**1-import pandas as pd**

**import numpy as np**

**df=pd.read\_csv("AirQuality.csv",encoding='cp1252')**

**2-df.head(5) 3-df.describe() 4-df.shape 5-df.info()**

**6-df.isnull().sum() 7-df.count() 8-df.describe()**

**9-df.info() 10-df = df.drop(['stn\_code','agency', 'location\_monitoring\_station'],axis=1)**

**11-df.isna().sum() 12-df = df.dropna(subset=['date']) 13-df.isna().sum() 14-df.columns**

**15-df['type'].unique()**

**16-types = {**

**"Residential" : "K",**

**"Residential and others" : "RO",**

**"Industrial Area":"I" ,**

**"Industrial Areas" : "I",**

**"Industrial" : "I" ,**

**"Sensitive Area": "s",**

**"Sensitive Areas":"s",**

**"Sensitive":"s",**

**"NaN":"PRO",**

**"Residential, Rural and other Areas":"MO" }**

**17-df.type = df.type.replace(types) 18-df['type'].unique() 19-df.head() 20-df.info()**

**21-df['date']=pd.to\_datetime(df['date'], errors="coerce")**

**df.head(5)**

**22-df['year']=df.date.dt.year**

**df.head()**

**23-COLS = ['so2','no2', 'rspm', 'spm', 'pm2\_5'] 24-df.info()**

**25-import numpy as np**

**from sklearn.impute import SimpleImputer**

**imputer = SimpleImputer(missing\_values = np.nan, strategy='mean')**

**26-df[COLS] = imputer.fit\_transform(df[COLS]) 27-df.head() 28-df.nunique() 29-df.duplicated().sum()**

**30-df.drop\_duplicates() 31-df.head() 32-df['type'].value\_counts()**

**33-df['type'].replace({ 'MO':1, 'I':2, 's':3 , 'RO':4, 'K':5, 'RIRUO':6 }, inplace=True)**

**34-df.info() 35-df['type']**

**35-from sklearn.preprocessing import LabelEncoder**

**labelencoder = LabelEncoder()**

**df['state'] =labelencoder.fit\_transform(df['state'])**

**df.head()**

**36-dfAndhra = df[df['state']==0] 37-dfAndhra 38-dfAndhra['location'].value\_counts()**

**39-from sklearn.preprocessing import OneHotEncoder**

**onehotencoder = OneHotEncoder(sparse=False, handle\_unknown='error', drop='first')**

**40-pd.DataFrame(onehotencoder.fit\_transform(dfAndhra[['location']]))**

**41-dfAndhra['location'].value\_counts() 42-df.isnull().sum()**

**43-df=df.fillna(df.max())**

**df.isnull().sum()**

**44-df.describe() 45-df[df['so2']>100]=0**

**46-import pandas as pd**

**df = pd.read\_csv('heart.csv')**

**47-df.shape 48-df.info() 49-df.dtypes 50-df.nunique() 51-df.info() 52-df['ca'].unique()**

**53-df.ca.value\_counts() 53-df.loc[df['ca']==4] 54-df.loc[df['ca']==4,'ca']=np.NaN**

**55-df['ca'].unique() 56-df.isna().sum()**

**57-df=df.fillna(df.max())**

**df.isnull().sum()**

**58-duplicates = df.duplicated(keep=False).sum()**

**Duplicates 59-df.describe()**

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

**from sklearn import svm**

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

**61-X = df.drop('target', axis=1)**

**y = df.target**

**X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.3, random\_state=1)**

**62-from sklearn import svm**

**clf = svm.SVC(kernel='linear')**

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

**y\_pred = clf.predict( X\_test)**

**63-from sklearn import metrics**

**accuracy = metrics.accuracy\_score(y\_test, y\_pred)**

**print("Accuracy:", accuracy)**

**64-print("Precision:",metrics.precision\_score(y\_test, y\_pred))**

**print("Recall:",metrics.recall\_score(y\_test, y\_pred))**