Missing data set

Never remove the observation

Take the mean of given column

From sklearn.preprocessing import imputer

Sklearn: library containing rich containing functions related to machine learning

Preprocessing: library of skleran

Imputer: allows us to tkake care of missing data

//create an object

Imputer = Imputer(missing\_values = ‘NaN’, stratergy = ’mean’, axis =0)

// cmd + I to inspect about the element

// fit the imputer to a dataset

Imputer = imputer.fit(X[:,1:3])

// now replace the missing values by new imputer

X[:,1:3] = imputer.transform(X[:,1:3])

GETTING STARTED WITH CATEGORICAL DATA SET

Categorical data set contains two values only : yes or no

Encode categorical data into numbers because ml models contains only numbers

Lib used: sklerarn.preprocessing library

Function used : labelEncoder

create object of labelencoder

labelEncoder\_X = LabelEncoder()

// converting text labels into numbers still cretes problems. ML models would think 1>0 and 2>1>0, however there is no relation between the two. Hence we create dummy variables

Function used for dummy variable : OneHotEncoder

Split data set into two set training set and test set

Library : crossvalidation library from sklearn

Import function : train\_test\_split

//begin training and import set

Create training and test variables

Make them equal to test\_train\_split(array(in our case the dataset), test\_size = 0.2, random\_state=0)

Test\_size = 0.2 means that if there are 10 columns in our dataset, then we assign 2 columns to test\_size and 8 columns to our training set

ML models make prediction based on training set to deduce the data in test set

Feature scaling

Very imporatint in ML

Eucludian distance by the formulae we observe that both the variables are not in the same scale. We observe that one dominates over the other very easily. Hence we need to scale the values accordingly so we use feature scaling for that.

Formulae used for feature scaling

Standardization

N = x – mean(x), D = standary deviation(x)

Normalization

N = x – min(x), D = max(x) – min(x)

Library : sklearn.preprocessing

Class: StandardScaler

Fit\_transform: fit the object to training set and then transform it

Not applicable to test set, we don’t need to fit the object first, we can just transform it.

Data preprocessing template