Data Preprocessing Techniques in Machine learning

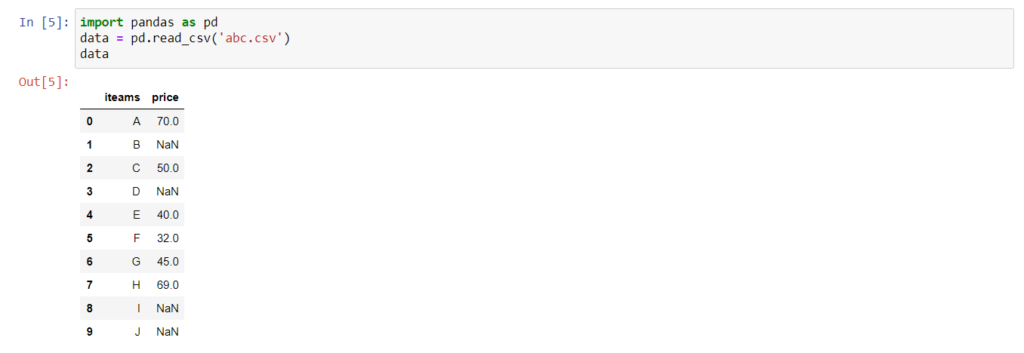
POSTED ON[MARCH 26, 2021](https://5minutesengineering.com/2021/03/26/data-preprocessing-techniques-in-machine-learning/)

Without data machine learning cannot be imagined, so its very important to provide appropriate data to machine learning algorithms. Firstly we should check the relevance of data, i.e. the data must be relevant with respect to the objective we want to achieve. For example I want to make groups of students based on grades, to achieve this if I provide students personal data like name, phone number, address will not help us, instead if I provide students academic data, that will surely help in achieving the objective. So here academic data is valid and personal data is irrelevant/invalid data. The following are the data preprocessing techniques that will make our data READY for machine learning algorithms.

Managing Missing Values

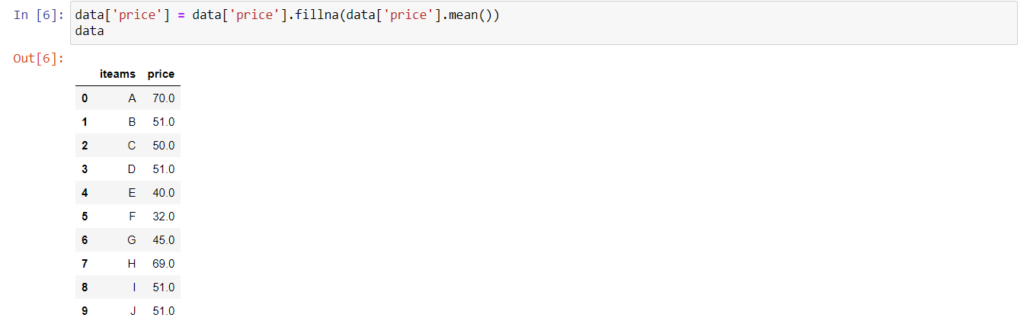
A data consisting of Missing values can be considered as incomplete data which is of NO use to machine learning algorithms, so we need to remove the NAN from dataset and replace it with SOMETHING. Now what can be this SOMETHING, it could be mean,median,standard deviation,min value or max values. Below mentioned codes explains each of the case.

[abcDOWNLOAD](https://5minutesengineering.com/wp-content/uploads/2021/04/abc-1.csv)



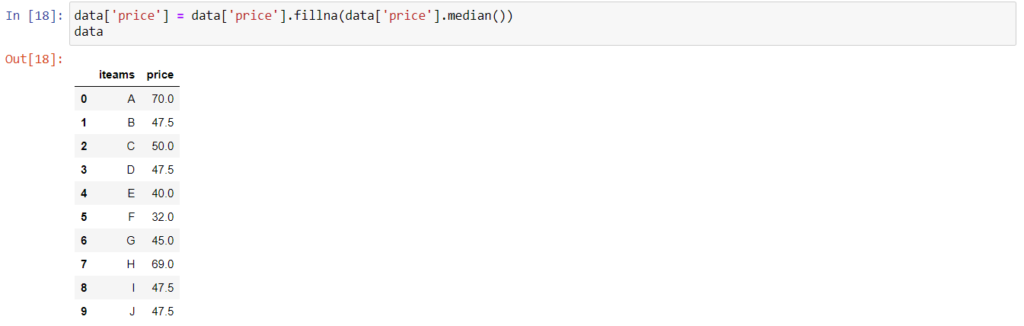
data['price'] = data['price'].fillna(data['price'].mean())

data



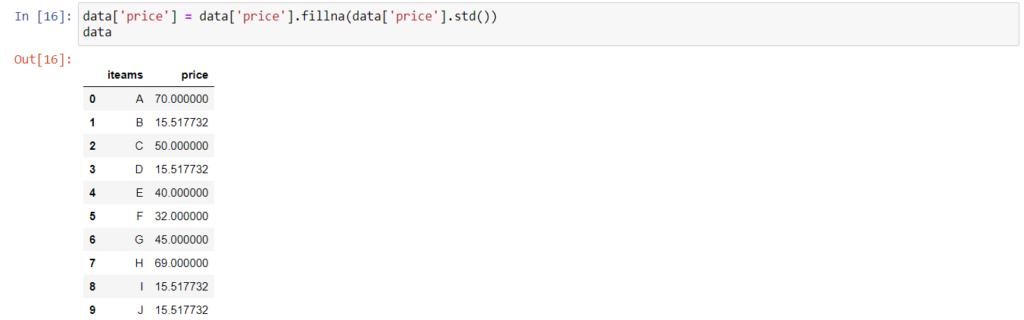
data['price'] = data['price'].fillna(data['price'].median())

data



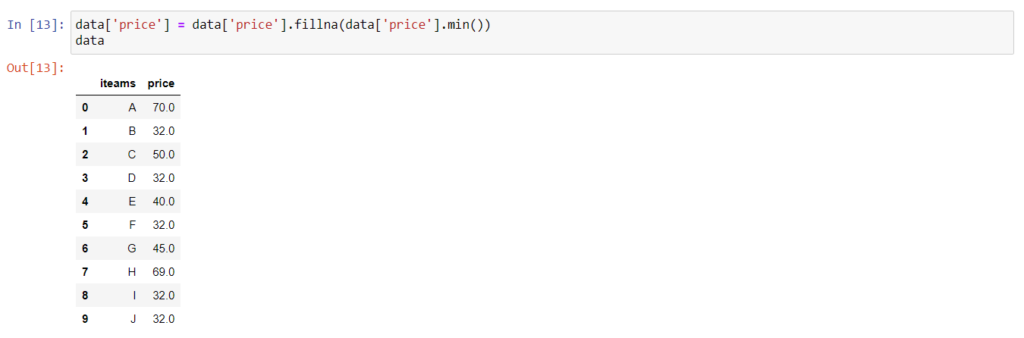
data['price'] = data['price'].fillna(data['price'].std())

data



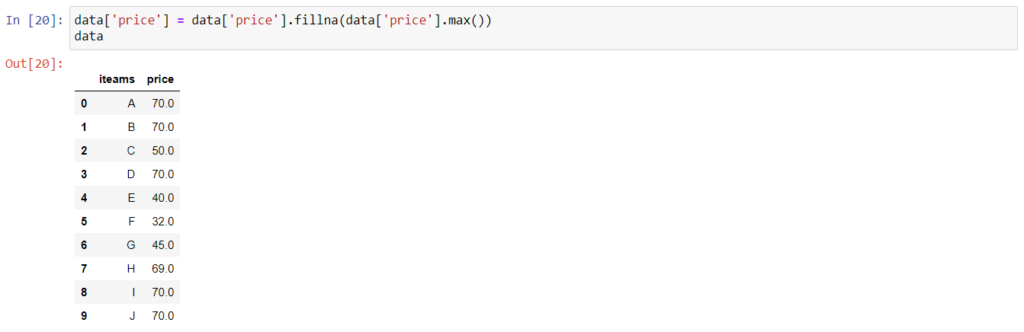
data['price'] = data['price'].fillna(data['price'].min())

data



data['price'] = data['price'].fillna(data['price'].max())

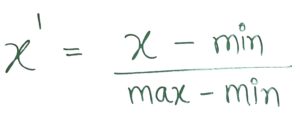
data



Normalization (Feature Scaling)

Normalization (also called, Min-Max normalization) is a scaling technique such that when it is applied the features will be rescaled so that the data will fall in the range of [0,1].

here x’ is the rescaled value which is calculated using the below mentioned formula.



from pandas import read\_csv

from numpy import set\_printoptions

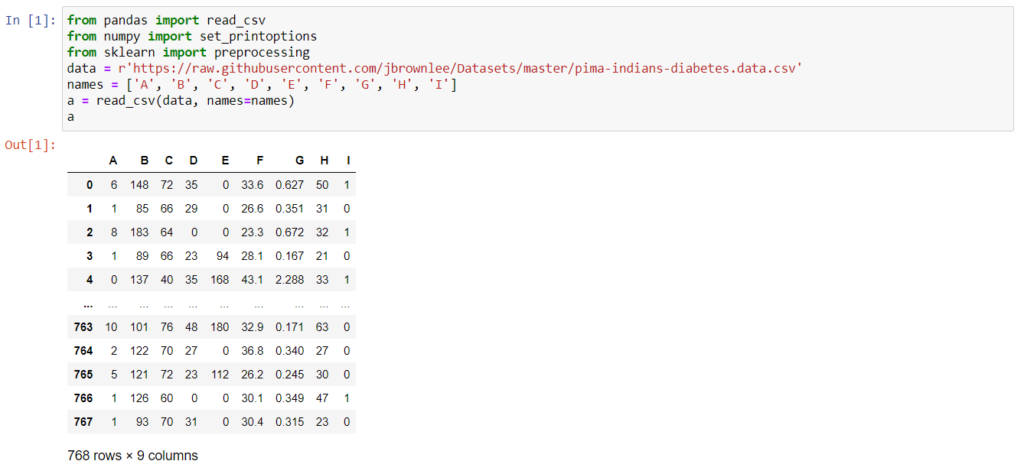
from sklearn import preprocessing

data = r'https://raw.githubusercontent.com/jbrownlee/Datasets/master/pima-indians-diabetes.data.csv'

names = ['A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I']

a = read\_csv(data, names=names)

a

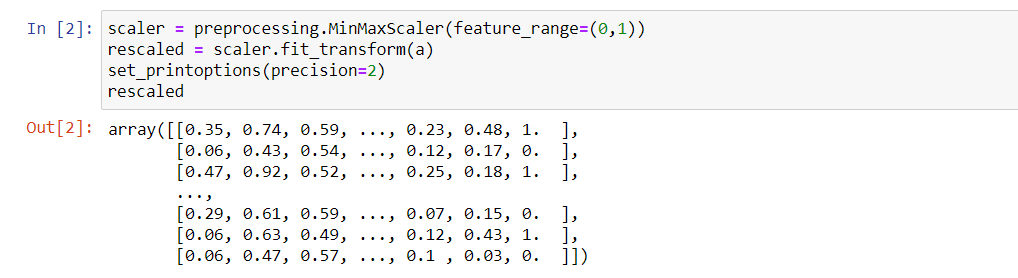


scaler = preprocessing.MinMaxScaler(feature\_range=(0,1))

rescaled = scaler.fit\_transform(a)

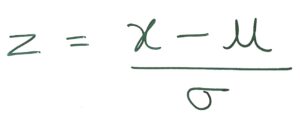
set\_printoptions(precision=2)

rescaled



Standardization

Standardization is a scaling technique such that when it is applied the features will be rescaled so that they’ll have the properties of a standard normal distribution with mean,μ=0 and standard deviation, σ=1, it is also called Z score normalization. Here z is the z score which is calculated using below mentioned formula.

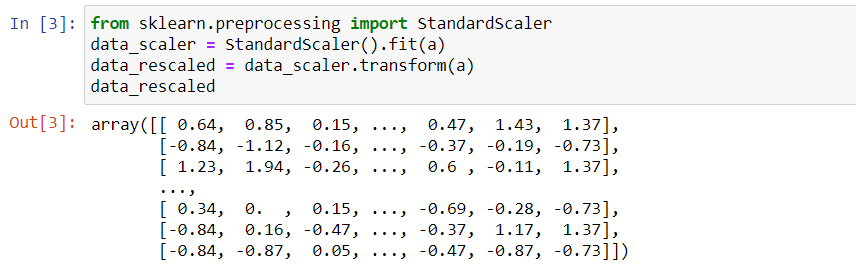


from sklearn.preprocessing import StandardScaler

data\_scaler = StandardScaler().fit(a)

data\_rescaled = data\_scaler.transform(a)

data\_rescaled



Binarization

Biarization as the name suggests here we try to convert number into either ‘1’ or ‘0’. When dataset contains probabilities and we want to convert the probabilities into crisp values we can use binarization.

from sklearn.preprocessing import Binarizer

binary = Binarizer(threshold=0.5)

binary1 = binary.transform(a)

binary1

