**EARLY DIABETES PREDICTION**

**(HEALTH CARE)**

**CODE:**

import pandas as pd

import matplotlib.pyplot as plt

import numpy as np

%matplotlib inline

df = pd.read\_csv(r"pima-indians-diabetes.csv")

df.shape

df.head()

df.describe()

df["Outcome"].value\_counts()

df.groupby("Outcome").mean()

X=df.drop(columns='Outcome',axis=1)

Y=df['Outcome']

print(X)

print(Y)

import seaborn as sns

import matplotlib.pyplot as plt

corrmat = df.corr()

top\_corr\_features = corrmat.index

plt.figure(figsize=(20,20))

g=sns.heatmap(df[top\_corr\_features].corr(),annot=True,cmap="RdYlGn")

Outcome\_true\_count = len(df.loc[df['Outcome'] == True])

Outcome\_false\_count = len(df.loc[df['Outcome'] == False])

(Outcome\_true\_count,Outcome\_false\_count)

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

feature\_columns = ['Pregnancies','Glucose','BloodPressure','SkinThickness','Insulin','BMI','DiabetesPedigreeFunction','Age']

predicted\_class = ['Outcome']

X = df[feature\_columns].values

y = df[predicted\_class].values

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size = 0.30,

random\_state=10)

from sklearn.ensemble import RandomForestClassifier

random\_forest\_model = RandomForestClassifier(random\_state=10)

random\_forest\_model.fit(X\_train, y\_train.ravel())

predict\_train\_df = random\_forest\_model.predict(X\_test)

from sklearn import metrics

print("Accuracy = {0:.3f}".format(metrics.accuracy\_score(y\_test,

predict\_train\_df)))

