INDEX

Name of the Laboratory :
Sub. Code :
Name of the Staff In-Charge :

Sl No.	Date of Experiment	Name of the Experiment	Page No.	Date of Sub-mission	Marks	Staff Signature with Date
7	14/08/23	Map Reduce using docker	24-26	07/09/23	10	
8.	14/08/23	HDFS Connectivity and working with HDFS	27-28	07/09/23	10	
9a	28/08/23	Wordcount in Pig	29	07/09/23	10	
9b	28/08/23	Wordcount in Hive	30	07/09/23	10	
10a	28/08/23	YARN Connectivity	31-32	07/09/23	10	
10b	04/08/23	YARN Program	33-34	07/09/23	10	
11	04/08/23	Working with Tableau	35-36	07/09/23	10	

completed on 8/9/23

Expt. No. 1 Page No. 01

Expt. Name Data Preprocessing - Building Good Datasets Date : 19/06/23

Aim :-

To implement data preprocessing using python.

Procedure :-

1. The pandas library is used to load the dataset from the CSV file into a data frame
2. Any necessary data cleaning steps can be performed. Additionally missing values in income column are handled by filling them with mean value.
3. The data set is split into features (x) and the target variable y.
4. The dataset is split into training and testing sets using the functions from the Sikit
5. Finally the shapes of the training and testing sets are printed to verify the sizes of the sets.

Expt. No. _____ Page No. 02

Expt. Name _____ Date : _____

Program :-

```
import csv
import pandas as pd
import numpy as np
from Sklearn.model_Selection import train_test_Split

data = [["Region", "Age", "Income", "Online Shopping"],
        ["India", 49, 86400, "no"],
        ["Brazil", 32, 57600, "yes"],
        ["USA", 35, 64800, "no"],
        ["India", 40, 69600, "yes"],
        ["Brazil", 43, 73200, "no"],
        ["USA", 45, 0, "yes"],
        ["Brazil", 0, 62400, "no"],
        ["India", 53, 94800, "yes"],
        ["USA", 55, 99600, "no"],
        ["India", 42, 80000, "yes"]]
f = open("Sample.csv", "w")
Writer = csv.writer(f)
Writer.writerows(f)
f.close()

data = pd.read_csv("sample.csv")
array = np.array(data)

def fillEmpty(arr, col):
    for i, j in enumerate(array):
```

Expt. No. _____ Page No. 03

Expt. Name _____ Date : _____

```
if j[col] == 0:
    array[i][col] = "NAN"

def NANToMean(arr, col):
    tot, count = 0, 0
    for i in array[:, col]:
        if i != "NAN":
            tot += i
    mean = tot // (len(array[:, col]) - list(array[:, col]).count("NAN"))
    for i, j in enumerate(array):
        if j[col] == "NAN":
            array[i][col] = mean

print(array)
fillEmpty(array, 1)
NANToMean(array, 1)

fillEmpty(array, 2)
NANToMean(array, 2)
print(array)

X = array[:, :-1]
y = array[:, -1]
X_train, X_test, Y_train, Y_test = train_test_split
(X, Y, test_size = 0.8, random_state = 1)

print
```

Output :-

```
[['India', 49, 86400, 'no']
 ['Brazil', 32, 57600, 'yes']
 ['USA', 35, 64800, 'no']
 ['Brazil', 43, 73200, 'no']
 ['USA', 45, 0, 'Yes']
 ['India', 40, 69600, 'Yes']
 ['Brazil', 0, 62400, 'no']
 ['India', 53, 94800, 'Yes']
 ['USA', 55, 99600, 'no']
 ['India', 42, 80000, 'Yes']]

[['India', 49, 86400, 'no']
 ['Brazil', 32, 57600, 'Yes']
 ['USA', 35, 64800, 'no']
 ['Brazil', 43, 73200, 'no']
 ['USA', 45, 0, 'Yes']
 ['India', 40, 69600, 'Yes']
 ['Brazil', 0, 62400, 'no']
 ['India', 53, 94800, 'Yes']
 ['USA', 55, 99600, 'no']
 ['India', 42, 80000, 'Yes']]

X train Shape : (2, 3)
X test Shape : (8, 3)
```

Expt. No. _____ Page No. 04

Expt. Name _____ Date : _____

print ("X train Shape:", X_train.shape)
print ("X train Shape:", X_test.shape)

Result :-

The data preprocessing for building good training sets is executed successfully.

Output :-

	Name	Twitter_Username	Account_Start_Date
0	A. Donald McEachin	RepMcEachin	2017-01-03T00:00:00Z
1	Aaron Michlewitz	RepMichlewitz	2010-06-27T00:00:00Z
2	Aaron Peskin	Aaronpeskin	2010-11-13T00:00:00Z
3	Aaron Peña	Aaronpena	2007-10-31T00:00:00Z
4	Aaron Schock	Aaronschock	2009-03-12T00:00:00Z

	Account_ID	Sex	Birthplace	Birthday
0	816181091673448448	male	Germany	1961-10-10T00:00:00Z
1	160249673	male	USA	1978-01-01T00:00:00Z
2	215369273	male	USA	1964-06-17T00:00:00Z
3	9843332	male	USA	1959-06-08T00:00:00Z
4	23951197	male	USA	1981-05-28T00:00:00Z

	Age	Instagram_Username	Political_party
0	59.0	repmceachin	Democratic party
1	42.0	NAN	Democratic party
2	56.0	apeskin52	Democratic party
3	61.0	NAN	Republican party
4	39.0	aaronshock	Republican party

Expt. No. 2 Page No. 05

Expt. Name Manipulation of Twitter Dataset Date : 26/06/23

Aim :

To manipulate a twitter dataset using python

Algorithm :

1. Import the required modules, dataset and load the dataset
2. Then drop a certain column from the dataset.
3. Drop a row from the dataset.
4. Then rename a column from the dataset.
5. Then group a data under a certain feature
6. Slicing the dataset to get required information
7. Describe the dataset and print it.

Program :

```
import pandas as pd
data = pd.read_csv("dataset.csv")
print(data.head())
```

	Name	Twitter_Username	Account_Start_time
0	A. Donald McEachin	RepMcEachin	2017-01-03T00:00:00Z
1	Aaron Michelwitz	Repmichelwitz	2010-06-27T00:00:00Z
2	Aaron Peskin	Aaronpeskin	2010-11-13T00:00:00Z
3	Aaron Pena	Aaronpena	2007-10-31T00:00:00Z
4	Aaron Schock	Aaron Schock	2009-03-12T00:00:00Z

	Account_ID	Sex	Birthday	Age	Instagram_Username	Political_Party
0	816181091673448448	Male	1961-10-10T00:00:00Z	59.0	repmceachin	Democratic party
1	160246973	male	1978-01-01T00:00:00Z	42.0	NAN	Democratic party
2	215369273	male	1964-06-17T00:00:00Z	56.0	apeskin52	(Democratic) party
3	9843332	male	1959-06-08T00:00:00Z	61.0	NAN	Republican party
4	23951197	male	1981-05-28T00:00:00Z	39.0	aaronschak	Republican party

Sex	Political_party
female	730
Male	1779
transgender female	4
transgender male	1

Expt. No. _____ Page No. 06

Expt. Name _____ Date : _____

```
df_dropped = data.drop("Birthplace", axis=1)
df_dropped.head()
```

```
data.groupby('Sex').agg({'Political_party':'count'})
```

```
0      repmclachin
1         NAN
2      apeskin52
3         NAN
4      aaronshock
```

Name: Instagram username, dtype: Object

```
count    2514
unique   1806
top      Ben Carson
freq     12
```

Name: Name, dtype: Object.

	Name	Twiter_Username	Account_Start_time
0	A. Donald McEachin	RepMcEachin	2017-01-03T00:00:00Z
1	Aaron Michelwitz	RepMichlewitz	2010-06-27T00:00:00Z
2	Aaron Pena	AaronPena	2007-10-31T00:00:00Z
3	Aaron Schock	aaronschock	2009-03-12T00:00:00Z
4	Abby Finkenauer	Abby4Iowa	2013-03-12T00:00:00Z

Account_ID	Sex	Birthplace	Birthday
816181091613448448	male	Germany	1961-10-10T00:00:00Z
160246973	male	USA	1978-01-01T00:00:00Z
964332	male	USA	1959-06-05T00:00:00Z
23951197	male	USA	1981-06-08T00:00:00Z
1262017122	female	USA	1988-12-27T00:00:00Z

```
data.iloc[:5,0]
desc = data["Name"].describe()
df_new = dropped = data.drop(2, axis = 0)
df_new = dropped.head()
```

Age	Instagram_Username	Political_party
59.0	repmreachin	Democratic party
42.0	NaN	Democratic party
61.0	NaN	Republican party
39.0	aaronschock	Republican party
31.0	abby4iowa	Democratic party

	Name	Twitter_username	Account_Start_time
0	A. Donald McEachin	RepMcEachin	2017-01-03T00:00:00Z
1	Aaron Michlewitz	RepMichlewitz	2010-06-27T00:00:00Z
2	Aaron Peskin	Aaronpeskin	2010-11-13T00:00:00Z
3	Aaron Pena	Aaronpena	2007-10-31T00:00:00Z
4	Aaron Schock	AaronSchock	2009-03-12T00:00:00Z

ID	Sex	Birthplace	Birthday
816181091673448448	male	Germany	1961-10-10T00:00:00Z
160246973	male	USA	1978-01-01T00:00:00Z
215369273	male	USA	1964-06-17T00:00:00Z
9843332	male	USA	1959-06-08T00:00:00Z
23951197	male	USA	1981-05-28T00:00:00Z

Age	Instagram_username	Political_Party
59.0	repmreachin	Democratic party
42.0	NaN	Democratic party
56.0	apeskin52	Democratic party
61.0	NaN	Republican party
39.0	aaronschock	Republican party

Expt. No. _____ Page No. 08

Expt. Name _____ Date . _____

```
data.columns
df_renamed = data.rename(columns = {'Account_ID': 'ID'})
df_renamed.head()
```

Result : 2/6

Thus the twitter dataset has been manipulated and result is successfully executed

Expt. No. 3 Page No. 09

Expt. Name Evaluating The Performance Of Results Of Machine Learning Algorithm Date : 26/06/23

Aim:-

To evaluate the performance of results of machine learning algorithm

Procedure:-

1. Import the required modules, dataset and load the dataset
2. Import the necessary libraries and model.
3. Import the model from Sklearn
4. The dataset is split into training and testing sets using the train_test_split() function from Sklearn
5. Use the predict() method from the model to examine our test dataset.
6. In this program, 75% of the data is allocated to the training set and 25% to the testing set.
7. Finally, the performance metrics are printed.

Program:-

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, precision_score,
recall_score, f1_score
dataset = pd.read_csv("diabetes.csv")
```

Expt. No. _____ Page No. 10

Expt. Name _____ Date : _____

```
x = dataset.iloc[:, [4, 7]].values
y = dataset.iloc[:, 8].values
xtrain, xtest, ytrain, ytest = train_test_split(x, y, test_size
    = 0.25, random_state = 0)
sc_x = StandardScaler()
xtrain = sc_x.fit_transform(xtrain)
xtest = sc_x.transform(xtest)
print(xtrain[0:10, :])
classifier = LogisticRegression(random_state = 0)
classifier.fit(xtrain, ytrain)
y_pred = classifier.predict(xtest)
accuracy = accuracy_score(ytest, y_pred)
precision = precision_score(ytest, y_pred)
recall = recall_score(ytest, y_pred)
f1 = f1_score(ytest, y_pred)
print("Accuracy:", accuracy)
print("Precision:", precision)
print("Recall:", recall)
print("F1 Score:", f1)
```

Result :

The evaluation of results of machine algorithm with diabetes dataset is executed successfully.

01

Output :-

0 20415624141743 55

	Cst_Cnt	Btl_Cnt	Sta_ID		
0	1	1	054.0	056.0	19-4903CR-HY-060-0930-05400560-0000A-3
1	1	2	054.0	056.0	19-4903CR-HY-060-0930-05400560-0008A-3
2	1	3	054.0	056.0	19-4903CR-HY-060-0930-05400560-0010A-7
3	1	4	054.0	056.0	19-4903CR-HY-060-0930-05400560-0019A-3
4	1	5	054.0	056.0	19-4903CR-HY-060-0930-05400560-0021A-7
...	...	...	...		...
864858	34404	864859	093.4	026.4	20-16115R-MX-310-2239-09340264-0000A-7
864859	34404	864860	093.4	026.4	20-16115R-MX-310-2239-09340264-0002A-3
864860	34404	864861	093.4	026.4	20-16115R-MX-310-2239-09340264-0005A-3
864861	34404	864862	093.4	026.4	20-16115R-MX-310-2239-09340264-0010A-3
864862	34404	864863	093.4	026.4	20-16115R-MX-310-2239-09340264-0015A-3

	T_deg	Salnty	RecInd	T_Prec	S_Prec	MeanAq	R_Depth	R_Temp
0	10.500	33.4400	1	2.0	3.0	9.0	0	10.50
1	10.460	33.4400	2	2.0	3.0	9.0	8	10.46
2	10.460	33.4370	7	2.0	3.0	9.0	10	10.46
3	10.450	33.4200	3	2.0	3.0	9.0	19	10.45
4	10.450	33.4210	7	2.0	3.0	9.0	20	10.45
...	...	...						...
864858	18.744	33.4083	7	2.0	3.0	9.0	0	18.74
864859	18.744	33.4083	3	2.0	3.0	9.0	2	18.74
864860	18.692	33.4150	3	2.0	3.0	9.0	5	18.69
864861	18.161	33.4062	3	2.0	3.0	9.0	10	18.16
864862	17.533	33.3880	3	2.0	3.0	9.0	15	17.53

Expt. No. 4(a) Page No. 11

Expt. Name Predicting Water Temperature based on Salinity Using Regression Date : 26/06/23

Aim :-

To predict water temperature based on Salinity using Regression.

Algorithm :-

1. Import the libraries, dataset and load the dataset.
2. Read the dataset
3. Remove the unwanted data from the dataset
4. Splitting the data into training and testing data.
5. Dropping any rows with Nan values
6. Data Scatter of predicted values Exploring our results.

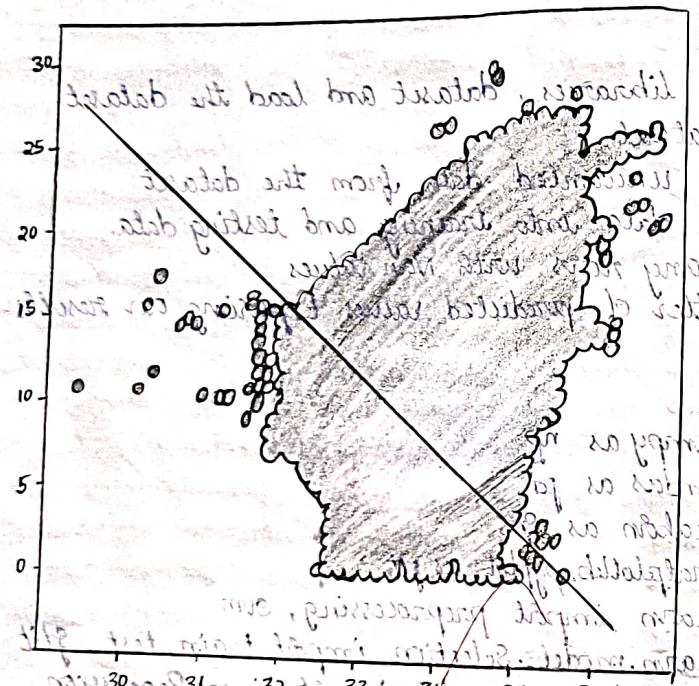
Program :-

```
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

df = pd.read_csv('bottle.csv')
df_binary = df[['Salnty', 'T_degC']]

df_binary.columns = ['Sal', 'Temp']

df_binary.head()
```



Expt. No. _____ Page No. 12

Expt. Name _____ Date : _____

```
df_binary.fillna(method='bfill',inplace=True)
X=np.array(df_binary['Sal']).reshape(-1,1)
y=np.array(df_binary['Temp']).reshape(-1,1)
df_binary.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))
print(df)
y_pred=regr.predict(X_test)
plt.scatter(X_test,y_test,color='b')
plt.plot(X_test,y_pred,color='k')
plt.show()
```

Result : Thus the water temperature can be predicted on Salinity using Regression and the result is Successfully executed.

Output :

Correlation Matrix for Iris dataset :

	Sepal length (cm)	Sepal width (cm)	Petal length (cm)	petal width (cm)
Sepal length (cm)	1.000000	-0.117570	0.871754	0.817941
Sepal width (cm)	-0.117570	1.000000	-0.428440	-0.366126
petal length (cm)	0.871754	-0.428440	1.000000	0.962865
petal width (cm)	0.817941	-0.366126	0.962865	1.000000

Correlation Matrix for boston Housing dataset :

	CRIM	ZN	INDUS	CHAS	NOX
CRIM	1.000000	-0.200469	0.406583	-0.055892	0.420972
ZN	-0.200469	1.000000	-0.533828	-0.042697	-0.516604
INDUS	0.406583	-0.533828	1.000000	0.062938	0.763651
CHAS	-0.055892	-0.042697	0.062938	1.000000	0.091203
NOX	0.420972	-0.516604	0.763651	0.091203	1.000000
RM	-0.219247	0.311991	-0.391676	0.091251	-0.302188
AGE	0.352734	-0.569537	0.644779	0.086518	0.731470
DIS	-0.379670	0.664408	-0.708027	-0.099176	-0.769230
RAD	0.625505	-0.311948	0.595129	-0.007368	0.611441
TAX	0.582764	-0.314563	0.720760	-0.035587	0.668023
PTRATIO	0.289946	-0.391679	0.383248	-0.121515	0.188933
B	-0.385064	0.175520	-0.356977	0.048788	-0.380051
LSTAT	0.455621	-0.412995	0.603800	-0.053929	0.590879

Expt. No. 4(b) Page No. 13

Expt. Name Implementation of Correlation Analysis of Date :

Iris And Boston Housing Datasets

Aim :

To perform correlation analysis on the iris dataset and the boston Housing dataset. It calculates the correlation matrix for each dataset and determines the pairwise correlation between two specific features in the boston Housing dataset.

Algorithm :-

1. Import the necessary files
2. Load the Iris dataset using function from Sklearn datasets
3. Convert the Iris data into a pandas DataFrame
4. Calculate the correlation matrix for the iris dataset.
5. Print the correlation matrix for the iris dataset
6. Load the boston Housing dataset using function from Sklearn.datasets
7. Convert the boston Housing data to a pandas DataFrame.
8. Calculate the correlation matrix for the boston Housing dataset.
9. Print the correlation matrix for the boston Housing dataset.
10. Calculate the pairwise correlation between the Selected features in the boston Housing dataset using the correlation function on the respective columns.
11. Print the calculated correlation value.

	RM	AGE	DIS	RAD	
	-0.219247	0.352734	-0.379670	0.625505	CRIM
	0.311991	-0.569537	-0.664408	-0.311948	ZN
	-0.391676	0.644779	-0.708027	0.595129	INDUS
	0.091251	0.086518	-0.099176	-0.007368	CHAS
	-0.302188	0.731470	-0.769230	0.611441	NOX
	1.000000	-0.240265	0.205246	-0.209847	RM
	-0.240265	1.000000	-0.747881	0.456022	AGE
	0.205246	-0.747881	1.000000	-0.494588	DIS
	-0.209847	0.456022	-0.494588	1.000000	RAD
	-0.292048	0.506456	-0.534432	0.910228	TAX
	-0.355501	0.261515	-0.232471	0.464741	PTRATIO
	0.128069	-0.273534	0.291512	-0.444413	B
	-0.613808	0.602339	-0.496996	0.488676	LSTAT

	TAX	PTRATIO	B	LSTAT
CRIM	0.582764	0.289946	-0.385064	0.455621
ZN	-0.314563	-0.391679	0.175520	-0.412995
INDUS	0.720760	0.383248	-0.356977	0.603800
CHAS	-0.035587	-0.121515	0.048788	-0.053929
NOX	0.668023	0.188933	-0.380051	0.590879
RM	-0.292048	-0.355501	0.128069	-0.613808
AGE	0.506456	0.261515	-0.273534	0.602339
DIS	-0.534432	-0.232471	0.291512	-0.496996
RAD	0.910228	0.464741	-0.444413	0.488676
TAX	1.000000	0.460853	-0.441808	0.543993
PTRATIO	0.460853	1.000000	-0.177383	0.374044
B	-0.441808	-0.177383	1.000000	-0.366087
LSTAT	0.543993	0.374044	-0.366087	1.000000

Expt. No. _____ Page No. 14

Expt. Name _____ Date : _____

Program :-

```
import numpy as np
import pandas as pd
from sklearn.datasets import load_iris, load_boston

iris = load_iris()
iris_data = iris.data
iris_feature_names = iris.feature_names

iris_df = pd.DataFrame(iris_data, columns=iris_feature_names)

iris_cor_matrix = iris_df.corr()

print("Correlation Matrix for iris dataset:")
print(iris_cor_matrix)
print()

boston = load_boston()
boston_data = boston.data
boston_feature_names = boston.feature_names

boston_df = pd.DataFrame(boston_data, columns=boston_
feature_names)

boston_cor_matrix = boston_df.corr()

print("Correlation Matrix for Boston Housing dataset:")
```

Expt. No. _____ Page No. 15

Expt. Name _____ Date : _____

```
print (boston_corr_matrix)
print()

Crim_corr = boston_df['CRIM'].corr(boston_df['TAX'])
print("Correlation between 'CRIM' and 'TAX' in Boston
Housing dataset:", Crim_corr)
```

Result :

Thus the correlation matrix of both iris and boston Housing Datasets are calculated using the correlation functions.

Expt. No. 5 Page No. 16

Expt. Name Classification Algorithm SVM Date :

Aim:-

To write a classification algorithm in Support vector Machine and execute it.

Procedure:-

1. Import the libraries
2. Use read_csv() function of pandas library; which is used to read a csv file and perform various operations on it.
3. Use iloc[] method of pandas library, used to extract the required rows and columns from the dataset.
4. To create the best line or decision boundary that can segregate n dimensional space into classes so that we easily put the new data point in the correct category in the future.
5. The SVM algorithm helps to find the best line or decision boundary, this best boundary or region is called as a hyperplane.
6. The dimensionals of hyperplane dependent on the features present in the dataset which means if they are 2 features then hyperplane will be a straight line.
7. If there are 3 features, then hyperplane will be a 2 dimensional plane.
8. To create the SVM classifier, import SVC class from Sklearn.svm library.
9. Predict the output for test set. for this, will create a new vector y-pred.
10. From the dataset, the SVM classifier has divided the users into two regions (purchased or not purchased)
11. Users who purchased the SVM are in red region with red scattered points. Find who did not purchase the SVM are in the green region with green scatter points.

Expt. No. _____ Page No. 17

Expt. Name _____ Date : _____

12. The hyperplane has divided the two classes into purchased and not purchased variable.

Program :-

```
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np
df = pd.read_csv("Social_Network_Ads.csv")

x = df.iloc[:, [2,3]].values
y = df.iloc[:,4:].values

from sklearn.model_selection import train_test_split

x_train, x_test, y_train, y_test = train_test_split(x,y,train_size=0.25,
random_state = 0)

from sklearn.preprocessing import svm, StandardScalers
sc = StandardScalers()

x_train = sc.fit_transform(x_train)
x_test = sc.transform(x_test)

from sklearn.svm import SVC
classifier = SVC(kernel='rbf', random_state=0)
classifier.fit(x_train, y_train)
y_pred = classifier.predict(x_test)
```

Expt. No. _____ Page No. 18

Expt. Name _____ Date : _____

```
from sklearn.metrics import confusion_matrix, accuracy_score
cm = confusion_matrix(y_test, y_pred)
print(cm)

accuracy_score(y_test, y_pred)

from matplotlib.colors import ListedColormap
x_set, y_set = x_test, y_test
x1, x2 = np.meshgrid(np.arange(start = x_set[:, 0].min() - 1,
                               stop = x_set[:, 0].max() + 1, step = 0.01),
                     np.arange(start = x_set[:, 1].min() - 1,
                               stop = x_set[:, 1].max() + 1, step = 0.01))

plt.contourf(x1, x2, classifier.predict(np.array([x1.ravel(),
x2.ravel()]).T).reshape(x1.shape), alpha = 0.75, cmap = ListedColormap
(('red', 'green')))
plt.xlim(x1.min(), x1.max())
plt.ylim(x2.min(), x2.max())
for i, j in enumerate(np.unique(y_set)):
    plt.scatter(x_set[y_set == j, 0], x_set[y_set == j, 1], c = ListedColormap
(('pink', 'green'))(i), label = j)
plt.title('SVM (Test set)')
plt.xlabel('Age')
plt.ylabel('Estimated Salary')
plt.legend()
plt.show()
```

Result: Thus, the classification algorithm of svm is executed successfully.

Output :

```
admin   0.000GB
config  0.000GB
local   0.000GB

Switched to db School

{ok: 1}

{
    "acknowledged" : true,
    "insertedId" : ObjectId("64db2084384ab13007584847c1")
}

{
    "_id" : ObjectId("64db2a1c033c78ee752bb6d7c") →
    name : Spongebob
    age : 30
    gpa : 3.2
}
```

Expt. No. 6 Page No. 19

Expt. Name NOSQL using MongoDB Date :

Aim :-

To Create, update and delete NO SQL documents using MongoDB.

Algorithm :-

1. Install MongoDB (Secured from the linked)
2. Install .msi file.
3. Install mongodb shell
4. Save it in C:
5. Set the path
   a. Goto Control panel -> System properties -> environmental variable
6. Show database using "show dbs"
7. Use database using use command
8. Create a School database
9. Insert many documents into School database
10. Display the documents in the database
11. Insert many documents on the collections
12. Sort the document in alphabetical oder, 1-alphabetical oder and -1 for reverse order.
13. Update One & many documents
14. Delete One & many documents

Program :-

```
test > Show dbs
test > Use School
School > db.Create collection ("Students")
School > db.Students.insertOne ({name: "Spongebob", age:30, gpa:3.2})
School > db.Students.find()
```

Expt. No. _____ Page No. 20

Expt. Name _____ Date : _____

School > db.Students.insertMany([{name: "patrick", age:38, gpa:1.5},
{name: "Sandy", age:27, gpa:4.0}, {name: "Gary", age:18, gpa:2.5}])

School > db.Students.find().sort({name:1})

```
{
  "acknowledged" : true,
  "insertIds" :
  [
    ObjectId("64db22a6066be78717b8de289"),
    ObjectId("64db22a6066be78717b8de28a"),
    ObjectId("64db22a6066be78717b8de28b"),
  ]
}

{
  "_id": objectId("64db23ecb69170a13e398f20c"),
  "name": "Gary", "age":18, "gpa":2.5
}
{
  "_id": objectId("64db23ecb69170a13e398f209"),
  "name": "Patrick", "age": 38, "gpa":1.5
}
{
  "_id": objectId("64db23ecb69170a13e398f20b"),
  "name": "Sandy", "age": 27, "gpa":4.0
}
{
  "_id": objectId("64db23ccb69170a13e398f206"),
  "name": "Spongbob", "age": 30, "gpa": 3.2
}
```

```
{
    "acknowledged": true,
    "matchedCount": 1,
    "modifiedCount": 1
}

{
    "_id": ObjectId("64db2abdf5bcd3678fd90f65"),
    "name": "spongebob", "age": 30, "gpa": 3.2, "fulltime": true
}
{
    _id: ObjectId("64db2ec4bfebf93643011b9c"),
    "name": "spongebob", "age": 30, "gpa": 3.2
}

{
    "acknowledged": true
    "matchedCount": 4
    "modifiedCount": 4
}

{
    "_id": ObjectId("64db309cbf47bdb732373b91"),
    "name": "spongebob", "age": 30, "gpa": 3.2, "fulltime": false
}
{
    "_id": ObjectId("64db309cbf47bdb732373b92"),
    "name": "Patrick", "age": 38, "gpa": 1.5, "fulltime": false
}
```

Expt. No. _____ Page No. 21

Expt. Name _____ Date : _____

School > db.Students.updateOne({name:"Spongebob"},{$set:{fulltime: true}})

School > db.Students.find({name: "Spongebob"})

School > db.Students.updateMany({},{$set:{fulltime: false}})

School > db.Students.find()

```
{
    "_id": ObjectId("64db309cbf47bd6732373b93"),
    "name": "Sandy", "age": 27, "gpa": 4, "fulltime": false
}
{
    "_id": ObjectId("64db309cbf47bd6732373b94"),
    "name": "Gary", "age": 18, "gpa": 2.5, "fulltime": false
}

{
    "acknowledged": true
    "deletecount": 1
}

{
    "_id": ObjectId("64db3cd18d7a5e7d2fa559f8"),
    "name": "Patrick", "age": 38, "gpa": 1.5, "fulltime": false
}
{
    "_id": ObjectId("64db3cd18d7a5e7d2fa559f9"),
    "name": "Sandy", "age": 27, "gpa": 4, "fulltime": false
}
{
    "_id": ObjectId("64db3cd18d7a5e7d2fa559fa"),
    "name": "Gary", "age": 18, "gpa": 2.5, "fulltime": false
}

{
    "acknowledged": true, "deletedcount": 3
}
```

Expt. No. _____ Page No. 22

Expt. Name _____ Date : _____

School > db.Students.deleteOne({name: "Spongebob"})

School > db.Students.find()

School > db.Students.deleteMany({fulltime: false})

Expt. No. _____ Page No. 23

Expt. Name _____ Date : _____

School >db.Students.find()

Result :-

Thus, the creation, updation and deletion commands of NO SQL documents using MongoDB is executed sucessfully.

Expt. No. 7 Page No. 24

Expt. Name MapReduce using cloudera Date : 14/08/23

Aim :-

To write a python code to print word count in mapreduce using cloudera.

Algorithm :-

1. Read lines from standard input ('stdin')
2. For each line, remove leading and trailing whitespace and split it into words.
3. For each word in the line, emit a key-value pair where the word is the key and '1' is the value.
4. This process generates a list of key-value pairs for each word in the input.
5. Now, for the "reducer.py" → Read lines from Standard input ('stdin')
6. Initialize variables 'current_word' and 'current_count' to 'None' and '0' respectively.
7. For each line, split it into a word and a count.
8. If the current word is the same as the new word, add the count to the current count.
9. If the current word changes, print previous word and its total count.
10. At the end of the input, make sure to print the last word and its total count if it exists.

Program :-

```
#!/usr/bin/python
"""mapper.py"""

import sys
for line in sys.stdin:
```

Output:

```
> gedit mapper.py
> gedit word.txt
> cat word.txt | python mapper.py
hello 1
BigData 1
operatingSystems 1
hello 1
mapreduce 1
pig 1
hive 1
BigData 1
operatingSystem 1
hello 1
hi 1
hadoop 1
hi 1
end 1
> gedit reducer.py
> cat word.txt | python mapper.py | sort | python reducer.py
BigData 2
end 1
hadoop 2
hello 3
hi 2
hive 1
mapreduce 1
```

o/p verified

Expt. No. _____ Page No. 25

Expt. Name _____ Date : _____

```
    line = line.strip()
    words = line.split()
    for word in words:
        print '%s\t%s' % (word,1)
```

```
#!/usr/bin/env python
""" reducer.py """

import sys

current_word = None
current_count = 0
word = None

#input comes from STDIN
for line in sys.stdin:
    #remove leading and trailing whitespace
    line = line.strip()

    #parse the input we got from mapper.py
    word, count = line.split('\t', 1)

    #convert count(currently a String) to int
    try:
        count = int(count)
    except ValueError:
        continue
```

operatingSystem 2
pig 1

Expt. No. _____ Page No. 26

Expt. Name _____ Date : _____

```
if current_word == word:
    current_count += count
else:
    if current_word:
        print('%s\t%s' % (current_word, current_count))
    current_count = count
    current_word = word
if current_word == word:
    print('%s\t%s' % (current_word, current_count))
```

Result: Thus the execution of word count in mapreduce using python in the cloudera is successfull verified.

Output :-

Re-format filesystem in Storage Directory root = /tmp/hadoop-root/dfs/name; location = null? (Y or N) Y

2023-09-01 04:16:59,475 INFO namenode.FSImage: Allocated new BlockPoolId: BP-150640189-172.28.0.12-1693541819454

2023-09-01 04:16:59, 475 INFO Common.Storage: will remove files: [/tmp/hadoop-root/dfs/name/current/seen_txid, /tmp/hadoop-root/dfs/name/current/fsimage_0000000000000000000, /tmp/hadoop-root/dfs/name/current/VERSION, /tmp/hadoop-root/dfs/name/current/fsimage_0000000000000000000.md5]

2023-09-01 04:16:59, 515 INFO Common.Storage: Storage directory /tmp/hadoop-root/dfs/name has been successfully formatted.

2023-09-01 04:16:59, 556 INFO namenode.FSImageFormatProtobuf: Saving image file /tmp/hadoop-root/dfs/name/current/fsimage.ckpt_0000000000000000000 using no compression

2023-09-01 04:16:59, 732 INFO namenode.FSImageFormatProtobuf: Image file /tmp/hadoop-root/dfs/name/current/fsimage.ckpt_0000000000000000000 Size 399 bytes saved in 0 seconds

2023-09-01 04:16:59, 745 INFO namenode.NNStorageRetentionManager: Going to retain 1 images with txid >=0

2023-09-01 04:16:59, 794 INFO namenode.FSNamesystem: Stopping services Started for active State

2023-09-01 04:16:59, 795 INFO namenode.FSNamesystem: Stopping services Started for Standby State

2023-09-01 04:16:59, 801 INFO namenode.FSImage: FSImageSaver clean Checkpoint: txid = 0 when meet shutdown

Expt. No. 8 Page No. 27

Expt. Name HDFS Connectivity and Working with HDFS Date : 14/08/23

Aim :-

To demonstrate the running Status of HDFS file System.

Procedure :-

1. Install JavaB if not already installed.
2. Download Hadoop 3.3.1
3. Extract the downloaded Hadoop archive.
4. Set environment variables for Java and Hadoop paths.
5. Configure HDFS by updating Core-Site.xml and hdfs-Site.xml
6. Format HDFS using hdfs namenode -format
7. Verify HDFS is running using jps command

Code :-

```
!apt-get install openjdk-8-jdk-headless -qq
!wget -q https://downloads.apache.org/hadoop/common/hadoop-3.3.1/hadoop-3.3.1.tar.gz
!tar xf "hadoop-3.3.1.tar.gz"
import os
os.environ["JAVA_HOME"] = "/usr/lib/jvm/java-8-openjdk-amd64"
os.environ['HADOOP_HOME'] = '/content/hadoop-3.3.1'
os.environ['HADOOP_CONF_DIR'] = '/content/hadoop-3.3.1/etc/hadoop'
os.environ['PATH'] = os.environ['HADOOP_HOME'] + '/bin:' +
os.environ['PATH']

# Configure HDFS
!cp /content/hadoop-3.3.1/etc/hadoop/core-site.xml /content/hadoop-3.3.1/etc/hadoop/core-site.xml.bak
```

```
2023-09-01 04:16:59,801 INFO namenode.NameNode: SHUTDOWN_MSG:
/************************************************
SHUTDOWN_MSG: Shutting down NameNode at ba73d57e1e43/172.28.0.12
************************************************/
2320 Jps
Re-format filesystem in Storage Directory root = /tmp/hadoop-root/dfs/name;
location = null ? (Y or N) N
Format aborted in Storage Directory root = /tmp/hadoop-root/dfs/name;
location = null
2023-09-01 04:38:05,514 INFO namenode.FSNamesystem: Stopping Services
Started for active State
2023-09-01 04:38:05,515 INFO namenode.FSNamesystem: Stopping Services
Started for Standby State
2023-09-01 04:38:05,517 INFO util.ExitUtil: Exiting with status 1: 0:
ExitException
2023-09-01 04:38:05,521 INFO namenode.NameNode: SHUTDOWN_MSG:
/************************************************
SHUTDOWN_MSG: Shutting down NameNode at ba73d57e1e43/172.28.0.12
************************************************/
7432 Jps
```

```
!cp /content/hadoop-3.3.1/etc/hadoop/hdfs-site.xml /content/hadoop-3.3.1/etc/hadoop/hdfs-site.xml.bak

!echo -e
'<configuration><property><name>fs.default.name</name><value>hdfs://localhost:9000</value></property></configuration>' > /content/hadoop-3.3.1/etc/hadoop/core-site.xml
!echo -e
'<configuration><property><name>dfs.replication</name><value>1</value></property></configuration>' > /content/hadoop-3.3.1/etc/hadoop/hdfs-site.xml

#Format HDFS
!hdfs namenode -format

# verify HDFS is running
!jps
```

Result:-

The demonstration of running status of HDFS file System is successfully verified.

output :-

```
>gedit wordcount.pig
>gedit textfile.txt
>pig -x local wordcount.pig
Input(s):
Successfully read records from: "/home/cloudera/Documents/textfile.txt"

Output(s):
Successfully stored records in: "file:/tmp/temp1540212539/tmp-406493
```

Job DAG:

job_local1253433066_0001

(a, 2)
(It, 1)
(in, 1)
(is, 3)
(to, 1)
(Pig, 1)
(and, 1)
(for, 1)
(fun, 1)
(pig, 1)
(This, 2)
(Word, 1)
(Pig., 1)
(test, 1)
(text, 1)

Expt. No. 9a
Page No. 29
Expt. Name Wordcount in Pig
Date : 21/08/23

Aim:-

To print the wordcount using pig in cloudera.

Algorithm:-

1. Load the input text file 'textfile.txt' into a relation called 'lines' and treat each line as a character array (chararray)
2. Tokenize each line into words using space (' ') as the delimiter and generate a relation called 'words' where each word is flattened.
3. Group the 'words' relation by the 'word' column
4. Calculate the count of words within each group and create a relation called 'wordcount' that contains the word and its corresponding count.
5. Dump the 'wordcount' relation to the console, displaying the word and its count.

Program:-

-wordcount.pig

```
lines = LOAD '/home/cloudera/Documents/textfile.txt' AS (line:chararray);
words = foreach lines GENERATE FLATTEN(TOKENIZE(line,' ')) AS word;
grouped = GROUP words BY word;
wordcount = foreach grouped GENERATE group, COUNT(words);
dump wordcount;
```

Result:-

Thus the execution of the wordcount using pig on cloudera is successful.

Output:-

word	Count
cloudera	1
Hive	1
is	1
Count	1
data	1
file	1
ffe	1
hello	1
in	1
input	1
is	1
on	1
Sample	1
testing	1
this	1
warehouse	1
World	1

Expt. No. 9b
Page No. 30
Expt. Name Wordcount in Hive
Date : 20/08/23

Aim:-

To print the wordcount using Hive.

Algorithm:-

1. Start
2. Create table files
3. Load data from input docs
4. Select word from files
5. Group and order the words
6. Stop

Program:-

```
Create TABLE FILES(LINE STRING);
LOAD DATA IN PATH 'docs' OVERWRITE INTO TABLE FILES;
CREATE TABLE WORD_COUNTS AS
SELECT word, count(1) AS Count FROM
(SELECT explode (split(line, ' ')) AS FROM FILES)
GROUP By word
ORDER By word
```

Result:-

Thus the above program has been executed Successfully.

Expt. No. 10(a)

Expt. Name YARN Connectivity

Page No. 31

Date : 28/09/23

Aim:-

To implement YARN Connectivity using python.

Algorithm:-

Step 1 : Download YARN

Step 2 : YARN Config get and Set

Step 3 : Create npm Auth Token

Program:-

Yarn Set version

Yarn Set version latest

Yarn Set version Canary

Yarn Set version classic

Yarn Set version 3.x

Yarn Set version 2.0.0-rc.10

Yarn Set version 1.22.1

YARN CONFIG GET:

Yarn Config get

Yarn Config get yarn path

Yarn Config get package Extension

Yarn Config get 'npm Scopes["my-company"].npm registor Server'

Yarn Config get npm Auth Token -no redacted

Yarn Config get package Extensions -json

YARN CONFIG SET:

Yarn Config Set

Yarn Config Set init Scope my Scope

Output :-

> latest/berry/stable → The most recent stable berry (>= 2.0.0) release

> canary → The most recent canary(release candidate) berry (>= 2.0.0) release

> classic → The most recent classic ($A_0.x||A_1.x$) release

> Yarnpath: "./scripts/yarn-2.0.0-rc.001.js"

> Success set "init-license" to "Bsn-2 clause"

> *,* Done in 0.05s

npmAuthToken: "FFFFFFFFF-FFFF-FFFF-FFFF-FFFFFFFFFFFF"

Yarn config set unsafe Http whitelist --- json '["*.example.com"]'

Yarn config set package Extensions - json '{"@babel/parser@*": {dependencies": {@babel Types": "}}}'

Yarn config set npm Scopes Company.npm Registry Server" https://npm.example.com"

Yarn config set 'npm Registers["//npm.example.com"].npm Auth Token "" "FFFFFFFF-FFFF-FFFF-----FFFFFFFFFFFFF"

Result:-

Thus the above program has been executed successfully.

Expt. No. 10(b) Page No. 33

Expt. Name YARN Program Date : 04/08/22

Aim :-

To implement yarn program using python

Algorithm :-

1. Start
2. Import required libraries in python.
3. Running the python Script
4. Exchanging data between Node and python
5. Stop.

Program :-

```
npm install python-shell
import {python Shell} from 'python_shell';
pythonshell.runstring ('x=1+1; print(x)', null, function (err){
if (err) through err;
Conside.log ('finished');
});
let pyshell = new pythonShell ('my_Script.py');
pyshell. Send ("hello");
pyshell. on ('message', function (message){
    Console.log (message);
});
pyshell. end (function (err. code. Signal){
    if (err) throw err;
Conside. log ('The exit code was :' + cache);
Conside. log ('The exit Signal was :' + Signal);
Conside. log ('finished');
});
```

Output :-

Filename : foo.py

Inside File : hello world

Expt. No. _____

Expt. Name _____

Page No. 34

Date : _____

```
pythonshell.default options = { Script path '---/Scripts' };
pythonShell.run ('script.py', null, function (era, results){
});
print ("hello wold", file = open (I, "w"))
impit { python shell, newline transformer, options } from
python_shell 'const options : options = {
    'Stdio' :
        [ 'pipe', 'pipe', 'pipe', 'pipe' ]
}
Const pyshell = new python shell ('Foo.py', options)
Const Custom pipe = pyshell. child process. stdio [3].
Custom pipe . pipe (new Newline Tnansformer ()). on ('data', Custom
Result: Buffer) => { Console. log (Custom Result to String ())
})
```

Result :-

Thus the above program has been executed Successfully.

Output :-

Marks.CSV

NAME	AGE	YEAR	MARKS
BALU	19	3RD	80
ANIL	20	2ND	37
KRISHNA	18	3RD	96
MAHESH	20	3RD	87
MADAV	15	2ND	74
SUNNY	20	3RD	98
NITISH	19	2ND	57
REDDY	20	2ND	65
SANATH	19	3RD	74
KANAN	18	2ND	95
KOUSHIK	19	2ND	78
HARI	20	2ND	84
ROHITH	19	3RD	98

Expt. No. 11

Page No. 35

Expt. Name Working with Tableau

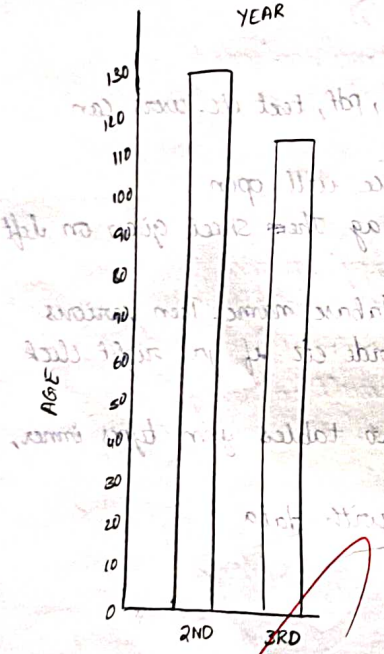
Date : 04/09/23

Aim :-

To work with Tableau and perform visualization of the dataset.

Procedure :-

1. Users can Connect to file like excel, pdf, text etc. users can Connect to Server.
2. Click on file to New, the workplace will open
3. Click on file, open, to Excel file. Drag the sheet given on left Side to "Drag Sheet".
4. After dragging we can see the database name. Then various options are there for rename, wide etc. if you right click the System.
5. We can perform joins between two tables join types inner, Full outer, left and right.
6. Goto Sheet1, to see visualization with data
   * Dimensions
   * Measures
   * Filters
   * Show Me
7. We will drag values from dimensions a measures and make a visulization.
8. We can change Sum to Average, Media.
9. We can change the size of the graph click on Size → adjust Size as per your wish.
10. Drag region field into the filter. We can filter by region wise. The options are General, wild card, condition, Top.



Expt. No. _____ Page No. 36

Expt. Name _____ Date : _____

11. Shortcut keys to play with data.
12. Tableu Dashboard Contains more than one Sheet, select Color, Size, rename as per the design we want.
13. The two Sheets are dragged into dashboard. In two bar graphs, one is for Sum of Sales other is for number of records from Sheets. 1,2,...

Result :-

Thus the data has been visualized using tableau and the required information has been acquired.