Slip1

Q. 1) Write a PHP script to keep track of number of times the web page has been accessed (Use Session Tracking).

.php file

```
<?php
   session_start();
   if (!isset($_SESSION['counter'])) {
      $_SESSION['counter'] = 1;
   } else {
      $_SESSION['counter']++; }
   echo ("Page Views: ".$_SESSION['counter']);
?>
```

Q. 2)Create 'Position_Salaries' Data set. Build a linear regression model by identifying independent and target variable. Split the variables into training and testing sets. then divide the training and testing sets into a 7:3 ratio, respectively and print them. Build a simple linear regression model.

```
import warnings
warnings.filterwarnings('ignore')
import numpy as np
import pandas as pd
advertising = pd.read_csv("D:/TY110/DA/Company_data.csv")
advertising
advertising.shape
advertising.info()
advertising.describe()
import matplotlib.pyplot as plt
import seaborn as sns
sns.pairplot(advertising, x_vars=['TV', 'Radio','Newspaper'],y_vars='Sales', size=4, aspect=1, kind='scatter')
plt.show()
X = advertising['TV']
y = advertising['Sales']
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, train_size = 0.7, test_size = 0.3, random_state = 100)
X_train
y_train
import statsmodels.api as sm
X_train_sm = sm.add_constant(X_train)
lr = sm.OLS(y_train, X_train_sm).fit()
lr.params
lr.summary()
Sales = 6.948 + 0.054 * X
plt.scatter(X_train, y_train)
plt.plot(X_train, 6.948 + 0.054*X_train, 'r')
plt.show()
y_train_pred = lr.predict(X_train_sm)
res = (y_train - y_train_pred)
fig = plt.figure()
sns.distplot(res, bins = 15)
plt.title('Error Terms', fontsize = 15)
plt.xlabel('y_train - y_train_pred', fontsize = 15)
plt.show()
X_test_sm = sm.add_constant(X_test)
y_test_pred = lr.predict(X_test_sm)
y_test_pred
from sklearn.metrics import r2_score
r_squared = r2_score(y_test, y_test_pred)
r_squared
plt.scatter(X_test, y_test)
plt.plot(X_test, y_test_pred, 'r')
plt.show()
from sklearn.model_selection import train_test_split
X_train_lm, X_test_lm, y_train_lm, y_test_lm = train_test_split(X, y, train_size = 0.7, test_size = 0.3,
random_state = 100)
X_train_lm.shape
X_train_lm = X_train_lm.values.reshape(-1,1)
X_test_lm = X_test_lm.values.reshape(-1,1)
print(X_train_lm.shape)
print(X_test_lm.shape)
from sklearn.linear_model import LinearRegression
lm = LinearRegression()
lm.fit(X_train_lm, y_train_lm)
print("Intercept :",lm.intercept_)
print('Slope :',lm.coef_)
y_train_pred = lm.predict(X_train_lm)
y_test_pred = lm.predict(X_test_lm)
```

```
print(r2_score(y_train,y_train_pred))
print(r2_score(y_test,y_test_pred))
```

slip2
Q. 1Write a PHP script to change the preferences of your web page like font style, font size, font color, background color using cookie. Display selected setting on next web page and actual implementation (with new settings) on third page (Use Cookies).
.htmlfile

```
<html><body><form action="second.php" method="get"><center>
<b>Select font style :</b><input type=text name=s1>
<b>Enter font size : </b><input type=text name=s>
<b>Enter font color :</b><input type=text name=c>
<b>Enter background color :</b><input type=text name=b>
<input type=submit value="Next">
</center></form></body></html>
```

Second.php

```
<?php
echo "style is ".$_GET['s1']."
color is ".$_GET['c']."
Background color is ".$_GET['b']."
size is ".$_GET['s'];
setcookie("set1",$_GET['s1'],time()+3600);
setcookie("set2",$_GET['c'],time()+3600);
setcookie("set3",$_GET['b'],time()+3600);
setcookie("set4",$_GET['s'],time()+3600);
?>
<a href="third.php">Show</a>
```

Third.php

```
<?php
$style=$_COOKIE['set1'];
$color=$_COOKIE['set2'];
$size=$_COOKIE['set4'];
$b_color=$_COOKIE['set3'];
$msg="Genus iT Solution";
echo "<body bgcolor=$b_color>";
echo "<font color=$color size=$size face=$style>$msg";
echo "</font></body>";
?>
```

Q. 2)Create 'Salary' Data set . Build a linear regression model by identifying independent and target variable. Split the variables into training and testing sets and print them. Build a simple linear regression model for predicting purchases.

```
import pandas as pd
import seaborn as sns
import sklearn
a=pd.read_csv("D:\CSV 1\Real estate.csv")
a
a.columns
a1=a.iloc[:,2:7]
a1
y=a.iloc[:,7:8]
y
a.info()
a.describe()
sns.pairplot(a)
from sklearn import preprocessing
a_nor=preprocessing.normalize(a)
a_nor=pd.DataFrame(a_nor)
a_nor
a_nor.columns=a.columns
a_nor
x=a_nor.iloc[:,2:7]
x
y=a_nor.iloc[:,7:8]
y
sns.pairplot(a_nor)
sns.pairplot(x)
x.corr()
a_nor.corr()
from sklearn import model_selection
from sklearn import linear_model
x_train,x_test,y_train,y_test=model_selection.train_test_split(x,y,test_size=0.2)
```

```
x_train.shape
x_test.shape
y_train.shape
y_test.shape
x_train
lm=linear_model.LinearRegression()
model=lm.fit(x_train,y_train)
model.coef_
model.intercept_
pred=lm.predict(x_train)
pred
from sklearn.metrics import r2_score
r2_score(pred,y_train)
predd=lm.predict(x_test)
r2_score(predd,y_test)
```

Slip3

Q. 1) Write a PHP script to accept username and password. If in the first three chances, username and password entered is correct then display second form with “Welcome message” otherwise display error message. [Use Session]

.php file

```
<?php
session_start();
// Set correct username and password
$correct_username = 'admin';
$correct_password = 'password';
// Check if form was submitted
if (isset($_POST['submit'])) {
 // Check if username and password are correct
 if ($_POST['username'] == $correct_username && $_POST['password'] == $correct_password) {
 // Display welcome message
 echo "Welcome, $correct_username!";
 } else {
 // Increment number of failed attempts
 if (isset($_SESSION['failed_attempts'])) {
 $_SESSION['failed_attempts']++;
 } else {
 $_SESSION['failed_attempts'] = 1;}
 // Check if user has exceeded maximum number of attempts
 if ($_SESSION['failed_attempts'] >= 3) {
 // Display error message
 echo "Error: You have exceeded the maximum number of login attempts.";
 } else {
 // Display login form again
 echo "Incorrect username or password. Please try again.<br>";
 echo "<form method='post' action=''>";
 echo "<label for='username'>Username:</label>";
 echo "<input type='text' name='username' id='username'><br>";
 echo "<label for='password'>Password:</label>";
 echo "<input type='password' name='password' id='password'><br>";
 echo "<input type='submit' name='submit' value='Submit'>";
 echo "</form>";}}
} else {
 // Display login form
 echo "<form method='post' action=''>";
 echo "<label for='username'>Username:</label>";
 echo "<input type='text' name='username' id='username'><br>";
 echo "<label for='password'>Password:</label>";
 echo "<input type='password' name='password' id='password'><br>";
 echo "<input type='submit' name='submit' value='Submit'>";
 echo "</form>";}
?>
```

Q. 2)Create ‘User’ Data set having 5 columns namely: User ID, Gender, Age, Estimated Salary and Purchased. Build a logistic regression model that can predict whether on the given parameter a person will buy a car or not.

```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
dataset = pd.read_csv(r'D:\TY110\DA\Social_Network.csv')
X = dataset.iloc[:, [2,3]].values
y = dataset.iloc[:, 4].values
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.25, random_state = 0)
from sklearn.preprocessing import StandardScaler
sc_X = StandardScaler()
```

```
X_train = sc_X.fit_transform(X_train)
X_test = sc_X.transform(X_test)
from sklearn.linear_model import LogisticRegression
classifier = LogisticRegression(random_state=0)
classifier.fit(X_train, y_train)
y_pred = classifier.predict(X_test)
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test, y_pred)
print(cm)
from matplotlib.colors import ListedColormap
X_set, y_set = X_train, y_train
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(('red', 'green'))(i), label = j)
plt.title('Logistic Regression (Training set)')
plt.xlabel('Age')
plt.ylabel('Estimated Salary')
plt.legend()
plt.show()
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(('red', 'green'))(i), label = j)
plt.title('Logistic Regression (Test set)')
plt.xlabel('Age')
plt.ylabel('Estimated Salary')
plt.legend()
plt.show()
```

slip4

Q. 1) Write a PHP script to accept Employee details (Eno, Ename, Address) on first page. On second page accept earning (Basic, DA, HRA). On third page print Employee information (Eno, Ename, Address, Basic, DA, HRA, Total) [Use Session]

.html

```
<html><body>
<form action="emp2.php" method="POST">
 Enter employee no:<input type="text" name="no"><br>
<br>
Enter name:<input type="text" name="name">
<br><br>
Enter adddress;<input type="text" name="addr">
<br>
<input type="submit" value="ok">
</form></body><html>
```

.php

```
<?php
setcookie('name',$_POST['name']);
setcookie('no',$_POST['no']);
setcookie('addr',$_POST['addr']);
?>
<html>
<form action="emp3.php" method="POST"><br>
Basic Salary:<input type="text" name="bas"><br>
HRA<input type="text" name="hra"><br>
DA:<input type="text" name="da"><br>
<input type= "submit" value="submit">
</form></body></html>
```

.php

```
<?php
echo "Salary Slip <br>";
echo "Name:".$_COOKIE['name']."<br>";
echo "Employee no.:".$_COOKIE['no']."<br>";
echo "<br>address:".$_COOKIE['addr']."<br>";
echo "Basic:".$_POST['bas']."<br>";
echo "HRA:".$_POST['hra']."<br>";
echo "DA:".$_POAT['da']."<br>";
$tot=$_POST['bas']+$_POST['da']+$_POST['hra'];
echo "<br>total salary:".$tot;
?>
```

Q. 2)Build a simple linear regression model for Fish Species Weight Prediction.

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn import linear_model
from sklearn.model_selection import train_test_split
df = pd.read_csv(r"D:\CSV 1\Fish.csv")
print('Shape of dataset= ', df.shape)
print(df.head(5))
df.rename(columns={'Length1':'VerticalLen','Length2':'DiagonalLen','Length3':'CrossLen'},inplace = True)
print(df.sample(5))
print(df.Species.value_counts())
df_sp = df.Species.value_counts()
df_sp = pd.DataFrame(df_sp)
print(df_sp.T)
X = df[['Height','Width']]
X.head()
y = df[['Weight']]
y.head(5)
X_train,X_test, y_train, y_test = train_test_split(X, y, test_size =0.2, random_state = 42)
print('X_train dimension= ', X_train.shape)
print('X_test dimension= ', X_test.shape)
print('y_train dimension= ', y_train.shape)
print('y_train dimension= ', y_test.shape)
model = linear_model.LinearRegression()
print(model.fit(X_train,y_train))
print('coef= ', model.coef_)
print('intercept= ', model.intercept_)
print('score= ', model.score(X_test,y_test))
predictedWeight = pd.DataFrame(model.predict(X_test), columns=['Predicted Weight'])
actualWeight = pd.DataFrame(y_test)
actualWeight = actualWeight.reset_index(drop=True)
df_actual_vs_predicted = pd.concat([actualWeight,predictedWeight],axis =1)
print(df_actual_vs_predicted.T)
plt.scatter(X_test['Height'], y_test, color='red', label = 'Actual Weight')
plt.scatter(X_test['Height'], model.predict(X_test), color='green', label = 'Prdicted Weight')
plt.xlabel('Height')
plt.ylabel('Weight')
plt.rcParams["figure.figsize"] = (10,6) # Custom figure size in inches
plt.title('Actual Vs Predicted Weight for Test Data')
plt.legend()
plt.show()
```

slip5

Q. 1) Create XML file named “Item.xml”with item-name, item-rate, item quantity Store the details of 5 Items of different Types

.xml file

```
<?xml version="1.0" encoding="UTF-8"?>
<items>
 <item><item-name>Item 1</item-name><item-rate>10</item-rate><item-quantity>5</item-quantity></item>
 <item><item-name>Item 2</item-name><item-rate>20</item-rate><item-quantity>10</item-quantity></item>
 <item><item-name>Item 3</item-name><item-rate>30</item-rate><item-quantity>15</item-quantity></item>
 <item><item-name>Item 4</item-name><item-rate>40</item-rate><item-quantity>20</item-quantity></item>
 <item><item-name>Item 5</item-name><item-rate>50</item-rate><item-quantity>25</item-quantity></item>
</items>
```

Q. 2)Use the iris dataset. Write a Python program to view some basic statistical details like percentile, mean, std etc. of the species of 'Iris-setosa', 'Iris-versicolor' and 'Iris-virginica'. Apply logistic regression on the dataset to identify different species (setosa, versicolor, verginica) of Iris flowers given just 4 features: sepal and petal lengths and widths.. Find the accuracy of the model.

```
import pandas as pd
import numpy as np
import os
import matplotlib.pyplot as plt
import seaborn as sns
df = pd.read_csv("D:\CSV 1\Iris.csv")
print(df.head(5))
df = df.drop(columns = ['Id'])
print(df.head(5))
```

```python
df.info()
print(df['Species'].value_counts())
print(df.isnull().sum())
plt.hist(df['SepalLengthCm'])
plt.hist(df['SepalWidthCm'])
plt.hist(df['PetalLengthCm'])
plt.hist(df['PetalWidthCm'])
print(df.corr())
from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
df['Species'] = le.fit_transform(df['Species'])
from sklearn.model_selection import train_test_split
X = df.drop(columns = ['Species'])
Y = df['Species']
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size = 0.25)
from sklearn.linear_model import LogisticRegression
model = LogisticRegression()
model.fit(X_train, Y_train)
print("Accuracy: ", model.score(X_test, Y_test) * 100)
```

slip6

Q. 1) Write PHP script to read “book.xml” file into simpleXML object. Display attributes and elements . (simple_xml_load_file() function)

Book.xml

```xml
<?xml version="1.0"?>
<ABC_Bookstore>
<Book category="Technical">
    <Book_PubYear>2012</Book_PubYear>
    <Book_Title>Let Us C</Book_Title>
    <Book_Author>Kanetkar</Book_Author>
</Book>

<Book category="Cooking">
    <Book_PubYear>2000</Book_PubYear>
    <Book_Title>Let Us Cook</Book_Title>
    <Book_Author>Sanjeev Kapoor</Book_Author>
</Book>
<Book category="Yoga">
    <Book_PubYear>2001</Book_PubYear>
    <Book_Title>Let Us Do Yoga</Book_Title>
    <Book_Author>Yoga Master Ritz</Book_Author>
</Book>
</ABC_Bookstore>
```

.php

```php
<?php
$xml=simplexml_load_file('Book.xml') or die("Cannot load");
echo "<br>";
foreach($xml->Book as $a)
{
echo "<br>".$a['category'];
echo "<br>".$a->Book_PubYear;
echo "<br>".$a->Book_Title;
echo "<br>".$a->Book_Author;
echo "<br><br>";
}
?>
```

Q. 2)Create the following dataset in python & Convert the categorical values into numeric format.Apply the apriori algorithm on the above dataset to generate the frequent itemsets and association rules. Repeat the process with different min_sup values.

```python
import pandas as pd
from mlxtend.frequent_patterns import apriori, association_rules
transactions = [['eggs', 'milk','bread'],['eggs', 'apple'],['milk', 'bread'],['apple', 'milk'],['milk', 'apple',
'bread']]
from mlxtend.preprocessing import TransactionEncoder
te=TransactionEncoder()
te_array=te.fit(transactions).transform(transactions)
df=pd.DataFrame(te_array, columns=te.columns_)
df
print("******************************")
freq_items = apriori(df, min_support = 0.5, use_colnames = True)
print(freq_items)
print("******************************")
rules = association_rules(freq_items, metric ='support', min_threshold=0.05)
rules = rules.sort_values(['support', 'confidence'], ascending =[False,False])
print(rules)
```

slip7

Q. 1) Write a PHP script to read “Movie.xml” file and print all MovieTitle and ActorName of file using DOMDocument Parser. “Movie.xml” file should contain following information with at least 5 records with values. MovieInfoMovieNo, MovieTitle, ActorName ,ReleaseYear

.php

```
<?php
$xml = new DOMDocument();
$xml->load('Movie.xml');
$movies = $xml->getElementsByTagName('MovieInfo');
foreach ($movies as $movie) {
 echo "Movie Title: " . $movie->getElementsByTagName('MovieTitle')[0]->textContent . "," ;
 echo "Actor Name: " . $movie->getElementsByTagName('ActorName')[0]->textContent . "<br>";
}
?>
```

Xml

```
<?xml version="1.0"?>
<MovieList>
 <MovieInfo>
 <MovieNo>1</MovieNo>
 <MovieTitle>The Shawshank Redemption</MovieTitle>
 <ActorName>Tim Robbins</ActorName>
 <ReleaseYear>1994</ReleaseYear>
 </MovieInfo>
 <MovieInfo>
 <MovieNo>2</MovieNo>
 <MovieTitle>The Godfather</MovieTitle>
 <ActorName>Marlon Brando</ActorName>
 <ReleaseYear>1972</ReleaseYear>
 </MovieInfo>
 <MovieInfo>
 <MovieNo>3</MovieNo>
 <MovieTitle>The Dark Knight</MovieTitle>
 <ActorName>Christian Bale</ActorName>
 <ReleaseYear>2008</ReleaseYear>
 </MovieInfo>
 <MovieInfo>
 <MovieNo>4</MovieNo>
 <MovieTitle>The Godfather: Part II</MovieTitle>
 <ActorName>Al Pacino</ActorName>
 <ReleaseYear>1974</ReleaseYear>
 </MovieInfo>
 <MovieInfo>
 <MovieNo>5</MovieNo>
 <MovieTitle>12 Angry Men</MovieTitle>
 <ActorName>Henry Fonda</ActorName>
 <ReleaseYear>1957</ReleaseYear>
 </MovieInfo>
</MovieList>
```

Q. 2)Download the Market basket dataset. Write a python program to read the dataset and display its information. Preprocess the data (drop null values etc.) Convert the categorical values into numeric format. Apply the apriori algorithm on the above dataset to generate the frequent itemsets and association rules.

```
import pandas as pd
import io
from mlxtend.frequent_patterns import apriori, association_rules
df = pd.read_csv("D:\CSV 1\Real estate.csv")
df.info()
df=df.dropna()
transactions = []
for i in range(0, len(df)):
    transactions.append([str(df.values[i,j]) for j in range(0, len(df.columns))])
from mlxtend.preprocessing import TransactionEncoder
te=TransactionEncoder()
te_array=te.fit(transactions).transform(transactions)
df=pd.DataFrame(te_array, columns=te.columns_)
df
print("******************************")
frequent_itemsets = apriori(df, min_support=0.02, use_colnames=True)
print(frequent_itemsets)
print("******************************")
rules = association_rules(frequent_itemsets, metric='support', min_threshold=0.02)
rules = rules.sort_values(['support', 'confidence'], ascending =[False,False])
print(rules)
```

slip8

Q. 1) Write a JavaScript to display message 'Exams are near, have you started preparing for?' (usealert box) and Accept any two numbers from user and display addition of two number .(Use Prompt and confirm box)

```
<!DOCTYPE html>
<html><head>
 <title>JavaScript Example</title>
</head>
<body>
 <script>
 alert('Exams are near, have you started preparing for?');
 let num1 = prompt('Enter the first number:');
 let num2 = prompt('Enter the second number:');
 num1 = Number(num1);
 num2 = Number(num2);
 let sum = num1 + num2;
 confirm(`The sum of ${num1} and ${num2} is ${sum}.`);
 </script></body></html>
```

Q. 2)Download the groceries dataset. Write a python program to read the dataset and display its information. Preprocess the data (drop null values etc.) Convert the categorical values into numeric format. Apply the apriori algorithm on the above dataset to generate the frequent itemsets and association rules.

```
import pandas as pd
import io
from mlxtend.frequent_patterns import apriori, association_rules
df = pd.read_csv("D:\CSV 1\Real estate.csv")
df.info()
df=df.dropna()
transactions = []
for i in range(0, len(df)):
    transactions.append([str(df.values[i,j]) for j in range(0, len(df.columns))])
from mlxtend.preprocessing import TransactionEncoder
te=TransactionEncoder()
te_array=te.fit(transactions).transform(transactions)
df=pd.DataFrame(te_array, columns=te.columns_)
df
print("******************************")
frequent_itemsets = apriori(df, min_support=0.02, use_colnames=True)
print(frequent_itemsets)
print("******************************")
rules = association_rules(frequent_itemsets, metric='support', min_threshold=0.02)
rules = rules.sort_values(['support', 'confidence'], ascending =[False,False])
print(rules)
```

slip9

Q. 1) Write a JavaScript function to validate username and password for a membership form

```
<html><body><script>
function validateform(){
var name=document.myform.name.value;
var password=document.myform.password.value;
if (name==null || name==""){
  alert("Name can't be blank");
  return false;
}else if(password.length<6){
  alert("Password must be at least 6 characters long.");
  return false;
  }
}
</script>  <body>
<form name="myform" method="post" action="s9_1.html" onsubmit="return validateform()" >
Name: <input type="text" name="name"><br/>
Password: <input type="password" name="password"><br/>
<input type="submit" value="register">
</form></body></html>
s9_1.html
<html><body><h1>Valid User</h1></body></html>
```

Q. 2)Create your own transactions dataset and apply the above process on your dataset.

```
import pandas as pd
from mlxtend.frequent_patterns import apriori, association_rules
transactions = [['eggs', 'milk','bread'],['eggs', 'apple'],['milk', 'bread'],['apple', 'milk'],['milk', 'apple',
'bread']]
from mlxtend.preprocessing import TransactionEncoder
te=TransactionEncoder()
te_array=te.fit(transactions).transform(transactions)
df=pd.DataFrame(te_array, columns=te.columns_)
```

```
df
print("******************************")
freq_items = apriori(df, min_support = 0.5, use_colnames = True)
print(freq_items)
print("******************************")
rules = association_rules(freq_items, metric ='support', min_threshold=0.05)
rules = rules.sort_values(['support', 'confidence'], ascending =[False,False])
print(rules)
```

slip10

Q. 1) Create a HTML fileto insert text before and after a Paragraph using jQuery. [Hint : Use before( ) and after( )]

```
<!DOCTYPE html>
<html><head>
<title>Insert text before and after paragraph using jQuery</title>
<script src=https://code.jquery.com/jquery-3.6.0.min.js></script>
</head><body>
<h1>Insert text before and after paragraph using jQuery</h1>
<p>This is a paragraph.</p>
<script>
$(document).ready(function() {
$("p").before("Text inserted before the paragraph." );
$("p").after( "Text inserted after the paragraph.");
});
</script></body></html>
```

Q. 2)Create the following dataset in python & Convert the categorical values into numeric format.Apply the apriori algorithm on the above dataset to generate the frequent itemsets and association rules. Repeat the process with different min_sup values.

```
import pandas as pd
from mlxtend.frequent_patterns import apriori, association_rules
transactions = [['eggs', 'milk','bread'],['eggs', 'apple'],['milk', 'bread'],['apple', 'milk'],['milk', 'apple',
'bread']]
from mlxtend.preprocessing import TransactionEncoder
te=TransactionEncoder()
te_array=te.fit(transactions).transform(transactions)
df=pd.DataFrame(te_array, columns=te.columns_)
print(df)
print("******************************")
freq_items = apriori(df, min_support = 0.5, use_colnames = True)
print(freq_items)
print("******************************")
rules = association_rules(freq_items, metric ='support', min_threshold=0.05)
rules = rules.sort_values(['support', 'confidence'], ascending =[False,False])
print(rules)
```

slip11

Q. 1) Write a Javascript program to accept name of student, change font color to red, font size to 18 if student name is present otherwise on clicking on empty text box display image which changes its size (Use onblur, onload, onmousehover, onmouseclick, onmouseup)

```
<!DOCTYPE html>
<html><head>
 <title>Student Name</title>
 <style>
 #studentName {
 font-size: 18px;
 }
 #myImage {
 display: none;
 width: 200px;
 }
 #myImage:hover {
 width: 300px;
 }
 </style></head>
<body>
 <h1>Student Name</h1>
 <input type="text" id="studentName" onblur="checkName()" onclick="showImage()">
 <img id="myImage" src="https://picsum.photos/200">
 <script>
 function checkName() {
 var studentName = document.getElementById('studentName');
 if (studentName.value) {
 studentName.style.color = 'red';
 } else {
 studentName.style.color = 'black';}}
 function showImage() {
 var studentName = document.getElementById('studentName');
 var myImage = document.getElementById('myImage');
 if (!studentName.value) {
 myImage.style.display = 'block';
 } else {
 myImage.style.display = 'none';}}
 </script></body></html>
```
Q. 2)Create the following dataset in python & Convert the categorical values into numeric format.Apply the apriori algorithm on the above dataset to generate the frequent itemsets and associationrules. Repeat the process with different min_sup values.
```
import pandas as pd
from mlxtend.frequent_patterns import apriori, association_rules
transactions = [['eggs', 'milk','bread'],['eggs', 'apple'],['milk', 'bread'],['apple', 'milk'],['milk', 'apple',
'bread']]
from mlxtend.preprocessing import TransactionEncoder
te=TransactionEncoder()
te_array=te.fit(transactions).transform(transactions)
df=pd.DataFrame(te_array, columns=te.columns_)
print(df)
print("******************************")
freq_items = apriori(df, min_support = 0.5, use_colnames = True)
print(freq_items)
print("******************************")
rules = association_rules(freq_items, metric ='support', min_threshold=0.05)
rules = rules.sort_values(['support', 'confidence'], ascending =[False,False])
print(rules)
```

slip12

Q. 1)Write AJAX program to read contact.dat file and print the contents of the file in a tabular format when the user clicks on print button. Contact.dat file should contain srno, name, residence number, mobile number, Address. [Enter at least 3 record in contact.dat file]

.html
```
<!DOCTYPE html>
<html><head>
 <title>Read Contact File</title>
 <style>
 table, th, td {
 border: 1px solid black;
 border-collapse: collapse;}
 th, td {
 padding: 5px;}
 </style></head><body>
 <h1>Read Contact File</h1>
 <button onclick="printContacts()">Print Contacts</button>
 <table id="contactsTable">
 <tr>
 <th>Sr. No.</th>
 <th>Name</th>
 <th>Residence Number</th>
 <th>Mobile Number</th>
 <th>Address</th>
 </tr></table><script>
 function printContacts() {
 var xhr = new XMLHttpRequest();
 xhr.onreadystatechange = function() {
 if (this.readyState == 4 && this.status == 200) {
 var contacts = JSON.parse(this.responseText);
 var table = document.getElementById('contactsTable');
 for (var i = 0; i < contacts.length; i++) {
 var row = table.insertRow(-1);
 var cell1 = row.insertCell(0);
 var cell2 = row.insertCell(1);
 var cell3 = row.insertCell(2);
 var cell4 = row.insertCell(3);
 var cell5 = row.insertCell(4);
 cell1.innerHTML = contacts[i].srno;
 cell2.innerHTML = contacts[i].name;
 cell3.innerHTML = contacts[i].residenceNumber;
 cell4.innerHTML = contacts[i].mobileNumber;
 cell5.innerHTML = contacts[i].address;
 }}};
 xhr.open('GET', 'contact.dat', true);
 xhr.send();}
 </script></body></html>
```

contact.dat
```
[
 {
 "srno": 1,
 "name": "Alice",
 "residenceNumber": "1234567890",
 "mobileNumber": "0987654321",
 "address": "123 Main St"
 },
 {
 "srno": 2,
 "name": "Bob",
 "residenceNumber": "2345678901",
 "mobileNumber": "1987654321",
 "address": "456 Elm St"
 },
 {
 "srno": 3,
 "name": "Charlie",
 "residenceNumber": "3456789012",
 "mobileNumber": "2987654321",
 "address": "789 Oak St"
 }
]
```
Q. 2)Create 'heights-and-weights' Data set . Build a linear regression model by identifying independent and target variable. Split the variables into training and testing sets and print them. Build a simple linear regression model for predicting purchases.

```python
import pandas as pd
import seaborn as sns
import sklearn
a=pd.read_csv("D:\CSV 1\Real estate.csv")
a
a.columns
a1=a.iloc[:,2:7]
a1
y=a.iloc[:,7:8]
y
a.info()
a.describe()
sns.pairplot(a)
from sklearn import preprocessing
a_nor=preprocessing.normalize(a)
a_nor=pd.DataFrame(a_nor)
a_nor
a_nor.columns=a.columns
a_nor
x=a_nor.iloc[:,2:7]
x
y=a_nor.iloc[:,7:8]
y
sns.pairplot(a_nor)
sns.pairplot(x)
x.corr()
a_nor.corr()
from sklearn import model_selection
from sklearn import linear_model
x_train,x_test,y_train,y_test=model_selection.train_test_split(x,y,test_size=0.2)
x_train.shape
x_test.shape
y_train.shape
y_test.shape
x_train
lm=linear_model.LinearRegression()
model=lm.fit(x_train,y_train)
model.coef_
model.intercept_
pred=lm.predict(x_train)
pred
from sklearn.metrics import r2_score
r2_score(pred,y_train)
predd=lm.predict(x_test)
r2_score(predd,y_test)
```

slip13

Q. 1) Write AJAX program where the user is requested to write his or her name in a text box, and the server keeps sending back responses while the user is typing. If the user name is not entered then the message displayed will be, “Stranger, please tell me your name!”. If the name is Rohit, Virat, Dhoni, Ashwin or Harbhajan , the server responds with “Hello, master !”. If the name is anything else, the message will be “, I don’t know you!”

.html

```
<!DOCTYPE html>
<html><head><title>AJAX Program</title>
<script src=https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js></script>
</head><body>
<label for="name">Enter your name:</label>
<input type="text" id="name" name="name">
<div id="response"></div>
<script src="n.js"></script>
</body></html>
```

.php

```
<?php
// Get the name entered by the user
$name = $_POST['name'];
// Check if the name is empty
if (empty($name)) {
echo 'Stranger, please tell me your name!';
}
// Check if the name is one of the master names
else if ($name == 'Rohit' || $name == 'Virat' || $name == 'Dhoni' || $name == 'Ashwin' || $name =='Harbhajan') {
echo 'Hello, master!';
}
// Otherwise, the server doesn’t know the user
else {
echo $name . ', I don\'t know you!';
}
```

.js

```
$(document).ready(function() {
    // Attach an event listener to the name input field
    $('#name').on('input', function() {
    // Get the name entered by the user
    var name = $(this).val();
    // Send an AJAX request to the server
    $.ajax({
    url: 'n.php',
    type: 'POST',
    data: { name: name },
    success: function(response) {
    // Update the response div with the server’s response
    $('#response').html(response);
    }});});});
```

Q. 2)Download nursery dataset from UCI. Build a linear regression model by identifying independent and target variable. Split the variables into training and testing sets and print them. Build a simple linear regression model for predicting purchases.

```
import pandas as pd
import seaborn as sns
import sklearn
a=pd.read_csv("D:\CSV 1\Real estate.csv")
a
a.columns
a1=a.iloc[:,2:7]
a1
y=a.iloc[:,7:8]
y
a.info()
a.describe()
sns.pairplot(a)
from sklearn import preprocessing
a_nor=preprocessing.normalize(a)
a_nor=pd.DataFrame(a_nor)
a_nor
a_nor.columns=a.columns
a_nor
x=a_nor.iloc[:,2:7]
x
y=a_nor.iloc[:,7:8]
y
sns.pairplot(a_nor)
sns.pairplot(x)
```

```
x.corr()
a_nor.corr()
from sklearn import model_selection
from sklearn import linear_model
x_train,x_test,y_train,y_test=model_selection.train_test_split(x,y,test_size=0.2)
x_train.shape
x_test.shape
y_train.shape
y_test.shape
x_train
lm=linear_model.LinearRegression()
model=lm.fit(x_train,y_train)
model.coef_
model.intercept_
pred=lm.predict(x_train)
pred
from sklearn.metrics import r2_score
r2_score(pred,y_train)
predd=lm.predict(x_test)
r2_score(predd,y_test)
```

slip14
Q. 1) Create TEACHER table as follows TEACHER(tno, tname, qualification, salary). Write Ajax program to select a teachers name and print the selected teachers details

```
<!DOCTYPE html>
<html><head>
 <title>AJAX Teacher Details</title>
</head><body>
 <h1>AJAX Teacher Details</h1>
 <label for="teacherSelect">Select a teacher:</label>
 <select id="teacherSelect" onchange="getTeacherDetails()">
 <option value="">--Select a teacher--</option>
 <option value="1">Alice</option>
 <option value="2">Bob</option>
 <option value="3">Charlie</option>
 </select>
 <div id="teacherDetails"></div>
 <script>
 function getTeacherDetails() {
 var tno = document.getElementById('teacherSelect').value;
 if (tno) {
 var xhr = new XMLHttpRequest();
 xhr.onreadystatechange = function() {
 if (this.readyState == 4 && this.status == 200) {
 document.getElementById('teacherDetails').innerHTML = this.responseText;}};
 xhr.open('GET', 'getTeacherDetails.php?tno=' + tno, true);
 xhr.send();
 } else {
 document.getElementById('teacherDetails').innerHTML = ''; } }
 </script></body></html>
```

.php
```
<?php
$tno = $_GET['tno'];
// Connect to the database
$db = new mysqli('hostname', 'username', 'password', 'database_name');
// Query the database for the details of the selected teacher
$query = "SELECT * FROM TEACHER WHERE tno = $tno";
$result = $db->query($query);
$row = $result->fetch_assoc();
// Send back the teacher details as an HTML table
echo '<table>';
echo '<tr><th>TNo</th><td>' . $row['tno'] . '</td></tr>';
echo '<tr><th>Name</th><td>' . $row['tname'] . '</td></tr>';
echo '<tr><th>Qualification</th><td>' . $row['qualification'] . '</td></tr>';
echo '<tr><th>Salary</th><td>' . $row['salary'] . '</td></tr>';
echo '</table>';
$db->close();
?>
```

```
sql
CREATE TABLE TEACHER (tno INT PRIMARY KEY,tname VARCHAR(255),qualification VARCHAR(255),salary INT);
-- Insert some example data
```

```
INSERT INTO TEACHER (tno, tname, qualification, salary) VALUES (1, 'Alice', 'PhD', 80000),(2, 'Bob', 'MSc',
60000),(3, 'Charlie', 'MA', 50000);
```

Q. 2)Create the following dataset in python & Convert the categorical values into numeric format.Apply the apriori algorithm on the above dataset to generate the frequent itemsets and association rules. Repeat the process with different min_sup values.

```
import pandas as pd
from mlxtend.frequent_patterns import apriori, association_rules
transactions = [['eggs', 'milk','bread'],['eggs', 'apple'],['milk', 'bread'],['apple', 'milk'],['milk', 'apple',
'bread']]
from mlxtend.preprocessing import TransactionEncoder
te=TransactionEncoder()
te_array=te.fit(transactions).transform(transactions)
df=pd.DataFrame(te_array, columns=te.columns_)
print(df)
print("******************************")
freq_items = apriori(df, min_support = 0.5, use_colnames = True)
print(freq_items)
print("******************************")
rules = association_rules(freq_items, metric ='support', min_threshold=0.05)
rules = rules.sort_values(['support', 'confidence'], ascending =[False,False])
print(rules)
```

slip15

Q. 1) Write Ajax program to fetch suggestions when is user is typing in a textbox. (eg like google suggestions. Hint create array of suggestions and matching string will be displayed)

```
<!DOCTYPE html>
<html><head><title>AJAX Auto Suggestions Example</title>
<script>
function fetchSuggestions(str) {
if (str.length == 0) {
document.getElementById("suggestions").innerHTML = "";
return;}
var suggestions = ["apple", "banana", "cherry", "dates", "elderberry", "fig","grape", "honeydew", "kiwi", "lemon"];
var matches = [];
for (var i = 0; i < suggestions.length; i++) {
if (suggestions[i].toLowerCase().startsWith(str.toLowerCase())) {
matches.push(suggestions[i]);}}
if (matches.length > 0) {
document.getElementById("suggestions").innerHTML = matches.join("<br>");
} else {
document.getElementById("suggestions").innerHTML = "No suggestions found";}}
</script></head>
<body>
<input type="text" onkeyup="fetchSuggestions(this.value)">
<div id="suggestions"></div>
</body></html>
```

Q. 2)Create the following dataset in python & Convert the categorical values into numeric format.Apply the apriori algorithm on the above dataset to generate the frequent itemsets and association rules. Repeat the process with different min_sup values.

```
import pandas as pd
from mlxtend.frequent_patterns import apriori, association_rules
transactions = [['eggs', 'milk','bread'],['eggs', 'apple'],['milk', 'bread'],['apple', 'milk'],['milk', 'apple',
'bread']]
from mlxtend.preprocessing import TransactionEncoder
te=TransactionEncoder()
te_array=te.fit(transactions).transform(transactions)
df=pd.DataFrame(te_array, columns=te.columns_)
print(df)
print("******************************")
freq_items = apriori(df, min_support = 0.5, use_colnames = True)
print(freq_items)
print("******************************")
rules = association_rules(freq_items, metric ='support', min_threshold=0.05)
rules = rules.sort_values(['support', 'confidence'], ascending =[False,False])
print(rules)
```

slip16

Q. 1) Write Ajax program to get book details from XML file when user select a book name. Create XML file for storing details of book(title, author, year, price).

Book1.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<catalog>
 <book>
 <title>Book 1</title><author>Author 1</author><year>2001</year><price>10.00</price>
 </book>
 <book>
 <title>Book 2</title><author>Author 2</author><year>2002</year><price>20.00</price>
 </book>
 <book>
<title>Book 3</title><author>Author 3</author><year>2003</year><price>30.00</price>
 </book>
</catalog>
```

.html

```
<!DOCTYPE html>
<html><head><title>Ajax Book Details</title></head>
<body>
 <h1>Ajax Book Details</h1>
 <label for="bookSelect">Select a book:</label>
 <select id="bookSelect" onchange="showBookDetails(this.value)">
 <option value="">Select a book:</option>
 <option value="Book 1">Book 1</option>
 <option value="Book 2">Book 2</option>
 <option value="Book 3">Book 3</option>
 </select>
 <div id="bookDetails"></div>
 <script>
 function showBookDetails(bookTitle) {
 if (bookTitle === "") {
 document.getElementById("bookDetails").innerHTML = "";
 return;}
 let xhttp = new XMLHttpRequest();
 xhttp.onreadystatechange = function() {
 if (this.readyState === 4 && this.status === 200) {
 let xmlDoc = this.responseXML;
 let books = xmlDoc.getElementsByTagName("book");
 for (let i = 0; i < books.length; i++) {
 let title = books[i].getElementsByTagName("title")[0].childNodes[0].nodeValue;
 if (title === bookTitle) {
 let author = books[i].getElementsByTagName("author")[0].childNodes[0].nodeValue;
 let year = books[i].getElementsByTagName("year")[0].childNodes[0].nodeValue;
 let price = books[i].getElementsByTagName("price")[0].childNodes[0].nodeValue;
 document.getElementById("bookDetails").innerHTML = `
 Title: ${title}<br />
 Author: ${author}<br />
 Year: ${year}<br />
 Price: ${price}
 `;
 break;}}}};
 xhttp.open("GET", "book1.xml", true);
 xhttp.send();}
</script></body></html>
```

Q. 2)Consider any text paragraph. Preprocess the text to remove any special characters and digits. Generate the summary using extractive summarization process

```
import nltk
import re
text="""Hello all, Welcome to Python * Programming Academy 3. Python Programming 3 Academy is a nice platform to
learn new programming skills. It is difficult to get enrolled in this Academy."""
text = re.sub(r'[[0-9]{}*]', ' ', text)
print(text)
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize, sent_tokenize
stopWords = set(stopwords.words("english"))
words = word_tokenize(text)
wordfreq = {}
for word in words:
    if word in stopWords:
        if word in wordfreq:
            wordfreq[word] += 1
        else:
            wordfreq[word] = 1
maximum_frequency = max(wordfreq.values())
for word in wordfreq.keys():
    wordfreq[word] = (wordfreq[word]/maximum_frequency)
    wordfreq
sentences = sent_tokenize(text)
sentenceValue = {}
```

```python
for sentence in sentences:
    for word, freq in wordfreq.items():
        if word in sentence.lower():
            if sentence in sentenceValue:
                sentenceValue[sentence] += freq
            else:
                sentenceValue[sentence] = freq
                sentenceValue
import heapq
summary = ''
summary_sentences = heapq.nlargest(4, sentenceValue, key=sentenceValue.get)
summary = ' '.join(summary_sentences)
print(summary)
```

slip17

Q. 1) Write a Java Script Program to show Hello Good Morning message onload event using alert box and display the Student registration from.

```html
<!DOCTYPE html>
<html><head> <title>Student Registration</title></head>
<body onload="showGreeting()">
 <h1>Student Registration</h1>
 <form>
 <label for="firstName">First Name:</label>
 <input type="text" id="firstName" name="firstName"><br><br>
 <label for="lastName">Last Name:</label>
 <input type="text" id="lastName" name="lastName"><br><br>
 <label for="email">Email:</label>
 <input type="email" id="email" name="email"><br><br>
 <input type="submit" value="Submit">
 </form>
 <script>
 function showGreeting() {
 alert("Hello Good Morning!");
 }
</script></body></html>
```

Q. 2)Consider text paragraph.So, keep working. Keep striving. Never give up. Fall down seven times, get up eight. Ease is a greater threat to progress than hardship. Ease is a greater threat to progress than hardship. So, keep moving, keep growing, keep learning. See you at work.Preprocess the text to remove any special characters and digits. Generate the summary using extractive summarization process.

```python
import nltk
import re
text="""Hello all, Welcome to Python * Programming Academy 3. Python Programming 3 Academy is a nice platform to
learn new programming skills. It is difficult to get enrolled in this Academy."""
text = re.sub(r'[[0-9]{}*]', ' ', text)
print(text)
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize, sent_tokenize
stopWords = set(stopwords.words("english"))
words = word_tokenize(text)
wordfreq = {}
for word in words:
    if word in stopWords:
        if word in wordfreq:
            wordfreq[word] += 1
        else:
            wordfreq[word] = 1
maximum_frequency = max(wordfreq.values())
for word in wordfreq.keys():
    wordfreq[word] = (wordfreq[word]/maximum_frequency)
    wordfreq
sentences = sent_tokenize(text)
sentenceValue = {}
for sentence in sentences:
    for word, freq in wordfreq.items():
        if word in sentence.lower():
            if sentence in sentenceValue:
                sentenceValue[sentence] += freq
            else:
                sentenceValue[sentence] = freq
                sentenceValue
import heapq
summary = ''
summary_sentences = heapq.nlargest(4, sentenceValue, key=sentenceValue.get)
summary = ' '.join(summary_sentences)
print(summary)
```

slip18
Q. 1) Write a Java Script Program to print Fibonacci numbers on onclick event.

```
<!DOCTYPE html>
<html><head> <title>Fibonacci Sequence Generator</title></head>
<body>
 <h1>Fibonacci Sequence Generator</h1>
 <button onclick="generateFibonacci()">Generate Fibonacci Sequence</button>
 <p id="output"></p>
 <script>
 function generateFibonacci() {
 let n1 = 0, n2 = 1, nextTerm;
 let output = 'Fibonacci Sequence: ';
 for (let i = 1; i <= 10; i++) {
 output += n1 + ', ';
 nextTerm = n1 + n2;
 n1 = n2;
 n2 = nextTerm;}
 document.getElementById('output').innerHTML = output;}
 </script></body></html>
```

Q. 2)Consider any text paragraph. Remove the stopwords. Tokenize the paragraph to extract words and sentences. Calculate the word frequency distribution and plot the frequencies. Plot the wordcloud of the text.

```
import nltk
nltk.download('all')
text="""Hello all, Welcome to Python Programming Academy. Python Programming Academy is a nice platform to learn
new programming skills. It is difficult to get enrolled in this Academy."""
from nltk.tokenize import word_tokenize
from nltk.corpus import stopwords
tokenized_words=word_tokenize(text)
tokenized_words
stop_words_data=set(stopwords.words("english"))
stop_words_data
filtered_words_list=[]
for words in tokenized_words:
     if words not in stop_words_data:
            filtered_words_list.append(words)
print("Tokenized Words : \n",tokenized_words,"\n")
print("Filtered Words : \n",filtered_words_list,"\n")
from nltk.probability import FreqDist
frequency_distribution=FreqDist(tokenized_words)
print(frequency_distribution)
print(frequency_distribution.most_common(2))
import matplotlib.pyplot as plt
frequency_distribution.plot(32,cumulative=False)
from wordcloud import WordCloud
wordcloud = WordCloud(width = 800, height = 800,
                background_color ='white',
                stopwords = stopwords,
                min_font_size = 10).generate(tokenized_words)

# plot the WordCloud image
plt.figure(figsize = (8, 8), facecolor = None)
plt.imshow(wordcloud)
plt.axis("off")
plt.tight_layout(pad = 0)
plt.show()
```

slip19
Q. 1) Write a Java Script Program to validate user name and password on onSubmit event.

```
<!DOCTYPE html>
<html><head> <title>Form Validation</title></head>
<body>
 <h1>Form Validation</h1>
 <form onsubmit="return validateForm()">
 <label for="username">Username:</label>
 <input type="text" id="username" name="username"><br><br>
 <label for="password">Password:</label>
 <input type="password" id="password" name="password"><br><br>
 <input type="submit" value="Submit">
 </form>
 <script>
 function validateForm() {
 let username = document.getElementById("username").value;
 let password = document.getElementById("password").value;
 if (username === "") {
 alert("Username must be filled out");
 return false;}
 if (password === "") {
 alert("Password must be filled out");
 return false;}
 if (password.length < 8) {
 alert("Password must be at least 8 characters long");
 return false;}
 return true;}
 </script></body></html>
```

Q. 2)Download the movie_review.csv dataset from Kaggle by using the following link :https://www.kaggle.com/nltkdata/movie-review/version/3?select=movie_review.csv to perform sentiment analysis on above dataset and create a wordcloud.

```
import pandas as pd
from nltk.corpus import stopwords
from nltk.sentiment import SentimentIntensityAnalyzer
from wordcloud import WordCloud
import matplotlib.pyplot as plt
# Load the dataset
df = pd.read_csv("D:\CSV 1\movie_review.csv")
# Initialize the sentiment analyzer
sia = SentimentIntensityAnalyzer()
# Calculate the sentiment scores for each review
df['sentiment'] = df['text'].apply(lambda x: sia.polarity_scores(x)['compound'])
# Generate the word cloud
stop_words = set(stopwords.words('english'))
wordcloud = WordCloud(width=800, height=800, background_color='white',
stopwords=stop_words,min_font_size=10).generate(' '.join(df['text']))
# Plot the word cloud
plt.figure(figsize=(8, 8), facecolor=None)
plt.imshow(wordcloud)
plt.axis('off')
plt.tight_layout(pad=0)
plt.show()
```

slip20

Q. 1) create a student.xml file containing at least 5 student information

```
<?xml version="1.0" encoding="UTF-8"?>
<students>
 <student><name>John Doe</name><age>20</age><major>Computer Science</major></student>
 <student><name>Jane Smith</name><age>21</age><major>Mathematics</major></student>
 <student><name>Bob Johnson</name><age>22</age><major>Physics</major></student>
 <student><name>Alice Williams</name><age>19</age><major>Biology</major></student>
 <student><name>Charlie Brown</name><age>23</age><major>Chemistry</major></student>
</students>
```

Q. 2)Consider text paragraph."""Hello all, Welcome to Python Programming Academy. Python Programming Academy is a nice platform to learn new programming skills. It is difficult to get enrolled in this Academy."""Remove the stopwords.

```
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
text = """Hello all, Welcome to Python Programming Academy. Python Programming Academy is a
nice platform to learn new programming skills. It is difficult to get enrolled in this Academy."""
# Tokenize the text into words
words = word_tokenize(text)
# Remove stopwords
stop_words = set(stopwords.words('english'))
filtered_words = [word for word in words if word not in stop_words]
# Join the filtered words back into a string
filtered_text = ' '.join(filtered_words)
print(filtered_text)
```

slip21

Q. 1)Add a JavaScript File in Codeigniter. The Javascript code should check whether a number is positive or negative.

```
<!DOCTYPE html>
<html><head><title>Number Check</title>
 <script src="<?php echo base_url('js/checkNumber.js'); ?>"></script>
</head>
 <body>
 <h1>Number Check</h1>
 <p>Enter a number to check:</p>
 <input type="number" id="num" />
 <button onclick="checkNumber(document.getElementById('num').value)">Check</button>
 </body></html>

//html
<!DOCTYPE html>
<html><head>
 <title>Check Number</title>
 <script src="<?php echo base_url('assets/js/checkNumber.js'); ?>"></script>
</head><body>
    <input type="number" id="numberInput">
    <button onclick="checkNumber(document.getElementById('numberInput').value)">Check
    Number</button></body></html>
```

```
//checkNumber.js
function checkNumber(num) {
    If (num > 0) {
    Alert("The number is positive.");
    } else if (num < 0) {
    Alert("The number is negative.");
    } else {
    Alert("The number is zero.");}}
   //checkNumber.js
function checkNumber(num) {
    if (num > 0) {
    console.log(num + " is positive");
    } else if (num < 0) {
    console.log(num + " is negative");
    } else {
    console.log(num + " is zero");}}
```

Q. 2)Build a simple linear regression model for User Data.

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn import linear_model
from sklearn.model_selection import train_test_split
df = pd.read_csv(r"D:\CSV 1\Fish.csv")
print('Shape of dataset= ', df.shape)
print(df.head(5))
df.rename(columns={'Length1':'VerticalLen','Length2':'DiagonalLen','Length3':'CrossLen'},inplace = True)
print(df.sample(5))
print(df.Species.value_counts())
df_sp = df.Species.value_counts()
df_sp = pd.DataFrame(df_sp)
print(df_sp.T)
X = df[['Height','Width']]
X.head()
y = df[['Weight']]
y.head(5)
X_train,X_test, y_train, y_test = train_test_split(X, y, test_size =0.2, random_state = 42)
print('X_train dimension= ', X_train.shape)
print('X_test dimension= ', X_test.shape)
print('y_train dimension= ', y_train.shape)
print('y_train dimension= ', y_test.shape)
model = linear_model.LinearRegression()
print(model.fit(X_train,y_train))
print('coef= ', model.coef_)
print('intercept= ', model.intercept_)
print('score= ', model.score(X_test,y_test))
predictedWeight = pd.DataFrame(model.predict(X_test), columns=['Predicted Weight'])
actualWeight = pd.DataFrame(y_test)
actualWeight = actualWeight.reset_index(drop=True)
df_actual_vs_predicted = pd.concat([actualWeight,predictedWeight],axis =1)
print(df_actual_vs_predicted.T)
plt.scatter(X_test['Height'], y_test, color='red', label = 'Actual Weight')
plt.scatter(X_test['Height'], model.predict(X_test), color='green', label = 'Prdicted Weight')
plt.xlabel('Height')
plt.ylabel('Weight')
plt.rcParams["figure.figsize"] = (10,6) # Custom figure size in inches
plt.title('Actual Vs Predicted Weight for Test Data')
plt.legend()
plt.show()
```

slip22

Q. 1)Create a table student having attributes(rollno, name, class). Using codeigniter, connect to the database and insert 5 recodes in it.

```
//student table in codeignitor
CREATE TABLE student (
 rollno INT PRIMARY KEY,
 name VARCHAR(255),
 class VARCHAR(255)
);
//php
<?php
class Student extends CI_Controller {
 public function __construct() {
 parent::__construct();
 $this->load->database();
 }
 public function insert_students() {
 $data = array(
 array(
 'rollno' => 1,
 'name' => 'John Doe',
 'class' => 'A'
 ),
 array(
 'rollno' => 2,
 'name' => 'Jane Smith',
 'class' => 'B'
 ),
 array(
 'rollno' => 3,
 'name' => 'Bob Johnson',
 'class' => 'C'
 ),
 array(
 'rollno' => 4,
 'name' => 'Alice Williams',
 'class' => 'D'
 ),
 array(
 'rollno' => 5,
 'name' => 'Charlie Brown',
 'class' => 'E'
 )
 );
 $this->db->insert_batch('student', $data);
 }
}
```

Q. 2)Consider any text paragraph. Remove the stopwords.

```
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
text = """In 2021, the world's first space hotel is set to open. The hotel, named Voyager Station, will
be operated by Orbital Assembly Corporation. It will have 24 modules connected by elevator shafts
that make up a rotating wheel orbiting the Earth. The hotel will have artificial gravity, and guests will
be able to participate in recreational activities such as low-gravity basketball and rock climbing."""
# Tokenize the text into words
words = word_tokenize(text)
# Remove stopwords
stop_words = set(stopwords.words('english'))
filtered_words = [word for word in words if word not in stop_words]
# Join the filtered words back into a string
filtered_text = ' '.join(filtered_words)
print(filtered_text)
```

slip23

Q. 1) Create a table student having attributes(rollno, name, class) containing atleast 5 recodes . Using codeigniter, display all its records.

```
//student table
CREATE TABLE student (
 rollno INT PRIMARY KEY,
 name VARCHAR(255),
 class VARCHAR(255)
);
```

```
//php
<?php
class Student extends CI_Controller {
 public function __construct() {
 parent::__construct();
 $this->load->database();}
 public function insert_students() {
 $data = array(
 array(
 'rollno' => 1,
 'name' => 'John Doe',
 'class' => 'A'
 ),
 array(
 'rollno' => 2,
 'name' => 'Jane Smith',
 'class' => 'B'
 ),
 array(
 'rollno' => 3,
 'name' => 'Bob Johnson',
 'class' => 'C'
 ),
 array(
 'rollno' => 4,
 'name' => 'Alice Williams',
 'class' => 'D'
 ),
 array(
 'rollno' => 5,
 'name' => 'Charlie Brown',
 'class' => 'E'
 )
 );
 $this->db->insert_batch('student', $data);
 }
 public function display_students() {
 $query = $this->db->get('student');
 $students = $query->result_array();
 foreach ($students as $student) {
 echo $student['rollno'] . ': ' . $student['name'] . ', Class: ' . $student['class'] . '<br>';}}}
```

Q. 2) Consider any text paragraph. Preprocess the text to remove any special characters and digits.

```
import re
text = """In 2021, the world's first space hotel is set to open. The hotel, named Voyager Station, will
be operated by Orbital Assembly Corporation. It will have 24 modules connected by elevator shafts
that make up a rotating wheel orbiting the Earth. The hotel will have artificial gravity, and guests will
be able to participate in recreational activities such as low-gravity basketball and rock climbing."""
# Preprocess the text to remove special characters and digits
text = re.sub(r'[^a-zA-Z\s]', '', text)
print(text)
```

slip24

Q. 1) Write a PHP script to create student.xml file which contains student roll no, name, address, college and course. Print students detail of specific course in tabular format after accepting course as input

```
<?php
// Create the XML document
$doc = new DOMDocument();
$doc->formatOutput = true;
// Create the root element
$root = $doc->createElement('students');
$doc->appendChild($root);
// Add some student elements
$students = array(
 array(
 'rollno' => 1,
 'name' => 'John Doe',
 'address' => '123 Main St',
 'college' => 'ABC College',
 'course' => 'Computer Science'
 ),
 array(
 'rollno' => 2,
 'name' => 'Jane Smith',
 'address' => '456 Elm St',
 'college' => 'XYZ College',
 'course' => 'Mathematics'
 ),
 array(
 'rollno' => 3,
 'name' => 'Bob Johnson',
 'address' => '789 Oak St',
 'college' => 'ABC College',
 'course' => 'Physics'
 ),
 array(
 'rollno' => 4,
 'name' => 'Alice Williams',
 'address' => '321 Pine St',
 'college' => 'XYZ College',
 'course' => 'Biology'
 ),
 array(
 'rollno' => 5,
 'name' => 'Charlie Brown',
 'address' => '654 Maple St',
 'college' => 'ABC College',
 'course' => 'Chemistry'
 )
);
foreach ($students as $student) {
 $studentElement = $doc->createElement('student');
 $root->appendChild($studentElement);
 foreach ($student as $key => $value) {
 $element = $doc->createElement($key, $value);
 $studentElement->appendChild($element);}}
// Save the XML document to a file
$doc->save('student.xml');
// Load the XML document from the file
$xml = simplexml_load_file('student.xml');
// Get the course to display from the user input
$course = $_GET['course'];
// Display the student details in a table
echo '<table border="1">';
echo '<tr><th>Roll
No</th><th>Name</th><th>Address</th><th>College</th><th>Course</th></tr>';
foreach ($xml->student as $student) {
 if ($student->course == $course) {
 echo '<tr>';
 echo '<td>' . $student->rollno . '</td>';
 echo '<td>' . $student->name . '</td>';
 echo '<td>' . $student->address . '</td>';
 echo '<td>' . $student->college . '</td>';
 echo '<td>' . $student->course . '</td>';
 echo '</tr>';}}
echo '</table>';
?>
```

Q. 2) Consider the following dataset : https://www.kaggle.com/datasets/datasnaek/youtubenew?select=INvideos.csv Write a Python script for the following : i. Read the dataset and perform data cleaning operations on it. ii. ii. Find the total views, total likes, total dislikes and comment count.

```
import pandas as pd
# Read the dataset
df = pd.read_csv("D:\INvideos.csv")
# Perform data cleaning operations
df = df.dropna()
df = df.drop_duplicates()
# Find the total views, total likes, total dislikes and comment count
total_views = df['views'].sum()
total_likes = df['likes'].sum()
total_dislikes = df['dislikes'].sum()
total_comments = df['comment_count'].sum()
print(f'Total Views: {total_views}')
print(f'Total Likes: {total_likes}')
print(f'Total Dislikes: {total_dislikes}')
print(f'Total Comments: {total_comments}')
```

slip25

Q. 1) Write a script to create “cricket.xml” file with multiple elements as shown below: ____ ______ ____ Write a script to add multiple elements in “cricket.xml” file of category, country=”India”

```php
<?php
// Create a new DOM document
$doc = new DOMDocument();
// Create the root element
$cricketTeam = $doc->createElement("CricketTeam");
// Create the first team element for Australia
$teamAustralia = $doc->createElement("Team");
$teamAustralia->setAttribute("country", "Australia");
// Create the player element and set its value
$player1 = $doc->createElement("player", "Steve Smith");
$teamAustralia->appendChild($player1);
// Create the runs element and set its value
$runs1 = $doc->createElement("runs", "7090");
$teamAustralia->appendChild($runs1);
// Create the wicket element and set its value
$wicket1 = $doc->createElement("wicket", "17");
$teamAustralia->appendChild($wicket1);
// Append the team element to the root element
$cricketTeam->appendChild($teamAustralia);
// Create the second team element for India
$teamIndia = $doc->createElement("Team");
$teamIndia->setAttribute("country", "India");
// Create the player element and set its value
$player2 = $doc->createElement("player", "Virat Kohli");
$teamIndia->appendChild($player2);
// Create the runs element and set its value
$runs2 = $doc->createElement("runs", "12169");
$teamIndia->appendChild($runs2);
// Create the wicket element and set its value
$wicket2 = $doc->createElement("wicket", "4");
$teamIndia->appendChild($wicket2);
// Create the category element and set its value
$category = $doc->createElement("category", "Captain");
$teamIndia->appendChild($category);
// Append the team element to the root element
$cricketTeam->appendChild($teamIndia);
// Append the root element to the document
$doc->appendChild($cricketTeam);
// Save the XML file
$doc->save("cricket.xml");
echo "Elements added successfully!";
?>
```

Q. 2) Consider the following dataset : https://www.kaggle.com/datasets/seungguini/youtube-commentsfor-covid19-relatedvideos?select=covid_2021_1.csv Write a Python script for the following : i. Read the dataset and perform data cleaning operations on it. ii. ii. Tokenize the comments in words. iii. Perform sentiment analysis and find the percentage of positive, negative and neutral comments..

```python
import pandas as pd
from nltk.sentiment import SentimentIntensityAnalyzer
from nltk.tokenize import word_tokenize
# Read the dataset
df = pd.read_csv("D:\covid_2021_1.csv")
# Perform data cleaning operations
df = df.dropna()
df = df.drop_duplicates()
# Tokenize the comments into words
df['tokens'] = df['comment_text'].apply(word_tokenize)
# Initialize the sentiment analyzer
sia = SentimentIntensityAnalyzer()
# Perform sentiment analysis on the comments
df['sentiment'] = df['comment_text'].apply(lambda x: sia.polarity_scores(x)['compound'])
# Find the percentage of positive, negative and neutral comments
positive_comments = df[df['sentiment'] > 0.05].shape[0]
negative_comments = df[df['sentiment'] < -0.05].shape[0]
neutral_comments = df[(df['sentiment'] >= -0.05) & (df['sentiment'] <= 0.05)].shape[0]
total_comments = positive_comments + negative_comments + neutral_comments
positive_percentage = positive_comments / total_comments * 100
negative_percentage = negative_comments / total_comments * 100
neutral_percentage = neutral_comments / total_comments * 100
print(f'Positive Comments: {positive_percentage:.2f}%')
print(f'Negative Comments: {negative_percentage:.2f}%')
print(f'Neutral Comments: {neutral_percentage:.2f}%')
```

slip26

Q. 1) Create employee table as follows EMP (eno, ename, designation, salary). Write Ajax program to select the employees name and print the selected employee's details.

```
<!DOCTYPE html>
<html><head><title>Ajax Example</title>
 <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
 <script>
 // Define the employee data
 const employees = [
 { eno: 1, ename: 'Alice', designation: 'Manager', salary: 80000 },
 { eno: 2, ename: 'Bob', designation: 'Developer', salary: 70000 },
 { eno: 3, ename: 'Charlie', designation: 'Designer', salary: 60000 }
 ];
 $(document).ready(function() {
 // Populate the dropdown menu with employee names
 employees.forEach(employee => {
 $('#employee-select').append(`<option
value="${employee.eno}">${employee.ename}</option>`);
 });
 // Handle the change event of the dropdown menu
 $('#employee-select').on('change', function() {
 // Get the selected employee number
 const eno = $(this).val();
 // Find the employee with the selected number
 const employee = employees.find(employee => employee.eno == eno);
 // Display the employee details
 $('#employee-details').html(`
 <p>Employee Number: ${employee.eno}</p>
 <p>Employee Name: ${employee.ename}</p>
 <p>Designation: ${employee.designation}</p>
 <p>Salary: $${employee.salary}</p>
 `);});});
 </script></head>
<body>
 <h1>Select an Employee</h1>
 <select id="employee-select">
 <option value="">--Select an Employee--</option>
 </select>
 <div id="employee-details"></div>
</body></html>
```

Q. 2)Consider text paragraph. """Hello all, Welcome to Python Programming Academy. Python Programming Academy is a nice platform to learn new programming skills. It is difficult to get enrolled in this Academy.""" Preprocess the text to remove any special characters and digits. Generate the summary using extractive summarization process.

```
import nltk
import re
text="""Hello all, Welcome to Python * Programming Academy 3. Python Programming 3 Academy is a nice platform to
learn new programming skills. It is difficult to get enrolled in this Academy."""
text = re.sub(r'[[0-9]{}*]', ' ', text)
print(text)
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize, sent_tokenize
stopWords = set(stopwords.words("english"))
words = word_tokenize(text)
wordfreq = {}
for word in words:
    if word in stopWords:
        if word in wordfreq:
            wordfreq[word] += 1
        else:
            wordfreq[word] = 1
maximum_frequency = max(wordfreq.values())
for word in wordfreq.keys():
    wordfreq[word] = (wordfreq[word]/maximum_frequency)
    wordfreq
sentences = sent_tokenize(text)
sentenceValue = {}
for sentence in sentences:
    for word, freq in wordfreq.items():
        if word in sentence.lower():
            if sentence in sentenceValue:
                sentenceValue[sentence] += freq
            else:
                sentenceValue[sentence] = freq
                sentenceValue
import heapq
summary = ''
summary_sentences = heapq.nlargest(4, sentenceValue, key=sentenceValue.get)
summary = ' '.join(summary_sentences)
print(summary)
```

slip27

Q. 1) Create web Application that contains Voters details and check proper validation for (name, age, and nationality), as Name should be in upper case letters only, Age should not be less than 18 yrs and Nationality should be Indian.(use HTML-AJAX-PHP)

```
//index.html
<!DOCTYPE html>
<html><head>
 <title>Voter Details</title>
 <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
 <script>
 $(document).ready(function() {
 // Handle the form submission
 $('#voter-form').on('submit', function(event) {
 event.preventDefault();
 // Get the form data
 const name = $('#name').val();
 const age = $('#age').val();
 const nationality = $('#nationality').val();
 // Send the form data to the server using Ajax
 $.ajax({
 url: 's27.php',
 type: 'POST',
 data: { name: name, age: age, nationality: nationality },
 success: function(response) {
 // Display the server response
 $('#response').html(response);}});});});
 </script></head><body>
 <h1>Voter Details</h1>
 <form id="voter-form">
 <label for="name">Name:</label>
 <input type="text" id="name" name="name" required><br><br>
 <label for="age">Age:</label>
 <input type="number" id="age" name="age" required><br><br>
 <label for="nationality">Nationality:</label>
 <input type="text" id="nationality" name="nationality" required><br><br>
 <input type="submit" value="Submit">
 </form>
 <div id="response"></div></body></html>
```

.php

```
<?php
// Get the form data
$name = $_POST['name'];
$age = $_POST['age'];
$nationality = $_POST['nationality'];
// Check for proper validation
if (strtoupper($name) != $name) {
 echo 'Name should be in upper case letters only.';
} elseif ($age < 18) {
 echo 'Age should not be less than 18 years.';
} elseif ($nationality != 'Indian') {
 echo 'Nationality should be Indian.';
} else {
 echo 'Validation successful!';}
?>
```

Q. 2) Create your own transactions dataset and apply the above process on your datase

```
import pandas as pd
from mlxtend.frequent_patterns import apriori, association_rules
transactions = [['eggs', 'milk','bread'],['eggs', 'apple'],['milk', 'bread'],['apple', 'milk'],['milk', 'apple',
'bread']]
from mlxtend.preprocessing import TransactionEncoder
te=TransactionEncoder()
te_array=te.fit(transactions).transform(transactions)
df=pd.DataFrame(te_array, columns=te.columns_)
```

```
print(df)
print("******************************")
freq_items = apriori(df, min_support = 0.5, use_colnames = True)
print(freq_items)
print("******************************")
rules = association_rules(freq_items, metric ='support', min_threshold=0.05)
rules = rules.sort_values(['support', 'confidence'], ascending =[False,False])
print(rules)
```

slip28

Q. 1) Write a PHP script using AJAX concept, to check user name and password are valid or Invalid (use database to store user name and password).

```
//index.html
<!DOCTYPE html>
<html><head>
 <title>Login</title>
 <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
 <script>
 $(document).ready(function() {
 // Handle the form submission
 $('#login-form').on('submit', function(event) {
 event.preventDefault();
 // Get the form data
 const username = $('#username').val();
 const password = $('#password').val();
 // Send the form data to the server using Ajax
 $.ajax({
 url: 's28.php',
 type: 'POST',
 data: { username: username, password: password },
 success: function(response) {
 // Display the server response
 $('#response').html(response);
 }});});});
 </script></head>
<body>
 <h1>Login</h1>
 <form id="login-form">
 <label for="username">Username:</label>
 <input type="text" id="username" name="username" required><br><br>
 <label for="password">Password:</label>
 <input type="password" id="password" name="password" required><br><br>
 <input type="submit" value="Login">
 </form>
 <div id="response"></div>
</body></html>
```

```
//login.php
<?php
// Connect to the database
$db = mysqli_connect('hostname', 'username', 'password', 'database_name');
// Check if the connection was successful
if (!$db) {
 die('Connection failed: ' . mysqli_connect_error());
}
// Get the form data
$username = $_POST['username'];
$password = $_POST['password'];
// Check if the username and password are valid
$query = "SELECT * FROM users WHERE username='$username' AND password='$password'";
$result = mysqli_query($db, $query);
if (mysqli_num_rows($result) == 1) {
 echo 'Login successful!';
} else {
 echo 'Invalid username or password.';
}
?>
```

Q. 2) Build a simple linear regression model for Car Dataset.

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

```
df = pd.read_csv(r"D:\CSV 1\Fish.csv")
print('Shape of dataset= ', df.shape)
print(df.head(5))
df.rename(columns={'Length1':'VerticalLen','Length2':'DiagonalLen','Length3':'CrossLen'},inplace = True)
print(df.sample(5))
print(df.Species.value_counts())
df_sp = df.Species.value_counts()
df_sp = pd.DataFrame(df_sp)
print(df_sp.T)
X = df[['Height','Width']]
X.head()
y = df[['Weight']]
y.head(5)
X_train,X_test, y_train, y_test = train_test_split(X, y, test_size =0.2, random_state = 42)
print('X_train dimension= ', X_train.shape)
print('X_test dimension= ', X_test.shape)
print('y_train dimension= ', y_train.shape)
print('y_train dimension= ', y_test.shape)
model = linear_model.LinearRegression()
print(model.fit(X_train,y_train))
print('coef= ', model.coef_)
print('intercept= ', model.intercept_)
print('score= ', model.score(X_test,y_test))
predictedWeight = pd.DataFrame(model.predict(X_test), columns=['Predicted Weight'])
actualWeight = pd.DataFrame(y_test)
actualWeight = actualWeight.reset_index(drop=True)
df_actual_vs_predicted = pd.concat([actualWeight,predictedWeight],axis =1)
print(df_actual_vs_predicted.T)
plt.scatter(X_test['Height'], y_test, color='red', label = 'Actual Weight')
plt.scatter(X_test['Height'], model.predict(X_test), color='green', label = 'Prdicted Weight')
plt.xlabel('Height')
plt.ylabel('Weight')
plt.rcParams["figure.figsize"] = (10,6) # Custom figure size in inches
plt.title('Actual Vs Predicted Weight for Test Data')
plt.legend()
plt.show()
```

slip29

Q. 1) Write a PHP script for the following: Design a form to accept a number from the user. Perform the operations and show the results. 1) Fibonacci Series. 2) To find sum of the digits of that number. (Use the concept of self processing page.)

.php

```
<?php
// Check if the form was submitted
if ($_SERVER['REQUEST_METHOD'] == 'POST') {
 // Get the form data
 $number = $_POST['number'];
 // Calculate the Fibonacci series
 $fibonacci = [0, 1];
 for ($i = 2; $i < $number; $i++) {
 $fibonacci[] = $fibonacci[$i - 1] + $fibonacci[$i - 2];
 }
 // Find the sum of the digits of the number
 $sum_of_digits = array_sum(str_split($number));
}
?>
<!DOCTYPE html>
<html><head> <title>PHP Form</title></head>
<body>
 <h1>PHP Form</h1>
 <form method="post">
 <label for="number">Number:</label>
 <input type="number" id="number" name="number" required><br><br>
 <input type="submit" value="Submit">
 </form>
 <?php if (isset($fibonacci)): ?>
 <h2>Fibonacci Series:</h2>
 <p><?php echo implode(', ', $fibonacci); ?></p>
 <?php endif; ?>
 <?php if (isset($sum_of_digits)): ?>
 <h2>Sum of Digits:</h2>
 <p><?php echo $sum_of_digits; ?></p>
 <?php endif; ?>
</body></html>
```

Q. 2) Build a logistic regression model for Student Score Dataset.

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
# Load the student score dataset
df = pd.read_csv('student_scores.csv')
# Define the features and target variable
X = df[['hours_studied']]
y = df['pass_exam']
# Split the data into training and test sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=0)
# Create a logistic regression model
model = LogisticRegression()
# Train the model on the training data
model.fit(X_train, y_train)
# Make predictions on the test data
y_pred = model.predict(X_test)
# Evaluate the model performance
accuracy_score = model.score(X_test, y_test)
print(f'Accuracy Score: {accuracy_score:.2f}')
```

slip30

Q. 1) Create a XML file which gives details of books available in "Bookstore" from following categories. 1) Yoga 2) Story 3) Technical and elements in each category are in the following format -------------- --------------- -------------- Save the file as "Bookcategory.xml"

```
<?xml version="1.0" encoding="UTF-8"?>
<Bookstore>
 <Yoga>
 <Book><Book_Title>Light on Yoga</Book_Title><Book_Author>B.K.S. Iyengar</Book_Author>
 <Book_Price>20.99</Book_Price></Book>
 <Book><Book_Title>The Yoga Bible</Book_Title><Book_Author>Christina
Brown</Book_Author><Book_Price>15.50</Book_Price></Book></Yoga>
 <Story><Book><Book_Title>The Alchemist</Book_Title><Book_Author>Paulo
Coelho</Book_Author><Book_Price>12.99</Book_Price></Book>
 <Book><Book_Title>The Da Vinci Code</Book_Title><Book_Author>Dan Brown</Book_Author>
<Book_Price>14.75</Book_Price></Book></Story>
 <Technical>
 <Book><Book_Title>Python for Data Science Handbook</Book_Title><Book_Author>Jake
VanderPlas</Book_Author><Book_Price>28.99</Book_Price></Book>
 <Book><Book_Title>Cracking the Coding Interview</Book_Title><Book_Author>Gayle Laakmann
McDowell</Book_Author><Book_Price>23.50</Book_Price> </Book></Technical>
</Bookstore>
```

Q. 2) Create the dataset . transactions = [['eggs', 'milk','bread'], ['eggs', 'apple'], ['milk', 'bread'], ['apple', 'milk'], ['milk', 'apple', 'bread']] . Convert the categorical values into numeric format.Apply the apriori algorithm on the above dataset to generate the frequent itemsets and association rules.

```
import pandas as pd
from mlxtend.frequent_patterns import apriori, association_rules
transactions = [['eggs', 'milk','bread'],['eggs', 'apple'],['milk', 'bread'],['apple', 'milk'],['milk', 'apple',
'bread']]
from mlxtend.preprocessing import TransactionEncoder
te=TransactionEncoder()
te_array=te.fit(transactions).transform(transactions)
df=pd.DataFrame(te_array, columns=te.columns_)
print(df)
print("******************************")
freq_items = apriori(df, min_support = 0.5, use_colnames = True)
print(freq_items)
print("******************************")
rules = association_rules(freq_items, metric ='support', min_threshold=0.05)
rules = rules.sort_values(['support', 'confidence'], ascending =[False,False])
print(rules)
```