

## ASSIGNMENT-2

### EMERGING METHODS FOR EARLY DETECTION OF FOREST FIRES

Student name	Mohamed Mohideen Afran
Student reg. no	<b>961819104056</b>
Team ID	PNT2022TMID34407
Maximum Mark	<b>2</b> mark

## # ASSIGNMENT- 2

### # Importing Libraries

```
import os
```

```
import numpy as np
```

```
import pandas as pd
```

```
Current Working Directory
```

```
os.getcwd()
```

### # Loading the dataset

```
data=pd.read_csv('Churn_Modelling.csv')
```

```
data
```

### # 3.Visualizations

#### # \*Univariate Analysis

```
import matplotlib.pyplot as plt
```

```
import seaborn as sns
```

```
from sklearn.preprocessing import LabelEncoder,MinMaxScaler
```

```
from sklearn.model_selection import train_test_split
```

```
sns.histplot(data["CreditScore"],color='red')
```

#### # \*Bivariate Analysis

```
sns.catplot(x='Gender', y='Age', hue='IsActiveMember', data=data,color='blue')
```

#### # \*Multi-Variate Analysis

```
sns.pairplot(data,hue="Age")
```

### # 4.Perform descriptive statistics on the dataset.

```
data.describe()
```

```
# 5.Handle the Missing values.
```

```
data.isnull().sum()
```

```
# 6.Find the outliers and replace the outliers
```

```
CreditsMedian = data.loc[data['CreditScore']<400, 'CreditScore'].median()
```

```
ProdMedian = data.loc[data['NumOfProducts']>=3.5,'NumOfProducts'].median()
```

```
data.loc[data.CreditScore < 400, 'CreditScore'] = np.nan
```

```
data.fillna(CreditsMedian,inplace=True)
```

```
data.loc[data.NumOfProducts > 3, 'NumOfProducts'] = np.nan
```

```
data.fillna(ProdMedian,inplace=True)
```

```
# 7. Check for Categorical columns and perform encoding.
```

```
from sklearn.preprocessing import LabelEncoder
```

```
le = LabelEncoder()
```

```
data['Surname'] = le.fit_transform(data['Surname'])
```

```
data['Gender'] = le.fit_transform(data['Gender'])
```

```
data.head(10)
```

```
# 8.Split the data into dependent and independent variables
```

```
independent = data.iloc[:, 0:1]
```

```
dependent = data.iloc[:,1:]
```

```
# 9.Scale the independent variables
```

```
nm =MinMaxScaler()
```

```
N_independent = nm.fit_transform(independent)
```

```
# 10. Split the data into training and testing
```

```
x=data.drop(columns = ['CreditScore'],axis = 1)
```

```
y = data['CreditScore']
```

```
from sklearn.model_selection import train_test_split
```

```
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size = 0.3, random_state = 0)
```

```
x_train
```

```
x_test
```

