**Introduction**

* Research question:
* Which variables contribute to subscription of term deposits?
* Relevance:
* Helps banks in deciding their marketing strategy, and which groups to reach out to
* Method used:

1. Support vector machine
2. Cross validation (with adjusted rand index)
3. Linear and non-homogenous kernels (to model non-linear effects)

* Results:

1. Which variables contribute the most?
2. Which model, in general, is best at predicting?

**Data**

* Describe the following:

1. Where does the data come from?

* Time, place (Portugal), institution (bank)

1. What is the sampling? We know our original exists of 10.000, work on sample of 1000

* We can also increase to more, but want to get all the results at 1000 first since it already takes forever to calculate everything
* What are the transformations we apply to the data?

1. Which do we put into dummy’s?
2. How do we scale the others?
3. What do we do with variables where some variables are missing?

**Method**

* **Describe in 2-3 sentences what a support vector machine is**

1. When is it appropriate to use it?
2. What is the general shape of the loss function?

* What are the type of hinge errors?

1. Trade-offs in terms of punishing outliers, computational efficiency (smooth function)

Should we pick one, or show all of them? If we pick one, I propose Huber since smooth but has the ability to punish outliers more severely

* What is the adjusted rand index?

1. Formula describing it (see: https://en.wikipedia.org/wiki/Rand\_index#Adjusted\_Rand\_index)
2. Why is it better than misclassification or F1 metric?

* What kernels do we use, and why?

I think we only should use linear and non-homogeneous polynomial kernel, since from there we can interpret the weights – do people agree? And more importantly, once we get the weights from these, how to exactly interpret them?

**Results**

* Briefly comment on how it yields the same results as svmmaj (plots in appendix)
* Show plots which hyper parameters (lambda, k\_huber) we ended up with
* 2D plot for absolute and quadratic error, 3D for huber error
* Then show result of applying these hyper parameters to a training and test set (70%/30% split)
* Confusion matrix
* Adjusted Rand Index
* Plot to show which ones were wrong
* Then show what the effect of adding kernels is(we should discuss what we want to show here)

Results

1. CV

* Adjusted Rand Index as comparison
* Which hyperparameters
* Grafiek om score per hyperparameter aan te tonen

1. Performance

* Train (70%) test (30%)
* Adjusted Rand, en confusion matrix
* Plot to show predicted Q and eventual class

1. Interpretation

* Weights
* Include non-linear relations from kernel comparison

1. Kernel comparison

* Linear, Non-Homogenous polynomial
* If improve: shows non-linear dynamics

To research

* What delta for the huber error?
* Pre-processing for data