**ML Lab Libraries and Functionalities**

**Lab 2 (Basics)**

Check and remove duplicates: df.duplicated()

df.drop\_duplicates()

Removing Columns: df = df.drop([‘c1’],axis=1)

Null Values: df.isna().sum()

df.dropna()

Statistics: mean = ukD.Age.mean()

Mode = ukD.Age.mode().iloc[0]

Median = ukD.Age.median()

To extract NAN-columns=> df.select\_dtypes(exclude=[‘number’]).columns.to\_list()

Fill null values => df = df.fillna( <val> )

**OUTLIER:**

df = ukD.sort\_values(by='Age',ascending=True)

q1 = np.percentile(df['Age'],25)

q3 = np.percentile(df['Age'],75)

iqr = q3-q1

lwr = q1 - (1.5\*iqr)

upr = q3 + (1.5\*iqr)

outliers=[]

for i in df['age']:

if(i<lwr\_bound or i>upr\_bound):

outliers.append(i)

for i in outliers:

df['age'] = np.where(df['age']==i,mean, df['age'])

**OR:** import seaborne as sns

Sns.boxplot( df[‘Age’] )

LabelEncoding: from sklearn.preprocessing import LabelEncoder

df = lbl.fit\_transform(df)

Standardization : from sklearn.preprocessing import StandardScalar

df = scl.fit\_transform(df)

**Data Splitting:** from sklearn.model\_selection import train\_test\_split

**Lab 3:** ANN => from sklearn.neural\_network import MLPClassifier

**Lab 4:** Naïve\_Bayes => from sklearn.naive\_bayes import GaussianNB

**Lab 5:**PCA => from sklearn.decomposition import PCA

pca = PCA(n\_components=3)

New\_x = pca.fit\_transform(x)

**Lab 6:** Decision Tree => from sklearn.tree import DecisionTreeClassifier

**Lab 7:** SVM => from svm import SVC,LinearSVC

**Lab 8:** Kmeans => from sklearn.cluster import KMeans

Km = KMeans(n\_clusters=3)

**Lab 9:** KNN => from sklearn.neighbors import KNeighborsClassifier