

## Slip 1

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

```
<?php
session_start();

if(isset($_SESSION['page_visits'])) {
    $_SESSION['page_visits']++;
} else {
    $_SESSION['page_visits'] = 1;
}

echo "This page has been visited " . $_SESSION['page_visits'] . " times.";
?>
```

**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 pandas as pd
df = pd.read_csv("C:/Users/Admin/Downloads/slip1 dataset/Position_Salaries.csv")
df
df.describe()
df.info()
df.size
df.shape
df.isnull
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
import seaborn as sns
import statsmodels.api as sm
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import r2_score, mean_squared_error
df.columns
df.corr()
sns.regplot(x="Level", y="Salary", data=df)
sns.heatmap(df.corr(), annot=True)
df[['Level', 'Salary']].describe()
x=df[['Level']]
y=df['Salary']
x
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.3, random_state=1)
x_train
y_train
lin_model = LinearRegression()
lin_model.fit(x_train, y_train)
lin_model.coef_
lin_model.intercept_
y_pred=lin_model.predict(x_test)
y_pred
y_test[0:5]
test_accuracy = r2_score(y_test, y_pred)
test_accuracy
mse = mean_squared_error(y_test, y_pred)
mse
plt.scatter(x_train, y_train)
plt.plot(x_train, -77417.58+50769.23*x_train, 'r')
plt.show()
```

## Slip 2

**Q. 1)** Write 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).

### 1.actual\_perference.php

```
<?php
// Get the selected preferences from cookies
$fontStyle = isset($_COOKIE['font_style']) ? $_COOKIE['font_style'] : 'Arial';
$fontSize = isset($_COOKIE['font_size']) ? $_COOKIE['font_size'] : '16px';
$fontColor = isset($_COOKIE['font_color']) ? $_COOKIE['font_color'] : '#000000';
$bgColor = isset($_COOKIE['bg_color']) ? $_COOKIE['bg_color'] : '#FFFFFF';
?>
<!DOCTYPE html>
<html>
<head>
  <title>Actual Implementation</title>
  <style>
    body {
      font-family: <?php echo $fontStyle; ?>;
      font-size: <?php echo $fontSize; ?>;
      color: <?php echo $fontColor; ?>;
      background-color: <?php echo $bgColor; ?>;
    }
  </style>
</head>
<body>
  <h1>Actual Implementation</h1>
  <p>This is an example web page with the selected preferences implemented:</p>

</body>
</html>
```

### 2.perference.php

```
<?php
if (isset($_POST['submit'])) {
  // Get the preferences from the form
  $font_style = $_POST['font_style'];
  $font_size = $_POST['font_size'];
  $font_color = $_POST['font_color'];
  $bg_color = $_POST['bg_color'];

  // Set the cookies for the preferences
  setcookie('font_style', $font_style, time() + (86400 * 30), "/");
  setcookie('font_size', $font_size, time() + (86400 * 30), "/");
  setcookie('font_color', $font_color, time() + (86400 * 30), "/");
  setcookie('bg_color', $bg_color, time() + (86400 * 30), "/");

  // Redirect to the display preferences page
  header("Location: display_preferences.php");
}
```

```
    exit();
}
?>
```

```
<!DOCTYPE html>
<html>
<head>
    <title>Preferences</title>
</head>
<body>
    <h1>Preferences</h1>
    <form action="" method="POST">
        <label for="font_style">Font Style:</label>
        <select id="font_style" name="font_style">
            <option value="Arial">Arial</option>
            <option value="Verdana">Verdana</option>
            <option value="Tahoma">Tahoma</option>
        </select><br>

        <label for="font_size">Font Size:</label>
        <select id="font_size" name="font_size">
            <option value="small">Small</option>
            <option value="medium">Medium</option>
            <option value="large">Large</option>
        </select><br>

        <label for="font_color">Font Color:</label>
        <input type="color" id="font_color" name="font_color"><br>

        <label for="bg_color">Background Color:</label>
        <input type="color" id="bg_color" name="bg_color"><br>

        <input type="submit" name="submit" value="Save Preferences">
    </form>
</body>
</html>
```

### 3.display\_preference.php

```
<?php
// Get the selected preferences from cookies
$fontStyle = isset($_COOKIE['font_style']) ? $_COOKIE['font_style'] : 'Arial';
$fontSize = isset($_COOKIE['font_size']) ? $_COOKIE['font_size'] : '16px';
$fontColor = isset($_COOKIE['font_color']) ? $_COOKIE['font_color'] : '#000000';
$bgColor = isset($_COOKIE['bg_color']) ? $_COOKIE['bg_color'] : '#FFFFFF';
?>
```

```
<!DOCTYPE html>
```

```

<html>
<head>
  <title>Selected Preferences</title>

</head>
<body>
  <h1>Selected Preferences</h1>
  <p>Font Style: <?php echo $fontStyle; ?></p>
  <p>Font Size: <?php echo $fontSize; ?></p>
  <p>Font Color: <?php echo $fontColor; ?></p>
  <p>Background Color: <?php echo $bgColor; ?></p>
  <p><a href="preferences.php">Change Preferences</a></p>
  <p><a href="actual_preferences.php">See Actual Implementation</a></p>
</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
df= pd.read_csv("C:/Users/Admin/Downloads/slip 2 dataset/Salary_Data.csv")
df
df.describe()
df.info()
df.size
df.shape
df.isnull
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
import seaborn as sns
import statsmodels.api as sm
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import r2_score, mean_squared_error
df.columns
sns.regplot(x="YearsExperience", y="Salary", data=df)
df.corr()
sns.heatmap(df.corr(), annot=True)
df[['YearsExperience', 'Salary']].describe()
x=df[['YearsExperience']]
y=df['Salary']
x
y
x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=0.2,random_state=1)
lin_model = LinearRegression()
lin_model.fit(x_train,y_train)
lin_model.coef_
lin_model.intercept_
y_pred=lin_model.predict(x_test)
y_pred
y_test[0:5]
test_accuracy = r2_score(y_test,y_pred)
test_accuracy
mse = mean_squared_error(y_test,y_pred)
mse
plt.scatter(x_train,y_train)
plt.plot(x_train,25609.89+9332.944*x_train,'r')
plt.show()

```

### Slip 3

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
session_start();
if(!isset($_SESSION['attempts']))
{
    $_SESSION['attempts']=0;
}
if(isset($_POST['username']) && isset($_POST['password']))
{
    if($_POST['username']=='username'&& $_POST['password']=='password')
    {
        $_SESSION['attempts']=0;
        echo "WELCOME";
    }
    else
    {
        $_SESSION['attempts']++;
        if($_SESSION['attempts']>=3)
        {
            echo "Maximum number of attempts reached....please try again!!";
        }
        else
        {
            echo "INVALID username or password..please try again";
        }
    }
}
if(!isset($_POST['username']) || !isset($_POST['password']) ||
$_SESSION['attempts']<3){
?>
<form method="post">
    <label for="username">USERNAME</label>
    <input type="text" name="username" id="username"><br>
    <label for="password">PASSWORD</label>
    <input type="password" name="password" id="password"><br>
    <input type="submit" value="LOGIN" >
</form>
<?php
}
?>
```

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 pandas as pd
import seaborn as sns
import numpy as np
import statsmodels.api as sm
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report
df = pd.read_csv("C:/Users/Admin/Downloads/slip 3 dataset/User_Data.csv")
df
df.head
df.shape
df.size
df.info()
df['Gender'].unique()
df['Gender'].replace({'Male':0, 'Female':1}, inplace=True)
df['Gender'].unique()
df['Purchased'].value_counts()
sns.countplot(x='Purchased', data=df)
sns.countplot(x='Purchased', hue='Gender', data=df)
sns.distplot(df['Age'])
df.head()
x=df.drop('Purchased', axis=1)
y=df['Purchased']
x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=0.2,random_state=1)
log_model = LogisticRegression()
log_model.fit(x_train,y_train)
y_pred=log_model.predict(x_test)
y_pred[:5]
y_test[:5]
y_prob=log_model.predict_proba(x_test)
y_prob[:5]
test_accuracy=accuracy_score(y_test,y_pred)
print("model accuracy",test_accuracy)
con_matrix = confusion_matrix(y_test,y_pred)
con_matrix
matrix = pd.crosstab(y_test,y_pred,rownames=['Actual'],colnames=['Predicted'])
sns.heatmap(matrix,annot=True)
cls_report = classification_report(y_test,y_pred)
print(cls_report)
```

#### Slip 4

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]

##### 1.accept.html

```
<html>
<body>
    <form action="earning.php" method="post">
        Employee Number
        <input type="number" name="eno"><br><br>
        Employee name
        <input type="text" name="ename"><br><br>
```

### Employee Address

```
<input type="textarea" name="eadd"><br><br>
<input type="submit" value="SUBMIT" name="submit">
```

```
</form>
```

```
</body>
```

```
</html>
```

### 2. earning.php

```
<?php
```

```
setcookie('eno',$_POST['eno']);
```

```
setcookie('ename',$_POST['ename']);
```

```
setcookie('eadd',$_POST['eadd']);
```

```
?>
```

```
<html>
```

```
<body>
```

```
<form action="information.php" method="post">
```

```
    BASIC :
```

```
    <input type="number" name="basic"><br><br>
```

```
    DA
```

```
    <input type="number" name="da"><br><br>
```

```
    HRA
```

```
    <input type="number" name="hra"><br><br>
```

```
    <input type="submit" value="SUBMIT" name="submit">
```

```
</form>
```

```
</body>
```

```
</html>
```

### 3.information.php

```
<?php
```

```
    echo"<br> NO: ".$_COOKIE["eno"];
```

```
    echo"<br> NAME: ".$_COOKIE["ename"];
```

```
    echo"<br> ADDRESS: ".$_COOKIE["eadd"];
```

```
    echo"<br><br>";
```

```
    echo "<table border=1><tr><th>Details</th><th>VALUES</th></tr>";
```

```
    echo "<tr><td>Basic</td><td>".$_POST['basic']. "</td></tr>";
```

```
    echo "<tr><td>HRA</td><td>".$_POST['hra']. "</td></tr>";
```

```
    echo "<tr><td>DA</td><td>".$_POST['da']. "</td></tr>";
```

```
    echo"<br><br><br>";
```

```
    $total = $_POST['basic']+$_POST['hra']+$_POST['da'];
```

```
    echo"<br> TOTAL: ".$total;
```

```
?>
```

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

Same as slip1

### Slip 5

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 version="1.0" encoding="UTF-8"?>
```

```
<items>
```

```

<item>
    <itemname>book</itemname>
    <itemprice>100</itemprice>
    <itemquantity>dfsa</itemquantity>
</item>
<item>
    <itemname>pen</itemname>
    <itemprice>10</itemprice>
    <itemquantity>fgvds</itemquantity>
</item>
<item>
    <itemname>pencil</itemname>
    <itemprice>1</itemprice>
    <itemquantity>grfsd</itemquantity>
</item>
<item>
    <itemname>nootbook</itemname>
    <itemprice>0</itemprice>
    <itemquantity>bdf</itemquantity>
</item>
<item>
    <itemname>goa</itemname>
    <itemprice>1</itemprice>
    <itemquantity>t</itemquantity>
</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, virginica) of Iris flowers given just 4 features: sepal and petal lengths and widths.. Find the accuracy of the model

### Slip 6

Q. 1) Write PHP script to read "book.xml" file into simpleXML object. Display attributes and elements . ( simple\_xml\_load\_file() function )

#### 1.Book.xml

```

<?xml version="1.0" encoding="UTF-8"?>
<bookinfo>
    <book>
        <booknumber>1</booknumber>
        <bookname>java</bookname>
        <authorname>balguruswami</authorname>
        <bookprice>250</bookprice>
        <year>2006</year>
    </book>
    <book>
        <booknumber>2</booknumber>

```



```

        <bookname>C</bookname>
        <authorname>Denid Ritchie</authorname>
        <bookprice>500</bookprice>
        <year>1971</year>
    </book>
</bookinfo>

```

## 2.book.php

```

<?php
$xml=simplexml_load_file("book.xml");
foreach($xml->book as $ban)
{
    echo "Book Number = $ban->booknumber "."<br> ";
    echo "Book Name = $ban->bookname"."<br> ";
    echo "Author name = $ban->authorname"."<br>";
    echo "Book Price = $ban->bookprice"."<br>";
    echo "Year = $ban->year"."<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.

TID	Items
1	Bread, Milk
2	Bread, Diaper, Beer, Eggs
3	Milk, Diaper, Beer, Coke
4	Bread, Milk, Diaper, Beer
5	Bread, Milk, Diaper, Coke

```

pip install mlxtend
import pandas as pd
from mlxtend.frequent_patterns import apriori, association_rules

```

```

transactions=[['bread','milk'],
               ['bread','diaper','beer','eggs'],
               ['milk','diaper','beer','coke'],
               ['bread','milk','diaper','beer'],
               ['bread','milk','diaper','coke']]

```

```

from mlxtend.preprocessing import TransactionEncoder

```

```

te = TransactionEncoder()
te_array = te.fit(transactions).transform(transactions)

```

```

df = pd.DataFrame(te_array, columns=te.columns_)
df
freq_items = apriori(df, min_support=0.5, use_colnames = True)
print(freq_items)
rules = association_rules(freq_items, metric = 'support', min_threshold=0.05)
rules = rules.sort_values(['support', 'confidence'], ascending=[False, False])
freq_items['length']=freq_items['itemsets'].apply(lambda x: len(x))
freq_items
print(rules)

```

## Slip 7

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

### 1.Movie.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<movieinfo>
    <movie>
        <movienumber>1</movienumber>
        <movietitle>Jab we met</movietitle>
        <actorname>Shahid Kapoor</actorname>
        <releaseyear>2013</releaseyear>
    </movie>
    <movie>
        <movienumber>2</movienumber>
        <movietitle>Bhoolbhulaiya</movietitle>
        <actorname>Akshay Kumar</actorname>
        <releaseyear>2008</releaseyear>
    </movie>
    <movie>
        <movienumber>3</movienumber>
        <movietitle>Sairat</movietitle>
        <actorname>Parshya and Archi</actorname>
        <releaseyear>2015</releaseyear>
    </movie>
    <movie>
        <movienumber>4</movienumber>
        <movietitle>Student of the year</movietitle>
        <actorname>Alia Bhatt</actorname>
        <releaseyear>2018</releaseyear>
    </movie>
    <movie>
        <movienumber>5</movienumber>
        <movietitle>Wednesday</movietitle>
        <actorname>Wednesday Aadam</actorname>
        <releaseyear>2023</releaseyear>
    </movie>
</movieinfo>
```

### 2.movies.php

```
<?php
$dom =new DomDocument();
$dom->load('movie.xml');
echo "<h2>Name of the movie</h2>";
$bname=$dom->getElementsByTagName("movietitle");
foreach($bname as $b)
```

```

{

    echo $b->textContent."<br>";
}
echo "<h2>Name of the Actor</h2>";
$aname=$dom->getElementsByTagName("actorname");
foreach($aname as $b)
{

    echo $b->textContent."<br>";
}

?>

```

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
df=pd.read_csv("C:/Users/Admin/Downloads/Market_Basket_Optimisation.csv")
df.info()
print(df.head(10))

```

```

records=[]
for i in range(0,7499):
    records.append([str(df.values[i,j]) for j in range(0,20)])

```

```

from mlxtend.preprocessing import TransactionEncoder
te=TransactionEncoder()
te_array=te.fit(records).transform(records)
df=pd.DataFrame(te_array,columns=te.columns_)
print(df)
freq=apriori(df,min_support=0.0043,use_colnames=True)
print(freq)
rules=association_rules(freq,metric='support',min_threshold=0.0043)
rules=rules.sort_values(['support','confidence'],ascending=[False,False])
rules

```

## Slip 8

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

```

<html>
<body>
<button onclick = "myfunction()">Click</button>
<script type="text/javascript">
function myfunction()
{
    var person=prompt("Please enter your name","Prajakta");

```

```

        if(person!=null)
        {
            var r=confirm("Hello "+person+" Exams are near,have you started preparing
for?");
            if(r==true)
            {
                alert("Good!!");
            }
            else
            {
                alert("Start preparing");
            }
        }
    }
</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.

```

df=pd.read_csv("C:\\Users\\Admin\\Desktop\\Groceries data.csv")
df.info()
print(df.head(10))

```

```

records=[]
for i in range(0,38764):
    records.append([str(df.values[i,j]) for j in range(0,7)])

```

```

from mlxtend.preprocessing import TransactionEncoder
te=TransactionEncoder()
te_array=te.fit(records).transform(records)
df=pd.DataFrame(te_array,columns=te.columns_)
print(df)
freq=apriori(df,min_support=0.0043,use_colnames=True)
print(freq)
rules=association_rules(freq,metric='support',min_threshold=0.0043)
rules=rules.sort_values(['support','confidence'],ascending=[False,False])
rules

```

### Slip 9

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

```

<html>
<body>
<form>
    <input type="text" placeholder="Username" id="text1" /><br />
    <input type="password" placeholder="Password" id="text2" /><br />
    <input type="button" value="Login" onclick="javascript:validate()" />

```

```

</form>
<script type="text/javascript">
function validate()
{
    if( document.getElementById("text1").value == "user"
        && document.getElementById("text2").value == "password" )
    {
        alert( "validation succeeded" );

    }
    else
    {
        alert( "validation failed" );

    }
}
</script>
</body>
</html>

```

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

### Slip 10

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

```

<!DOCTYPE html>
<html>
<body>
    <script src = "https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js">
    </script>
    <script>
        $(document).ready(function()
        {
            $("#btn1").click(function()
            {
                $("#im").before("text");
            });
            $("#btn2").click(function()
            {
                $("#im").after("text");
            });
        });
    </script>
    <img id="im" src = "p.png" width=200px height=200px></img>
    <button id="btn1"> Before </button>
    <button id="btn2"> After </button>
</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.

TID	Items
1	'eggs', 'milk','bread'
2	'eggs', 'apple'
3	'milk', 'bread'
4	'apple', 'milk'
5	'milk', 'apple', 'bread'

Same as slip 6

### Slip 12

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]

#### 1.Info.html

```
<html>
<head>
<style>
spam
{
    font-size:25px;
}
table
{
    color:black;
}
</style>

<script type="text/javascript">
    function print()
    {
        var ob=false;
        ob=new XMLHttpRequest();

        ob.open("GET","setA1.php?");
        ob.send();

        ob.onreadystatechange=function()
        {
            if(ob.readyState==4 && ob.status==200)
```

```

        {
            document.getElementById("i").innerHTML=ob.responseText;
        }
    }
}
</script>
</head>
<body>
<body>
<center>
<h3>Display the content of a Contact.dat file</h3>
<br><input type="button" value=Print onclick="print()">
<span id="i"></span>
</center>
</body>
</html>
2.info.php
<?php

```

```

    $fp=fopen("Contact.dat",'r');//create a dat file
    echo"<table border=1>";
    echo"<tr><th>Sr_no</th><th>Name</th><th>R_no</th><th>Mobile_no</th><th>
Address</th></tr>";
    while($row=fscanf($fp, "%s %s %s %s %s"))
    {
        echo"<tr>";
        foreach($row as $r)
        {
            echo"<td>$r</td>";
        }
        echo"</tr>";
    }
    echo"</table>";
    fclose($fp);
?>

```

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.

Same as slip 1

### Slip 17

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

```

<!DOCTYPE html>
<html>
<head>
    <title>Student Registration Form</title>
</head>
<body onload="showGreeting()">

```

```

<!-- Student Registration Form -->
<h1>Student Registration Form</h1>
<form>
    <label for="name">Name:</label>
    <input type="text" id="name" name="name"><br><br>

    <label for="email">Email:</label>
    <input type="email" id="email" name="email"><br><br>

    <label for="phone">Phone:</label>
    <input type="tel" id="phone" name="phone"><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.

```

import nltk
import re

```

```

paragraph_text="""So,keep working.Keep striving.Never give up.Fall down seven times,get up e

```

```

formatted_article_text=re.sub("[^a-zA-Z0-9 ]",'', paragraph_text)

```

```

print(formatted_article_text)

```



## Slip 20

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

```
<Students>
  <Student>
    <Name>John Doe</Name>
    <Age>20</Age>
    <Gender>Male</Gender>
    <Grade>10</Grade>
    <City>New York</City>
  </Student>
  <Student>
    <Name>Jane Smith</Name>
    <Age>19</Age>
    <Gender>Female</Gender>
    <Grade>12</Grade>
    <City>Los Angeles</City>
  </Student>
  <Student>
    <Name>Michael Johnson</Name>
    <Age>21</Age>
    <Gender>Male</Gender>
    <Grade>11</Grade>
    <City>Chicago</City>
  </Student>
  <Student>
    <Name>Sarah Wilson</Name>
    <Age>18</Age>
    <Gender>Female</Gender>
    <Grade>9</Grade>
    <City>Miami</City>
  </Student>
  <Student>
    <Name>David Lee</Name>
    <Age>19</Age>
    <Gender>Male</Gender>
    <Grade>12</Grade>
    <City>San Francisco</City>
  </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.

```

import nltk

import re
nltk.download('all')

paragraph_text="""Hello all,Welcome to Python Programming Academy.Python Programming Academy i

from nltk.corpus import stopwords
stop_words = set(stopwords.words('english'))
print(stopwords.words('english'))

from nltk.tokenize import word_tokenize
word_tokens = word_tokenize(paragraph_text)
filtered_sentence = [w for w in word_tokens if not w.lower() in stop_words]
filtered_sentence = []
for w in word_tokens:
    if w not in stop_words:
        filtered_sentence.append(w)

print(word_tokens)
print(filtered_sentence)

```

### Slip 30

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

<Book>

<Book\_Title> </Book\_Title>

<Book\_Author> </Book\_Author>

<Book\_Price> </Book\_Price>

</Book>

Save the file as “Bookcategory.xml”

<Bookstore>

<Category>

<Category\_Name>Yoga</Category\_Name>

<Book>

<Book\_Title>The Yoga Bible</Book\_Title>

<Book\_Author>Christina Brown</Book\_Author>

<Book\_Price>19.99</Book\_Price>

</Book>

<Book>

<Book\_Title>Light on Yoga</Book\_Title>

<Book\_Author>B.K.S. Iyengar</Book\_Author>

<Book\_Price>14.99</Book\_Price>

</Book>

</Category>

<Category>

<Category\_Name>Story</Category\_Name>

<Book>

<Book\_Title>The Alchemist</Book\_Title>

```

    <Book_Author>Paulo Coelho</Book_Author>
    <Book_Price>12.99</Book_Price>
  </Book>
  <Book>
    <Book_Title>The Hobbit</Book_Title>
    <Book_Author>J.R.R. Tolkien</Book_Author>
    <Book_Price>10.99</Book_Price>
  </Book>
</Category>
<Category>
  <Category_Name>Technical</Category_Name>
  <Book>
    <Book_Title>Clean Code</Book_Title>
    <Book_Author>Robert C. Martin</Book_Author>
    <Book_Price>29.99</Book_Price>
  </Book>
  <Book>
    <Book_Title>JavaScript: The Good Parts</Book_Title>
    <Book_Author>Douglas Crockford</Book_Author>
    <Book_Price>24.99</Book_Price>
  </Book>
</Category>
</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.

Same as 6