Nature Inspired Search & Optimization Algorithm

STOCHASTIC DIFFUSION SEARCH

Solving the “Curse of Dimensionality” in Machine Learning

Author: Sotiris Ftiakas - 3076

TABLE OF CONTENTS

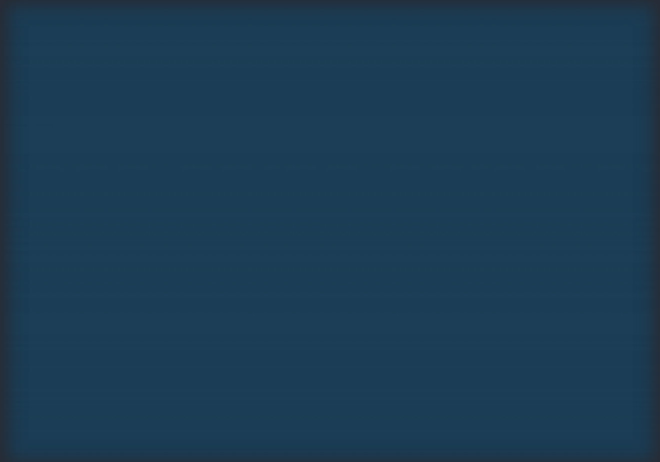
1

INTRODUCTION

What is Machine Learning and the Curse of Dimensionality?

MACHINE LEARNING

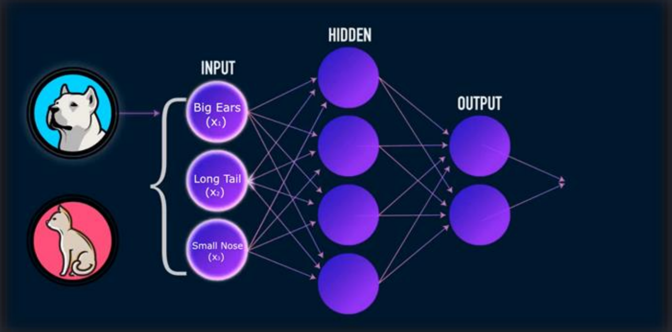
- Method of Data Analysis that automates predictive model building
- Computers learn from data, identify patterns and make decisions.
- Train, Predict, Improve
- More Data = More Accuracy
- Minimal human intervention



EXAMPLE – CAT VS DOG

- Supervised Learning – Labelled Data to train the model
- Input in a table format
- Rows = Examples, Columns = Features
- A model is simply a target function (f) that best maps input variables (X) to output variable (Y)

	FEATURES / INPUT VARIABLES			OUTPUT VARIABLE
	Big Ears (x1)	Long tail (x2)	Small Nose (x3)	Label (y)
1	1	1	0	DOG
2	0	1	1	CAT
3	0	0	0	DOG
4	1	1	1	DOG
5	0	1	0	CAT
6	0	1	1	CAT



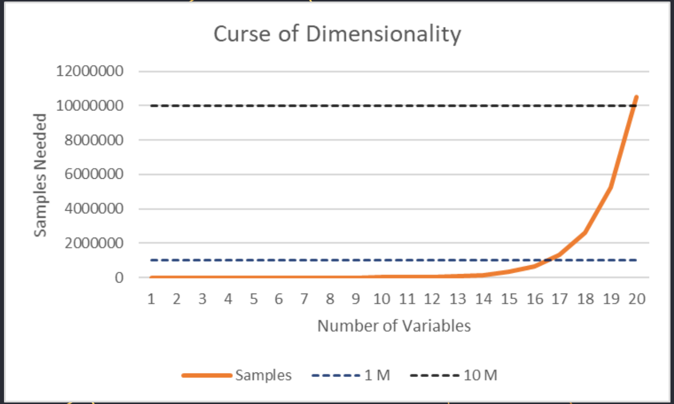
CURSE OF DIMENSIONALITY

"The problem caused by the exponential increase in volume associated with adding extra dimensions to Euclidean space"

—R. BELLMAN, 1957

CURSE OF DIMENSIONALITY

- In programming, this means that the error increases with the increase in the number of features
- Algorithms are harder to design in high dimensions, often having high running times
- Theoretically more information, practically higher possibility of noise and redundancy



Assume we want 10 samples per unique combination of variables:

1 Binary Variable → 2 Unique Combinations → 20 Samples

2 Binary Variables → 4 Unique Combinations → 40 Samples

.
.
.

k Binary Variables → 2^k Unique Combinations → 10 x 2^k Samples

2

ABOUT THE ALGORITHM

What is SDS and how does it work?

STOCHASTIC DIFFUSION SEARCH

- Proposed in 1989 as a population-based pattern-matching algorithm
- Uses a form of direct communication between agents
- Each agent poses a hypothesis about the possible solution and evaluates it partially
- Successful agents repeatedly test their hypothesis, while recruiting unsuccessful agents by direct communication
- Positive feedback mechanism - Agents converge onto promising solutions
- Global solution is constructed from agents forming the largest cluster.

STOCHASTIC DIFFUSION SEARCH

- Based on *partial evaluation* of fitness functions to save on the computational cost of repeated evaluations
- Still holds enough information for optimization purposes
- Variation and selection mechanisms in SDS solve the population homogeneity problem
- Wide *exploration* of all feasible solutions
- Detailed *exploitation* of a small number of them

3

SIMULATION

The Restaurant Game

SIMULATION

The Restaurant Game

"A group of agents attends a long conference in an unfamiliar town.

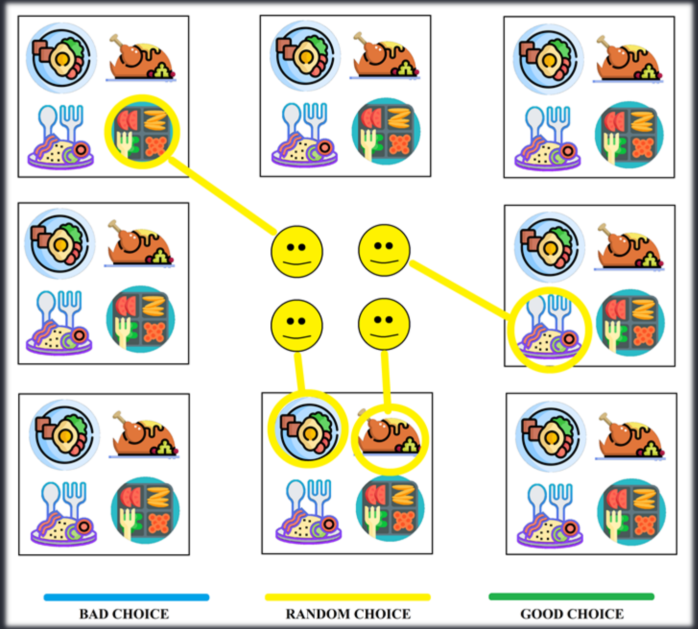
Each night they have to find somewhere to dine. There is a large choice of restaurants, each of which offers a large variety of meals.

The problem the group faces is to find the best restaurant, that is the restaurant where the maximum number of agents would enjoy dining. Even a parallel exhaustive search through the restaurant and meal combinations would take too long to accomplish.

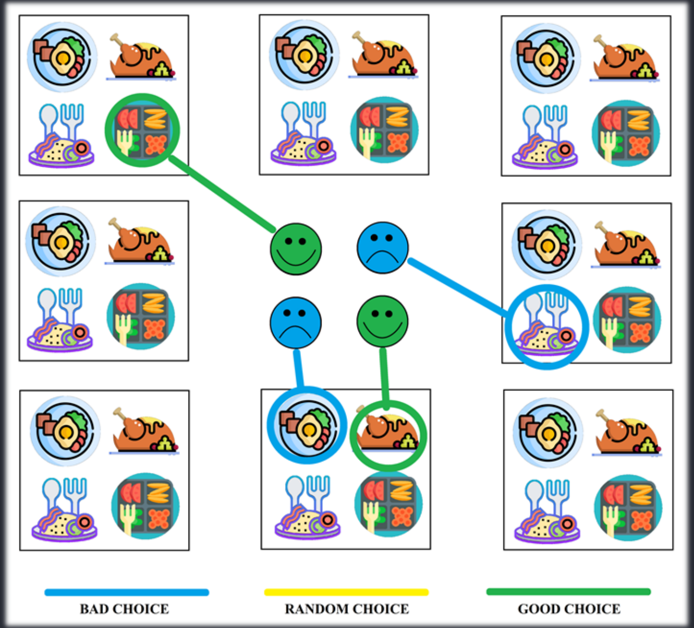
To solve the problem, agents decide to employ a Stochastic Diffusion Search."

THE RESTAURANT GAME – ITERATION 1

1) Initialization Phase

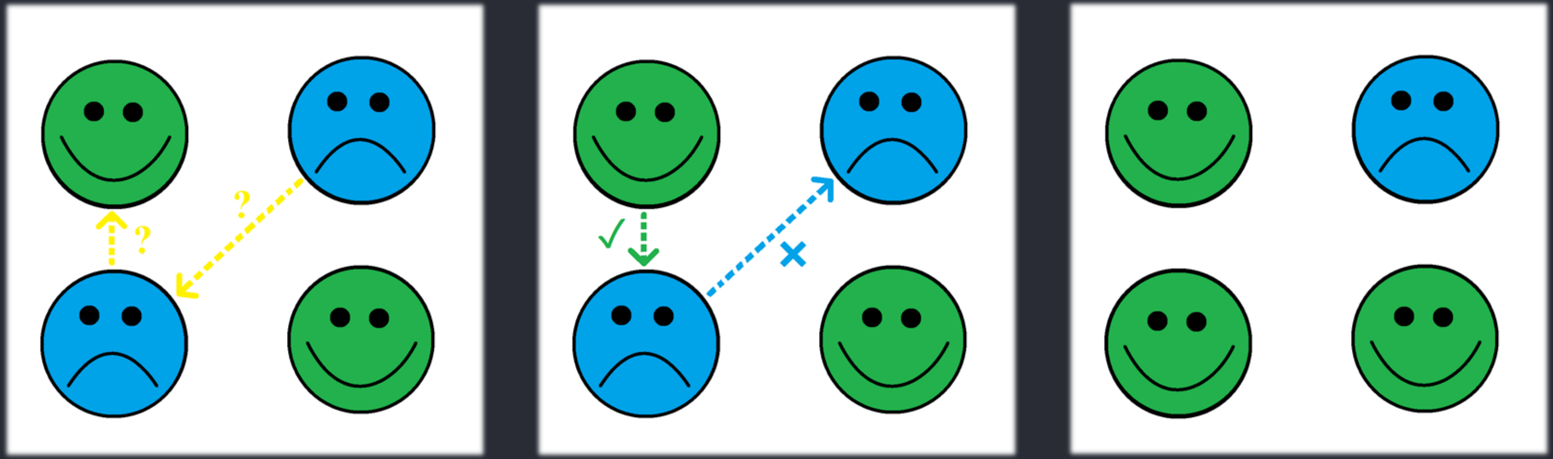


2) Testing Phase



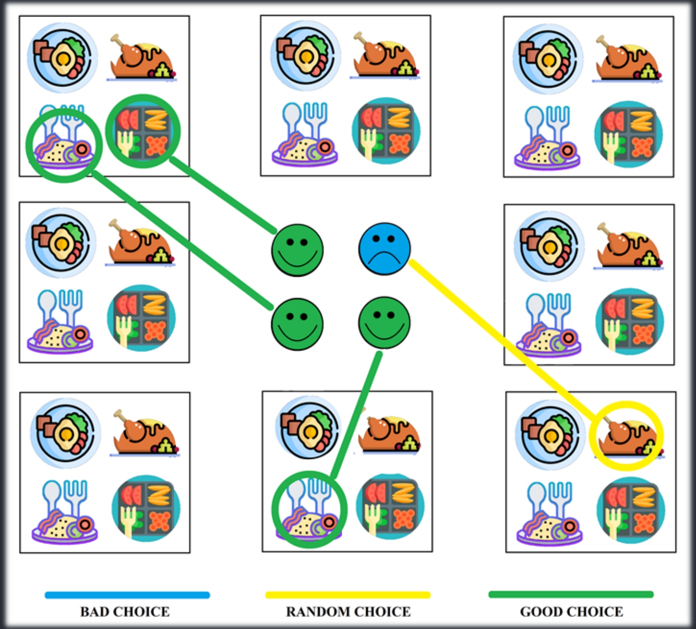
THE RESTAURANT GAME – ITERATION 1

3) Diffusion Phase

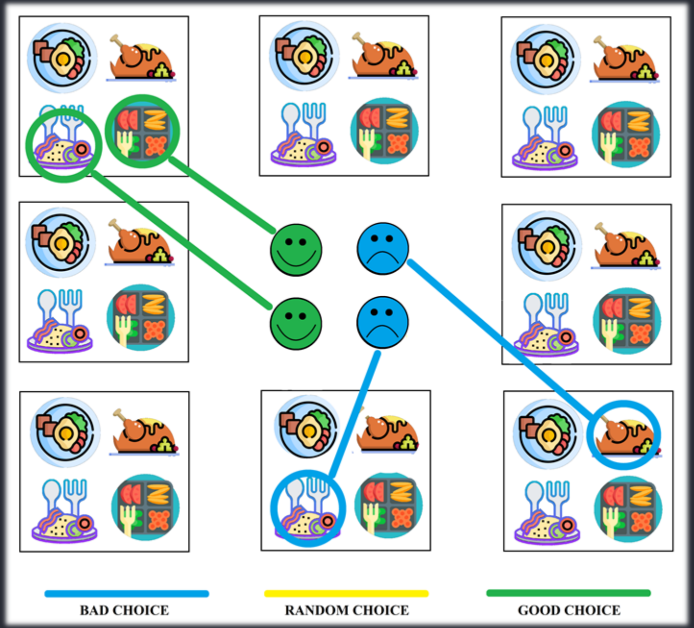


THE RESTAURANT GAME – ITERATION 2

1) *Initialization Phase*

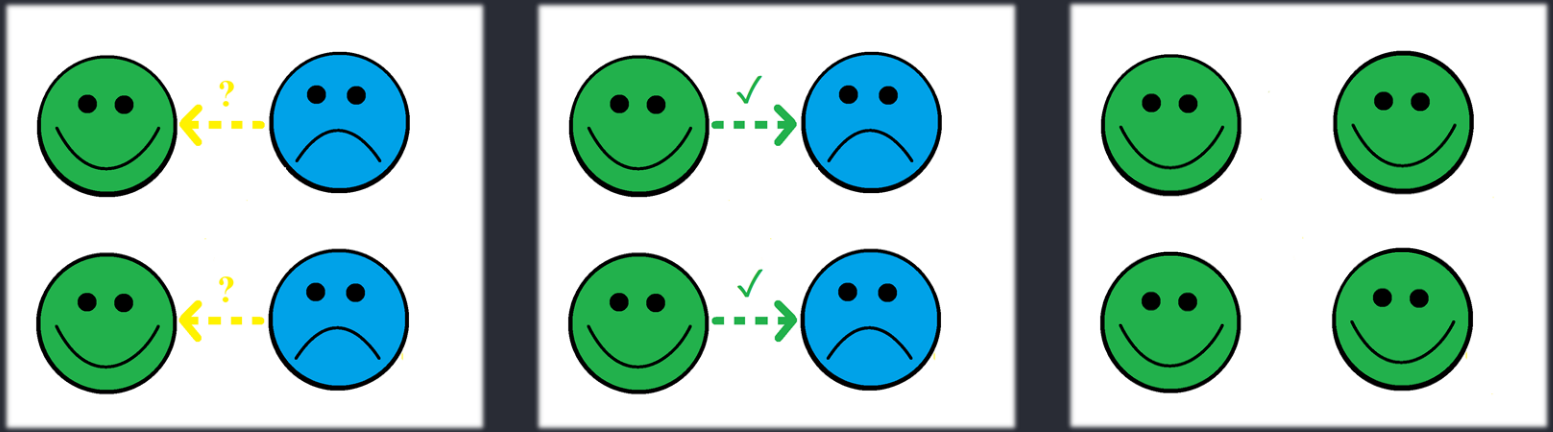


2) *Testing Phase*



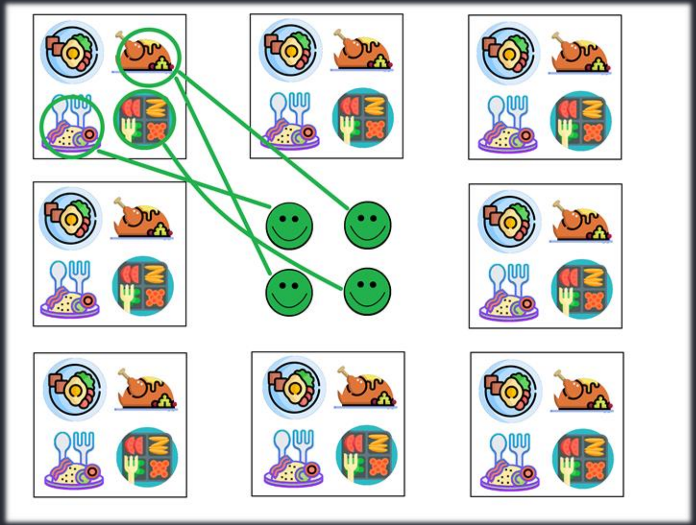
THE RESTAURANT GAME – ITERATION 2

3) *Diffusion Phase*



THE RESTAURANT GAME – ITERATION 3

1) *Initialization & Testing Phase*



2) *Halting Phase*

4

SDS IN THEORY

What is the theoretical background of SDS?

STOCHASTIC DIFFUSION SEARCH

- All feasible solutions to the problem form the *solution space* ***S***
- Each point in ***S*** has an associated *objective* value
- The objective values taken over the entire solution space form an *objective function* ***f***
- For simplicity, we assume that the objective is to *minimize* the sum of *n {0,1}-valued component functions* $\boldsymbol{f_i}$

$$\min_{\forall s \in \boldsymbol{S}} f(s) = \min_{\forall s \in \boldsymbol{S}} \sum_{i=1}^{n} f_i(s) , \qquad f_i : \boldsymbol{S} \rightarrow \{0.1\}$$

STOCHASTIC DIFFUSION SEARCH

- During operation, each agent maintains a hypothesis about the best solution to the problem
- A hypothesis is thus a candidate solution, and designates a point in the solution space
- Hypotheses can be binary strings, integer numbers or even real numbers

STOCHASTIC DIFFUSION SEARCH

1) Initialization Phase

- For the curse of dimensionality, our partial hypotheses are binary strings, indicating which features we should keep
  - E.g. *h= 10100010*, means we should keep the 1^{st} , 3^{rd} and 7^{th} feature only. (3 out of 8)
  - Training with this new dataset gives an accuracy score of 80%, so $s_h = 80$, $s_h \in S$

2) Testing Phase

- Agents randomly select a component function f_i , $i \in \{1, \dots, n\}$ and evaluate it for their particular hypothesis.
  - E.g. *Agent No.1* <u>randomly</u> selects *Agent No.5* and evaluates with component function $f_5 \left(s_h{}^{agent1} \right)$

STOCHASTIC DIFFUSION SEARCH

- Agents are divided into 2 groups:

Happy – Active

$$f_i(s_h) = 0$$

Unhappy – Inactive

$$f_i(s_h) = 1$$

- For our machine learning problems, these component functions can be modeled as:

$$f_i\left(s_h^{agentj}\right) = unit\_step\left(s_h^{agenti} - s_h^{agentj}\right), \qquad i, j \in \{1, \dots, n\}$$

STOCHASTIC DIFFUSION SEARCH

3) Diffusion Phase

- Each unhappy agent chooses at random another agent for communication
  - If the selected agent is happy, the unhappy agent copies its hypothesis *(diffusion)*
  - If the selected agent is unhappy, there is no flow of information, and the selecting agent adopts a new random hypothesis.
- Happy agents do not initiate a communication and repeat their hypothesis *(in standard SDS)*

4) Halting Phase

- Many different halting criterions *(Threshold of active agents, Number of iterations, etc.)*

CODING

You can find my coding repository on my Github Profile

https://github.com/SotirisFtiakas/Stochastic-Diffusion-Search-for-Feature-Selection

5

EXPERIMENTS

Maximization with constraints.
(Sonar signals, Image pixels)

EXPERIMENT 1 – SONAR SIGNALS

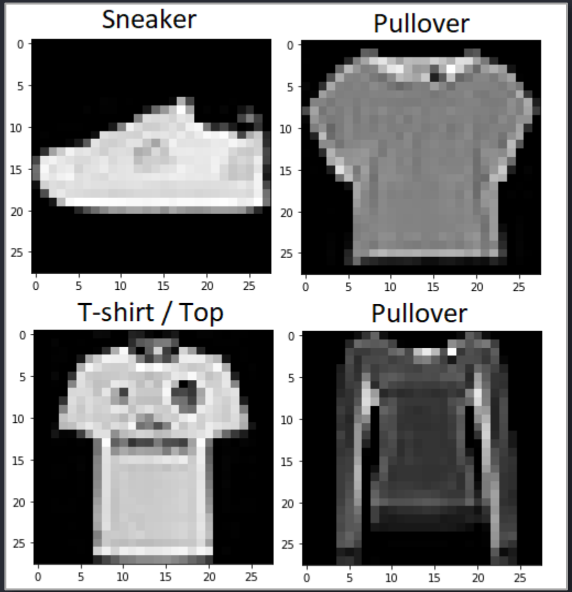
Sonar: Mines vs Rocks

- Dataset used to discriminate between sonar signals bounced off a *metal cylinder (mines)* and those bounced off a roughly *cylindrical rock*.
- 111 patterns by bouncing sonar signals off a metal cylinder at various angles
- 7 patterns obtained from rocks under similar conditions
- Each pattern is a set of 60 numbers from 0.0 to 1.0

EXPERIMENT 2 – IMAGE PIXELS

Fashion – MNIST Dataset

- 28 x 28 grayscale clothing images (784 pixels total)
- Pixels take values from 0 - 255, representing their darkness
- 10 different labels (T-shirt/top, Trouser, Pullover, etc.)
- 45.000 training examples



6

RESULTS

Compare with baseline models.

SONAR SIGNALS – RESULTS

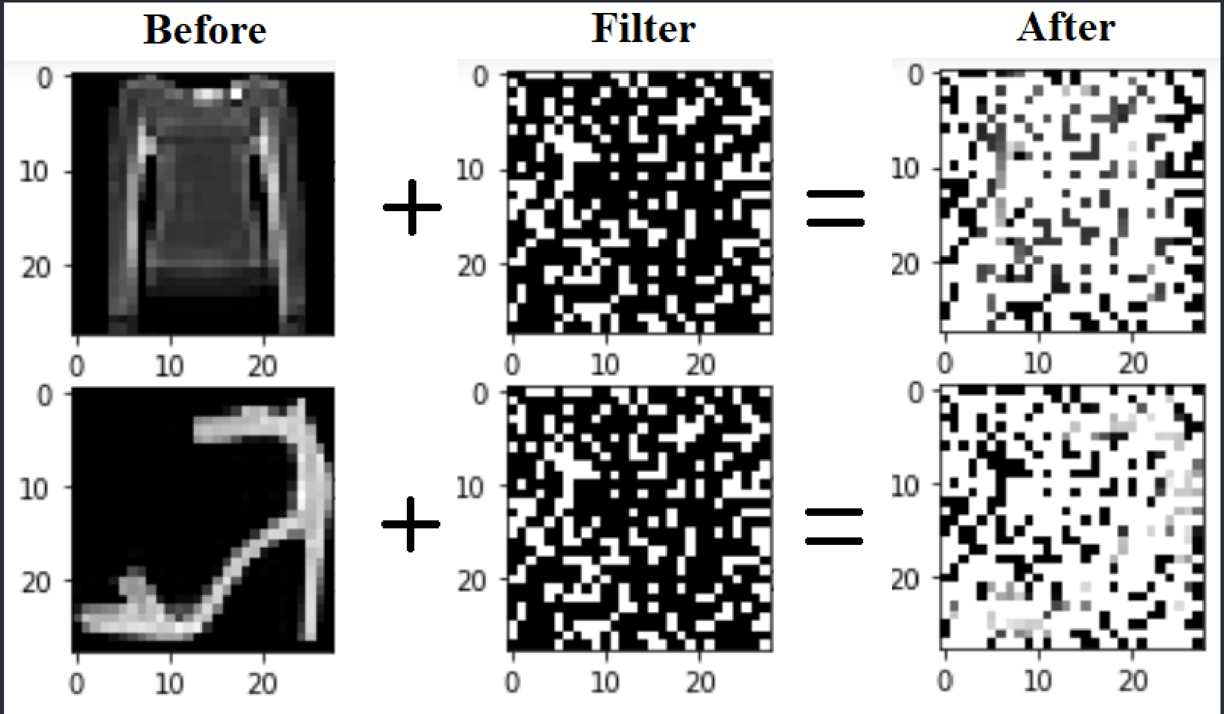
	Logistic Regression	Random Forest	Decision Tree
Initial Dataset, 60 Cols	0.77	0.85	0.75
SDS Subset, 36 Cols	0.85	**0.88**	0.77
SDS Subset, 8 Cols	0.83	0.85	**0.81**

IMAGE PIXELS – RESULTS

45000 rows	Logistic Regression	Random Forest	Decision Tree
Initial Dataset, **784 Cols**	0.85	**0.88**	0.71
SDS Subset, **181 Cols**	0.83	**0.87**	0.70

1000 rows	Logistic Regression	Random Forest	Decision Tree
Initial Dataset, **784 Cols**	0.79	**0.79**	0.67
SDS Subset, **145 Cols**	0.79	**0.81**	0.69
SDS Subset, **71 Cols**	0.75	**0.80**	0.70
SDS Subset, **18 Cols**	0.68	**0.74**	0.66

IMAGE PIXELS – RESULTS



THANKS!

DO YOU HAVE ANY QUESTION?

sftiakas@csd.auth.gr
github.com/SotirisFtiakas