# Machine Learning for Exploration Geophysics
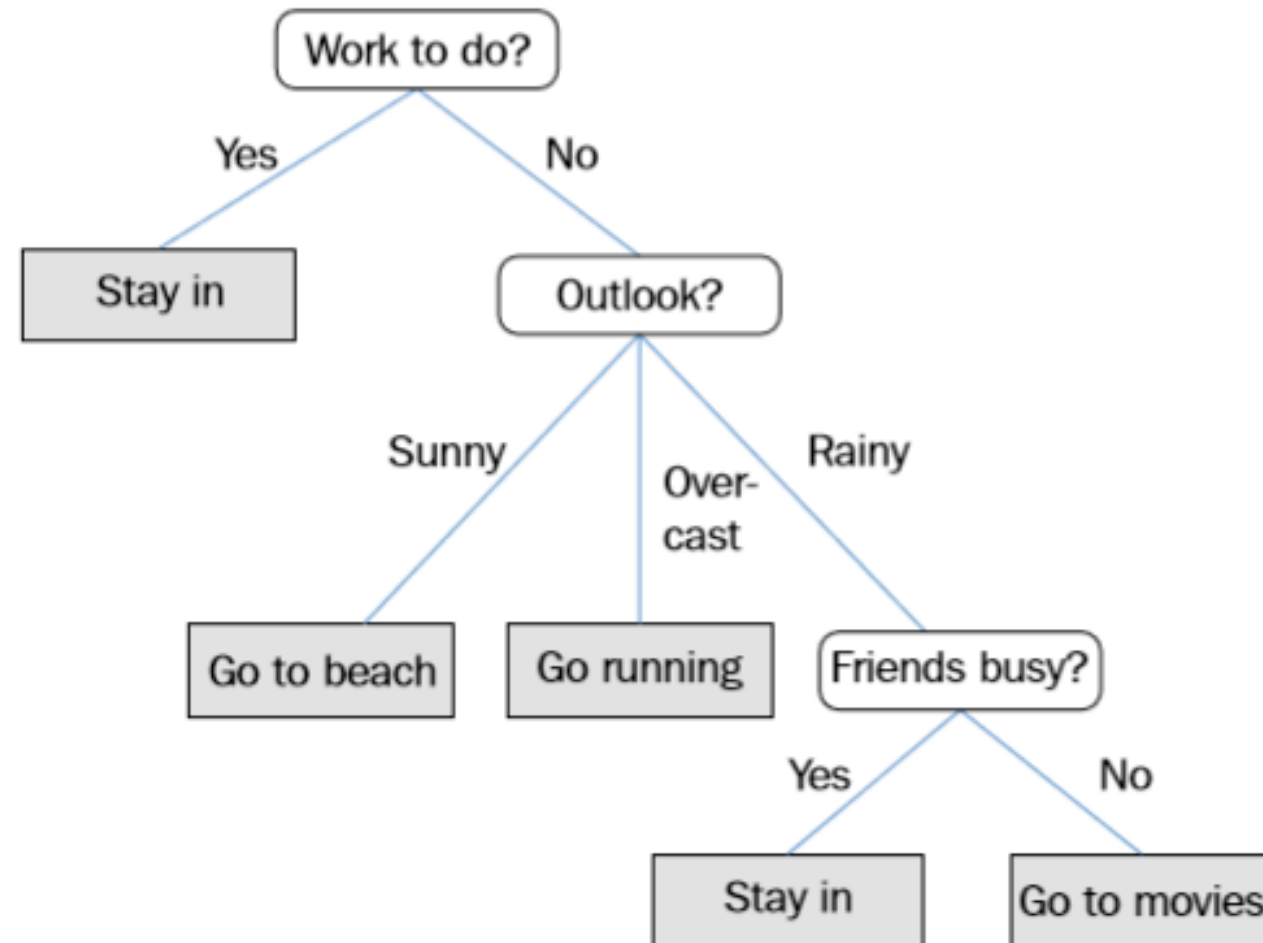
## Th3: Ensemble Learning

10. - 12. March 2020
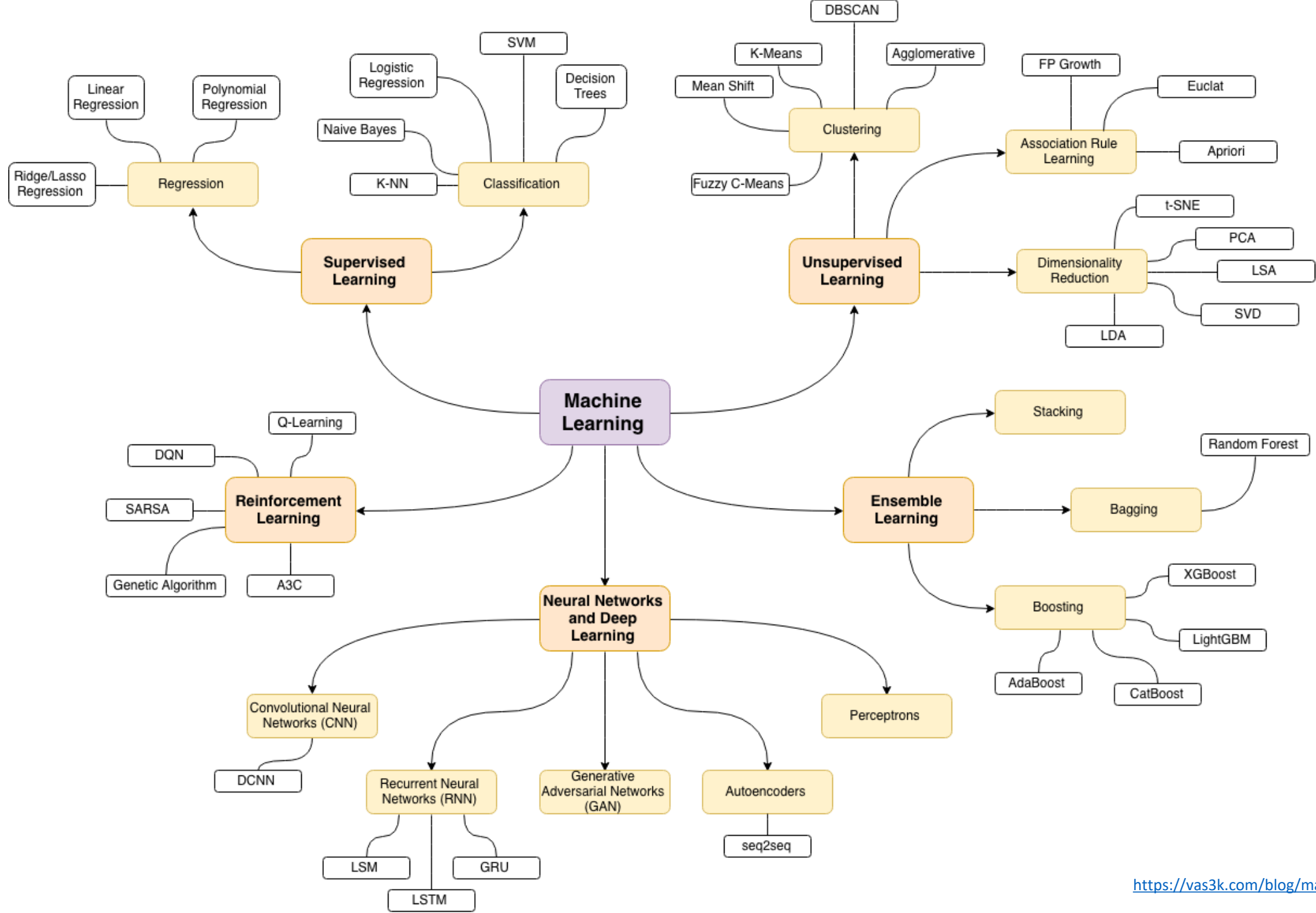
Hamburg
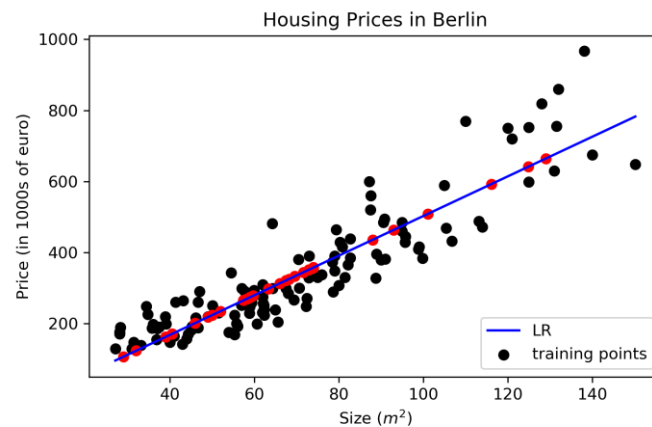
# Outline

- Ensemble Learning

- Decision Tree

- Stacking

- Bagging
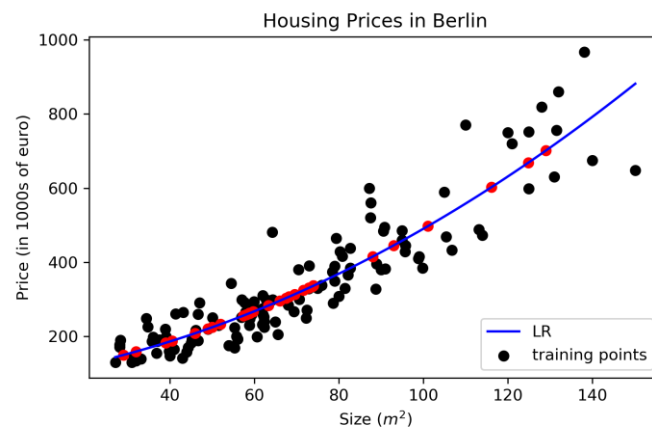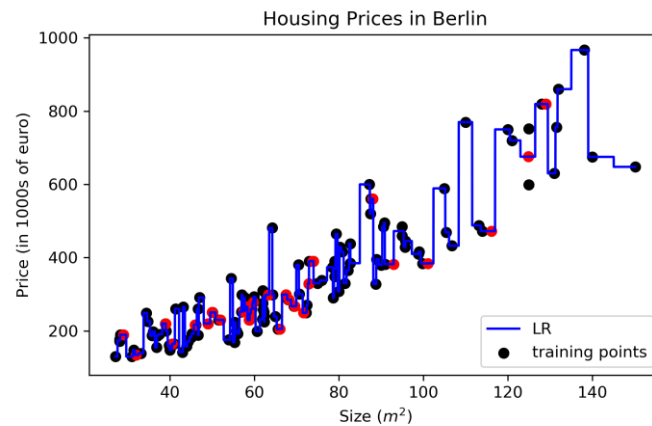    - Random forest

- Boosting
    - AdaBoost
    - XGboost

Ivan Abakumov

# Decision Tree

Image source: https://www.commonlounge.com/discussion/ee8b074936a041f2b5a57d2054dc3701
Ivan Abakumov

Machine Learning

**Supervised Learning**

Regression
- Linear Regression
- Polynomial Regression
- Ridge/Lasso Regression

Classification
- Logistic Regression
- SVM
- Decision Trees
- Naive Bayes
- K-NN

**Unsupervised Learning**

Clustering
- K-Means
- DBSCAN
- Agglomerative
- Mean Shift
- Fuzzy C-Means

Association Rule Learning
- FP Growth
- Euclat
- Apriori

Dimensionality Reduction
- t-SNE
- PCA
- LSA
- SVD
- LDA

**Reinforcement Learning**
- Q-Learning
- DQN
- SARSA
- Genetic Algorithm
- A3C

**Ensemble Learning**

Stacking

Bagging
- Random Forest

Boosting
- XGBoost
- LightGBM
- AdaBoost
- CatBoost

**Neural Networks and Deep Learning**

Convolutional Neural Networks (CNN)
- DCNN

Recurrent Neural Networks (RNN)
- LSM
- LSTM
- GRU

Generative Adversarial Networks (GAN)

Autoencoders
- seq2seq

Perceptrons

$X_{train}$

$X_{train}$

$X_{train}$

Housing Prices in Berlin

Price (in 1000s of euro)

Size ($m^2$)

LR
training points

$y_{pred}^1$

$y_{pred}^2$

$y_{pred}^3$

Stacking

final prediction

$X^1_{train}$
[1,1,3,4]

$X^2_{train}$
[1,2,2,4]

$X^3_{train}$
[2,3,4,4]

$y^1_{pred}$

$y^2_{pred}$

$y^3_{pred}$
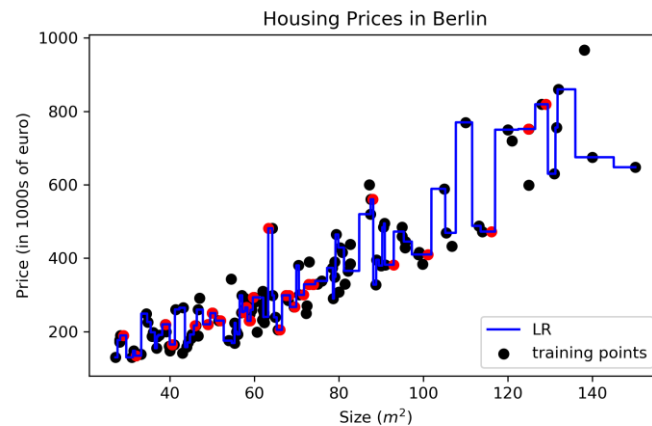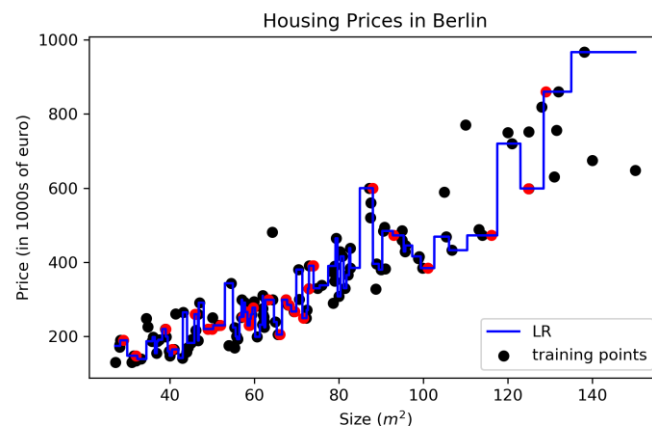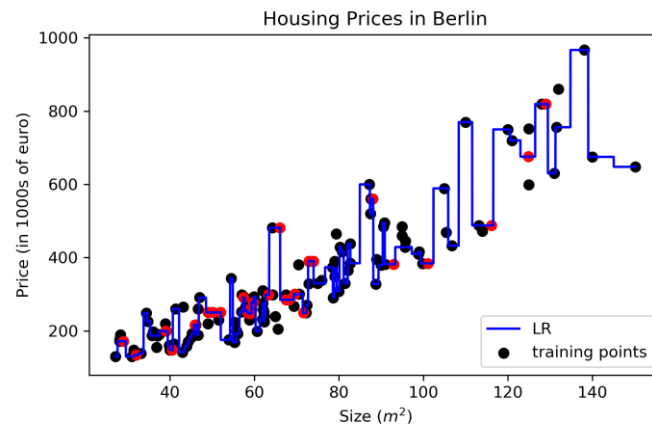
$final\ prediction$

Bagging
Bootstrap
AGGregatING

**function** AdaBoost(X_train, Y_train):

**for**  i = 1..m  **do**:
$$D_i^1 = \frac{1}{m}$$
**end for**

**for** t = 1..T **do**:
$$h_t = \operatorname*{arg\,min}_{h_j \in H} \; \epsilon_j = \sum_{i=1}^{m} D_i^t [\![ y_i \neq h_j(x_i) ]\!]$$
$$\alpha_t = \frac{1}{2} \ln \frac{1 - \epsilon_t}{\epsilon_t}$$

    **for** i = 1..m **do**:
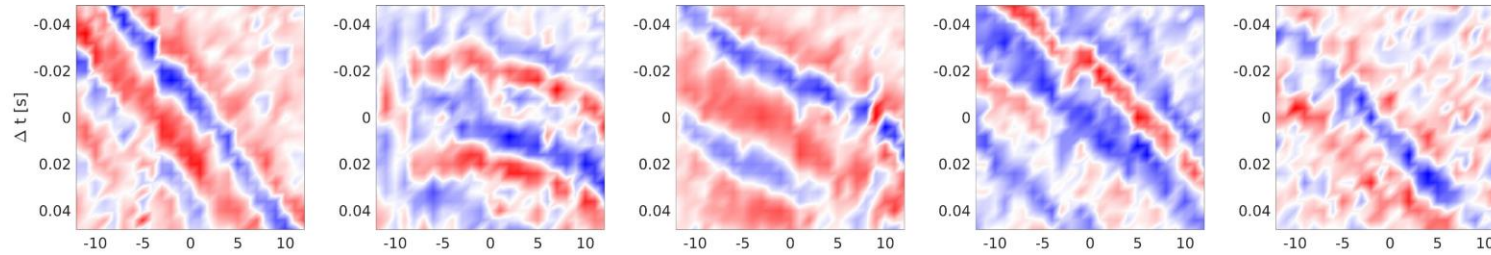$$D_i^{t+1} = D_i^t \; e^{(-\alpha_t \, y_i \, h_t(x_i))} / Z_t$$
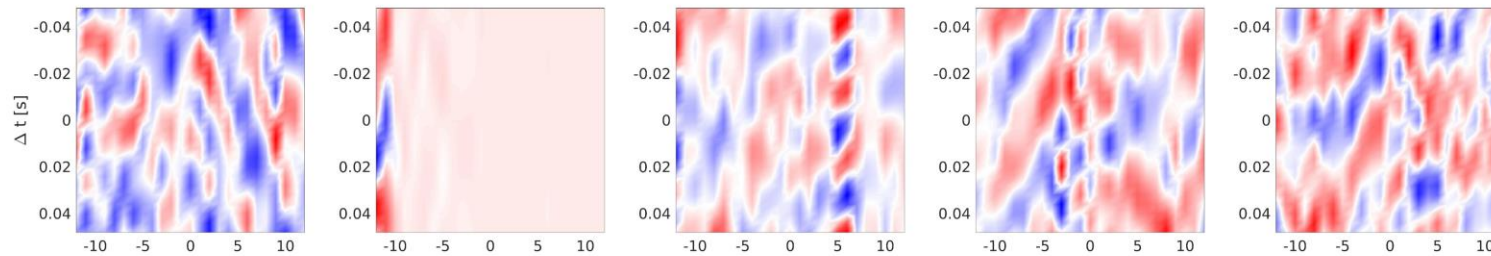
    **end for**
**end for**
$$H(x) = \operatorname{sign} \sum_{t=1}^{T} \alpha_t h_t(x)$$
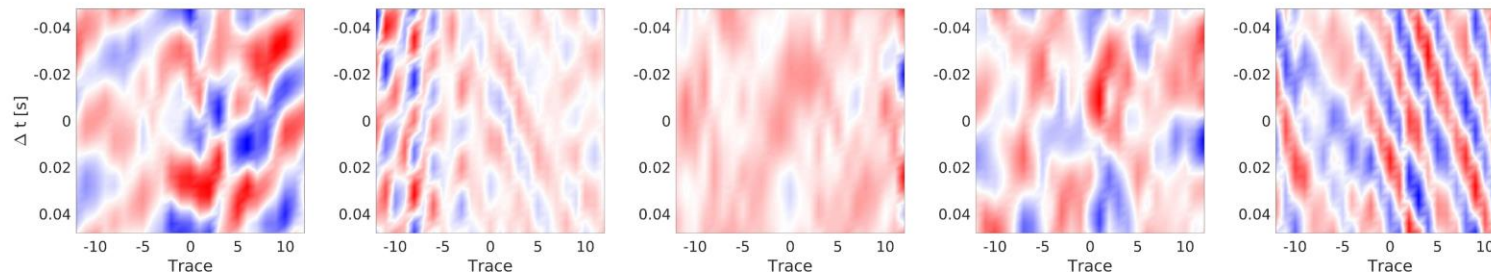
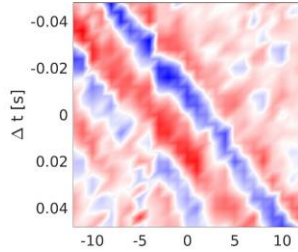**return** $H$

# Select the best features and train classifiers



Signal examples

Noise examples

# Select the best features and train classifiers



$$= \mathbf{x_i} \qquad y_i = \begin{cases} +1, & if\ x_i\ is\ event \\ -1, & if\ x_i\ is\ noise \end{cases}$$

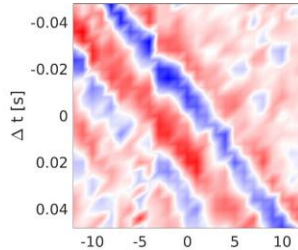$$f_j(\mathbf{x}) = S_c\left(\mathbf{x}, \tau\left(p_x^j, p_y^j, V_{NMO}^j\right)\right)$$

# Select the best features and train classifiers



$$= \mathbf{x}_i \qquad y_i = \begin{cases} +1, & if\ x_i\ is\ event \\ -1, & if\ x_i\ is\ noise \end{cases}$$
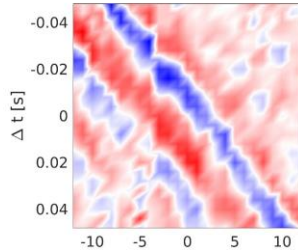
- Find "weak" classifiers

$$f_j(\mathbf{x}) = S_c\left(\mathbf{x}, \tau\left(p_x^j, p_y^j, V_{NMO}^j\right)\right)$$

where $\theta_j$ is a threshold, $s_j = \pm 1$ is polarity

$$h_j(\mathbf{x}) = \begin{cases} +s_j, & if\ f_j(\mathbf{x}) > \theta_j \\ -s_j, & otherwise \end{cases}$$

# Select the best features and train classifiers



$$= \boldsymbol{x_i} \qquad y_i = \begin{cases} +1, & if \ x_i \ is \ event \\ -1, & if \ x_i \ is \ noise \end{cases}$$

- Find "strong" classifiers

$$f_j = S_c\left(\mathbf{x}, \tau\left(p_x^j, p_y^j, V_{NMO}^j\right)\right)$$

where $\alpha_j$ is a weight

$$h(\mathbf{x}) = \text{sign}\left(\sum_{j=1}^{M} \alpha_j h_j(\mathbf{x})\right)$$

# Cascade architecture

$h_1(\mathbf{x}), h_2(\mathbf{x})$
(is it an event?)

No → Stop

Yes → $h_3(\mathbf{x}), .., h_{12}(\mathbf{x})$
(is it an event?)

No → Stop

Yes → $h_{13}(\mathbf{x}), .., h_{22}(\mathbf{x})$
(is it an event?)

No → Stop

Yes → $h_{23}(\mathbf{x}), .., h_{33}(\mathbf{x})$
(is it an event?)

No → Stop

Yes → Face!