Data Generation using Modelling & Simulation for Machine Learning

Dr. Prashant Singh Rana

Assistant Professor,

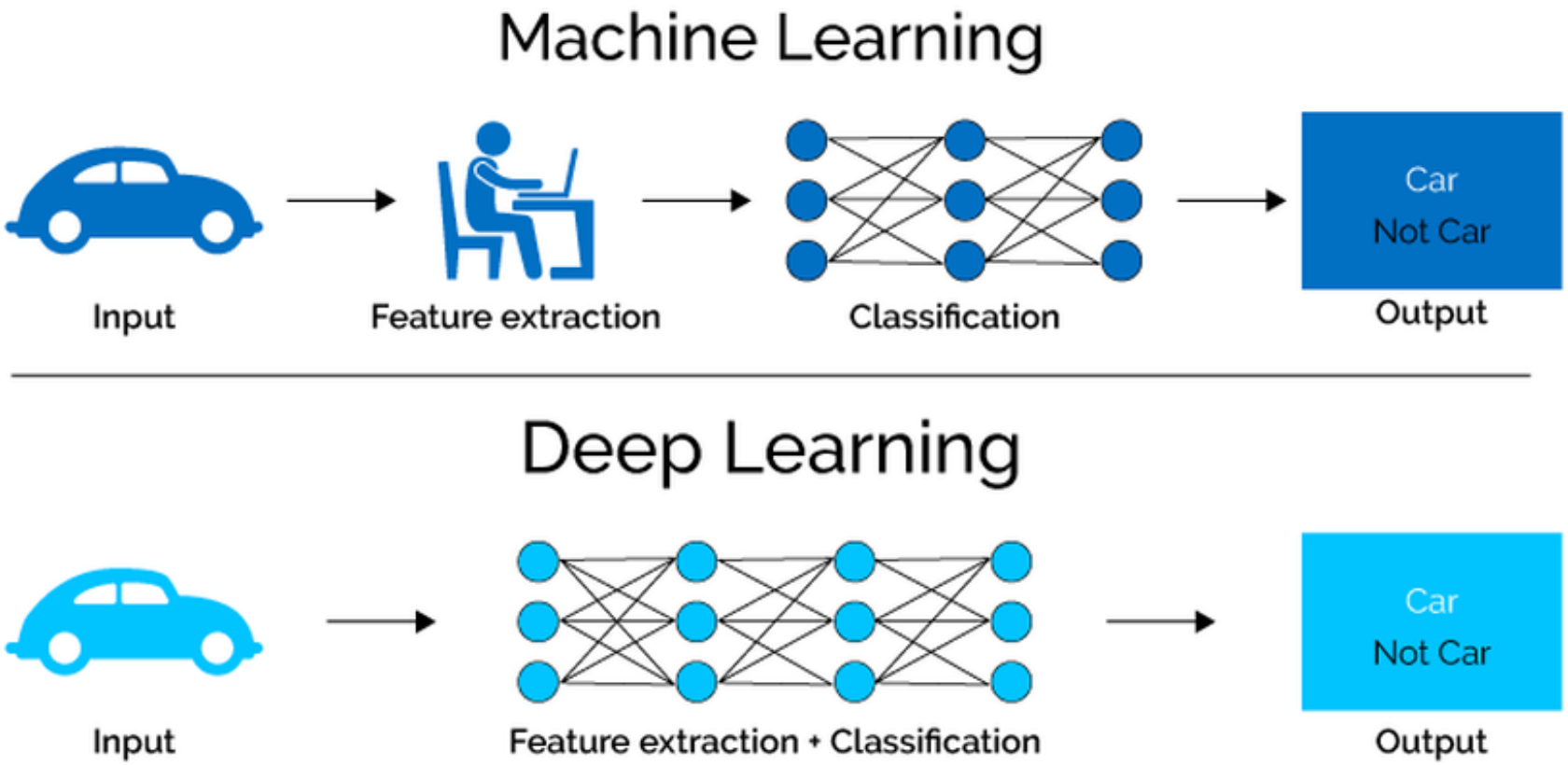
Computer Science and Engineering Department,

Thapar Institute of Engineering and Technology,

Patiala, Punjab.

www.psrana.com | psrana@gmail.com

Difference between Machine Learning & Deep Learning ??


Machine Learning
Input
Feature extraction
Classification
Car
Not Car
Output
Deep Learning
Input
Feature extraction + Classification
Car
Not Car
Output

What is Structured, Semi-Structured and Unstructured Data??

Unstructured Data

- **No format is defined**
- **Example:**
  - **1000 movies in a folder**
  - **10k docs in a folder**
  - **500 images in a folder**
  - **…….. many more**

Structured Data

- **Stored in well defined format**
- **Example: Song**

Song ID	Language	Genre	Singers	Likes	Dislikes	.	.
1	En	Rock	2	10k	1k	.	.
2	En	Jazz	2	11k	1.5k	.	.
3	Hi	Pop	3	20k	2k	.	.
4	Hi	Jazz	1	15k	1.2k	.	.
.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.

Semi-structured Data

- **Little format is defined**
- **Example:**
  - **1000 movies in three folder (12+, 16+, 18+)**
  - **10k docs in 4 folder (A, B, C, D)**
  - **500 photos in two folder (Male, Female)**
  - **…….. many more**

Data Set format for Machine Learning

Data Set format for Machine Learning

- **Multiple columns and one column is labelled (Strength)**

x1	x2	x3	x4	x5	Strength
17	0	-5	0.784245	37	26
12	0	-10	0.587296	25	27
18	0	-7	0.876622	40	25
11	0	-7	0.80826	24	23
18	0	-4	0.83215	37	28
10	1	-9	0.62842	27	28
19	0	7	0.522811	44	30
19	-1	4	0.548609	37	23
15	0	-6	0.177904	46	20

Biggest issue/ hurdle/ problem in Machine Learning ??

Availability of Structured Dataset….

Solution

Data Generation using Simulations……

What is Simulation ??

What is Simulation ??

Executing real world process in controlled environment….

Dam.....

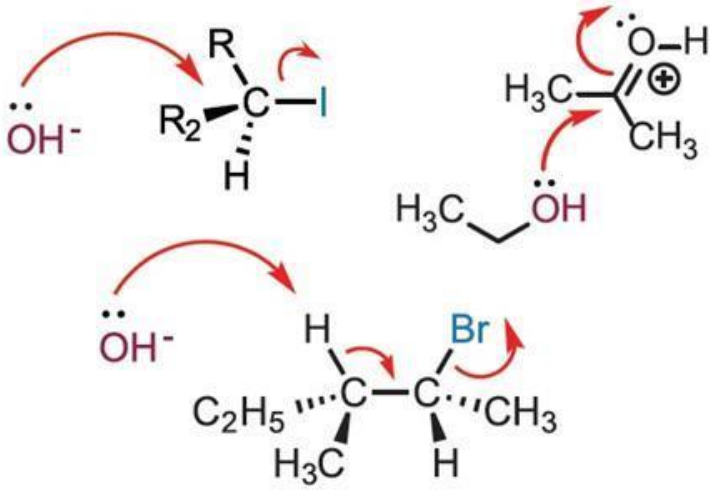
Dam.....

Dam parameters

- **Height of wall**
- **Width of wall**
- **Water Quantity**
- **Quality of concrete used**
- **Water flow**
- **Number of Gates**
- **…….. many more**

Car Crash.....

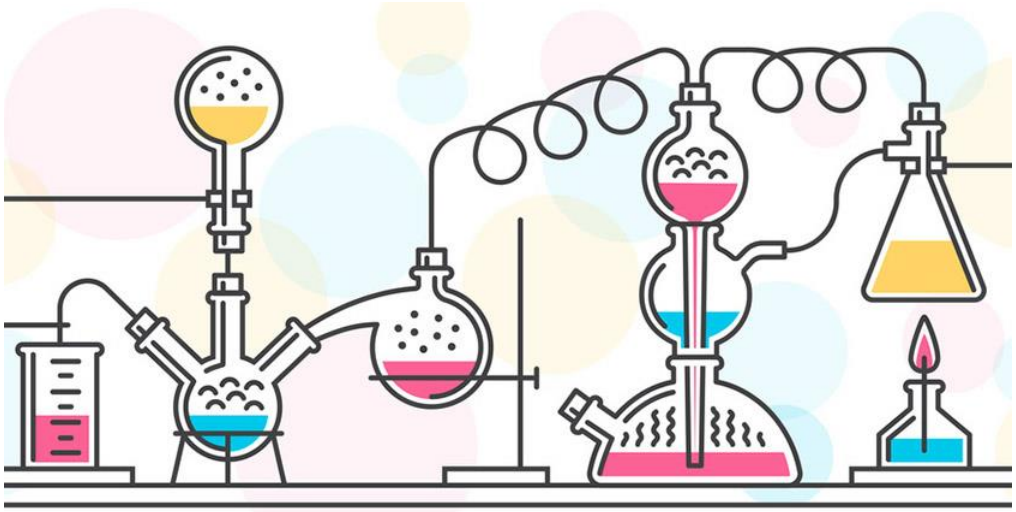
Chemical Reaction.....



Parameters

-Temperature

-Pressure



What is this ?

Testing of Bullet Proof Jacket

One bullet proof jacket cost ~2 Lac

Simulations Software

Google it "List of computer simulation software"

https://en.wikipedia.org/wiki/List_of_computer_simulation_software

WIKIPEDIA
The Free Encyclopedia

Main page
Contents
Featured content
Current events
Random article
Donate to Wikipedia
Wikipedia store

Interaction

Help
About Wikipedia
Community portal
Recent changes
Contact page

Tools

What links here
Related changes
Upload file
Special pages
Permanent link

Article | Talk

List of computer simulation software

From Wikipedia, the free encyclopedia

The following is a list of notable computer simulation software.

Contents [hide]

1 Free or open-source
2 Proprietary
3 See also
4 References

Free or open-source [edit]

- Advanced Simulation Library - open-source hardware accelerated multiphysics simulatio
- Algodoo - 2D physics simulator
- ASCEND - open-source equation-based modelling environment.
- Cantera - chemical kinetics package
- Celestia - a 3D astronomy program.
- CP2K - Open-source ab-initio molecular dynamics program
- DWSIM - an open-source CAPE-OPEN compliant chemical process simulator.
- Elmer - an open-source multiphysical simulation software for Windows/Mac/Linux.

Simulations Software

Software Domains:

- molecular dynamics
- astronomy
- chemical kinetics
- 2D physics simulator
- scientific prototyping
- network simulator
- Biomechanical
- algebra and geometry
- medical simulation
- transportation and environmental planning
- …….. many more

Example: Dam

library (dam) or import dam

Strength = Sim(x1, x2, x3, x4, x5)

	A	B	C	D	E	F	G
1	[10,20]	[-1,1]	[-10,10]	[0,1]	[20,50]		
2	x1	x2	x3	x4	x5	Strength	
3	17	0	-5	0.784245	37	26	1
4	12	0	-10	0.587296	25	27	1
5	18	0	-7	0.876622	40	25	1
6	11	0	-7	0.80826	24	23	0
7	18	0	-4	0.83215	37	28	1
8	10	1	-9	0.62842	27	28	1
9	19	0	7	0.522811	44	30	1
10	19	-1	4	0.548609	37	23	0
11	15	0	-6	0.177904	46	20	0

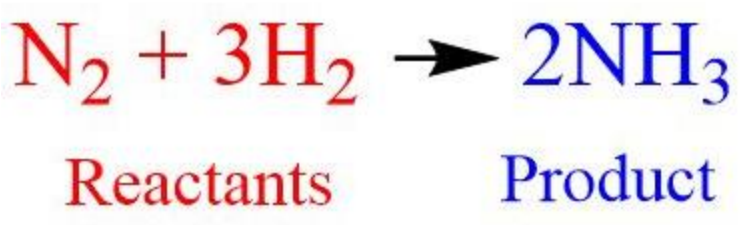
Cantera

Chemical Kinetics Package

Cantera

library (Cantera) or import Cantera

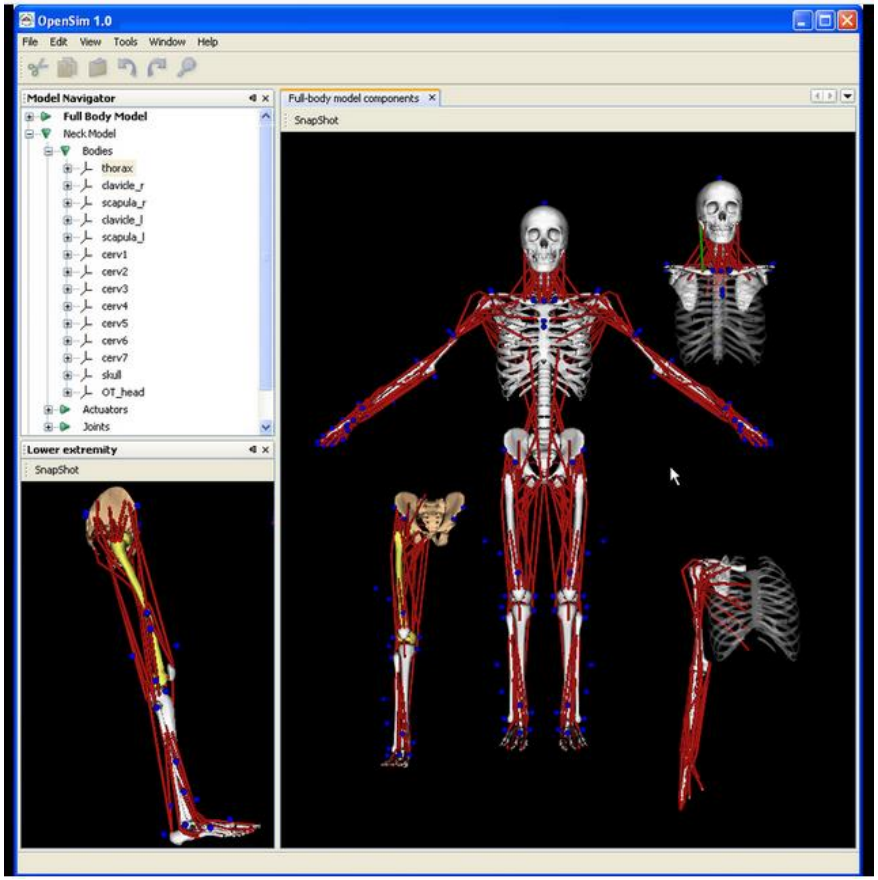
Rate = Cantera(Temp, Pressure)



	A	B	C
1	[-40, 40]	[10, 50]	
2	Temp	Pressure	Rate of Reaction
3	-29	18	10
4	-5	11	6
5	25	13	3
6	-32	34	6
7	-25	38	8
8	26	25	6
9	9	11	2
10	-3	44	7

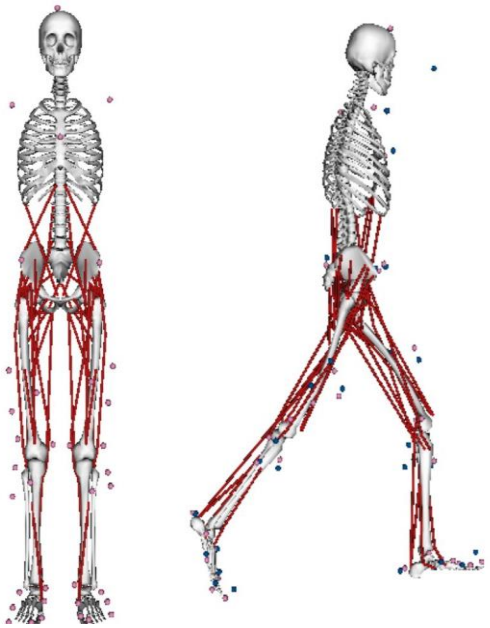
OpenSim

Software System for Biomechanical Modeling.



Opensim

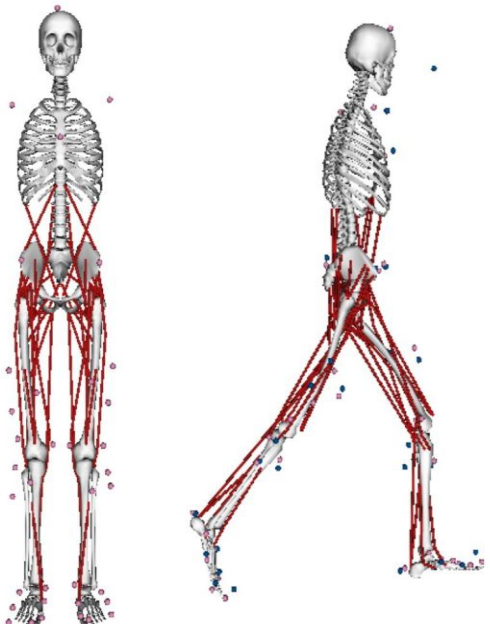
library (OpenSim) or import OpenSim
(F1,F2,F3,F4) = OpenSim(A, H, W, S)



	A	B	C	D	E	F	G	H	I
1	[10,80]	[100,200]	[40,120]	[2,4]					
2	Age	Height	Weight	Speed		F1	F2	F3	F4
3	50	102	68	4		44	50	39	26
4	58	162	62	3		21	37	24	39
5	36	148	120	2		39	30	44	42
6	23	158	98	4		50	48	44	46
7	41	183	64	2		23	27	23	50
8	17	182	106	2		22	27	43	33
9	42	138	82	4		38	25	42	30
10	49	104	69	4		44	27	22	33
11	34	114	112	3		26	27	23	30

Opensim

library (OpenSim) or import OpenSim
(F1,F2,F3,F4) = OpenSim(A, H, W, S)



	A	B	C	D	E	F	G	H	I
1	[10,80]	[100,200]	[40,120]	[2,4]					
2	Age	Height	Weight	Speed		F1	F2	F3	F4
3	50	102	68	4		44	50	39	26
4	58	162	62	3		21	37	24	39
5	36	148	120	2		39	30	44	42
6	23	158	98	4		50	48	44	46
7	41	183	64	2		23	27	23	50
8	17	182	106	2		22	27	43	33
9	42	138	82	4		38	25	42	30
10	49	104	69	4		44	27	22	33
11	34	114	112	3		26	27	23	30

Case Study - I

Title: Audio Genre Classification

- **GTZAN Audio** Dataset contain 696 audio files classified in 5 Genre (Rock, Pop, Metal, Country and Jazz)
- Python library "**librosa**" is used to extract the features of each song.
- **Number of features: 193**

S. No.	Feature Group	Number of Features
1	MFCCS	40
2	Chroma	12
3	Mel	128
4	Contrast	7
5	Tonnetz	6
	Total	**193**

Case Study - I

- Pass every song to the "**librosa**" library, create feature table and table them.

FEATURE TABLE

AUDIO NO.	F1	F2	F3	F4	F5	—	F189	F190	F191	F192	F193	LABEL
1	-113.571	121.5718	-19.1681	42.36642	-6.36466	—	0.009556	0.010512	-0.02046	0.001493	-0.00643	pop
2	-207.502	123.9913	8.955128	35.87765	2.907321	—	0.018907	0.070679	0.014551	0.009352	-0.00866	pop
.												
.						—						
98	-20.4705	53.68523	5.986029	10.14361	17.07146	—	0.010361	0.024009	-0.03452	0.004169	-0.00781	classical
99	-58.9489	68.86537	-8.46514	3.622923	5.078615	—	-0.01272	0.001894	0.026377	0.004	-0.00349	classical
.												
.						—						
357	-108.521	69.97168	14.8811	45.51458	-5.37834	—	-0.0032	0.007805	0.022346	-0.00182	0.001521	rock
358	-226.288	78.31248	7.799703	53.84219	-1.16246	—	0.003306	-0.00244	0.070924	0.006931	0.000406	rock
.												
.						—						
547	-100.384	104.6881	-57.2479	56.5685	-5.5517	—	-0.00487	0.029969	-0.01257	0.002505	0.003393	metal
548	-93.5559	89.86496	-55.8847	51.63797	-5.57456	—	0.009625	0.040979	0.057395	-0.01102	0.006867	metal
.												
.						—						
695	-111.547	85.55908	3.526411	16.37183	2.21108	—	0.03479	-0.01803	0.044246	0.000682	0.014066	hip-hop
696	-63.5241	79.02744	42.74856	16.09584	15.27049	—	0.027467	0.035715	-0.04266	-0.00558	0.013449	hip-hop

- Apply classification models.

Big Question??

If simulator is available then why to generate data and use Machine Learning??

Answer:

Simulation took huge time for single simulation.

1hr, 12hr, 1Day, 10Days, ...

Basics Idea for Data Generation

Basics Idea for Data Generation

1. **Find a suitable tool/ software/ library/ package.**
2. **Install it and try to run it**
3. **Study different parameters [Most Imp]**
4. **Generate random set of parameters, pass to the simulator and record the value.**

Random Parameter Generation

Random Parameter Generation

Variables and its range:

x1 = [10, 100], x2 = [-9, 0]

x3 = [-50, -10], x4 = [10, 50]

Outcome:

Generate 100 rows and save to a file:

x1	x2	x3	x4
76	-3	-16	36
51	-7	-29	15
29	0	-49	19
45	-4	-43	16

Parameter Generation

```
import random as r
fp=open("result.csv","w")   # Open the file in writing mode
fp.write("x1,x2,x3,x4\n")
for  i in range(100):
        x1=r.randint(10,100)
        x2=r.randint(-9,0)
        x3=r.randint(-50,-10)
        x4=r.randint(10,50)
        s="%d,%d,%d,%d\n"%(x1,x2,x3,x4)
        fp.write(s)      # Writing to the file line by line
fp.close()
print ("Done !! \nOpen result.csv to view the content")
```

Objective Function

Objective Function

A function take some parameters and return single or multiple values. Example:

```
Sim (x1, x2, x3, x4, x5):
  // Perform simulation
    return value
```

Objective Function

Write the Python Code for

Minimize the Schewel Function

$$f_5(x) = \sum_{i=1}^{D} |x_i| + \prod_{i=1}^{D} |x_i|$$

D = 10

Range = {-10, 10}

Output: List parameters x0, x1, ... x9 and Min value.

Solution (version 1)

```
#Write a python code and compile it
def FitnessFunction (x):
    s=0
    p=1
    for i in range(0, D):
        s = s + abs(x[i])
    for i in range(0, D):
        p= p * (x[i])
    return s + p
```

Solution (version 2)

```
#Write a python code and compile it
def FitnessFunction (x):
    s=0
    p=1
    for i in range(0, D):
        s = s + abs(x[i])
        p= p * (x[i])
    return s + p
```

Calculate fitness for

#……write after the above code

```
D = 5
C1=[1,4,2,3,5]
C2=[2,3,2,3,5]
C3=[1,1,1,5,1]
C4=[1,1,1,1,1]
print (“C1”, FitnessFunction(C1))
print (“C2”, FitnessFunction(C2))
print (“C3”, FitnessFunction(C3))
print (“C4”, FitnessFunction(C4))
```

Objective Function 2

Write the Python Code for

Minimize the Bohachevsky Function

$$f(\boldsymbol{x}) = x_1^2 + 2x_2^2 - 0.3\cos(3\pi x_1) - 0.4\cos(4\pi x_2) + 0.7$$

D = 2

Range = {-100, 100}

Output: List parameters x1 and x2 and Min value.

Solution

#Write a python code and compile it

```
import random as r
def FitnessFunction (x):
    s=x[0]**2 + 2*x[1]**2 –
        0.3*m.cos(3*m.pi*x[0] ) –
        0.4*m.cos(4*m.pi*x[1] ) +
        0.7
     return s
```

Data Generation using Simulation

Write the Python program for

f(x) = x1 + x2 + x3 + x4 + x5 + x6 + x7 + x8

x1 = [10, 100],

x2 = [-9, 0]

x3 = [-50, -10],

x4 = [-20, -10]

x5 = [0, 1],

x6 = [0.1, 0.5]

x7 = [-1, 0],

x8 = [-5.0, -1.0]

Data Generation using Simulation

```
import math as m
def FitnessFunction (x):
    return sum(x)
```

Data Generation using Simulation

```
fp=open("dataSet.csv","w")# Open the file in writing mode
fp.write("Target,x1,x2,x3,x4,x5,x6,x7,x8\n")
for  i in range(100):
        x1=r.randint(10, 100) ;  x2=r.randint(-9, 0);
        x3=r.randint(-50, -10);   x4=r.randint(-20, -10);
        x5=r.random();              x6=r.uniform(0.1, 0.5);
        x7=r.uniform(-1,0);       x8=r.uniform(-5.0, -1.0);
        target=FitnessFunction (x1, x2, x3, x4, x5, x6, x7, x8)
        s="%f%d,%d,%d,%d,%f,%f,%f,%f\n"
                        %(target,x1,x2,x3,x4,x5,x6,x7,x8)
        fp.write(s)      # Writing to the file line by line
fp.close()
print ("Done !! \nOpen dataSet.csv to view the content")
```

Data Generation using Simulation

Output File

Target	x1	x2	x3	x4	x5	x6	x7	x8
-11.71	63	-5	-49	-20	0.95	0.05	-0.02	-1.68
22.79	57	-8	-10	-16	0.84	0.16	-0.15	-1.06
-15.42	21	-1	-21	-14	0.54	0.46	-0.19	-1.23
-34.09	36	-8	-44	-17	0.51	0.49	-0.19	-1.90
12.18	45	-2	-10	-20	0.11	0.11	-0.60	-0.44

Data Generation using Simulation

Case 1: OpenSim

```
import OpenSim
def FitnessFunction (x):
        force=OpenSim.sim(x)
        return force
```

Data Generation using Simulation

Case 2: OpenSim

 # Simulation result is saved in a file

```
import OpenSim
def FitnessFunction (x):
        OpenSim.sim(x)
      # force/result is saved in forceFile file
        force=read(forceFile)
        return force
```

Data Generation using Simulation

Case 3: Cantera

```
import cantera
def FitnessFunction (x):
        rate=cantera.sim(x)
        return rate
```

Call External Commands

How to call external commands ???

Call External Commands

In python

```
import os
os.system("calc")
os.system("notepad")
os.system("mspaint")
os.system("python Random.py")
os.system("python GA.py")
os.system("rscript decisionTree.R")
os.system("rscript randomForest.R")
```

Call External Commands

In R

```
system("calc")
system("notepad")
system("mspaint")
system("python Random.py")
system("python GA.py")
system("rscript decisionTree.R")
system("rscript randomForest.R")
```

Call External Commands

In Matlab / Octave

```
system("calc")
system("notepad")
system("mspaint")
system("python Random.py")
system("python GA.py")
system("rscript decisionTree.R")
system("rscript randomForest.R")
```

Call External Commands

In C/C++ / Java (not confirm)

```
system("calc")
system("notepad")
system("mspaint")
system("python Random.py")
system("python GA.py")
system("rscript decisionTree.R")
system("rscript randomForest.R")
```

How to use

How to use calling external commands ???

How to use Calling External Commands??

Task

1. Download 100 images of bike from “Google Images”
2. Resize all the images to 50%
3. Convert all the images to b/w (grey scale)
4. Zip all the files
5. Mail to email id

Develop an automated pipeline for above Task

How to use Calling External Commands??

Interface: Download bulk images

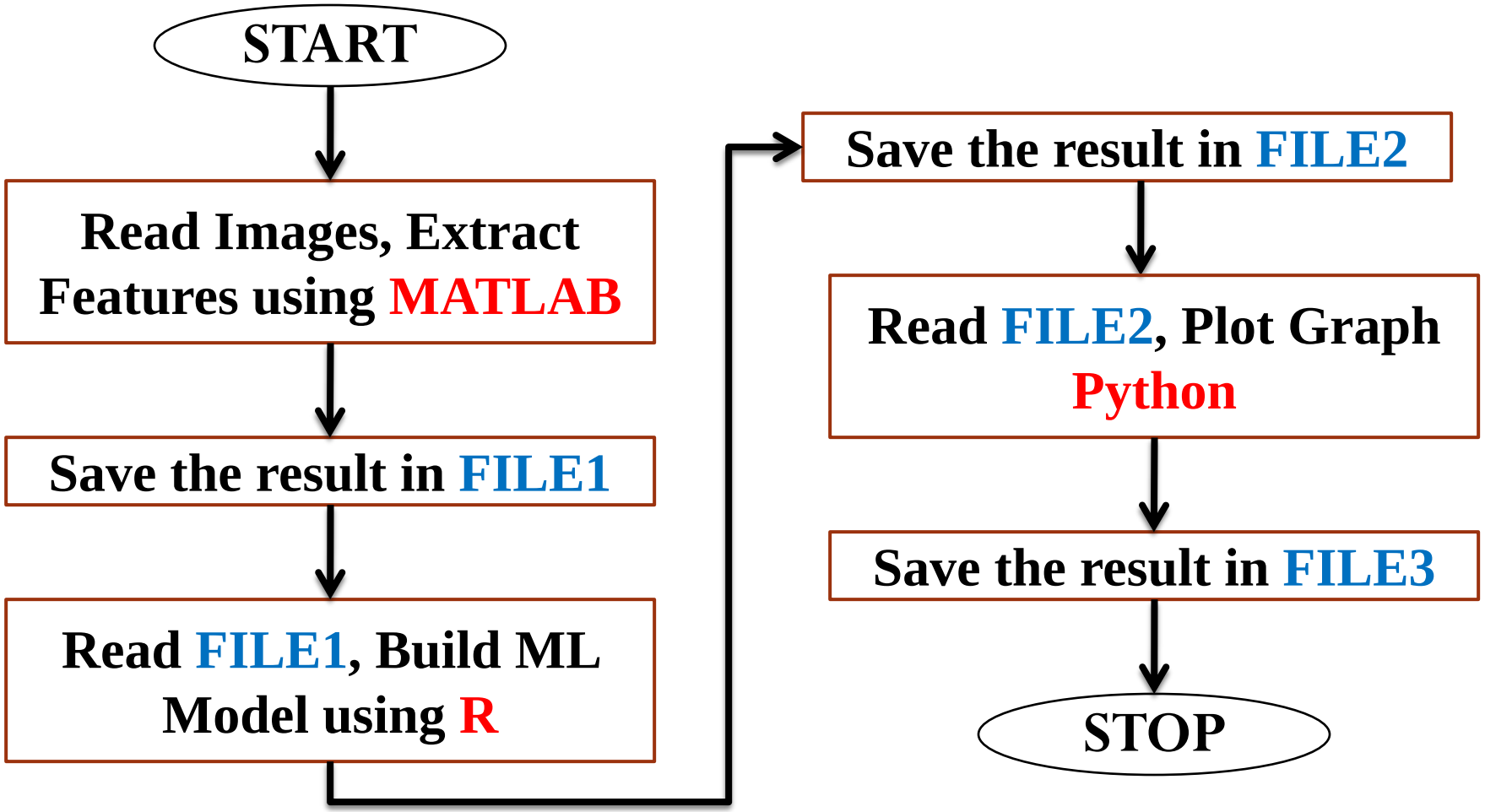
Data Generation using Simulation

Python Program: automate.py

```
import os
os.system ("python downloadGoogleImg.py")
os.system ("matlab resizeImg.m")
os.system ("matlab convertGrey.m")
os.system ("java zipImg")
os.system ("mailZipFile")
```

How to use Calling External Commands??

Example: Image Processing



ML Pipeline

```
# Program Name → modelBuilding.py
import os
# Phase I: Data Preprocessing
os.system("python readDataFile.py")
os.system("python removeDuplicate.py")
os.system("python removeOutlier.py")
os.system("python handleMissingValues.py")
os.system("python featureSelection.py")

# Phase II: Model Building and Result Analysis
os.system("python randomForest.py")
os.system("python decisionTree.py")
os.system("python svm.py")
os.system("python mergeResults.py")
```

Assignment 1

Develop an automated pipeline for below Task

Task

1. Download 50 videos of “sharry maan” from “Youtube”.
2. Convert all the videos to audio.
3. Cut first 2 minutes audios from all files.
4. Zip all the files
5. Mail to email id

Assignment 1

Develop an automated pipeline for below Task

Task

1. Download 50 videos of “sharry maan” from “Youtube”.
2. Convert all the videos to audio.
3. Cut first 2 minutes audios from all files.
4. Zip all the files
5. Mail to email id

Thanks

Learning by Doing

www.psrana.com | psrana@gmail.com