**VOICE RECOGNITION USING GENERATIVE ARTIFICIAL INTELLIGENCE**

**SOURCE CODE:**

**import numpy as np**

**import pandas as pd**

**import matplotlib.pyplot as plt**

**import seaborn as sns**

**df = pd.read\_csv('../input/voice-dataset/voice.csv')**

**df.head()**

**df.shape**

**df.isna().sum()**

**df['label'].value\_counts()**

**val= [1584,1584]**

**label = ['male','female']**

**plt.figure(figsize=(6,8))**

**plt.pie(val,labels=label)**

**plt.legend()**

**plt.show()**

**corr = df.corr()**

**sns.set(font\_scale=1.5)**

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

**sns.heatmap(corr,annot = True,cmap='coolwarm')**

**plt.show()**

**corr = df.corr()**

**corr = corr[corr>0.85]**

**plt.figure(figsize=(15,15))**

**sns.heatmap(corr,annot = True,cmap='coolwarm')**

**plt.show()**

**df = df.drop(['meanfreq','centroid',],axis=1)**

**#we removed meanfreq and centroid as it had the maximum correlated columns**

**sns.boxplot(x=df.maxdom,y=df.label)**

**plt.show()**

**sns.boxplot(x=df.dfrange,y=df.label)**

**plt.show()**

**df = df.drop(['dfrange','maxdom'],axis=1)**

**df = df.drop(['kurt','skew'],axis=1)**

**sns.boxplot(x=df['skew'],y=df.label)**

**plt.show()**

**sns.boxplot(x=df['kurt'],y=df.label)**

**plt.show()**

**sns.boxplot(x=df['sd'],y=df.label)**

**plt.show()**

**sns.boxplot(x=df['IQR'],y=df.label)**

**plt.show()**

**df = df.drop(['IQR'],axis=1)**

**sns.boxplot(x=df['sfm'],y=df.label)**

**plt.show()**

**sns.boxplot(x=df['sp.ent'],y=df.label)**

**plt.show()**

**df = df.drop('sp.ent',axis=1)**

**#We will remove sp.ent as it has more outliers**

**sns.pairplot(df,kind = 'boxplot',hue='label')**

**plt.show()**

**df = df.drop(['maxfun','modindx','minfun'],axis=1)**

**#these columns have too many outliers**

**corr = df.corr()**

**corr = corr[corr>0.85]**

**plt.figure(figsize=(5,5))**

**sns.heatmap(corr,annot = True,cmap='coolwarm')**

**plt.show()**

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

**X\_train,x\_test,y\_train,y\_test = train\_test\_split(X,y,test\_size=0.2,random\_state = 1)**

**X = df.iloc[:,:-1]**

**y = df.label**

**from sklearn.metrics import classification\_report,confusion\_matrix**

**def eval(y\_pred,ytest):**

**print("Confusion matrix:\n")**

**cm = confusion\_matrix(y\_pred,ytest)**

**sns.heatmap(cm,annot = True,xticklabels=["Female","Male"],yticklabels=["Female","Male"])**

**plt.show()**

**print("Classification Report\n",classification\_report(y\_pred,ytest))**

**def score(model):**

**print("Training score: ",model.score(X\_train,y\_train))**

**print("Test score: ",model.score(x\_test,y\_test))**

**from sklearn.tree import DecisionTreeClassifier**

**DTmodel = DecisionTreeClassifier(min\_samples\_split = 5,max\_depth = 10,random\_state = 0)**

**ypred1 = DTmodel.predict(x\_test)**

**ypred1[:5]**

**DTmodel.fit(X\_train,y\_train)**

**eval(ypred1,y\_test)**

**IMPLEMENTATION IS CODED BY SURIYAN.PA**