoopuser@ubuntu:~$
```

```
hadoopuser@ubuntu:~$ sudo nano /home/hadoopuser/Downloads/wcpract/input/WordCount.java
```

```
hadoopuser@ubuntu:~$ sudo mkdir /home/hadoopuser/Downloads/wcpract
hadoopuser@ubuntu:~$ sudo mkdir /home/hadoopuser/Downloads/wcpract/input
hadoopuser@ubuntu:~$ sudo nano /home/hadoopuser/Downloads/wcpract/input/WordCount.java
```

Write the java code

```
import java.io.IOException;
import java.util.StringTokenizer;

import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Job;
import org.apache.hadoop.mapreduce.Mapper;
import org.apache.hadoop.mapreduce.Reducer;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;

public class WordCount {

  public static class TokenizerMapper
       extends Mapper<Object, Text, Text, IntWritable>{

    private final static IntWritable one = new IntWritable(1);
    private Text word = new Text();

    public void map(Object key, Text value, Context context
                    ) throws IOException, InterruptedException {
      StringTokenizer itr = new StringTokenizer(value.toString());
      while (itr.hasMoreTokens()) {
        word.set(itr.nextToken());
        context.write(word, one);
      }
    }
  }

  public static class IntSumReducer
       extends Reducer<Text,IntWritable,Text,IntWritable> {
    private IntWritable result = new IntWritable();
```

```java
    private IntWritable result = new IntWritable();

    public void reduce(Text key, Iterable<IntWritable> values,
                       Context context
                       ) throws IOException, InterruptedException {
      int sum = 0;
      for (IntWritable val : values) {
        sum += val.get();
      }
      result.set(sum);
      context.write(key, result);
    }
  }

  public static void main(String[] args) throws Exception {
    Configuration conf = new Configuration();
    Job job = Job.getInstance(conf, "word count");
    job.setJarByClass(WordCount.class);
    job.setMapperClass(TokenizerMapper.class);
    job.setCombinerClass(IntSumReducer.class);
    job.setReducerClass(IntSumReducer.class);
    job.setOutputKeyClass(Text.class);
    job.setOutputValueClass(IntWritable.class);
    FileInputFormat.addInputPath(job, new Path(args[0]));
    FileOutputFormat.setOutputPath(job, new Path(args[1]));
    System.exit(job.waitForCompletion(true) ? 0 : 1);
  }
}
```

```
hadoopuser@ubuntu:~$ sudo nano /home/hadoopuser/Downloads/wcpract/input/input.txt
hadoopuser@ubuntu:~$
```

```
Hello how are you
I am fine
Tell me about yourself
Hadoop
hadoop mapred mapreduce mapred
hadoop wordcount
hadoop
bye bye bye
```

```
root@ubuntu:/home/hadoopuser# hadoop com.sun.tools.javac.Main /home/hadoopuser/Downloads/wcpract/input/WordCount.java
/opt/hadoop_env/hadoop/libexec/hadoop-functions.sh: line 2401: HADOOP_COM.SUN.TOOLS.JAVAC.MAIN_USER: invalid variable name
/opt/hadoop_env/hadoop/libexec/hadoop-functions.sh: line 2366: HADOOP_COM.SUN.TOOLS.JAVAC.MAIN_USER: invalid variable name
/opt/hadoop_env/hadoop/libexec/hadoop-functions.sh: line 2461: HADOOP_COM.SUN.TOOLS.JAVAC.MAIN_OPTS: invalid variable name
root@ubuntu:/home/hadoopuser#
```

```
root@ubuntu:/home/hadoopuser/Downloads/wcpract/input# hadoop com.sun.tools.javac.Main WordCount.java
/opt/hadoop_env/hadoop/libexec/hadoop-functions.sh: line 2401: HADOOP_COM.SUN.TOOLS.JAVAC.MAIN_USER: invalid variable name
/opt/hadoop_env/hadoop/libexec/hadoop-functions.sh: line 2366: HADOOP_COM.SUN.TOOLS.JAVAC.MAIN_USER: invalid variable name
/opt/hadoop_env/hadoop/libexec/hadoop-functions.sh: line 2461: HADOOP_COM.SUN.TOOLS.JAVAC.MAIN_OPTS: invalid variable name
```

```
root@ubuntu:/home/hadoopuser# jar cf /home/hadoopuser/Downloads/wcpract/input/wc.jar /home/hadoopuser/Downloads/wcpract/input/WordCount*.class
```

```
root@ubuntu:/home/hadoopuser/Downloads/wcpract/input# jar cf wc.jar WordCount*.class
root@ubuntu:/home/hadoopuser/Downloads/wcpract/input# ls
 input.txt   wc.jar  'WordCount$IntSumReducer.class'  'WordCount$TokenizerMapper.class'   WordCount.class   WordCount.java
```

```
root@ubuntu:/home/hadoopuser# jar cf /home/hadoopuser/Downloads/wcpract/input/wc.jar /home/hadoopuser/Downloads/wcpract/input/WordCount*.class
root@ubuntu:/home/hadoopuser#
```

```
root@ubuntu:/home/hadoopuser# hdfs dfs -ls /
Found 3 items
drwxr-xr-x   - root supergroup          0 2022-03-27 22:16 /opt
drwxrwxr-x   - root supergroup          0 2022-03-27 23:25 /tmp
drwxr-xr-x   - root supergroup          0 2022-03-27 22:50 /user
```

```
root@ubuntu:/home/hadoopuser# hdfs dfs -put /home/hadoopuser/Downloads/wcpract /
root@ubuntu:/home/hadoopuser# hdfs dfs -ls /
Found 4 items
drwxr-xr-x   - root supergroup          0 2022-03-27 22:16 /opt
drwxrwxr-x   - root supergroup          0 2022-03-27 23:25 /tmp
drwxr-xr-x   - root supergroup          0 2022-03-27 22:50 /user
drwxr-xr-x   - root supergroup          0 2022-05-20 15:39 /wcpract
```

```
root@ubuntu:/home/hadoopuser# hadoop jar /home/hadoopuser/Downloads/wcpract/input/wc.jar WordCount /wcpract/input/input.txt /wcpract/output
```

```
root@ubuntu:/home/hadoopuser# hadoop jar /home/hadoopuser/Downloads/wcpract/input/wc.jar WordCount /wcpract/input/input.txt /wcpract/output
2022-05-20 16:08:32,816 INFO client.RMProxy: Connecting to ResourceManager at /0.0.0.0:8032
2022-05-20 16:08:34,873 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not performed. Implement the Tool interface and
 execute your application with ToolRunner to remedy this.
2022-05-20 16:08:34,978 INFO mapreduce.JobResourceUploader: Disabling Erasure Coding for path: /tmp/hadoop-yarn/staging/root/.staging/job_1653
042898358_0001
2022-05-20 16:08:38,371 INFO input.FileInputFormat: Total input files to process : 1
2022-05-20 16:08:38,594 INFO mapreduce.JobSubmitter: number of splits:1
2022-05-20 16:08:39,098 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1653042898358_0001
2022-05-20 16:08:39,100 INFO mapreduce.JobSubmitter: Executing with tokens: []
2022-05-20 16:08:39,512 INFO conf.Configuration: resource-types.xml not found
2022-05-20 16:08:39,516 INFO resource.ResourceUtils: Unable to find 'resource-types.xml'.
2022-05-20 16:08:42,203 INFO impl.YarnClientImpl: Submitted application application_1653042898358_0001
2022-05-20 16:08:42,343 INFO mapreduce.Job: The url to track the job: http://ubuntu:8088/proxy/application_1653042898358_0001/
2022-05-20 16:08:42,345 INFO mapreduce.Job: Running job: job_1653042898358_0001
2022-05-20 16:09:15,461 INFO mapreduce.Job: Job job_1653042898358_0001 running in uber mode : false
2022-05-20 16:09:15,503 INFO mapreduce.Job:  map 0% reduce 0%
2022-05-20 16:10:04,395 INFO mapreduce.Job:  map 100% reduce 0%
2022-05-20 16:10:34,342 INFO mapreduce.Job:  map 100% reduce 100%
2022-05-20 16:10:36,098 INFO mapreduce.Job: Job job_1653042898358_0001 completed successfully
2022-05-20 16:10:37,285 INFO mapreduce.Job: Counters: 54
        File System Counters
                FILE: Number of bytes read=204
                FILE: Number of bytes written=469677
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=235
                HDFS: Number of bytes written=130
                HDFS: Number of read operations=8
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=2
                HDFS: Number of bytes read erasure-coded=0
        Job Counters
                Launched map tasks=1
                Launched reduce tasks=1
                Data-local map tasks=1
```

```
                Total time spent by all maps in occupied slots (ms)=41737
                Total time spent by all reduces in occupied slots (ms)=24746
                Total time spent by all map tasks (ms)=41737
                Total time spent by all reduce tasks (ms)=24746
                Total vcore-milliseconds taken by all map tasks=41737
                Total vcore-milliseconds taken by all reduce tasks=24746
                Total megabyte-milliseconds taken by all map tasks=42738688
                Total megabyte-milliseconds taken by all reduce tasks=25339904
        Map-Reduce Framework
                Map input records=8
                Map output records=22
                Map output bytes=213
                Map output materialized bytes=204
                Input split bytes=110
                Combine input records=22
                Combine output records=17
                Reduce input groups=17
                Reduce shuffle bytes=204
                Reduce input records=17
                Reduce output records=17
                Spilled Records=34
                Shuffled Maps =1
                Failed Shuffles=0
                Merged Map outputs=1
                GC time elapsed (ms)=202
                CPU time spent (ms)=2850
                Physical memory (bytes) snapshot=305930240
                Virtual memory (bytes) snapshot=4991074304
                Total committed heap usage (bytes)=153751552
                Peak Map Physical memory (bytes)=196018176
                Peak Map Virtual memory (bytes)=2491080704
                Peak Reduce Physical memory (bytes)=109912064
                Peak Reduce Virtual memory (bytes)=2499993600
        Shuffle Errors
                BAD_ID=0
                CONNECTION=0
```

```
        Shuffle Errors
                BAD_ID=0
                CONNECTION=0
                IO_ERROR=0
                WRONG_LENGTH=0
                WRONG_MAP=0
                WRONG_REDUCE=0
        File Input Format Counters
                Bytes Read=125
        File Output Format Counters
                Bytes Written=130
```

```
root@ubuntu:/home/hadoopuser# hdfs dfs -ls /wcpract/output
Found 2 items
-rw-r--r--   3 root supergroup          0 2022-05-20 16:10 /wcpract/output/_SUCCESS
-rw-r--r--   3 root supergroup        130 2022-05-20 16:10 /wcpract/output/part-r-00000
```

```
root@ubuntu:/home/hadoopuser# hdfs dfs -cat /wcpract/output/part-r-00000
Hadoop  1
Hello   1
I       1
Tell    1
about   1
am      1
are     1
bye     3
fine    1
hadoop  3
how     1
mapred  2
mapreduce       1
me      1
wordcount       1
you     1
yourself        1
```

Practical No : 3

Implement the program in practical 3 using Pig.

Step 1 :- Open new terminal and type ssh localhost and provide the user name and password

```
hadoopuser@ubuntu:~$ ssh localhost
Welcome to Ubuntu 20.04.4 LTS (GNU/Linux 5.13.0-37-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

2 updates can be applied immediately.
To see these additional updates run: apt list --upgradable


The list of available updates is more than a week old.
To check for new updates run: sudo apt update
Your Hardware Enablement Stack (HWE) is supported until April 2025.
Last login: Wed Jun  8 21:13:55 2022 from 127.0.0.1
hadoopuser@ubuntu:~$ su
Password:
root@ubuntu:/home/hadoopuser# start-all.sh
Starting namenodes on [localhost]
Starting datanodes
Starting secondary namenodes [ubuntu]
Starting resourcemanager
Starting nodemanagers
```

Step 1 :- Again Open second terminal for pig and type ssh localhost and provide the user name and password

```
root@ubuntu:/home/hadoopuser# pig
2022-06-08 21:31:34,587 INFO pig.ExecTypeProvider: Trying ExecType : LOCAL
2022-06-08 21:31:34,741 INFO pig.ExecTypeProvider: Trying ExecType : MAPREDUCE
2022-06-08 21:31:34,742 INFO pig.ExecTypeProvider: Picked MAPREDUCE as the ExecT
ype
2022-06-08 21:31:36,806 [main] INFO  org.apache.pig.Main - Apache Pig version 0.
17.0 (r1797386) compiled Jun 02 2017, 15:41:58
2022-06-08 21:31:36,862 [main] INFO  org.apache.pig.Main - Logging error message
s to: /home/hadoopuser/pig_1654704096714.log
2022-06-08 21:31:39,194 [main] INFO  org.apache.pig.impl.util.Utils - Default bo
otup file /root/.pigbootup not found
2022-06-08 21:31:45,685 [main] INFO  org.apache.hadoop.conf.Configuration.deprec
ation - mapred.job.tracker is deprecated. Instead, use mapreduce.jobtracker.addr
ess
2022-06-08 21:31:45,686 [main] INFO  org.apache.pig.backend.hadoop.executionengi
ne.HExecutionEngine - Connecting to hadoop file system at: hdfs://localhost:9820
2022-06-08 21:31:58,892 [main] INFO  org.apache.pig.PigServer - Pig Script ID fo
r the session: PIG-default-a3185837-b4f1-4dbe-97f3-996f384b8556
2022-06-08 21:31:58,938 [main] WARN  org.apache.pig.PigServer - ATS is disabled
since yarn.timeline-service.enabled set to false
grunt>
```

Step:- 3 In Hadoop terminal write hdfs dfs -mkdir hdfs://localhost:9820/pig-data

```
root@ubuntu:/home/hadoopuser# hdfs dfs -mkdir hdfs://localhost:9820/pig-data
```

Step 4: Now add the new text file by writing : nano /home/hadoopuser/Downloads/students.txt

```
root@ubuntu:/home/hadoopuser# nano /home/hadoopuser/Downloads/students.txt
root@ubuntu:/home/hadoopuser#
```

And write the below data in student.txt file

```
101,John,7.5
102,ALex,10.0
103,Philip,6.6
104,Terry,8.5
105,Jessi,8.0
106,Terrence,7.5
```

Step 5: Now create second txt file : nanp/home/hadoopuser/Download/Department.txt

```
root@ubuntu:/home/hadoopuser# nano /home/hadoopuser/Downloads/departments.txt
root@ubuntu:/home/hadoopuser#
```

And write the below data in student.txt file

```
101,10,MSC
102,11,MBA
103,12,MCA
104,10,MSC
105,11,MBA
106,12,MCA
```

Step 6: hdfs dfs -put /home/hadoopuser/**Download/ student.txt hdfs://localhost:9820/pig-data/** student.txt

: hdfs dfs -put /home/hadoopuser/**Download/ Department.txt hdfs://localhost:9820/pig-data/ Department.txt**

```
root@ubuntu:/home/hadoopuser# hdfs dfs -put /home/hadoopuser/Downloads/students.txt hdfs://localhost:9820/pig-data/students.txt
root@ubuntu:/home/hadoopuser# hdfs dfs -put /home/hadoopuser/Downloads/departments.txt hdfs://localhost:9820/pig-data/departments.txt
root@ubuntu:/home/hadoopuser#
```

Step 7 : hdfs dfs -ls /pig-data

```
root@ubuntu:/home/hadoopuser# hdfs dfs -ls /pig-data/
Found 2 items
-rw-r--r--   3 root supergroup         66 2022-06-09 00:23 /pig-data/departments.txt
-rw-r--r--   3 root supergroup         87 2022-06-09 00:22 /pig-data/students.txt
```

```
grunt> fs -cat /pig-data/students.txt
101,John,7.5
102,ALex,10.0
103,Philip,6.6
104,Terry,8.5
105,Jessi,8.0
106,Terrence,7.5
```

```
grunt> fs -cat /pig-data/departments.txt
101,10,MSC
102,11,MBA
103,12,MCA
104,10,MSC
105,11,MBA
106,12,MCA
```

Filter:

Find the tuples of those student where the GPA is greater than 8.0.

```
grunt> A = load '/pig-data/students.txt' USING PigStorage(',') as (rollno:int, name:chararray, gpa:float);
grunt> B = filter A by gpa > 8.0;
2022-06-09 01:19:36,793 [main] WARN  org.apache.pig.newplan.BaseOperatorPlan - Encountered Warning IMPLICIT_CAST_TO_DOUBLE 1 time(s).
grunt> DUMP B;
```

```
2022-06-09 01:23:50,211 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MapReduceLauncher - Success!
2022-06-09 01:23:50,262 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - yarn.resourcemanager.system-metrics-publisher.enabled
is deprecated. Instead, use yarn.system-metrics-publisher.enabled
2022-06-09 01:23:50,268 [main] INFO  org.apache.pig.data.SchemaTupleBackend - Key [pig.schematuple] was not set... will not generate code.
2022-06-09 01:23:50,367 [main] INFO  org.apache.hadoop.mapreduce.lib.input.FileInputFormat - Total input files to process : 1
2022-06-09 01:23:50,368 [main] INFO  org.apache.pig.backend.hadoop.executionengine.util.MapRedUtil - Total input paths to process : 1
(102,ALex,10.0)
(104,Terry,8.5)
grunt>
```

FOREACH:- Display the name of all students in uppercase.

```
grunt> A1 = load '/pig-data/students.txt' USING PigStorage(',') as (rollno:int, name:chararray, gpa:float);
2022-06-09 01:31:44,071 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - yarn.resourcemanager.system-metrics-publisher.enabled
is deprecated. Instead, use yarn.system-metrics-publisher.enabled
2022-06-09 01:31:44,112 [main] WARN  org.apache.pig.newplan.BaseOperatorPlan - Encountered Warning IMPLICIT_CAST_TO_DOUBLE 1 time(s).
grunt> B1 = foreach A1 generate UPPER (name);
2022-06-09 01:31:53,568 [main] WARN  org.apache.pig.newplan.BaseOperatorPlan - Encountered Warning IMPLICIT_CAST_TO_DOUBLE 1 time(s).
grunt> DUMP B1;
```

```
2022-06-09 01:35:22,428 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MapReduceLauncher - Success!
2022-06-09 01:35:22,452 [main] INFO  org.apache.pig.data.SchemaTupleBackend - Key [pig.schematuple] was not set... will not generate code.
2022-06-09 01:35:22,498 [main] INFO  org.apache.hadoop.mapreduce.lib.input.FileInputFormat - Total input files to process : 1
2022-06-09 01:35:22,498 [main] INFO  org.apache.pig.backend.hadoop.executionengine.util.MapRedUtil - Total input paths to process : 1
(JOHN)
(ALEX)
(PHILIP)
(TERRY)
(JESSI)
(TERRENCE)
grunt>
```

Group :Group tuples of students based on their GPA.

```
grunt> A3 = load '/pig-data/students.txt' USING PigStorage(',') as (rollno:int, name:chararray, gpa:float);
2022-06-09 01:36:23,940 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - yarn.resourcemanager.system-metrics-publisher.enabled
is deprecated. Instead, use yarn.system-metrics-publisher.enabled
2022-06-09 01:36:23,966 [main] WARN  org.apache.pig.newplan.BaseOperatorPlan - Encountered Warning IMPLICIT_CAST_TO_DOUBLE 1 time(s).
grunt> B3 = GROUP A3 BY gpa;
2022-06-09 01:36:38,346 [main] WARN  org.apache.pig.newplan.BaseOperatorPlan - Encountered Warning IMPLICIT_CAST_TO_DOUBLE 1 time(s).
grunt> DUBP B3;
```

```
o compute warning aggregation.
2022-06-09 01:41:34,745 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MapReduceLauncher - Success!
2022-06-09 01:41:34,885 [main] INFO  org.apache.pig.data.SchemaTupleBackend - Key [pig.schematuple] was not set... will not generate code.
2022-06-09 01:41:35,345 [main] INFO  org.apache.hadoop.mapreduce.lib.input.FileInputFormat - Total input files to process : 1
2022-06-09 01:41:35,346 [main] INFO  org.apache.pig.backend.hadoop.executionengine.util.MapRedUtil - Total input paths to process : 1
(6.6,{(103,Philip,6.6)})
(7.5,{(106,Terrence,7.5),(101,John,7.5)})
(8.0,{(105,Jessi,8.0)})
(8.5,{(104,Terry,8.5)})
(10.0,{(102,ALex,10.0)})
grunt>
```

Distinct : To remove duplicate tuples of students.

```
o compute warning aggregation.
2022-06-09 01:49:07,713 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MapReduceLauncher - Success!
2022-06-09 01:49:07,716 [main] INFO  org.apache.pig.data.SchemaTupleBackend - Key [pig.schematuple] was not set... will not generate code.
2022-06-09 01:49:07,740 [main] INFO  org.apache.hadoop.mapreduce.lib.input.FileInputFormat - Total input files to process : 1
2022-06-09 01:49:07,741 [main] INFO  org.apache.pig.backend.hadoop.executionengine.util.MapRedUtil - Total input paths to process : 1
(ALEX)
(JOHN)
(JESSI)
(TERRY)
(PHILIP)
(TERRENCE)
grunt>
```

```
grunt> A4 = load '/pig-data/students.txt' USING PigStorage(',') as (rollno:int, name:chararray, gpa:float);
2022-06-09 01:44:55,058 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - yarn.resourcemanager.system-metrics-publisher.enabled
is deprecated. Instead, use yarn.system-metrics-publisher.enabled
2022-06-09 01:44:55,080 [main] WARN  org.apache.pig.newplan.BaseOperatorPlan - Encountered Warning IMPLICIT_CAST_TO_DOUBLE 1 time(s).
grunt> B4 = foreach A4 generate UPPER (name);
2022-06-09 01:45:02,508 [main] WARN  org.apache.pig.newplan.BaseOperatorPlan - Encountered Warning IMPLICIT_CAST_TO_DOUBLE 1 time(s).
grunt> C4 = DISTINCT B4;
2022-06-09 01:45:08,206 [main] WARN  org.apache.pig.newplan.BaseOperatorPlan - Encountered Warning IMPLICIT_CAST_TO_DOUBLE 1 time(s).
grunt> DUMP C4;
```

Join : To join two relations namely, “student” and “department” based on the values contained in the “rollno” column.

```
grunt> A5 = load '/pig-data/students.txt' USING PigStorage(',') as (rollno:int, name:chararray, gpa:float);
2022-06-09 01:53:06,849 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - yarn.resourcemanager.system-metrics-publisher.enabled
is deprecated. Instead, use yarn.system-metrics-publisher.enabled
2022-06-09 01:53:06,859 [main] WARN  org.apache.pig.newplan.BaseOperatorPlan - Encountered Warning IMPLICIT_CAST_TO_DOUBLE 1 time(s).
grunt> B5 = load '/pig-data/departments.txt' USING PigStorage(',') as (rollno:int, deptno:int, deptname:chararray);
2022-06-09 01:53:13,294 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - yarn.resourcemanager.system-metrics-publisher.enabled
is deprecated. Instead, use yarn.system-metrics-publisher.enabled
2022-06-09 01:53:13,315 [main] WARN  org.apache.pig.newplan.BaseOperatorPlan - Encountered Warning IMPLICIT_CAST_TO_DOUBLE 1 time(s).
grunt> C5 = JOIN A5 BY rollno, B5 BY rollno;
2022-06-09 01:53:21,065 [main] WARN  org.apache.pig.newplan.BaseOperatorPlan - Encountered Warning IMPLICIT_CAST_TO_DOUBLE 1 time(s).
grunt> DUMP C5;
```

```
o compute warning aggregation.
2022-06-09 01:58:14,012 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MapReduceLauncher - Success!
2022-06-09 01:58:14,148 [main] INFO  org.apache.pig.data.SchemaTupleBackend - Key [pig.schematuple] was not set... will not generate code.
2022-06-09 01:58:14,528 [main] INFO  org.apache.hadoop.mapreduce.lib.input.FileInputFormat - Total input files to process : 1
2022-06-09 01:58:14,529 [main] INFO  org.apache.pig.backend.hadoop.executionengine.util.MapRedUtil - Total input paths to process : 1
(101,John,7.5,101,10,MSC)
(102,ALex,10.0,102,11,MBA)
(103,Philip,6.6,103,12,MCA)
(104,Terry,8.5,104,10,MSC)
(105,Jessi,8.0,105,11,MBA)
(106,Terrence,7.5,106,12,MCA)
grunt>
```

```
o compute warning aggregation.
2022-06-09 02:02:57,496 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MapReduceLauncher - Success!
2022-06-09 02:02:57,498 [main] INFO  org.apache.pig.data.SchemaTupleBackend - Key [pig.schematuple] was not set... will not generate code.
2022-06-09 02:02:57,522 [main] INFO  org.apache.hadoop.mapreduce.lib.input.FileInputFormat - Total input files to process : 1
2022-06-09 02:02:57,523 [main] INFO  org.apache.pig.backend.hadoop.executionengine.util.MapRedUtil - Total input paths to process : 1
(101,10,MSC)
(102,11,MBA)
(103,12,MCA)
(104,10,MSC)
(105,11,MBA)
(106,12,MCA)
grunt>
```

Split : To partition a relation based on the GPAs acquired by the students.

- GPA = 8.0, place it into relation X.
- GPA is < 8.0, place it into relation Y.

```
grunt> A6 = load '/pig-data/students.txt' USING PigStorage(',') as (rollno:int, name:chararray, gpa:float);
2022-06-09 02:04:22,497 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - yarn.resourcemanager.system-metrics-publisher.enabled
is deprecated. Instead, use yarn.system-metrics-publisher.enabled
2022-06-09 02:04:22,546 [main] WARN  org.apache.pig.newplan.BaseOperatorPlan - Encountered Warning IMPLICIT_CAST_TO_DOUBLE 1 time(s).
grunt> SPLIT A6 INTO X6 IF gpa==8.0, Y6 IF gpa<8.0;
2022-06-09 02:04:27,108 [main] WARN  org.apache.pig.newplan.BaseOperatorPlan - Encountered Warning IMPLICIT_CAST_TO_DOUBLE 3 time(s).
grunt> DUMP X6;
```

```
o compute warning aggregation.
2022-06-09 02:07:49,457 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MapReduceLauncher - Success!
2022-06-09 02:07:49,463 [main] INFO  org.apache.pig.data.SchemaTupleBackend - Key [pig.schematuple] was not set... will not generate code.
2022-06-09 02:07:49,508 [main] INFO  org.apache.hadoop.mapreduce.lib.input.FileInputFormat - Total input files to process : 1
2022-06-09 02:07:49,508 [main] INFO  org.apache.pig.backend.hadoop.executionengine.util.MapRedUtil - Total input paths to process : 1
(105,Jessi,8.0)
```

Avg : To calculate the average marks for each student.

```
grunt> A7 = load '/pig-data/students.csv' USING PigStorage(',') as (studname:chararray,marks:int);
2022-06-09 02:17:49,432 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - yarn.resourcemanager.system-metrics-publisher.enabled
is deprecated. Instead, use yarn.system-metrics-publisher.enabled
2022-06-09 02:17:49,441 [main] WARN  org.apache.pig.newplan.BaseOperatorPlan - Encountered Warning IMPLICIT_CAST_TO_DOUBLE 7 time(s).
grunt> B7 = GROUP A7 BY studname;
2022-06-09 02:17:54,964 [main] WARN  org.apache.pig.newplan.BaseOperatorPlan - Encountered Warning IMPLICIT_CAST_TO_DOUBLE 7 time(s).
grunt> C7 = FOREACH B7 GENERATE A7.studname,AVG(A7.marks);
2022-06-09 02:18:05,778 [main] WARN  org.apache.pig.newplan.BaseOperatorPlan - Encountered Warning IMPLICIT_CAST_TO_DOUBLE 7 time(s).
grunt> DUMP C7;
```

```
o compute warning aggregation.
2022-06-09 02:22:25,228 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MapReduceLauncher - Success!
2022-06-09 02:22:25,236 [main] INFO  org.apache.pig.data.SchemaTupleBackend - Key [pig.schematuple] was not set... will not generate code.
2022-06-09 02:22:25,277 [main] INFO  org.apache.hadoop.mapreduce.lib.input.FileInputFormat - Total input files to process : 1
2022-06-09 02:22:25,277 [main] INFO  org.apache.pig.backend.hadoop.executionengine.util.MapRedUtil - Total input paths to process : 1
({(ALex)},40.0)
({(John)},80.0)
({(Jessi)},100.0)
({(Terry)},90.0)
({(Philip)},60.0)
({(Terrence)},65.0)
grunt>
```

Max

To calculate the maximum marks for each student.

```
grunt> A8 = load '/pig-data/students.csv' USING PigStorage(',') as (studname:chararray,marks:int);
2022-06-09 02:24:08,697 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - yarn.resourcemanager.system-metrics-publisher.enabled
is deprecated. Instead, use yarn.system-metrics-publisher.enabled
2022-06-09 02:24:08,775 [main] WARN  org.apache.pig.newplan.BaseOperatorPlan - Encountered Warning IMPLICIT_CAST_TO_DOUBLE 7 time(s).
grunt> B8 = GROUP A8 BY studname;
2022-06-09 02:24:16,942 [main] WARN  org.apache.pig.newplan.BaseOperatorPlan - Encountered Warning IMPLICIT_CAST_TO_DOUBLE 7 time(s).
grunt> C8 = FOREACH B8 GENERATE A8.studname,MAX(A8.marks);
2022-06-09 02:24:22,400 [main] WARN  org.apache.pig.newplan.BaseOperatorPlan - Encountered Warning IMPLICIT_CAST_TO_DOUBLE 7 time(s).
grunt> DUMP C8;
```

```
o compute warning aggregation.
2022-06-09 02:28:28,667 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MapReduceLauncher - Success!
2022-06-09 02:28:28,670 [main] INFO  org.apache.pig.data.SchemaTupleBackend - Key [pig.schematuple] was not set... will not generate code.
2022-06-09 02:28:28,771 [main] INFO  org.apache.hadoop.mapreduce.lib.input.FileInputFormat - Total input files to process : 1
2022-06-09 02:28:28,771 [main] INFO  org.apache.pig.backend.hadoop.executionengine.util.MapRedUtil - Total input paths to process : 1
({(ALex)},40)
({(John)},80)
({(Jessi)},100)
({(Terry)},90)
({(Philip)},60)
({(Terrence)},65)
grunt>
```

Practical No : 4

Configure the Hive and implement the application in Hive.

Create table stud

Create table stud (sid integer , sname string , roll_n integer , class string , department string)

Hive Add a name... Add a description...

```
create table stud (sid integer,sname string,roll_n integer,class string, department string )
```

default.stud

Filter...

Column (5)	Type	Description	Sample	
sid	int		NULL	NULL
sname	string		NULL	NULL
roll_n	int		NULL	NULL
class	string		NULL	NULL
department	string		NULL	NULL

Insert the values in table stud

Hive Add a name... Add a description...

```
insert into stud (sid,sname,roll_n ,class , department) VALUES
 (1,'vishal',11,'MSC','IT'),
 (2,'kunal',12,'MCA','Computer science'),
 (3,'rutuja',13,'MSC','IT'),
 (4,'avinash',14,'MSC','IT'),
 (5,'jitendra',15,'MSC','CS')
```

Execute 5000

	col_name	data_type	comment
1	sid	int	
2	sname	string	
3	roll_n	int	
4	class	string	
5	department	string	

select * from stud;

11	1	vishal	11	MSC	IT
12	2	kunal	12	MCA	Computer science
13	3	rutuja	13	MSC	IT
14	4	avinash	14	MSC	IT
15	5	jitendra	15	MSC	CS

Let's rename our table name from the stud to the stud12;

ALTER TABLE stud RENAME TO stud12;

Query History | Saved Queries | Results | Chart | Execution Analysis

	stud12.sid	stud12.sname	stud12.roll_n	stud12.class	stud12.department
9	NULL	NULL	NULL	NULL	NULL
10	NULL	NULL	NULL	NULL	NULL
11	1	vishal	11	MSC	IT
12	2	kunal	12	MCA	Computer science
13	3	rutuja	13	MSC	IT
14	4	avinash	14	MSC	IT
15	5	jitendra	15	MSC	CS

Let's add a column gender to the stud12 table that we have obtained after renaming the stud.

ALTER TABLE stud12 ADD COLUMNS(gender string);

Query History | Saved Queries | Results | Chart | Execution Analysis

	stud12.sid	stud12.sname	stud12.roll_n	stud12.class	stud12.department	stud12.gender
9	NULL	NULL	NULL	NULL	NULL	NULL
10	NULL	NULL	NULL	NULL	NULL	NULL
11	1	vishal	11	MSC	IT	NULL
12	2	kunal	12	MCA	Computer science	NULL
13	3	rutuja	13	MSC	IT	NULL
14	4	avinash	14	MSC	IT	NULL
15	5	jitendra	15	MSC	CS	NULL

Lets ALTER TABLE existing Column to new column name datatype ;

ALTER TABLE stud12 CHANGE gender DOB STRING;

Query History | Saved Queries | Results | Chart | Execution Analysis

	stud12.sid	stud12.sname	stud12.roll_n	stud12.class	stud12.department	stud12.dob
9	NULL	NULL	NULL	NULL	NULL	NULL
10	NULL	NULL	NULL	NULL	NULL	NULL
11	1	vishal	11	MSC	IT	NULL
12	2	kunal	12	MCA	Computer science	NULL
13	3	rutuja	13	MSC	IT	NULL
14	4	avinash	14	MSC	IT	NULL
15	5	jitendra	15	MSC	CS	NULL

Drop table

DROP TABLE IF EXISTS stud12;

Create new table Employee Table :-

create table emp1(eid integer , ename string , salary integer , dept string)

Hive Add a name... Add a description...

```
1 create table emp1(eid integer , ename string , salary integer , dept string)
```

Insert data into table emp1

insert into emp1(eid , ename , salary , dept) Values

(101 , 'Vishal' , 1234 , 'data anylystics'),

(102 , 'Rutuja' , 3216 , 'developer'),

(103 , 'Kunal' , 7894 , 'website manager '),

(104 , 'Shyam' , 65269 , 'android devpls'),

(105 , 'Pillu' , 21504 , 'teacher')

Hive Add a name... Add a description...

```
1 insert into emp1(eid , ename , salary , dept) Values
2 (101 , 'Vishal' , 1234 , 'data anylystics'),
3 (102 , 'Rutuja' , 3216 , 'developer'),
4 (103 , 'Kunal' , 7894 , 'website manager '),
5 (104 , 'Shyam' , 65269 , 'android devpls'),
6 (105 , 'Pillu' , 21504 , 'teacher')
```

Execute 5000

To show the values in emp1 table :

select * from emp1;

Query History | Saved Queries | Results | Chart | Execution Analysis

	emp1.eid	emp1.ename	emp1.salary	emp1.dept
1	101	Vishal	1234	data anylystics
2	102	Rutuja	3216	developer
3	103	Kunal	7894	website manager
4	104	Shyam	65269	android devpls
5	105	Pillu	21504	teacher

Using ORDER By Clause :-

SELECT eid, ename,salary,dept FROM emp1 ORDER BY dept ;

Query History | Saved Queries | Results | Chart | Execution Analysis

	eid	ename	salary	dept
1	104	Shyam	65269	android devpls
2	101	Vishal	1234	data anylystics
3	102	Rutuja	3216	developer
4	105	Pillu	21504	teacher
5	103	Kunal	7894	website manager

Using GROUP By Clause :-

SELECT dept,count(*) FROM emp1 GROUP BY dept;

Query History | Saved Queries | Results | Chart | Execution Analysis

	dept	_c1
1	android devpls	1
2	data anylystics	1
3	developer	1
4	teacher	1
5	website manager	1

Perform the JOIN operator :

Now create one more table “emp_data” and perform the join

create table cust (id integer , name string , age string , address string , salary integer);

insert the value in cust

insert into cust (id,name,age,address,salary) values

(1,'vishal',21,'mumbai',200),

(2,'rutuja',22,'vashi',600),

(3,'kunal',23,'pune',400),

(4,'noone ',24,'pen',200)

Query History | Saved Queries | Results | Chart | Exec

	cust.id	cust.name	cust.age	cust.address	cust.salary
1	1	vishal	21	mumbai	200
2	2	rutuja	22	vashi	600
3	3	kunal	23	pune	400
4	4	noone	24	pen	200

```
create table ord (oid integer , cid integer , amount integer );
insert into ord(oid , cid , amount) values
(10,2,33),
(11,3,66),
(12,1,55)
```

Query History Saved Queries Resu

	ord.oid	ord.cid	ord.amount
1	10	2	33
2	11	3	66
3	12	1	55

```
SELECT c.ID, c.NAME, c.AGE, o.AMOUNT
FROM cust c JOIN ord o
ON (c.id = o.cid);
```

Query History Saved Queries R

	c.id	c.name	c.age	o.amount
1	1	vishal	21	55
2	2	rutuja	22	33
3	3	kunal	23	66

Illustrating some built-in functions

SELECT round(2.6) FROM student;

```
21 SELECT round(2.6) FROM student1;
```

Execute 5000

Query History | Saved Queries | Results

	_c0
1	3
2	3
3	3
4	3
5	3
6	3
7	3

SELECT floor(2.6) FROM student;

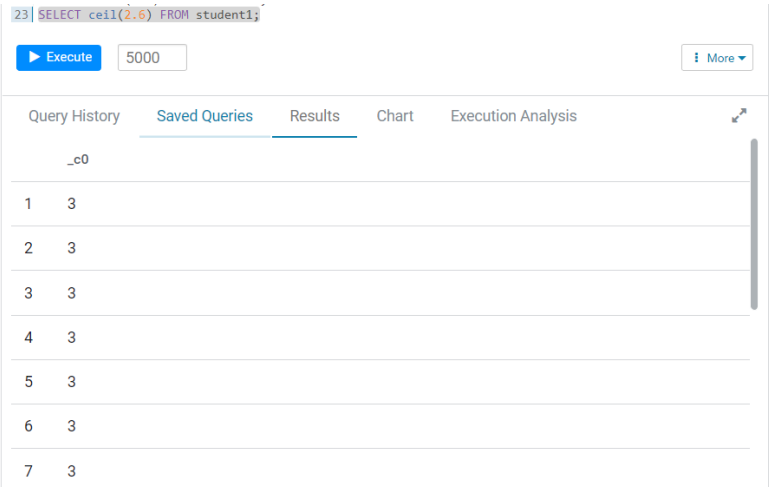
```
22 SELECT floor(2.6) FROM student1;
```

Execute 5000 More

Query History | Saved Queries | Results | Chart | Execution Analysis

	_c0
1	2
2	2
3	2
4	2
5	2
6	2
7	2

SELECT ceil(2.6) FROM student;



Practical No :5

Implement K-Mean classification techniques

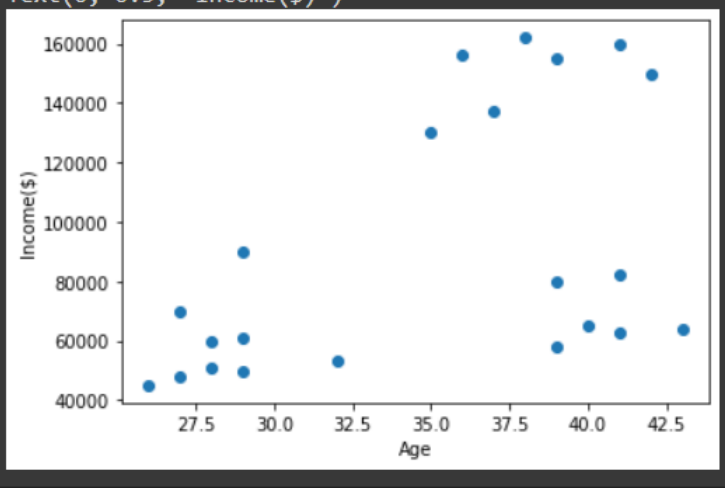
```
from sklearn.cluster import KMeans
import pandas as pd
from sklearn.preprocessing import MinMaxScaler
from matplotlib import pyplot as plt
%matplotlib inline
```

```
df = pd.read_csv("income.csv")
df.head()
```

	Name	Age	Income($)
0	Rob	27	70000
1	Michael	29	90000
2	Mohan	29	61000
3	Ismail	28	60000
4	Kory	42	150000

```
plt.scatter(df.Age,df['Income($)'])
plt.xlabel('Age')
plt.ylabel('Income($)')
```

```
Text(0, 0.5, 'Income($)')
```




```
km = KMeans(n_clusters=3)
y_predicted = km.fit_predict(df[['Age','Income($)']])
y_predicted
```

```
array([2, 2, 0, 0, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 2, 2, 0],
      dtype=int32)
```

```
df['cluster'] = y_predicted
df.head()
```

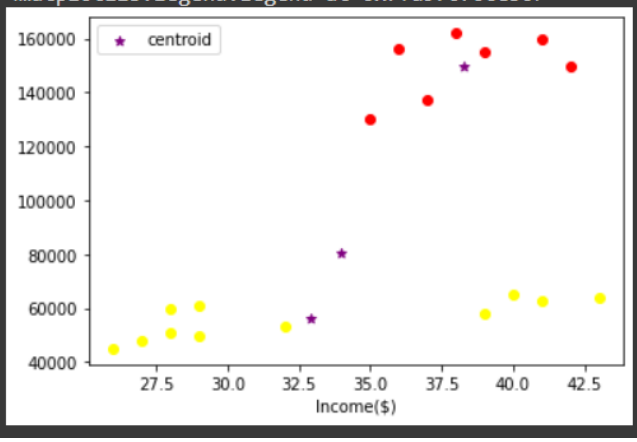
	Name	Age	Income($)	cluster
0	Rob	27	70000	2
1	Michael	29	90000	2
2	Mohan	29	61000	0
3	Ismail	28	60000	0
4	Kory	42	150000	1

```
km.cluster_centers_
```

```
array([[3.29090909e+01, 5.61363636e+04],
       [3.82857143e+01, 1.50000000e+05],
       [3.40000000e+01, 8.05000000e+04]])
```

```
df1 = df[df.cluster==0]
df2 = df[df.cluster==1]
df3 = df[df.cluster==3]
plt.scatter(df1.Age,df1['Income($)'],color='yellow')
plt.scatter(df2.Age,df2['Income($)'],color='red')
plt.scatter(df3.Age,df3['Income($)'],color='blue')
plt.scatter(km.cluster_centers_[:,0],km.cluster_centers_[:,1],color='purple',marker='*',label='centroid')
plt.xlabel('Age')
plt.xlabel('Income($)')
plt.legend()
```

```
<matplotlib.legend.Legend at 0x7fa97076ce50>
```




```
[ ] scaler = MinMaxScaler()

    scaler.fit(df[['Income($)']])
    df['Income($)'] = scaler.transform(df[['Income($)']])

    scaler.fit(df[['Age']])
    df['Age'] = scaler.transform(df[['Age']])
```

```
[ ] df.head()
```

```
[ ]
```

Practical No : 6

Perform Apriori Algorithm

```
!pip install apyori
```

```
Collecting apyori
  Downloading apyori-1.1.2.tar.gz (8.6 kB)
Building wheels for collected packages: apyori
  Building wheel for apyori (setup.py) ... done
  Created wheel for apyori: filename=apyori-1.1.2-py3-none-any.whl size=5974 sha256=4981d8d85a4df274c57a82d2b98fddd00efb4c1f33c82daaf9aab9b2129a010b
  Stored in directory: /root/.cache/pip/wheels/cb/f6/e1/57973c631d27efd1a2f375bd6a83b2a616c4021f24aab84080
Successfully built apyori
Installing collected packages: apyori
Successfully installed apyori-1.1.2
```

```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
from apyori import apriori
```

```
store_data = pd.read_csv("/content/store_data.csv")
store_data.head()
```

	shrimp	almonds	avocado	vegetables mix	green grapes	whole weat flour	yams	cottage cheese	energy drink	tomato juice	low fat yogurt	green tea	honey	salad	mineral water	salmon	antioxydant juice	frozen smoothie	spinach	olive oil
0	burgers	meatballs	eggs	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1	chutney	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2	turkey	avocado	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
3	mineral water	milk	energy bar	whole wheat rice	green tea	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
4	low fat yogurt	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

```
store_data = pd.read_csv("/content/store_data.csv",header=None)
```

```
[ ] num_records=len(store_data)
    print(num_records)
```

```
7501
```

```
[ ] records=[]
    for i in range(0,num_records):
      records.append([str(store_data.values[i,j])for j in range(0,20)])
```

```
[ ] association_rules = apriori(records, min_support=0.0053, min_confidence=0.2, min_lift=3, min_length=2)
    association_results = list(association_rules)
```

```
[ ] print(len(association_results))  #to check the Total Number of Rules mined
    print(association_results[0])  # to print the first item the association_rules list to see the first rule
```

```
32
RelationRecord(items=frozenset({'mushroom cream sauce', 'escalope'}), support=0.005732568990801226, ordered_statistics=[OrderedStatistic(items_base=frozenset({'mushroom cream sauce'}),
```

```python
results=[]
for item in association_results:
  pair = item[0]
  items = [x for x in pair]
  value0 = str(items[0])
  value1 = str(items[1])
  value2 = str(item[1])[:7]
  value3 = str(item[2][0][2])[:7]
  value4 = str(item[2][0][3])[:7]

  rows = (value0, value1,value2, value3, value4)
  results.append(rows)
  Label = ['Title1','Title2','Support','Confidence','Lift']
  store_suggestion = pd.DataFrame.from_records(results,columns=Label)
  print(store_suggestion)
```

```
                 Title1    Title2  Support Confidence     Lift
0  mushroom cream sauce  escalope  0.00573    0.30069  3.79083
                 Title1    Title2  Support Confidence     Lift
0  mushroom cream sauce  escalope  0.00573    0.30069  3.79083
1                 pasta  escalope  0.00586    0.37288  4.70081
                 Title1         Title2  Support Confidence     Lift
0  mushroom cream sauce       escalope  0.00573    0.30069  3.79083
1                 pasta       escalope  0.00586    0.37288  4.70081
2           ground beef  herb & pepper  0.01599    0.32345  3.29199
                 Title1         Title2  Support Confidence     Lift
0  mushroom cream sauce       escalope  0.00573    0.30069  3.79083
1                 pasta       escalope  0.00586    0.37288  4.70081
2           ground beef  herb & pepper  0.01599    0.32345  3.29199
3           ground beef   tomato sauce  0.00533    0.37735  3.84065
                 Title1         Title2  Support Confidence     Lift
0  mushroom cream sauce       escalope  0.00573    0.30069  3.79083
1                 pasta       escalope  0.00586    0.37288  4.70081
2           ground beef  herb & pepper  0.01599    0.32345  3.29199
3           ground beef   tomato sauce  0.00533    0.37735  3.84065
4     whole wheat pasta      olive oil  0.00799    0.27149  4.12241
                 Title1         Title2  Support Confidence     Lift
0  mushroom cream sauce       escalope  0.00573    0.30069  3.79083
1                 pasta       escalope  0.00586    0.37288  4.70081
2           ground beef  herb & pepper  0.01599    0.32345  3.29199
3           ground beef   tomato sauce  0.00533    0.37735  3.84065
4     whole wheat pasta      olive oil  0.00799    0.27149  4.12241
5                 shrimp      chocolate  0.00533    0.23255  3.25451
                 Title1         Title2  Support Confidence     Lift
0  mushroom cream sauce       escalope  0.00573    0.30069  3.79083
1                 pasta       escalope  0.00586    0.37288  4.70081
2           ground beef  herb & pepper  0.01599    0.32345  3.29199
3           ground beef   tomato sauce  0.00533    0.37735  3.84065
4     whole wheat pasta      olive oil  0.00799    0.27149  4.12241
5                 shrimp      chocolate  0.00533    0.23255  3.25451
```

Practical No : 7

Perform the Linear Regression

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn import linear_model
```

```
df = pd.read_csv("homeprices1.csv")
df.head()
```

	area	price
0	2600	550000
1	3000	565000
2	3200	610000
3	3600	660000
4	4000	725000

```
df.columns
```

```
Index(['area', 'price'], dtype='object')
```

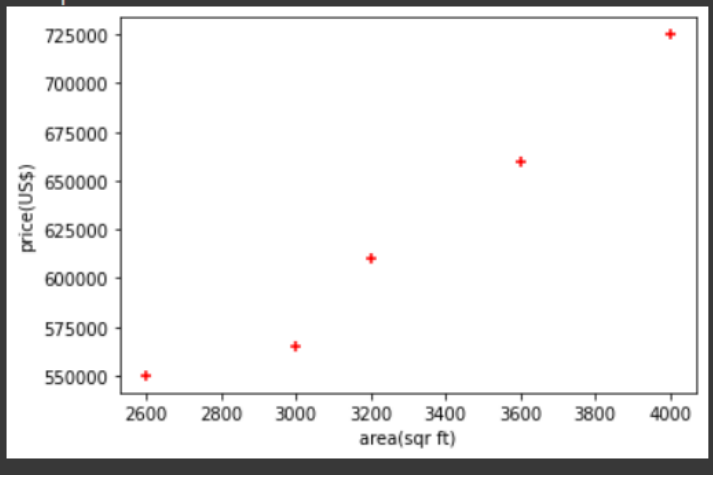
```
%matplotlib inline
```

```
df['area']
```

```
0    2600
1    3000
2    3200
3    3600
4    4000
Name: area, dtype: int64
```

```
plt.xlabel('area(sqr ft)')
plt.ylabel('price(US$)')
plt.scatter(df['area'],df['price'],color ='red',marker='+')
```

```
<matplotlib.collections.PathCollection at 0x7f16f8694d90>
```




```
reg =  linear_model.LinearRegression()
```

```
[ ] reg.fit(df[['area']],df.price)
```

```
LinearRegression()
```

```
[ ] reg.predict([[3300]])
```

```
/usr/local/lib/python3.7/dist-packages/sklearn/base.py:451: UserWarning: X does not have valid feature names, but LinearRegression was fitted with feature names
  "X does not have valid feature names, but"
array([624606.16438356])
```

```
[ ] reg.coef_
```

```
array([130.30821918])
```

```
[ ] reg.intercept_
```

```
194589.0410958904
```

```
[ ] %matplotlib
```

```
130.308219183300+194589.0410958904
```

```
194719.3493150737
```

```
reg.predict([[5000]])
```

```
/usr/local/lib/python3.7/dist-packages/sklearn/base.py:451: UserWarning: X does not have valid feature names,
  "X does not have valid feature names, but"
array([846130.1369863])
```

```python
d = pd.read_csv("area.csv")
d.head()
```

	area
0	1000
1	1500
2	2300
3	3540
4	4120

```python
p = reg.predict(d)
```

```python
pd
```

```
array([ 324897.26027397,  390051.36986301,  494297.94520548,
        655880.1369863 ,  731458.90410959,  788794.52054795,
        909981.16438356,  655880.1369863 ,  731458.90410959,
        788794.52054795,  909981.16438356,  645455.47945205,
        813553.08219178,  494297.94520548, 1367363.01369863,
       1315239.7260274 , 1119777.39726027])
```

```python
d.to_csv("prediction.csv",index=False)
```

Perform the Linear Regression

```
import pandas as pd
from matplotlib import pyplot as plt
%matplotlib inline
```

```
[ ] df = pd.read_csv("insurance_data.csv")
    df.head()
```

	age	bought_insurance
0	22	0
1	25	0
2	47	1
3	52	0
4	46	1

```
[ ] df.columns
```

```
Index(['age', 'bought_insurance'], dtype='object')
```

```
[ ] %matplotlib inline
```

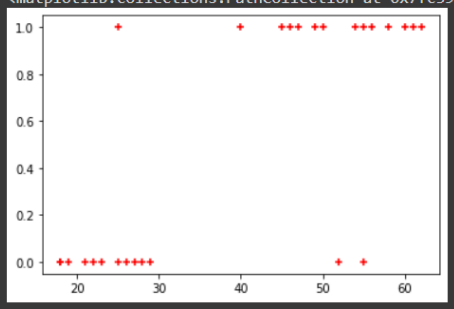


```
[ ] df['age']
```

```
0    22
1    25
2    47
3    52
4    46
5    56
6    55
7    60
8    62
9    61
10   18
11   28
12   27
13   29
14   49
15   55
16   25
17   58
18   19
19   18
20   21
21   26
22   40
23   45
24   50
25   54
26   23
Name: age, dtype: int64
```

```
[ ] plt.scatter(df['age'], df['bought_insurance'], color='red', marker='+')
```

```
<matplotlib.collections.PathCollection at 0x7fc3938171d0>
```




```
[ ] from sklearn.model_selection import train_test_split
```

```
[ ] x_train, x_test,y_train, y_test= train_test_split(df[['age']], df.bought_insurance, train_size=0.8)
```

```
[ ] x_test
```

	age
2	47
14	49
1	25
5	56
16	25
13	29

```
[ ] from sklearn.linear_model import LogisticRegression
    model = LogisticRegression()
```

```
[ ] model.fit(x_train, y_train)
```

```
LogisticRegression()
```

```
[ ] x_test
```

	age
2	47
14	49
1	25
5	56
16	25
13	29

```
[ ] y_predicted= model.predict(x_test)
```

```
[ ] model.predict_proba(x_test)
```

```
array([[0.33443244, 0.66556756],
       [0.27023439, 0.72976561],
       [0.93519109, 0.06480891],
       [0.11287215, 0.88712785],
       [0.93519109, 0.06480891],
       [0.8868389 , 0.1131611 ]])
```

```
[ ] y_predicted
```

```
array([1, 1, 0, 1, 0, 0])
```

```
x_test
```

	age
2	47
14	49
1	25
5	56
16	25
13	29

```
[ ] model.coef_
```

```
array([[0.15261423]])
```

```
[ ] model.intercept_
```

```
array([-6.48466354])
```

```
[ ] import math
    def sigmoid (x):
      return 1/(1+math.exp(-x))
```

```
[ ] def prediction_function(age):
      z=0.15 * age - 6.5
      y= sigmoid(z)
      return y
```

```
[ ] age = 35
    prediction_function(age)
```

```
    0.22270013882530884
```

```
[ ] age = 43
    prediction_function(age)
```

```
    0.48750260351578967
```

Practical No : 8

Decision Tree

```
import pandas as pd
```

```
[ ] df= pd.read_csv("salaries.csv")
    df.head()
```

	company	job	degree	salary_more_than_100k
0	google	sales executive	bacholers	0
1	google	sales executive	masters	0
2	google	business manager	bacholers	1
3	google	business manager	masters	1
4	google	computer programmer	bacholers	0

```
[ ] inputs= df.drop(['salary_more_than_100k'] , axis='columns')
    target= df['salary_more_than_100k']
```

```
[ ] inputs
```

	company	job	degree
0	google	sales executive	bacholers
1	google	sales executive	masters
2	google	business manager	bacholers
3	google	business manager	masters
4	google	computer programmer	bacholers
5	google	computer programmer	masters
6	abc pharma	sales executive	masters
7	abc pharma	computer programmer	bacholers
8	abc pharma	business manager	bacholers
9	abc pharma	business manager	masters
10	facebook	sales executive	bacholers
11	facebook	sales executive	masters
12	facebook	business manager	bacholers
13	facebook	business manager	masters
14	facebook	computer programmer	bacholers
15	facebook	computer programmer	masters

```
[ ] target
```

```
0     0
1     0
2     1
3     1
4     0
5     1
6     0
7     0
8     0
9     1
10    1
11    1
12    1
13    1
14    1
15    1
Name: salary_more_than_100k, dtype: int64
```

```
[ ] from sklearn.preprocessing import LabelEncoder
```

```
[ ] le_company=  LabelEncoder()
    le_job=  LabelEncoder()
    le_degree=  LabelEncoder()
```

```
[ ] inputs['company_n']= le_company.fit_transform(inputs['company'])
    inputs['job_n']= le_job.fit_transform(inputs['job'])
    inputs['degree_n']= le_degree.fit_transform(inputs['degree'])
```

```
inputs
```

	company	job	degree	company_n	job_n	degree_n
0	google	sales executive	bacholers	2	2	0
1	google	sales executive	masters	2	2	1
2	google	business manager	bacholers	2	0	0
3	google	business manager	masters	2	0	1
4	google	computer programmer	bacholers	2	1	0
5	google	computer programmer	masters	2	1	1
6	abc pharma	sales executive	masters	0	2	1
7	abc pharma	computer programmer	bacholers	0	1	0
8	abc pharma	business manager	bacholers	0	0	0
9	abc pharma	business manager	masters	0	0	1
10	facebook	sales executive	bacholers	1	2	0
11	facebook	sales executive	masters	1	2	1
12	facebook	business manager	bacholers	1	0	0
13	facebook	business manager	masters	1	0	1
14	facebook	computer programmer	bacholers	1	1	0
15	facebook	computer programmer	masters	1	1	1

```python
inputs_n= inputs.drop(['company', 'job', 'degree'], axis='columns')
inputs_n
```

	company_n	job_n	degree_n
0	2	2	0
1	2	2	1
2	2	0	0
3	2	0	1
4	2	1	0
5	2	1	1
6	0	2	1
7	0	1	0
8	0	0	0
9	0	0	1
10	1	2	0
11	1	2	1
12	1	0	0
13	1	0	1
14	1	1	0
15	1	1	1

```python
target
```

```
0     0
1     0
2     1
3     1
4     0
5     1
6     0
7     0
8     0
9     1
10    1
11    1
12    1
13    1
14    1
15    1
Name: salary_more_than_100k, dtype: int64
```

```python
from sklearn import tree
model= tree.DecisionTreeClassifier()
```

```python
model.fit(inputs_n, target)
```

```
DecisionTreeClassifier()
```

```
[ ] model.score(inputs_n,target)

    1.0

[ ] model.predict([[2,1,0]])

    /usr/local/lib/python3.7/dist-packages/sklearn/base.py:451: UserWarning: X does not have valid feature names, but DecisionTreeClassifier was fitted with feature names
      "X does not have valid feature names, but"
    array([0])

[ ] model.predict([[2,1,1]])

    /usr/local/lib/python3.7/dist-packages/sklearn/base.py:451: UserWarning: X does not have valid feature names, but DecisionTreeClassifier was fitted with feature names
      "X does not have valid feature names, but"
    array([1])

[ ]
```

Practical No : 9

NAIVE BAYE'S ALGORITHM

```
import pandas as pd
```

```
df = pd.read_csv("/content/titanic1.csv")
df.head()
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	S

```
df = pd.read_csv("/content/titanic1.csv")
df.head()
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	S

```
df.drop(['PassengerId', 'Name', 'SibSp', 'Parch', 'Ticket', 'Cabin', 'Embarked', ], axis='columns', inplace=True)
df.head()
```

	Survived	Pclass	Sex	Age	Fare
0	0	3	male	22.0	7.2500
1	1	1	female	38.0	71.2833
2	1	3	female	26.0	7.9250
3	1	1	female	35.0	53.1000
4	0	3	male	35.0	8.0500

```
inputs= df.drop('Survived', axis='columns')
target= df.Survived
```

```
dummies= pd.get_dummies(inputs.Sex)
dummies.head(3)
```

	female	male
0	0	1
1	1	0
2	1	0

```
inputs = pd.concat([inputs, dummies], axis='columns')
inputs.head()
```

	Pclass	Sex	Age	Fare	female	male
0	3	male	22.0	7.2500	0	1
1	1	female	38.0	71.2833	1	0
2	3	female	26.0	7.9250	1	0
3	1	female	35.0	53.1000	1	0
4	3	male	35.0	8.0500	0	1

```
[ ] inputs.drop(['Sex', 'male'], axis='columns', inplace=True)
    inputs.head(3)
```

	Pclass	Age	Fare	female
0	3	22.0	7.2500	0
1	1	38.0	71.2833	1
2	3	26.0	7.9250	1

```
[ ] inputs.columns[inputs.isna().any()]
```

```
Index(['Age'], dtype='object')
```

```
[ ] inputs.Age[:10]
```

```
0    22.0
1    38.0
2    26.0
3    35.0
4    35.0
5     NaN
6    54.0
7     2.0
8    27.0
9    14.0
Name: Age, dtype: float64
```

```
[ ] inputs.Age = inputs.Age.fillna(inputs.Age.mean())
    inputs.head(6)
```

	Pclass	Age	Fare	female
0	3	22.000000	7.2500	0
1	1	38.000000	71.2833	1
2	3	26.000000	7.9250	1
3	1	35.000000	53.1000	1
4	3	35.000000	8.0500	0
5	3	29.699118	8.4583	0

```
[ ] from sklearn.model_selection import train_test_split
    X_train, X_test, y_train, y_test = train_test_split(inputs, target, test_size=0.3)
```

```
[ ] from sklearn.naive_bayes import GaussianNB
    model= GaussianNB()
```

```
[ ] model.fit(X_train, y_train)
```

```
GaussianNB()
```

```
[ ] model.score(X_test, y_test)
```

```
0.7052238805970149
```

```
[ ] X_test[0:10]
```

	Pclass	Age	Fare	female
681	1	27.0	76.7292	0
721	3	17.0	7.0542	0
69	3	26.0	8.6625	0
813	3	6.0	31.2750	1
357	2	38.0	13.0000	1
623	3	21.0	7.8542	0
884	3	25.0	7.0500	0
534	3	30.0	8.6625	1
309	1	30.0	56.9292	1
636	3	32.0	7.9250	0

```
[ ] y_test[0:10]
```

```
681    1
721    0
69     0
813    0
357    0
623    0
884    0
534    0
309    1
636    0
Name: Survived, dtype: int64
```

```
[ ] model.predict(X_test[0:10])
```

```
array([1, 0, 0, 1, 1, 0, 0, 1, 1, 0])
```

```
[ ] model.predict_proba(X_test[0:10])
```

```
array([[0.41949954, 0.58050046],
       [0.96468028, 0.03531972],
       [0.97015121, 0.02984879],
       [0.10154324, 0.89845676],
       [0.0932169 , 0.9067831 ],
       [0.96761773, 0.03238227],
       [0.96951127, 0.03048873],
       [0.17961422, 0.82038578],
       [0.01167526, 0.98832474],
       [0.97166848, 0.02833152]])
```

```
[ ] from sklearn.model_selection import cross_val_score
    cross_val_score(GaussianNB(), X_train, y_train, cv=5)
```

```
array([0.832     , 0.808     , 0.768     , 0.80645161, 0.82258065])
```