# -\*- coding: utf-8 -\*-

"""

Created on Sun Mar 24 00:07:21 2022

@author: Mudit Mahajan

"""

# coding: utf-8

# In[57]:

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

from sklearn import metrics

import pprint

# In[58]:

train = pd.read\_csv('pubg\_final\_data\_labels.csv')

train.Gender[train.Gender == 'Male'] = 1

train.Gender[train.Gender == 'Female'] = 0

# train = train['Gender'].replace(0, 'Female')

# train = train['Gender'].replace(1, 'Male')

# In[59]:

from sklearn.preprocessing import Imputer

imputer = Imputer(missing\_values='NaN' ,strategy='mean')

imputer = imputer.fit(train[['Do you have high self-esteem?']])

train[['Do you have high self-esteem?']] = imputer.transform(train[['Do you have high self-esteem?']])

# In[60]:

train.drop(['Unnamed: 0','Country','Do you feel preoccupied with your gaming behaviour?','Do you feel more irritability, anxiety or even sadness when you try to either reduce or stop your gaming activity?','Do you feel the need to spend increasing amount of time engaged gaming in order to achieve satisfaction or pleasure?','Do you systematically fail when trying to control or cease your gaming activity?','Have you lost interests in previous hobbies and other entertainment activities as a result of your engagement with the game?','Have you continued your gaming activity despite knowing it was causing problems between you and other people?','Have you deceived any of your family members, therapists or others because the amount of your gaming activity?','Do you play in order to temporarily escape or relieve a negative mood (e.g., helplessness, guilt, anxiety)?','Have you jeopardised or lost an important relationship, job or an educational or career opportunity because of your gaming activity?','How often do you have trouble wrapping up the final details of a project, once the challenging parts have been done?','How often do you have difficulty getting things in order when you have to do a task that requires organization?','How often do you have problems remembering appointments or obligations?','When you have a task that requires a lot of thought, how often do you avoid or delay getting started?','How often do you fidget or squirm with your hands or feet when you have to sit down for a long time','How often do you feel overly active and compelled to do things, like you were driven by a motor?', 'Do you feel nervous, anxious or on edge?','Are you not being able to stop or control worrying?','Do you worry too much about different things?','Do you have trouble relaxing?','Do you ever feel so restless that it is hard to sit still?','Do you become easily annoyed or irritable?','Do you feel afraid as if something awful might happen?'],axis=1,inplace=True)

train.drop(['solo\_headshotKills','solo\_kills','solo\_longestTimeSurvived','solo\_roundsPlayed','solo\_roundMostKills','solo\_top10s','solo\_wins','solo\_AvgSurvival','solo\_Top10%','solo\_WinRatio','duo\_headshotKills','duo\_kills','duo\_longestTimeSurvived','duo\_roundsPlayed','duo\_roundMostKills','duo\_top10s','duo\_wins','duo\_AvgSurvival','duo\_Top10%','duo\_WinRatio','squad\_headshotKills','squad\_kills','squad\_longestTimeSurvived','squad\_roundsPlayed','squad\_roundMostKills','squad\_top10s','squad\_wins','squad\_AvgSurvival','squad\_Top10%','squad\_WinRatio'],axis=1,inplace=True)

#This is for calculating result for ADHD.

#train.drop(['username','IGD\_score','anxiety\_result','IGD\_result','anxiety\_score'],axis=1,inplace=True)

#This is for calculating result for anxiety result.

#train.drop(['username','IGD\_score','ADHD\_result','IGD\_result','anxiety\_score'],axis=1,inplace=True)

#This is for calculating result for IGD result

train.drop(['username','IGD\_score','ADHD\_result','anxiety\_result','anxiety\_score'],axis=1,inplace=True)

# In[61]:

print(train.columns)

# In[62]:

print(train.head)

# In[63]:

from sklearn.model\_selection import train\_test\_split

X\_train, X\_test, y\_train, y\_test = train\_test\_split(train.drop('IGD\_result',axis=1),

train['IGD\_result'], test\_size=0.20,

random\_state=101)

# In[64]:

from sklearn.preprocessing import StandardScaler

std = StandardScaler()

X\_train = std.fit\_transform(X\_train)

X\_test = std.transform(X\_test)

# In[66]:

print(X\_train)

# In[67]:

from sklearn.linear\_model import LogisticRegression

logmodel = LogisticRegression()

logmodel.fit(X\_train,y\_train)

predictions = logmodel.predict(X\_test)

# In[69]:

print(predictions)

from sklearn.metrics import classification\_report

print(classification\_report(y\_test,predictions))

y\_pred = logmodel.predict(X\_test)

print('Accuracy of logistic regression classifier on test set: {:.2f}'.format(logmodel.score(X\_test, y\_test)))