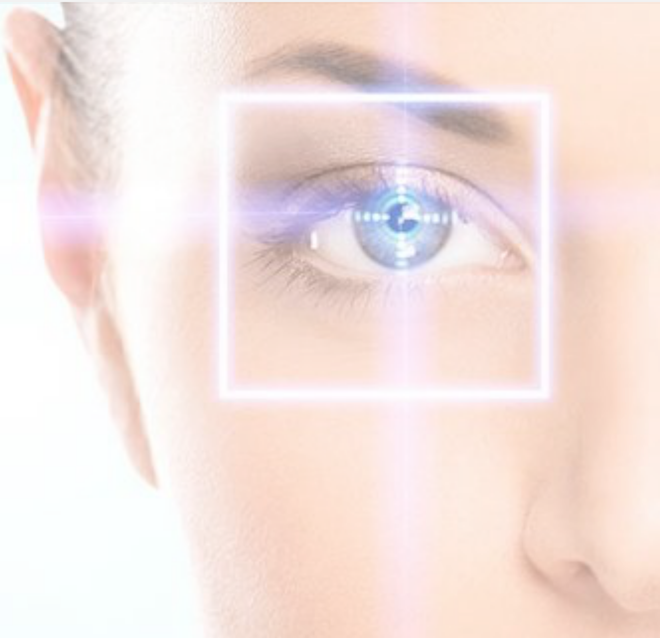


Exam: M. Streicher, September 24, 2024, Potsdam

Intelligent Data Analysis - Medical lasers

Problem Setting

Input data D:

$D = \{(x_i, y_i)\}_{i=1}^{200}$,

where $x_i = [x_{i1}, x_{i2}, ..., x_{i60}]$ is the frequency time series of laser i and $y_i \in \{-1,1\}$ the binary label.

$y = 1$, Suitable for sale

$y = -1$, Not suitable for sale

Model:

$f_\theta : \mathbb{R}^{60} \to \{1, -1\}$, where θ is the model parameter vector.

Problem Setting

Task:

Binary classification problem

Type of learning:

Supervised Learning

Goal:

Classify each laser as belonging to one of two classes,
1 or -1, based on the frequency behavior over time.

Data analysis

<u>Input data D:</u>

$$D = \{(x_i, y_i)\}_{i=1}^{200}$$

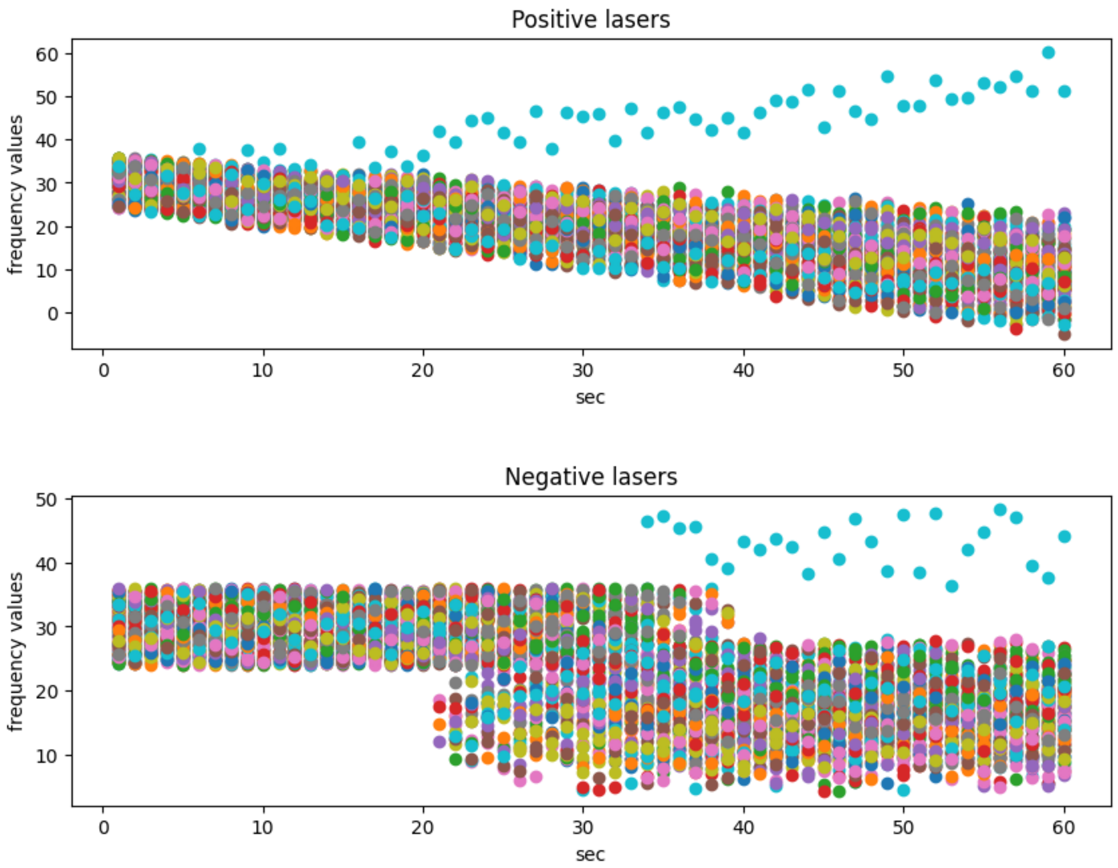
<u>Label distribution:</u>

$$D_{-1} = \{(x_i, y_i) \mid y_i = -1\} = 100$$

$$D_1 = \{(x_i, y_i) \mid y_i = 1\} = 100$$

<u>Missing data:</u>

No missing data found.



Evaluation Protocol

Train and Test:

BaseDataset(Dataset) #pytorch

data_split_indices.pkl #pickle

Test inputs,
Test labels

Train inputs,
Train labels

Train_validation inputs,
Train_validation labels

Validation inputs,
Validation labels

Tripple Cross Validation:

ParameterGrid() #sklearn.model_selection

Models

Linear Classifier, DTW Kernel, Polynomial Kernel, RBF Kernel

Linear Classifier

Model:

$$argmin_\theta \sum_{i=1}^{n} l(max(0,1 - y_i f_\theta(x_i)), y_i) + \lambda\Omega_2(\theta), \text{ with } f_\theta(x_i) = x_i * \theta_i$$

ERM using Gradient Descent Method:

$$\nabla_\theta L(\theta) = \begin{cases} \frac{2\lambda}{n}\theta, & \text{if } 1 - y_i(\mathbf{x}_i \cdot \theta) \leq 0 \\ -y_i\mathbf{x}_i + \frac{2\lambda}{n}\theta, & \text{if } 1 - y_i(\mathbf{x}_i \cdot \theta) > 0 \end{cases}$$

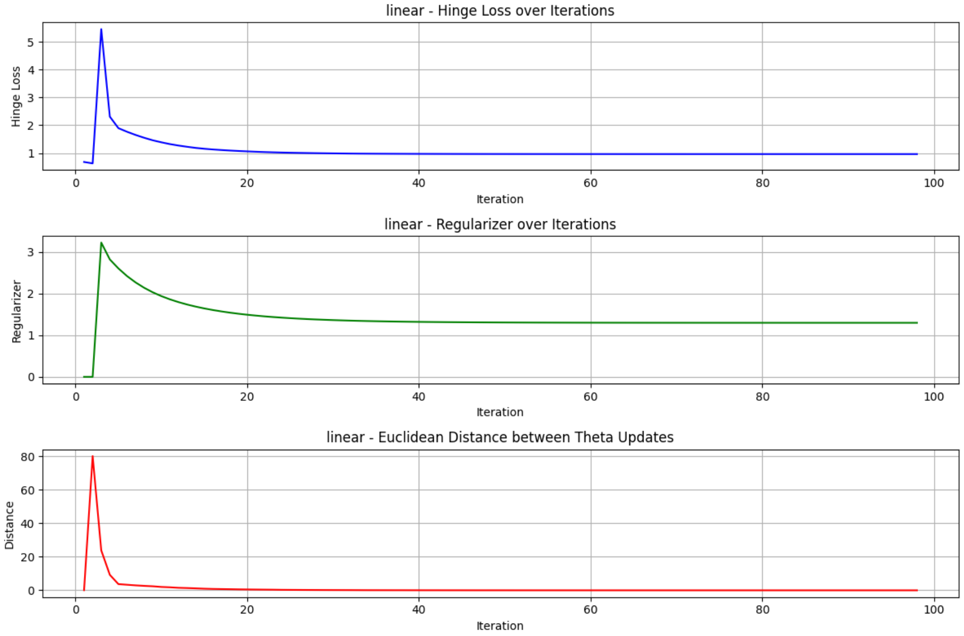
Feature Engineering:

Feature I: R^2

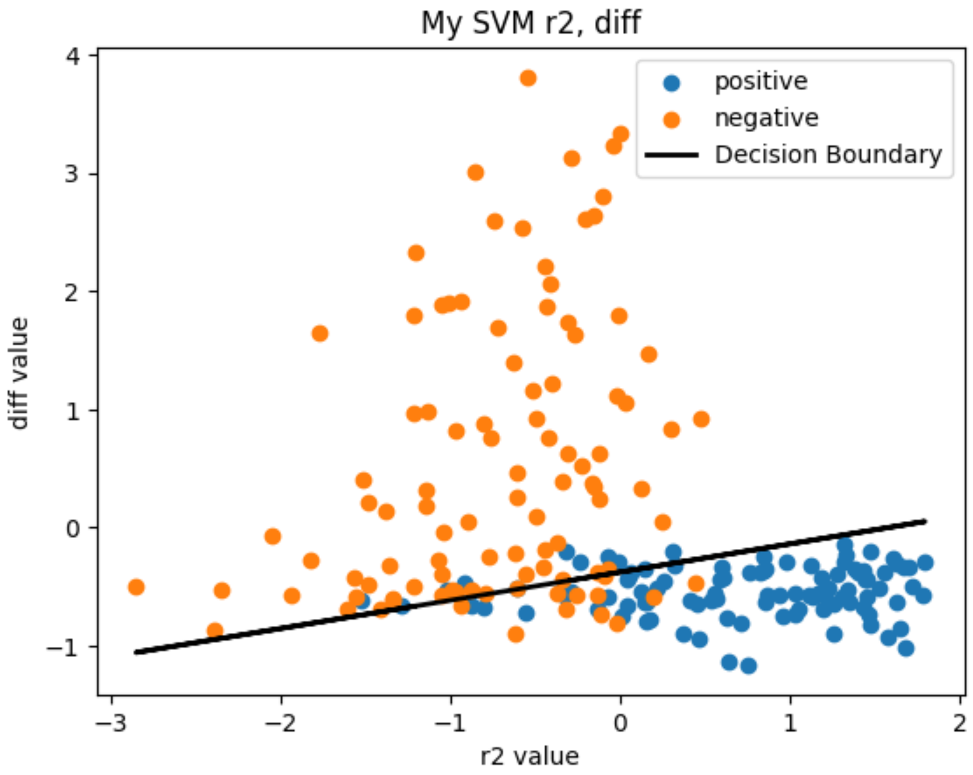
Feature II: Maximal difference to a subsequent measuring point.

Linear Classifier

```
Best parameters found:
{'alpha_0': 0.001, 'decay': 0.9, 'epsilon': 0.0001, 'lambda_value': 0.001}
Best score: 0.925
```



Linear Classifier



DTW Kernel

$$k_{\mathrm{DTW}}(x, x') = e^{-\lambda d_{\mathrm{DTW}}(x, x'; d)}$$

$$d_{DTW}(x, x'; d) = \gamma(\,|\,x\,|\,,|\,x'\,|\,)$$

$$\gamma(i, j) = \begin{cases} d(x_i, x_j') + \min\left(\gamma(i-1, j-1), \gamma(i-1, j), \gamma(i, j-1)\right) & (1 \le i \le |x|, 1 \le j \le |x'|), \\ \infty & i = 0 \vee j = 0, \\ 0 & (i, j) = (0, 0)\,. \end{cases}$$

DTW Kernel:

$$k_{\mathrm{DTW}}(x, x') = e^{-\lambda d_{\mathrm{DTW}}(x,x';d)}$$

Polynomial Kernel:

$$k_{poly}(x, x') = (\alpha * x^T x' + c)^d$$

RBF Kernel:

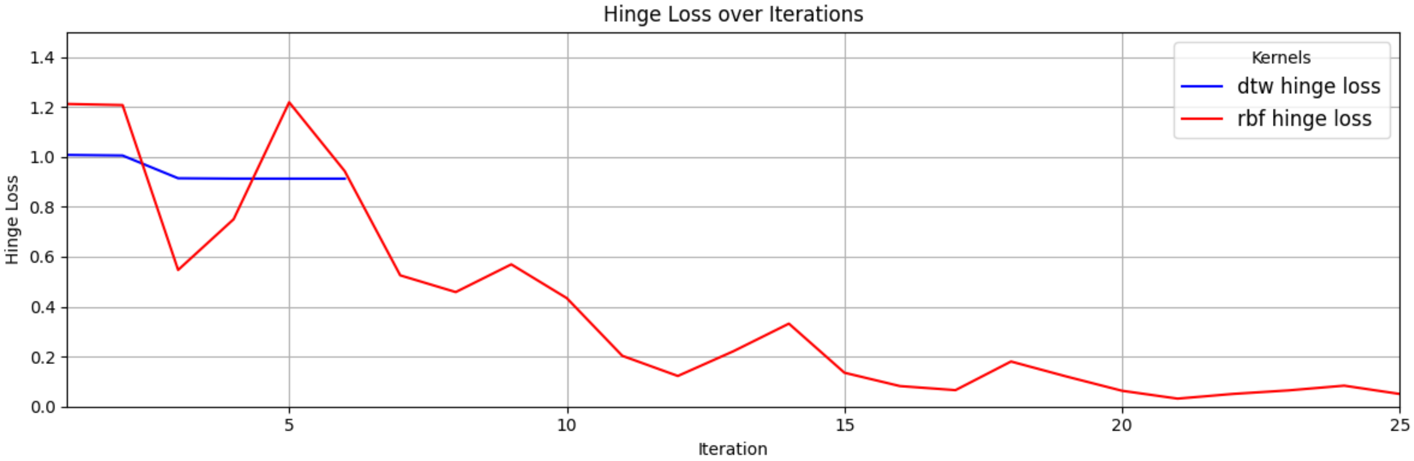
$$k_{RBF}(x, x') = e^{-\lambda * \|x - x'\|^2}$$

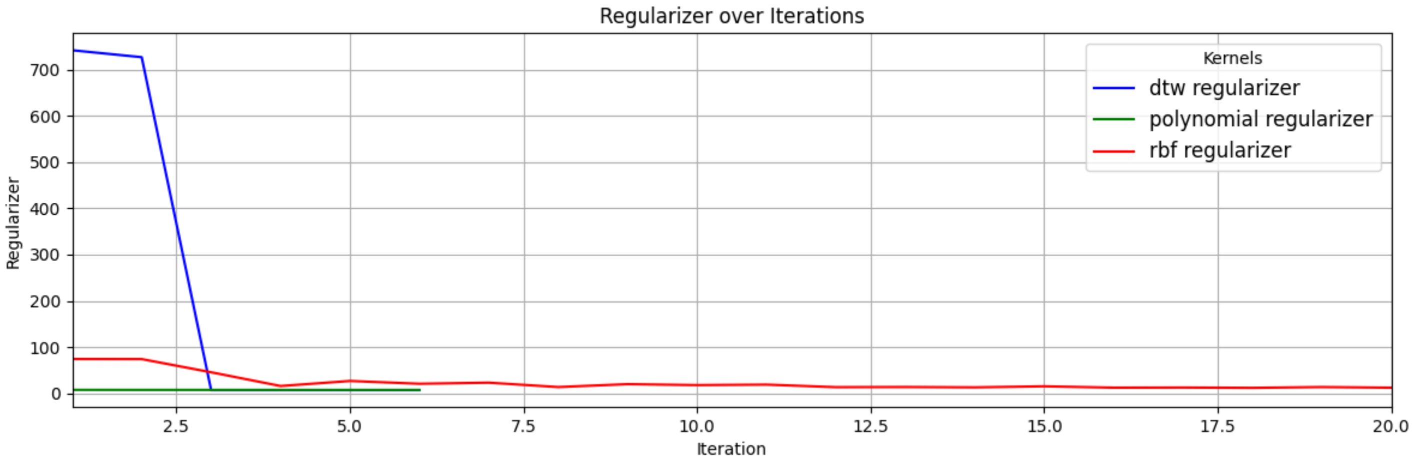
Results

Linear Classifier, DTW Kernel, Polynomial Kernel, RBF Kernel

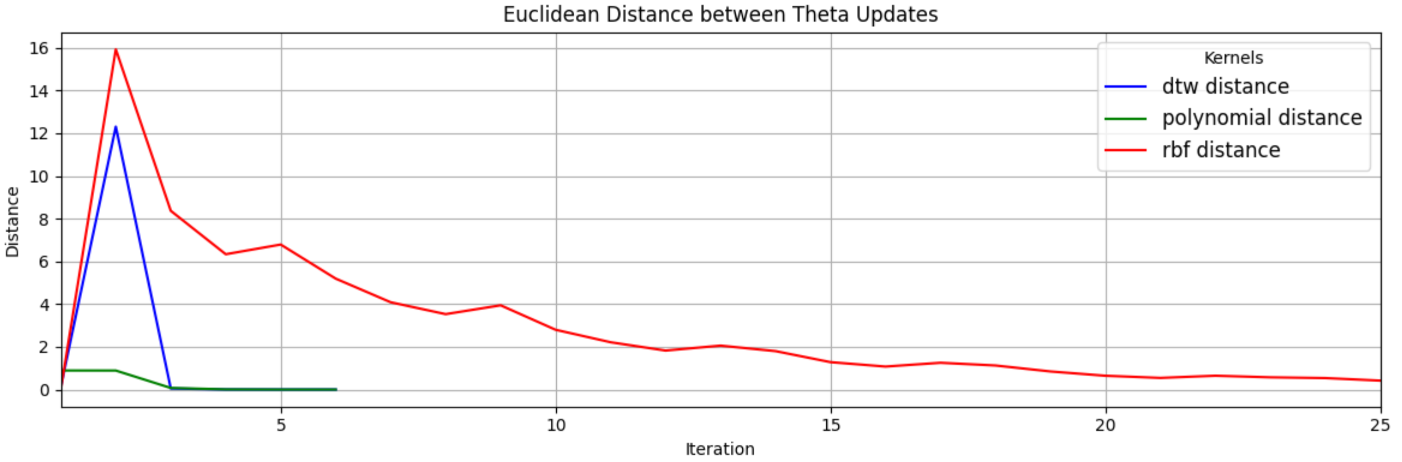
	DTW	Polynomial	RBF
Alpha	0,001	0,1	0,001
Decay	0,1	0,1	0,9
Epsilon	0,0001	0,0001	0,0001
Lambda	10	0,1	1
Kernel specific		C = 0 Degree = 5 Alpha_Poly = 1	Gamma = 0,001
Accuracy	**0,975**	**0,675**	**0,975**


Hinge Loss over Iterations
1e23
Hinge Loss
Iteration
Kernels
polynomial hinge loss

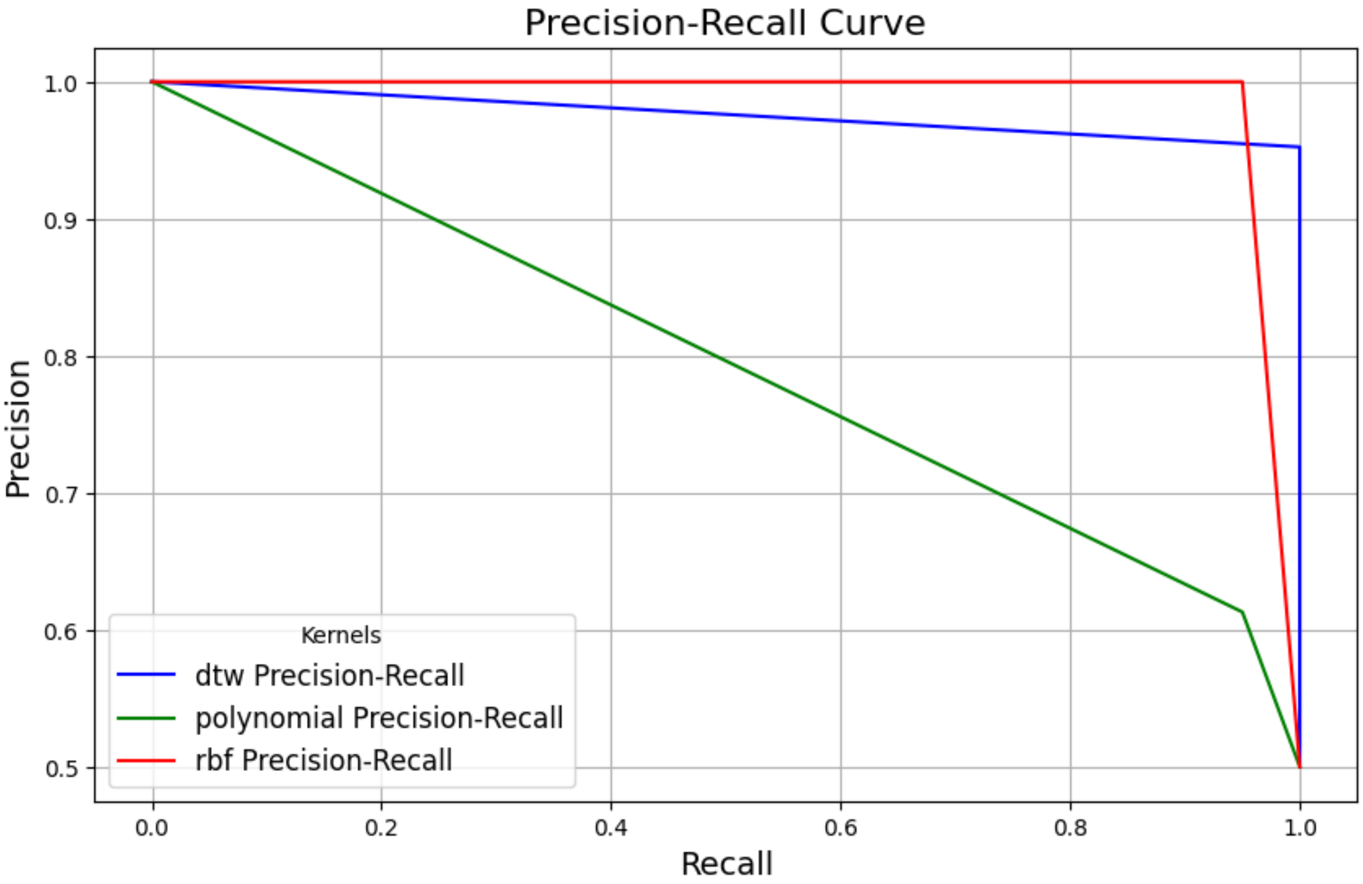

Hinge Loss over Iterations
Hinge Loss
Iteration
Kernels
dtw hinge loss
rbf hinge loss

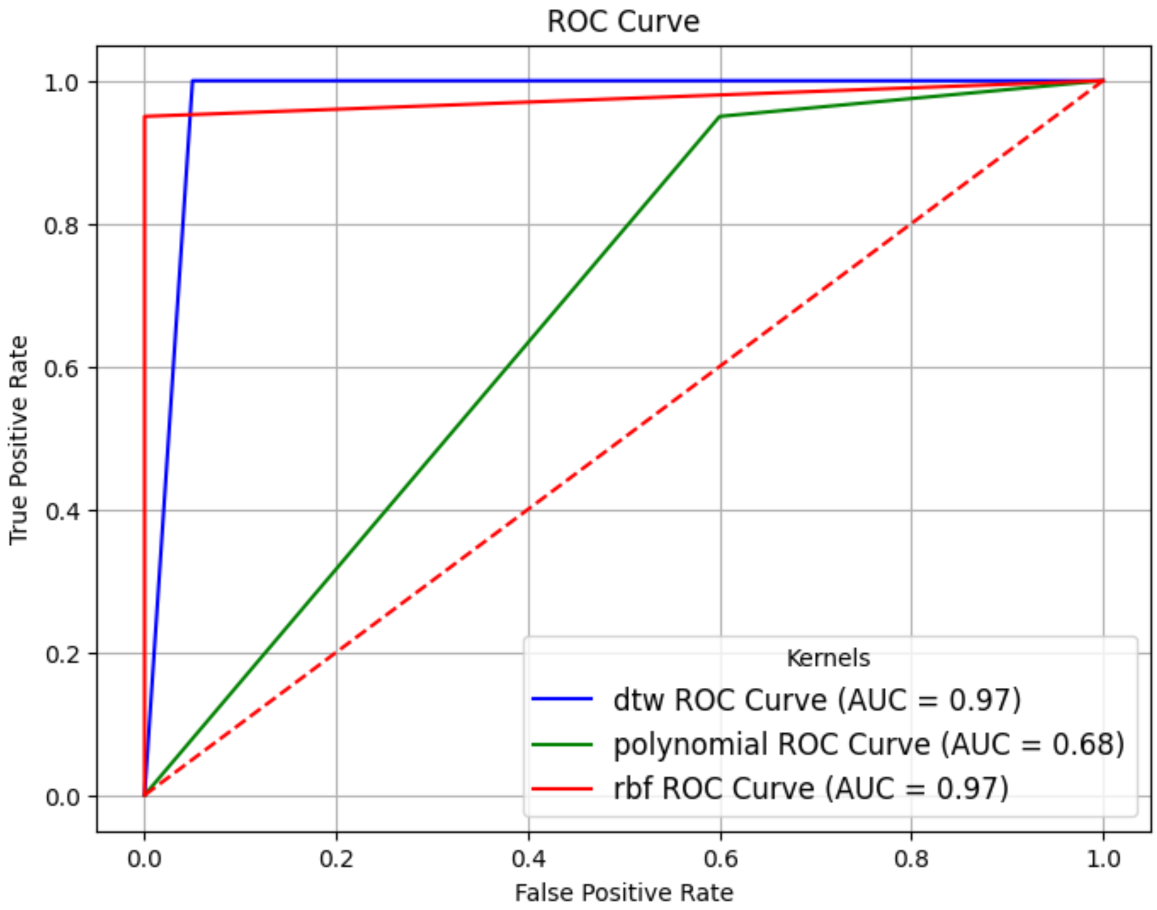


Regularizer over Iterations
Kernels
dtw regularizer
polynomial regularizer
rbf regularizer
Regularizer
0
100
200
300
400
500
600
700
2.5
5.0
7.5
10.0
12.5
15.0
17.5
20.0
Iteration



Euclidean Distance between Theta Updates
Kernels
dtw distance
polynomial distance
rbf distance
Distance
0
2
4
6
8
10
12
14
16
5
10
15
20
25
Iteration


Precision-Recall Curve
Precision
Recall
1.0
0.9
0.8
0.7
0.6
0.5
0.0
0.2
0.4
0.6
0.8
1.0
Kernels
dtw Precision-Recall
polynomial Precision-Recall
rbf Precision-Recall


ROC Curve
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0
Kernels
dtw ROC Curve (AUC = 0.97)
polynomial ROC Curve (AUC = 0.68)
rbf ROC Curve (AUC = 0.97)

Conclusion

Linear Classifier, DTW Kernel, Polynomial Kernel, RBF Kernel

Conclusion

- **Polynomial Kernel does not fit the data**
- **Lowest accuracy** with 0.675
- **Followed by the linear model** with 0.95
- **Positive**: Hardly any risk of overfitting
- **Negative**: Requires preprocessing of the data
- **Best kernels with the same accuracy of 0.975**: DTW, RBF
- **RBF shows significantly better precision** in the Precision-Recall curve and ROC

Conclusion: RBF would be the model of choice.

Sources

[1] Image cover slide: rbb, 2016, Retrieved from: https://www.rbb-online.de/content/dam/rbb/rbb/fernsehen/rbb_praxis_bilder/2016/05/25/COLOURBOX8440750.jpg.jpg/size=966x543.jpg

[2] Image question slide: Gekonnt wirken, Retrieved from: https://www.gekonnt-wirken.de/wp-content/uploads/2018/07/AdobeStock_496887170-scaled-82625_1080x675.jpeg

[3] Source Code: https://github.com/MarStreicher/IDA_Laser/tree/main/