**ASSIGNMENT-2**

**EMERGING METHODS FOR EARLY**

**DETECTION OF FOREST FIRES**

| Student name | delber |
| --- | --- |
| Student reg. no | **961819104028** |
| Team ID | PNT2022TMID34407 |
| Maximum Mark | **2** 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