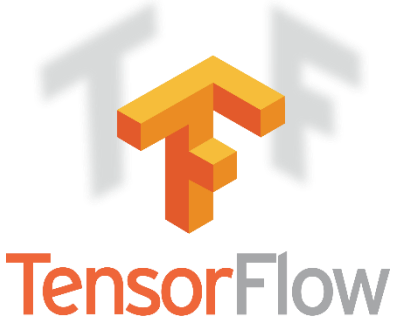
Data Processing

Most Commonly Used Libraries



Getting Started with Python Program

DataFrame

- Two-dimensional size-mutable, potentially heterogeneous tabular data structure with labeled axes (rows and columns).

Col1	Col2	Col3	.	.	.	Col n
Row 1						
Row 2						
….						

DataFrame

- Two-dimensional size-mutable, potentially heterogeneous tabular data structure with labeled axes (rows and columns).

```
Loan_ID,Gender,ApplicantIncome,CoapplicantIncome,LoanAmount,Area,Loan_Status
LP001002,,5849,0,,urban,Y
LP001003,Male,4583,,128,semi,N
LP001005,Male,3000,0,66,,Y
LP001006,Female,2583,2358,120,semi,Y
LP001008,Male,,0,141,urban,Y
LP001011,Male,5417,4196,267,semi,Y
LP001013,Male,2333,1516,,semi,Y
LP001014,Female,3036,2504,158,semi,N
LP001018,Male,4006,1526,168,semi,Y
LP001020,Male,12841,10968,349,semi,N
LP001024,Female,3200,700,70,urban,Y
LP001027,Male,2500,1840,109,urban,Y
LP001028,Female,,8106,,urban,Y
LP001029,Male,1853,2840,114,urban,N
LP001030,Male,1299,1086,17,semi,Y
LP001032,Male,4950,0,125,semi,Y
```

DataFrame

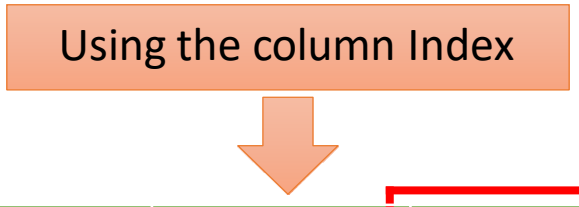
- Two-dimensional size-mutable, potentially heterogeneous tabular data structure with labeled axes (rows and columns).

```
1 Loan_ID,Gender,ApplicantIncome,CoapplicantIncome,LoanAmount,Area,Loan_Status
2 LP001002,,5849,0,,urban,Y
3 LP001003,Male,4583,,128,semi,N
4 LP001005,Male,3000,0,66,,Y
5 LP001006,Female,2583,2358,120,semi,Y
6 LP001008,
7 LP001011,
8 LP001013,Male,2333,1516,,semi,Y
9 LP001014,Female,3036,2504,158,s
10 LP001018,Male,4006,1526,168,semi,
11 L
12 L
```

pd.read_csv("student_data.csv")

	0	1	2	3	4	5	6
Index	**Loan_ID**	**Gender**	**ApplicantIncome**	**CoapplicantIncome**	**LoanAmount**	**Area**	**LoanStatus**
0	LP001002		5849	0		Urban	Y
1	LP001003	Male	4583		128	Semi	N
2	LP001004	Male	3000	0	66		Y

DataFrame – Select the data



Index	0	1	2	3	4	5	6
	ID	Name	Age	Study	Subject	Tests Taken	Marks
0	ST01	Jitesh	16	4	Math	0	70
1	ST02	John	17	7	Physics	1	89
2	ST03	Alia	16	5	Math	1	77

iloc [row range, column range]

iloc [from_row_index : to_row_index+1, from_column_index : to_column_index+1]

iloc [0 : 2, 2:5]

DataFrame – Select the data

Using the column names

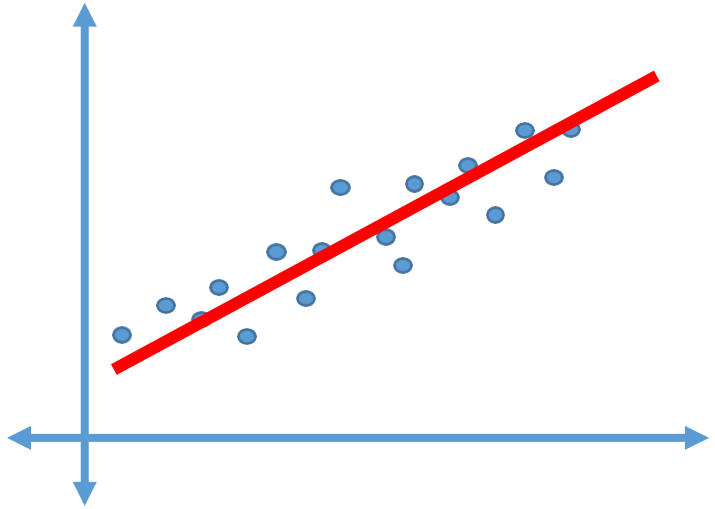
Index	ID	Name	Age	Study	Subject	Tests Taken	Marks
0	ST01	Jitesh	16	4	Math	0	70
1	ST02	John	17	7	Physics	1	89
2	ST03	Alia	16	5	Math	1	77

dataframeName [[“Age”, “Study”, “Subject”]] [: 2]

Categorical Variables

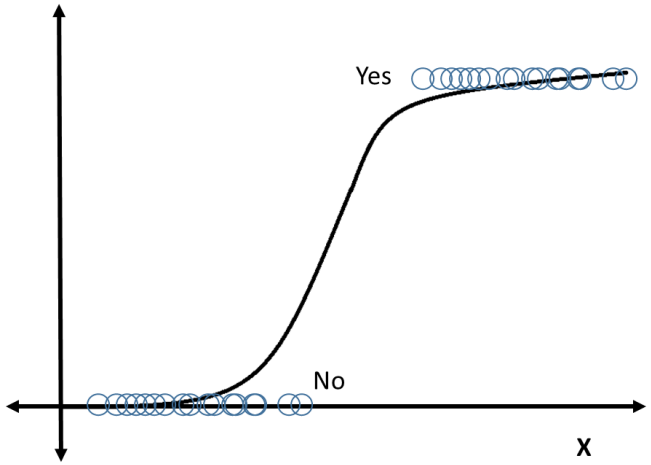
Mathematics as basis of Machine Learning

Regression



$$Y = a + bX$$

Classification



$$\text{Log}\left(\frac{P}{1-P}\right) = b0 + b1X$$

Categorical Variables

Loan_ID	Gender	ApplicantIncome	oapplicantIncom	LoanAmount	Area	Loan_Status
LP001002	Male	5849.00	0.00	nan	urban	Y
LP001003	Male	4583.00	nan	128.00	semi	N
LP001005	Male	00.00	0.00	66.00	semi	Y
LP001006	Female	2 00	2358.00	120.00	semi	Y
LP001008	Male	nan	0.00	141.00	urban	Y
LP001011	Male	5417	4196.00	267.00	semi	Y
LP001013	Male	2		n	semi	Y
LP001014	Female	3		8.00	semi	N
LP001018	Male	4		8.00	semi	Y
LP001020	Male	12841.00	10968.00	349.00	semi	N
LP001024	Female	3200.00	700.00	70.00	urban	Y
LP001027	Male	2500.00	1840.00	109.00	urban	Y
LP001028	Female	nan	8106.00	nan	urban	Y
LP001029	Male	1853.00	2840.00	114.00	urban	N
LP001030	Male	1299.00	1086.00	17.00	semi	Y
LP001032	Male	4950.00	0.00	125.00	semi	Y

Male → 0
Female → 1

Y → 0
N → 1

Urban → 0
Semi → 1
Rural → 2

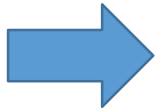
LabelEncoder

```
22
23 # Categorical to Numeric label encoding using Pandas
24 dt.dtypes
25
26 dt[cols] = dt[cols].astype('category')
27 dt.dtypes
28
29 for columns in cols:
30     dt[columns] = dt[columns].cat.codes
```

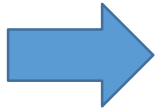
LP001002	1	5849.00	0.00	140.92	1	1
LP001003	1	4583.00	2509.33	128.00	0	0
LP001005	1	3000.00	0.00	66.00	0	1
LP001006	0	2583.00	2358.00	120.00	0	1
LP001008	1	4103.57	0.00	141.00	1	1
LP001011	1	5417.00	4196.00	267.00	0	1
LP001013	1	2333.00	1516.00	140.92	0	1
LP001014	0	3036.00	2504.00	158.00	0	0
LP001018	1	4006.00	1526.00	168.00	0	1
LP001020	1	12841.00	10968.00	349.00	0	0
LP001024	0	3200.00	700.00	70.00	1	1
LP001027	1	2500.00	1840.00	109.00	1	1
LP001028	0	4103.57	8106.00	140.92	1	1
LP001029	1	1853.00	2840.00	114.00	1	0
LP001030	1	1299.00	1086.00	17.00	0	1
LP001032	1	4950.00	0.00	125.00	0	1

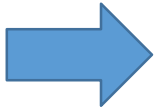
Problem with LabelEncoder ONLY

Area	LabelEncoder
Urban	1
Semi-Urban	2
Rural	3

3 > 2 > 1  Rural > Semi-Urban > Urban

City	LabelEncoder
London	1
New York	2
Delhi	3

1 + 2 = 3  London + New York = Delhi

(1 + 3)/2 = 2  (London + Delhi)/2 = New York

One-Hot Encoder

City	LabelEncoder	London	New York	Delhi
London	1	1	0	0
New York	2	0	1	0
Delhi	3	0	0	1

Data Normalization

What is Normalization?

"In the simplest cases, normalization of ratings means adjusting values measured on different scales to a notionally common scale, often prior to averaging"

--Wikipedia

What is Normalization?

City	Temperature
New York	92 °F
Chicago	87 °F
Boston	94 °F
Detroit	91 °F

City	Temperature
London	28 °C
Paris	24 °C
Delhi	34 °C
Tokyo	31 °C

City	Temperature
New York	92 °F
Chicago	87 °F
Boston	94 °F
Detroit	91 °F
London	82.4 °F
Paris	75.2 °F
Delhi	93.2 °F
Tokyo	87.8 °F

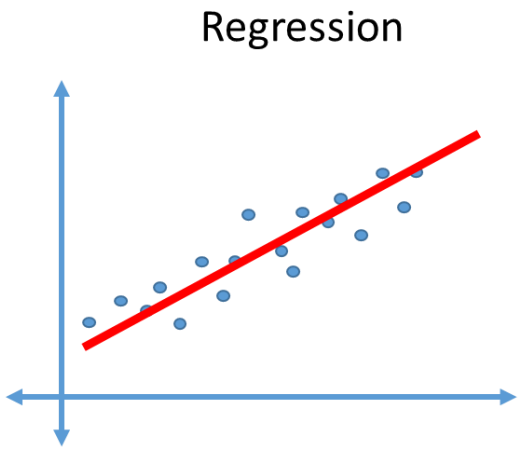
City	Temperature
New York	33.3 °C
Chicago	30.5 °C
Boston	34.4 °C
Detroit	32.8 °C
London	28 °C
Paris	24 °C
Delhi	34 °C
Tokyo	31 °C

What is Normalization?

"In the simplest cases, normalization of ratings means adjusting values measured on different scales to a notionally common scale, often prior to averaging"

--Wikipedia

Why should we normalize the data?

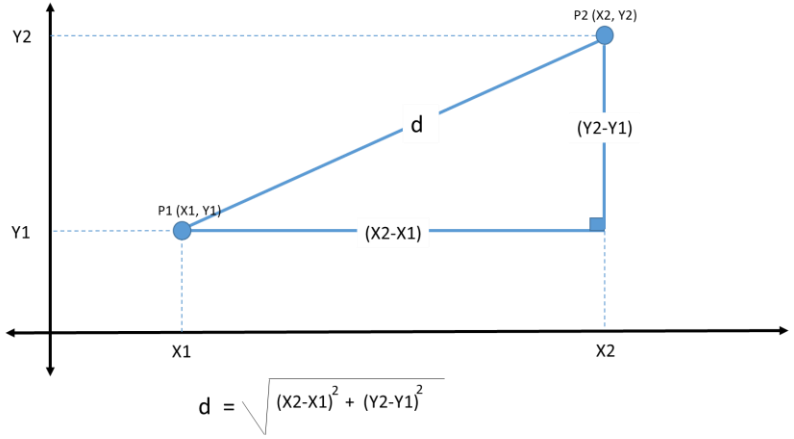


$$Y = a + bX1 + cX2$$

LoanTerm	Salary
12	$ 68,000
60	$ 85,000
120	$ 115,000

Why should we normalize the data?

Clustering



Euclidean Distance

Normalization Defined

- A method to standardise the range of independent variables or features of data
- Variables are fitted within a certain range (Generally between 0 and 1)
- Applied on numeric columns

Normalize data – Transformation Methods

ZScore

$$Z = \frac{X - mean(x)}{stdev\ (x)}$$

MinMax

$$Z = \frac{X - min(x)}{Max\ (x) - min(x)}$$

Logistic

$$Z = \frac{1}{1 + exp(-x)}$$

Most commonly used transformation methods

Train and Test

Train and Test Data