Data Preprocessing

Dr. R. Bala Krishnan

Assistant Professor

National Institute of Technology
Tiruchirappalli, Tamil Nadu – 620 015

February 1, 2021

Overview

- Basics of Machine Learning
- Load Dataset
- Need for Data Preprocessing
- Objectives of Data Preprocessing
- Data Preprocessing Steps
- Need for Feature Scaling
- Summary

Basics of Machine Learning

Sample Bike Dataset

Features / Attributes

Dataset

Sample

Bike	Length *(in mm)*	Width *(in mm)*	Height *(in mm)*	Wheel Base (in mm)	Ground Clearance *(in mm)*	Fuel Tank Capacity *(in L)*	Price *(in Rs)*	Value for Price
Hondac Activa	1814	704	1151	1260	155	5.3	78,000	Yes
Yamahas Fascino	1820	675	1120	1270	130	5.2	85,000	Yes
Heros Maestro	1841	695	1190	1261	155	5.2	70,000	No
Suzukis Access	1870	655	1160	1265	160	5.6	65,000	Yes
TVSK Jupiter	1834	650	1115	1275	150	5.6	78,000	No
TVSK Scooty	1834	650	1115	1230	135	5	55,000	No

Target

Load Dataset

Possible Ways

- **Numpy**
  - ✓ Arrays & Matrices
  - ✓ Used for scientific computing

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$$

- **Pandas**
  - ✓ Used for data manipulation and analysis
  - ✓ Best for handling tabular datasets comprising different variable types

X	Y	Z
1	1	1.2
2	4	2.2
4	16	3.1

Numpy vs Pandas Dataframe

data - NumPy array (read only)

	0	1	2	3
0	5.1	3.5	1.4	0.2
1	4.9	3	1.4	0.2
2	4.7	3.2	1.3	0.2
3	4.6	3.1	1.5	0.2
4	5	3.6	1.4	0.2
5	5.4	3.9	1.7	0.4
6	4.6	3.4	1.4	0.3
7	5	3.4	1.5	0.2
8	4.4	2.9	1.4	0.2
9	4.9	3.1	1.5	0.1
10	5.4	3.7	1.5	0.2
11	4.8	3.4	1.6	0.2
12	4.8	3	1.4	0.1

Format Resize Background color

Close

df - DataFrame

Index	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)
0	5.1	3.5	1.4	0.2
1	4.9	3	1.4	0.2
2	4.7	3.2	1.3	0.2
3	4.6	3.1	1.5	0.2
4	5	3.6	1.4	0.2
5	5.4	3.9	1.7	0.4
6	4.6	3.4	1.4	0.3
7	5	3.4	1.5	0.2
8	4.4	2.9	1.4	0.2
9	4.9	3.1	1.5	0.1
10	5.4	3.7	1.5	0.2
11	4.8	3.4	1.6	0.2
12	4.8	3	1.4	0.1
13	4.3	3	1.1	0.1

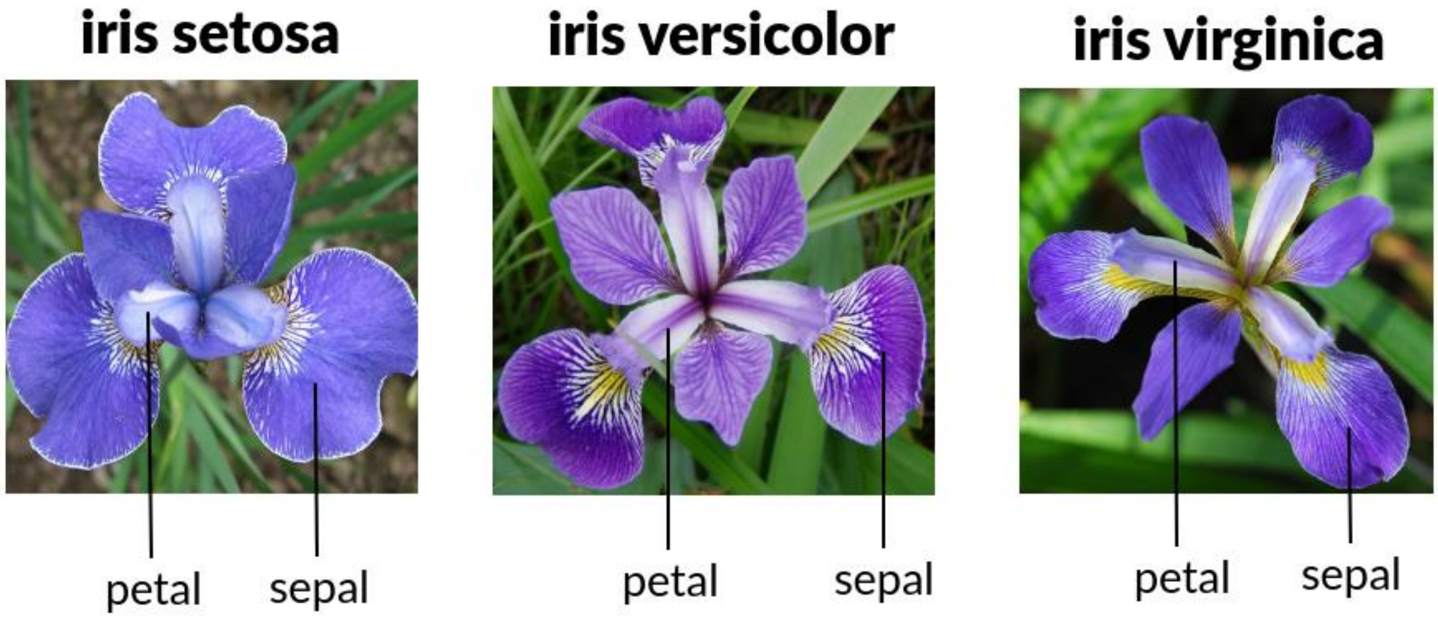
Format Resize Background color Column min/max Save and Close Close

Built-In Datasets

Dataset	Usage
Boston	Regression
Iris	Classification
Diabetes	Regression
Digits	Classification
Linnerud	Multivariate Regression
Wine	Classification
Breast Cancer	Classification

Iris Dataset

- Is a multivariate data set
- Consists of 50 samples from each — *Setosa*, *Virginica* and *Versicolor*
- Length and Width of Sepals and Petals



Iris Dataset

```
'target_names': array(['setosa', 'versicolor', 'virginica'], dtype='|S10'), 'feature_names':
['sepal length (cm)', 'sepal width (cm)', 'petal length (cm)', 'petal width (cm)'], 'data':
array([[5.1, 3.5, 1.4, 0.2],
       [4.9, 3. , 1.4, 0.2],
       [4.7, 3.2, 1.3, 0.2],
       [4.6, 3.1, 1.5, 0.2],
       [5. , 3.6, 1.4, 0.2],
       [5.4, 3.9, 1.7, 0.4],
       [4.6, 3.4, 1.4, 0.3],
       [5. , 3.4, 1.5, 0.2],
       [4.4, 2.9, 1.4, 0.2],
       [4.9, 3.1, 1.5, 0.1],
       [5.4, 3.7, 1.5, 0.2],
       [4.8, 3.4, 1.6, 0.2],
       [4.8, 3. , 1.4, 0.1],
       [4.3, 3. , 1.1, 0.1],
       [5.8, 4. , 1.2, 0.2],
       [5.7, 4.4, 1.5, 0.4],
       [5.4, 3.9, 1.3, 0.4],
       [5.1, 3.5, 1.4, 0.3],
       [5.7, 3.8, 1.7, 0.3],
       [5.1, 3.8, 1.5, 0.3],
```

```
{'target': array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
       2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
       2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2]), 'DESCR': '.. _iris_d
```

Load Iris Dataset

Normal

```
from sklearn import datasets
iris = datasets.load_iris()
```

Using Pandas

```
from sklearn import datasets
import pandas as pd
df=pd.DataFrame(datasets.load_iris()['data'],
    columns=datasets.load_iris()['feature_names'])
```

Pandas Dataframe

Index	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)
0	5.1	3.5	1.4	0.2
1	4.9	3	1.4	0.2
2	4.7	3.2	1.3	0.2
3	4.6	3.1	1.5	0.2
4	5	3.6	1.4	0.2
5	5.4	3.9	1.7	0.4
6	4.6	3.4	1.4	0.3
7	5	3.4	1.5	0.2
8	4.4	2.9	1.4	0.2
9	4.9	3.1	1.5	0.1
10	5.4	3.7	1.5	0.2
11	4.8	3.4	1.6	0.2
12	4.8	3	1.4	0.1
13	4.3	3	1.1	0.1

Load Dataset from URL

```
6,148,72,35,0,33.6,0.627,50,1
1,85,66,29,0,26.6,0.351,31,0
8,183,64,0,0,23.3,0.672,32,1
1,89,66,23,94,28.1,0.167,21,0
0,137,40,35,168,43.1,2.288,33,1
5,116,74,0,0,25.6,0.201,30,0
3,78,50,32,88,31.0,0.248,26,1
10,115,0,0,0,35.3,0.134,29,0
2,197,70,45,543,30.5,0.158,53,1
8,125,96,0,0,0.0,0.232,54,1
4,110,92,0,0,37.6,0.191,30,0
10,168,74,0,0,38.0,0.537,34,1
10,139,80,0,0,27.1,1.441,57,0
1,189,60,23,846,30.1,0.398,59,1
5,166,72,19,175,25.8,0.587,51,1
7,100,0,0,0,30.0,0.484,32,1
0,118,84,47,230,45.8,0.551,31,1
7,107,74,0,0,29.6,0.254,31,1
1,103,30,38,83,43.3,0.183,33,0
1,115,70,30,96,34.6,0.529,32,1
3,126,88,41,235,39.3,0.704,27,0
8,99,84,0,0,35.4,0.388,50,0
7,196,90,0,0,39.8,0.451,41,1
9,119,80,35,0,29.0,0.263,29,1
11,143,94,33,146,36.6,0.254,51,1
10,125,70,26,115,31.1,0.205,41,1
7,147,76,0,0,39.4,0.257,43,1
1,97,66,15,140,23.2,0.487,22,0
```

```
import numpy as np
import urllib
url =
https://raw.githubusercontent.com/jbrownlee/Datasets/master/pima-indians-diabetes.data.csv
raw_data = urllib.urlopen(url)
dataset = np.loadtxt(raw_data, delimiter=",")
X = dataset[:,0:7]
y = dataset[:,8]
```

Load Dataset from File

	A
1	Car;MPG;Cylinders;Displacement;Horsepower;Weight;Acceleration;Model;Origin
2	STRING;DOUBLE;INT;DOUBLE;DOUBLE;DOUBLE;DOUBLE;INT;CAT
3	Chevrolet Chevelle Malibu;18.0;8;307.0;130.0;3504.;12.0;70;US
4	Buick Skylark 320;15.0;8;350.0;165.0;3693.;11.5;70;US
5	Plymouth Satellite;18.0;8;318.0;150.0;3436.;11.0;70;US
6	AMC Rebel SST;16.0;8;304.0;150.0;3433.;12.0;70;US
7	Ford Torino;17.0;8;302.0;140.0;3449.;10.5;70;US
8	Ford Galaxie 500;15.0;8;429.0;198.0;4341.;10.0;70;US
9	Chevrolet Impala;14.0;8;454.0;220.0;4354.;9.0;70;US
10	Plymouth Fury iii;14.0;8;440.0;215.0;4312.;8.5;70;US
11	Pontiac Catalina;14.0;8;455.0;225.0;4425.;10.0;70;US
12	AMC Ambassador DPL;15.0;8;390.0;190.0;3850.;8.5;70;US
13	Citroen DS-21 Pallas;0;4;133.0;115.0;3090.;17.5;70;Europe
14	Chevrolet Chevelle Concours (sw);0;8;350.0;165.0;4142.;11.5;70;US
15	Ford Torino (sw);0;8;351.0;153.0;4034.;11.0;70;US
16	Plymouth Satellite (sw);0;8;383.0;175.0;4166.;10.5;70;US
17	AMC Rebel SST (sw);0;8;360.0;175.0;3850.;11.0;70;US
18	Dodge Challenger SE;15.0;8;383.0;170.0;3563.;10.0;70;US
19	Plymouth 'Cuda 340;14.0;8;340.0;160.0;3609.;8.0;70;US
20	Ford Mustang Boss 302;0;8;302.0;140.0;3353.;8.0;70;US
21	Chevrolet Monte Carlo;15.0;8;400.0;150.0;3761.;9.5;70;US
22	Buick Estate Wagon (sw);14.0;8;455.0;225.0;3086.;10.0;70;US
23	Toyota Corolla Mark ii;24.0;4;113.0;95.00;2372.;15.0;70;Japan
24	Plymouth Duster;22.0;6;198.0;95.00;2833.;15.5;70;US
25	AMC Hornet;18.0;6;199.0;97.00;2774.;15.5;70;US
26	Ford Maverick;21.0;6;200.0;85.00;2587.;16.0;70;US
27	Datsun PL510;27.0;4;97.00;88.00;2130.;14.5;70;Japan
28	Volkswagen 1131 Deluxe Sedan;26.0;4;97.00;46.00;1835.;20.5;70;Europe

```
import pandas as pd
dataset = pd.read_csv('C:\Users\bala\Desktop\cars.csv', delimiter=";")
```

Data Preprocessing

Need for Data Preprocessing

#	Cushion Color	Cushion Diameter *(in cm)*	Cushion Length *(in cm)*	Cushion Area *(in mm)*	Cap Color	Cap diameter *(in cm)*	Cap Length *(in cm)*	Tail Color	Tail Length *(in cm)*	Target
1	Black	0.8	3	2.4	Black	1	4	Black	3	Black
2	Black	0.8	3	2.4	Black	1	4	Black	3	Black
3	Blue	0.8	3	2.4	Blue		4	Blue	3	Blue
4	Blue	0.8%	3	2.4	Blue	1	4	Blue	3	Blue
5	Blue	0.8	3	2.4	Blue	1	4	Blue	3	Blue
6	Black	0.8	3	2.4	Black	1	4.	Black	3	Black
7	Black	0.8	3	2.4	Black	NULL	4	Black	3	Black
8	Blue	0.8	3	2.4	Blue	1	4	Blue	3	Blue
9	Blue	0.8	3	2.4	Blue	1	4	Blue	3	Blue

Objectives of Data Preprocessing

- Format the data to make it suitable for running ML algorithms
- Clean the data to remove incomplete variables
- Sample the data further to reduce running times for algorithms and memory requirements

Data Preprocessing Steps

- Preliminary Steps
  - ✓ Drop duplicate rows
  - ✓ Drop columns with only-one/less unique values
- Intermediate Steps
  - ✓ Drop columns with NULL values
  - ✓ Drop rows with NULL values
  - ✓ Drop redundant columns
  - ✓ Find target column
- Advanced Steps
  - ✓ Handle missing values
  - ✓ Investigate categorical columns
- More Advanced Step
  - ✓ Feature scaling

Commands for Data Preprocessing

Step 1: Find & Drop Duplicate Rows

```
print(any(loans_2007_repeated['id'].duplicated()))
loans_2007_repeated.drop_duplicates(subset=['id'], keep='first')
```

Step 2: No. of NULL

```
row_NULL = loans_2007.isnull().sum(axis = 1)
column_NULL = loans_2007.isnull().sum(axis = 0)
```

Step 3: Drop Columns

```
half_count = len(loans_2007) / 2
loans_2007 = loans_2007.dropna(thresh=half_count, axis=1)
loans_2007 = loans_2007.drop(['url', 'desc'], axis=1)
```

Commands for Data Preprocessing

Step 4: Drop Columns with Only One Value

```
loans_2007.apply(pd.Series.nunique)

loans_2007 = loans_2007.loc[:,loans_2007.apply(pd.Series.nunique) != 1]
```

Step 5: Drop Columns with Less Unique Values

```
for col in loans_2007.columns:
    if (len(loans_2007[col].unique()) < 4):
        loans_2007 = loans_2007.drop(col,axis=1)
```

Sl. No.	Length	Width
1	55	50
2	55	45
3	55	25
4	55	11
5	55	25
6	55	11

Handle Missing Values

- **Drop rows**
- **Replace with default value**

loans_2007['loan_amnt'].fillna(0, inplace=True)

- **Mean**

loans_2007['loan_amnt'].fillna(loans_2007['loan_amnt'].mean(), inplace=True)

- **Median**

loans_2007['loan_amnt'].fillna(loans_2007['loan_amnt'].median(), inplace=True)

- **Mode**

loans_2007['loan_amnt'].fillna(loans_2007['loan_amnt'].mode()[0], inplace=True)

Missing values will be displayed as NaN

"inplace = True" -> Updates the dataframe in which you are working on

Handle Categorical Values

- A < B < C < D < E < F < G
- mapping_dict = {"grade":{ "A": 1, "B": 2, "C": 3, "D": 4, "E": 5, "F": 6, "G": 7 } }
- filtered_loans = filtered_loans.replace(mapping_dict)

Feature Scaling

Need for Feature Scaling / Normalization / Standardization

Id	Cushion Color	Cushion Diameter *(in cm)*	Cushion Length *(in mm)*	Cap ~~Color~~	Cap diameter *(in cm)*	Cap Length *(in mm)*	Tail Color	Tail Length *(in cm)*	Target
1	Black	0.8	3.1	1000	1.4	4.4	0	3.2	Black
2	Black	5.8	3.6	1001	1.2	4.6	0	3.1	Black
3	Blue	4.3	3.2	1005	1.9	4.2	1	3.0	Blue
4	Blue	1.6	3.9	1000	1.2	4.5	1	3.5	Blue
5	Blue	9.3	3.0	1004	1.5	4.0	1	3.2	Blue
6	Black	4.2	3.4	1003	1.0	4.2	0	3.5	Black
7	Black	3.7	3.8	1002	1.2	4.3	0	3.6	Black
8	Blue	2.4	3.4	1003	1.5	4.1	1	3.8	Blue
9	Blue	2.0	3.2	1000	1.2	4.0	1	3.2	Blue

Summary

- Dataset
- Load dataset — In-built, URL, File
- Need & Objectives of Preprocessing
- Preprocessing Steps
  - ✓ Drop duplicate rows
  - ✓ Drop columns with only-one/less unique values
  - ✓ Drop columns with NULL values
  - ✓ Drop rows with NULL values
  - ✓ Drop redundant columns
  - ✓ Handle missing values
  - ✓ Investigate categorical columns
  - ✓ Feature scaling

Feature Engineering

- Process of transforming raw data into features that better represent the underlying problem to the machine learning algorithm

CITY 1 LAT.	CITY 1 LNG.	CITY 2 LAT.	CITY 2 LNG.	Driveable?
123.24	46.71	121.33	47.34	Yes
123.24	56.91	121.33	55.23	Yes
123.24	46.71	121.33	55.34	No
123.24	46.71	130.99	47.34	No

=>

DISTANCE (MI.)	Driveable?
14	Yes
28	Yes
705	No
2432	No

- Exceptionally difficult for a machine learning algorithm to learn the relationships between these four attributes and the class label
- Compute the distance between the source and destination and use it as a feature
- Results in improved model accuracy on unseen data

Feature Engineering

- Feature engineering is when you use your knowledge about the data to create fields that make a machine learning algorithm work better
- Engineered features that enhance the training provide information that better differentiate the patterns in the data
- Should strive to add a feature that provides additional information that is not clearly captured or easily apparent in the original or existing feature set

Feature Engineering

- Decompose Categorical Attributes
  - Item_Color -> Red, Blue, Unknown
  - Binary Feature -> Has_Color
  - Binary Features -> Is_red, Is_Blue, Is_Unknown
- Decompose Date and Time
  - 2014-09-20T20:45:40Z
  - Numerical Feature -> Hour_of_Day
- Reframe Numerical Quantities
  - Transform into a new unit or the decomposition into multiple components
  - Quantity like weight, distance, timing
  - A linear transform may be useful to regression and other scale dependent

Thank You