# liquid state machines

- Adrian Chervinchuk
- Magdalena Zembrzuska Kuba Wroński
- Vatsal patel

Problem

Data

Solution

Results and Model Evaluation

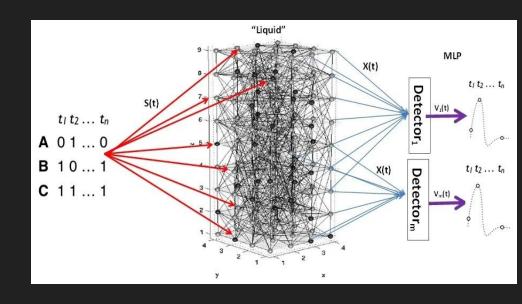
Reflections

Agenda

# Problem

## The problem

- LSM is a machine learning model or system that is part of a series of particular neural network models
- these models build on traditional designs to introduce new and innovative ways of processing information
- neural networks that is based around the neurobiology of the human brain

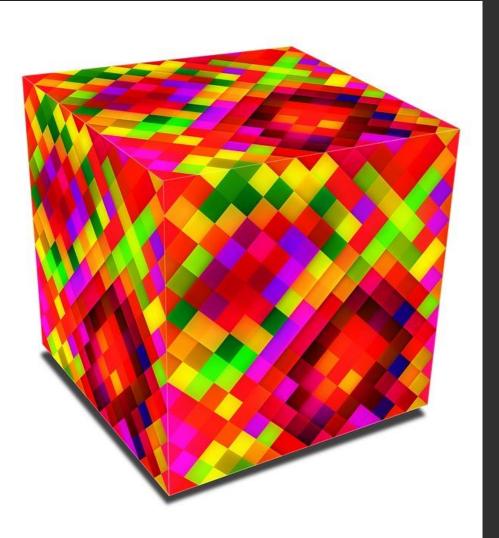


## Who can benefit?

the Neumann machines also consume tremendous power energy power every and resources, however the brain is spared neuromorphic computing provides an appealing attack resolution which shows great efficiency in terms of both hardware cost and also the runtime in

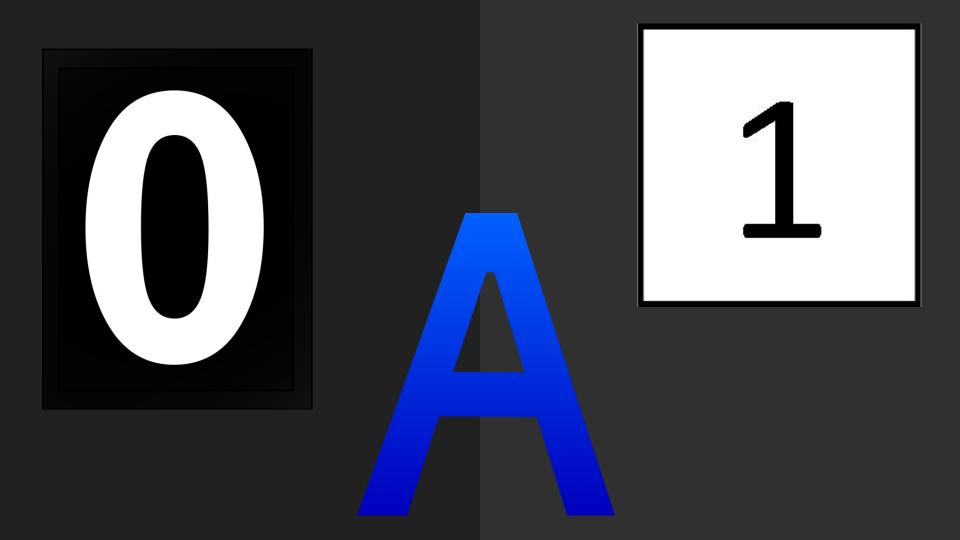
## Why interesting to AS?

This problem creates some interest for our team because of its non triviality and flexibility. We believe that it can be applied to different varieties of topics, starting from robot control and ending with speech recognition.



# how can this data help solve the problem?

 After the solving the problem we will potentially be able to see the which of the 3 characters was chosen by liquid state machine(LSM)



# DATA



# Sources of data, assessment of its reliability

 Data is generated by LSM spiking neural network program

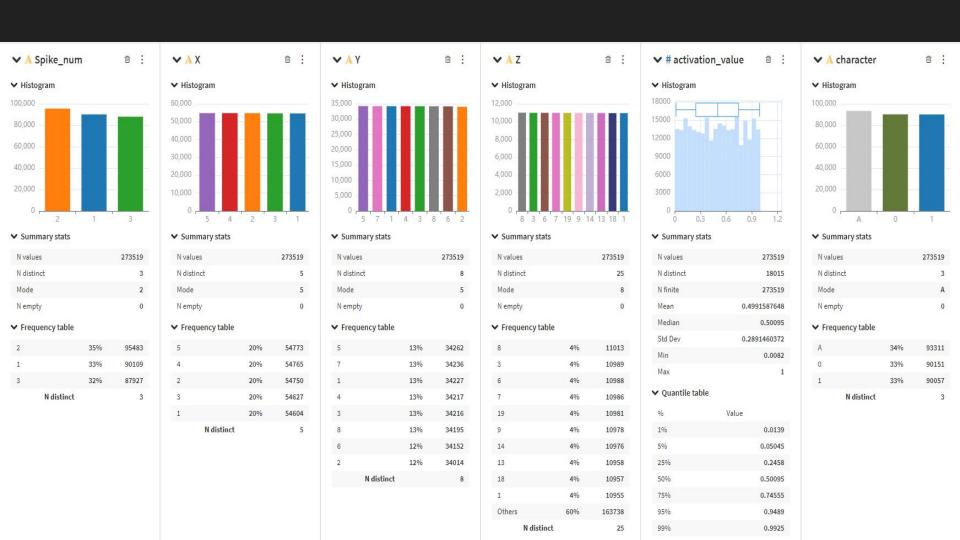
https://github.com/KarolChlasta/rnap/tree/main/dat/raw/ex2022.04.24-RetNet40-25PUM

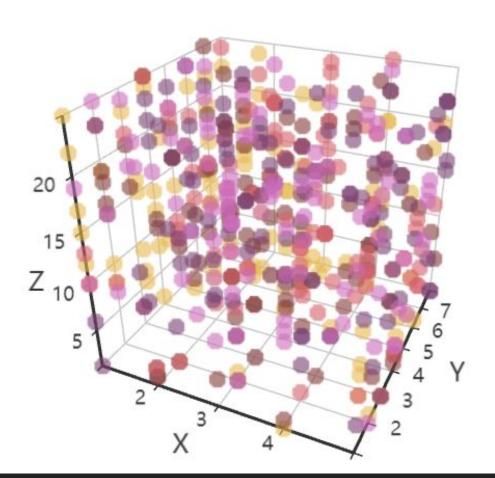
# Analysis of the data

### Dataset

- No rows -273519
- No columns -6

6	https://github.com/KarolChlasta/rnap/tree/main/dat/raw/ex2022.04.24-RetNet40-25PUM		$\forall_{\mathcal{V}}$	Q	to	ø	*	æ	(3	٤
	stats	Added RenNet40-25PUM						1 minu	ute ago	
	RetNet40- <mark>0</mark> -column.spike1	Added RenNet40-25PUM						1 min	ute ago	
D	RetNet40-0-column.spike2	Added RenNet40-25PUM						1 min	ute ago	
	RetNet40- <mark>0</mark> -column.spike3	Added RenNet40-25PUM						1 minu	ute ago	1
	RetNet40-0-retina.spike1	Added RenNet40-25PUM						1 minu	ute ago	
	RetNet40-0-retina.spike2	Added RenNet40-25PUM						1 minu	ute ago	
	RetNet40-0-retina.spike3	Added RenNet40-25PUM						1 min	ute ago	
	RetNet40-1-column.spike1	Added RenNet40-25PUM						1 min	ute ago	
	RetNet40-1-column.spike2	Added RenNet40-25PUM						1 minu	ute ago	
	RetNet40-1-column.spike3	Added RenNet40-25PUM						1 minu	ute ago	
	RetNet40-1-retina.spike1	Added RenNet40-25PUM						1 minu	ute ago	
	RetNet40-1-retina.spike2	Added RenNet40-25PUM						1 min	ute ago	-
	RetNet40-1-retina.spike3	Added RenNet40-25PUM						1 min	ute ago	
٥	RetNet40-A-column.spike1	Added RenNet40-25PUM						1 minu	ute ago	
	RetNet40- <mark>A</mark> -column.spike2	Added RenNet40-25PUM						1 minu	ute ago	
٥	RetNet40-A-column.spike3	Added RenNet40-25PUM						1 minu	ute ago	
	RetNet40-A-retina.spike1	Added RenNet40-25PUM						1 min	ute ago	
	RetNet40-A-retina.spike2	Added RenNet40-25PUM						1 minu	ute ago	
D	RetNet40-A-retina.spike3	Added RenNet40-25PUM						1 minu	ute ago	





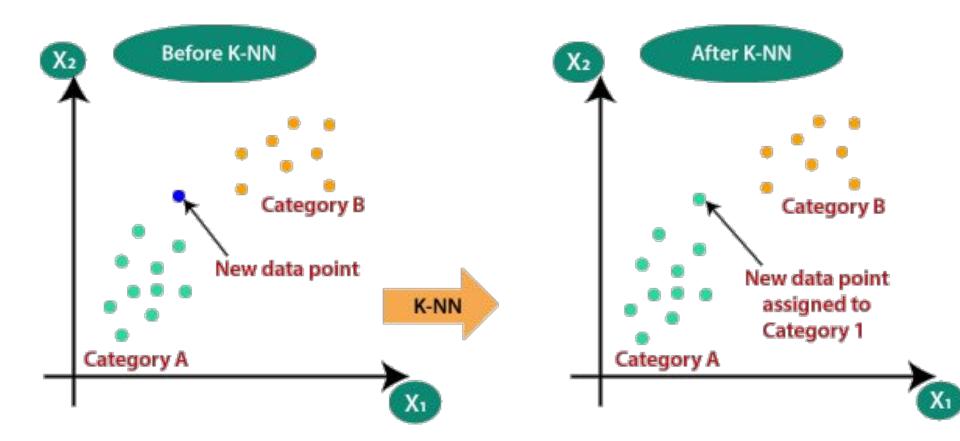
Let's go to dataiku

# Solution

# k-Nearest Neighbors (KNN)

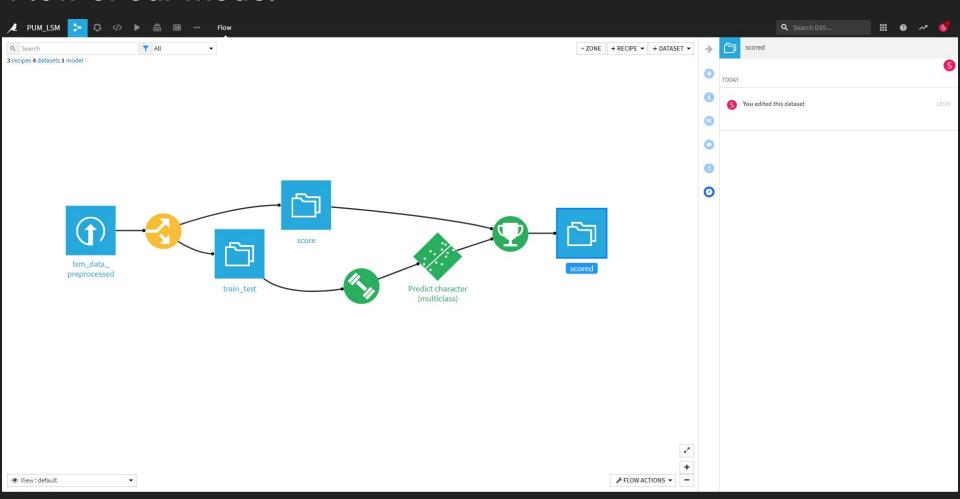






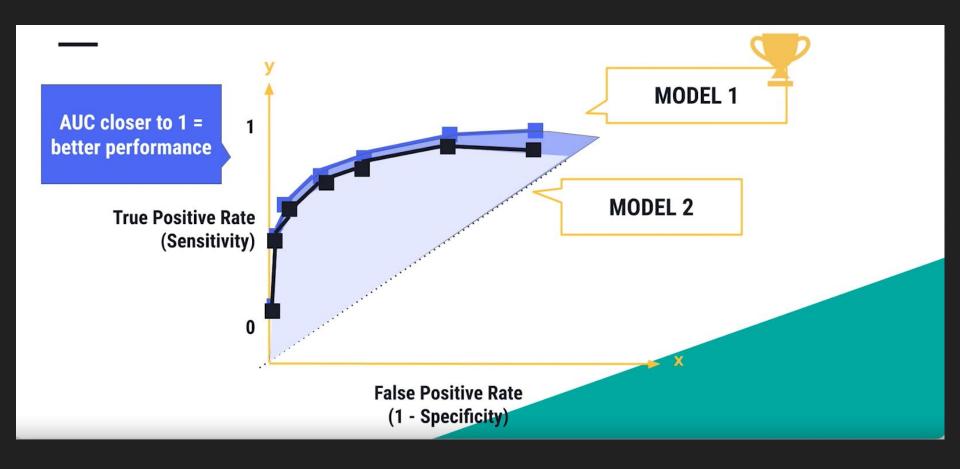
# Result

## Flow of our model



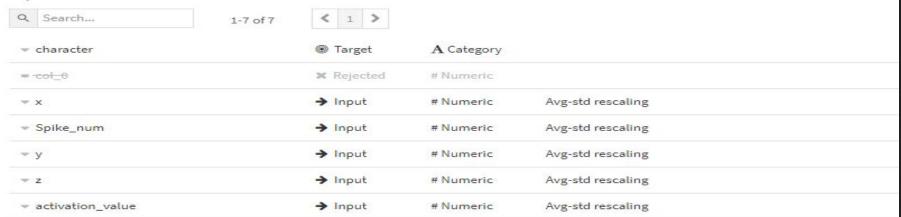


### receiver operating characteristic curve (ROC)



## Modeling results 1

#### Input features



#### Preprocessed features (5)

•x •Spike\_num •y •z •activation\_value

#### Feature generation

None enabled

#### K Nearest Neighbors (k=5) (s11)

ROC AUC: 0.463

■ Dataset◆ Model✓ Custom

#### Model

Model ID	A-PUM_LSM-PXrEi8we-jwmZx0IY-s11-pp1-m1				
Backend	Python (in memory)				
Algorithm	Knn				
Trained on	2022/05/30 18:24				
Columns (train set)	7				
Rows (train set)	131278				
Calibration method	No calibration				

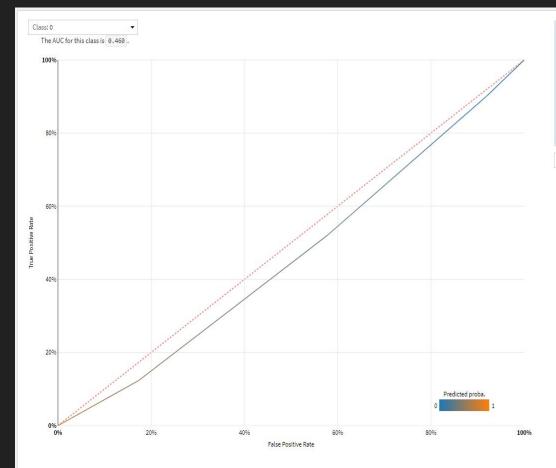
#### Metadata

9	trainDataset:dataset-name	$\rightarrow$	train_test	Û
9	testDataset:dataset-name	$\rightarrow$	train_test	
9	evaluationDataset:dataset-name	$\rightarrow$	train_test	
4	model:algorithm	$\rightarrow$	KNN	Û
4	model:name	$\rightarrow$	K Nearest Neighbors (k=5) (s11)	Û
•	model:date	$\rightarrow$	2022-05-30T16:24:16.611+0000	Û
dis	evaluation:date	$\rightarrow$	2022-05-30T16:24:16.611+0000	

#### + ADD A LABEL

Optional. Informative labels for the model. The model:algorithm, model:date, model:name, trainDataset:dataset-name, testDataset:dataset-name labels, evaluation:date and evaluationDataset:dataset-name are automatically added.





The Receiver Operating Characteristic (or ROC) curve shows the true positive rate vs. the false positive resulting from different cutoffs in the predictive model. The "faster" the curve climbs, the better it is.

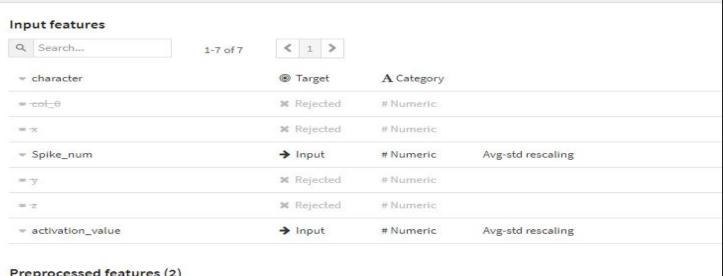
On the contrary, a curve close to the diagonal line is worse.

The MAUC (Multi-class Area Under the Curve) for this model is 0.463, which is worse than random guess:(.

EXPORT DATA

The faster the curve climbs ,the better but in this case the line is close to the diagonal so its worse

## Modeling results 2



#### Preprocessed features (2)

• Spike\_num • activation\_value

#### Feature generation

None enabled

## Little modification

Remove some column

#### K Nearest Neighbors (k=5) (s10)

ROC AUC: 0.963

#### Model

Model ID	A-PUM_LSM-PXrEi8we-jwmZx0IY-s10-pp1-m1 Python (in memory)			
Backend				
Algorithm	Knn			
Trained on	2022/05/30 18:19			
Columns (train set)	7			
Rows (train set)	131278			
Calibration method	No calibration			

#### Metadata

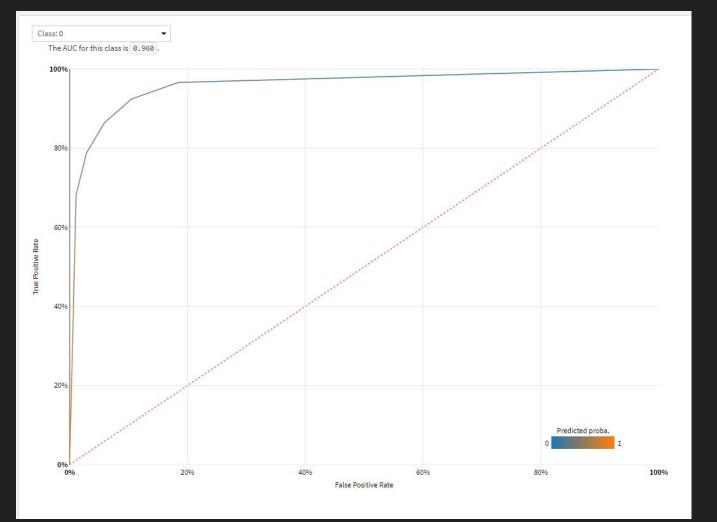
trainDataset:dataset-name	$\rightarrow$	train_test	
testDataset;dataset-name	$\rightarrow$	train_test	ê
evaluationDataset:dataset-name	-	train_test	
♦ model:algorithm	$\rightarrow$	KNN	Û
♦ model:name	<b>→</b>	K Nearest Neighbors (k=5) (s10)	ê
♦ model:date	<b>→</b>	2022-05-30T16:19:56.494+0000	Ē
i evaluation:date	<b>→</b>	2022-05-30T16:19:56.494+0000	Û

#### + ADD A LABEL

Optional. Informative labels for the model. The model:algorithm, model:alee, model:name, trainDataset:dataset-name, testDataset:dataset-name labels, evaluation:date and evaluationDataset:dataset-name are automatically added.



Accuracy 0.963

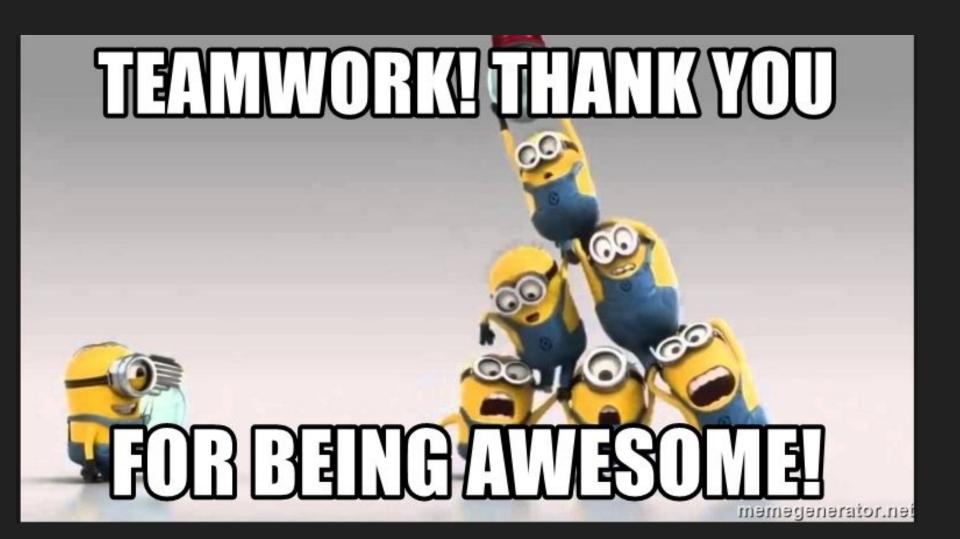


The faster the curve climbs ,the better this case our curve is not to the diagonal faster than previous example

## Reflections

 We have created a simple dashboard for reflection so let's switch to dataiku





# Thank you for attention Dziękuję za uwagę Спасибо за внимание

:)