Preprocessing Steps	**R**	**Python**
Reading_file	read.csv(file = "file_name" , header=TRUE , sep="," , stringsAsFactors = F , na.strings=" ")	import pandas as pd dataframe = pd.read_csv("file_name ")
Statistics for columns	summary(dataframe)	dataframe.describe()
Summary of dataframe	str(dataframe)	dataframe.info()
Column names	colnames(dataframe)	dataframe.columns
Changing Index	rownames(dataframe) = dataframe$column dataframe$column=NULL	dataframe.set_index('column_name', inplace=True)
Checking columns datatypes	is.numeric(dataframe$column)	dataframe.column.dtype
Changing the datatypes	as.numeric(dataframe$column)	dataframe.column.astype('int')

Finding missing values	colSums(is.na(dataframe))
	# Returns no. of missing values per column
	sum(is.na(dataframes))
	# Returns total number of missing values per dataframe
		dataframe.isnull().sum()
		# Returns no. of missing values per column

Imputing Missing values (Numeric)	library(DMwR)
	df_imputed = centralImputation(dataframe)
		from sklearn.preprocessing import **Imputer**
		imputer = Imputer(missing_values='NaN', strategy='mean',axis =0)
		imputer.fit_transform(dataframe.column)
		# Returns numpy array

Imputing Missing values (Categorical)	NA	dataframe.column.apply(lambda x : x.fillna(x.value_counts().index[0]))

Categorical to Numeric	library(dummies)
	dummy(dataframe$column)
		pd.get_dummies(dataframe.column, prefix= 'column_name')

Scaling the Numerical Attribute (Range)	library(vegan)
	dataframe = decostand(x =dataframe[, c(all_numeric_columns)], method = "range",MARGIN = 2)
		from sklearn.preprocessing import **MinMaxScaler**
		minmax_scaler = MinMaxScaler()
		minmax_scaler.fit_transform(dataframe[all_numeric_columns])
		# Returns Numpy array

Scaling the Numerical Attribute (Standardize) (Mean = 0, SD = 1)	dataframe = decostand(x =dataframe[, c(all_numeric_columns)],
	method ="standardize",MARGIN = 2)
		from sklearn.preprocessing import **StandardScaler**
		std_scalar = StandardScaler()
		std_scalar.fit_transform(dataframe[all_numeric_columns])
		# Returns Numpy array