

# ML Exercise 1

Classification

## **Group 8:**

Alexander Leitner, 01525882

Mario Hiti, 01327428

Peter Holzner, 01426733



## Datasets

- Heart disease
- Amazon
- Congressional voting
- Polish company bankruptcy



## Classifiers

- K-Nearest Neighbors (KNN)
- Random Forest (RF or RFC)
- Multilayer Perceptron (MLP)

Dataset	samples	dimensions	Nominal/ordinal	# of classes	missing values
Heart Disease	303	14	mixed	5	very few
Amazon	750	10000	ordinal	40	no
Companies	6k	64	ordinal	binary	~1,5%
Congress	218	17	nominal/binary	binary	



# Heart Disease

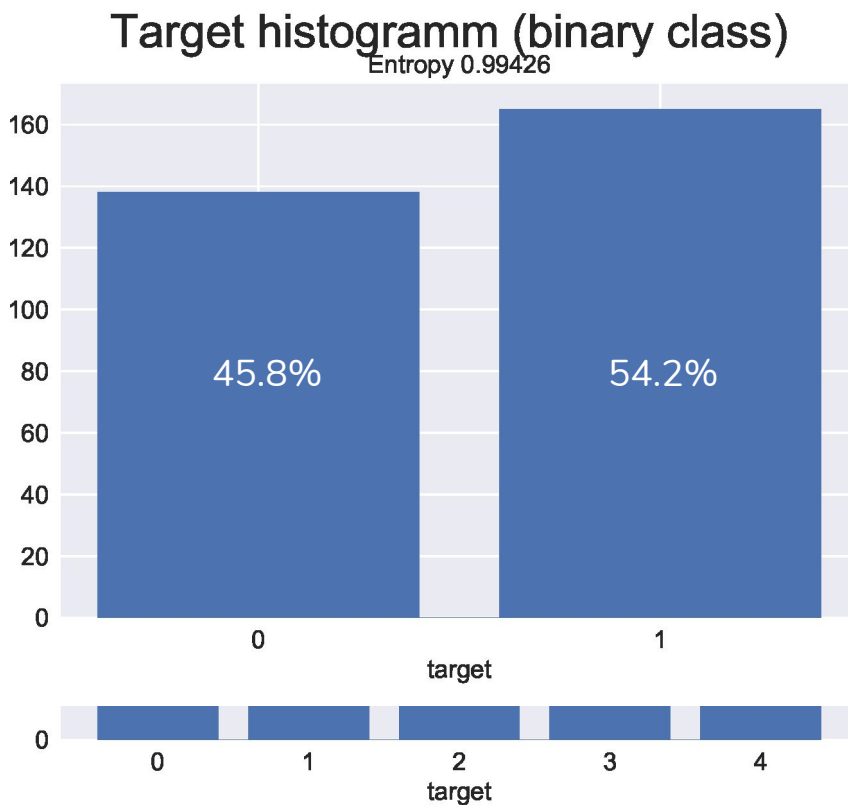


# Heart Disease



Small + low dimensional:

- Samples: 303
- Features: 13 + 1 target
- Target (5 classes):
  - 0... no disease
  - 1,2,3,4... different diseases
  - Uneven target distribution
- Evaluation: Accuracy + F1-score (balanced model)
  - Custom cost matrix: domain expert needed



# Data preparation



Features			
numerical		categorical	
name	range	name	range
age	29 - 77	sex	0, 1
trestbps	94 - 200	cp	0, 1, 2, 3
chol	126 - 564	fbs	0, 1, 2
thalach	71 - 202	restecg	0, 1, 2
oldpeak	0 - 6.2	exang	0, 1
-	-	slope	0, 1, 2
-	-	ca	0, 1, 2, 3, 4
-	-	thal	3, 6, 7 → 0, 2, 1

Features:

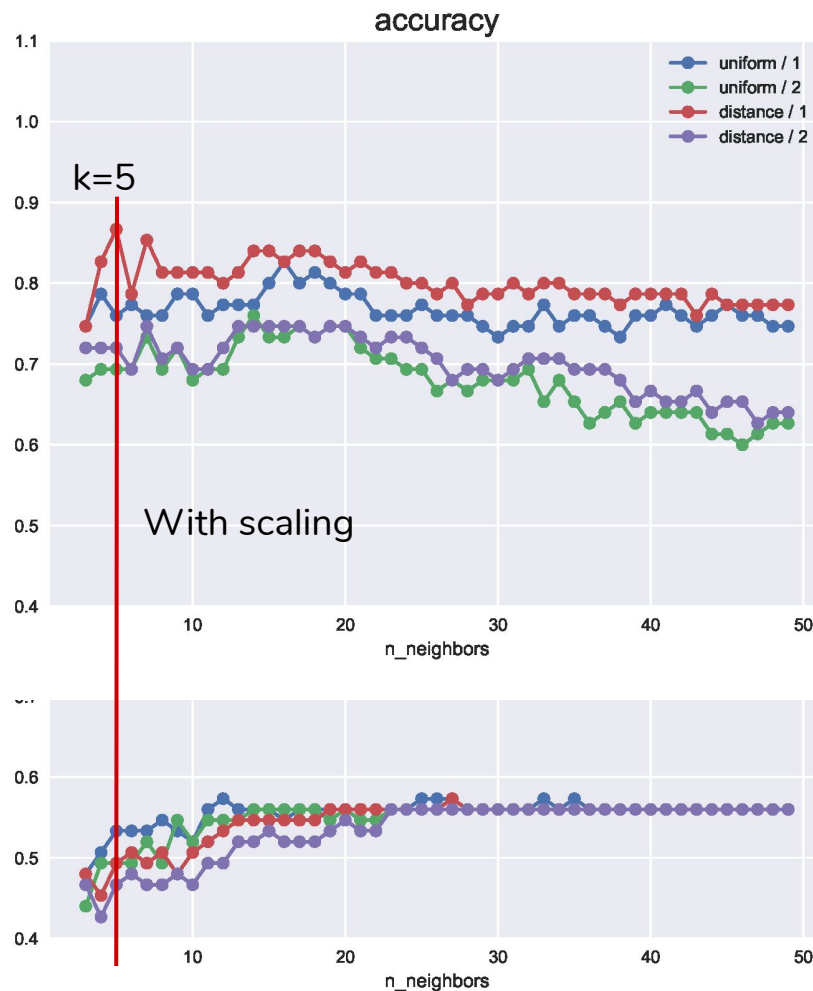
- 5 numerical
  - different ranges → scale
- 8 categorical
  - 2 - 5 encoded categories per feature
- 5 samples with missing values in “ca” or “thal” (entry “?” in csv → hard to spot)
  - 1.5% of samples: no gain if imputed → simply drop samples

# KNN Classifier



## Results:

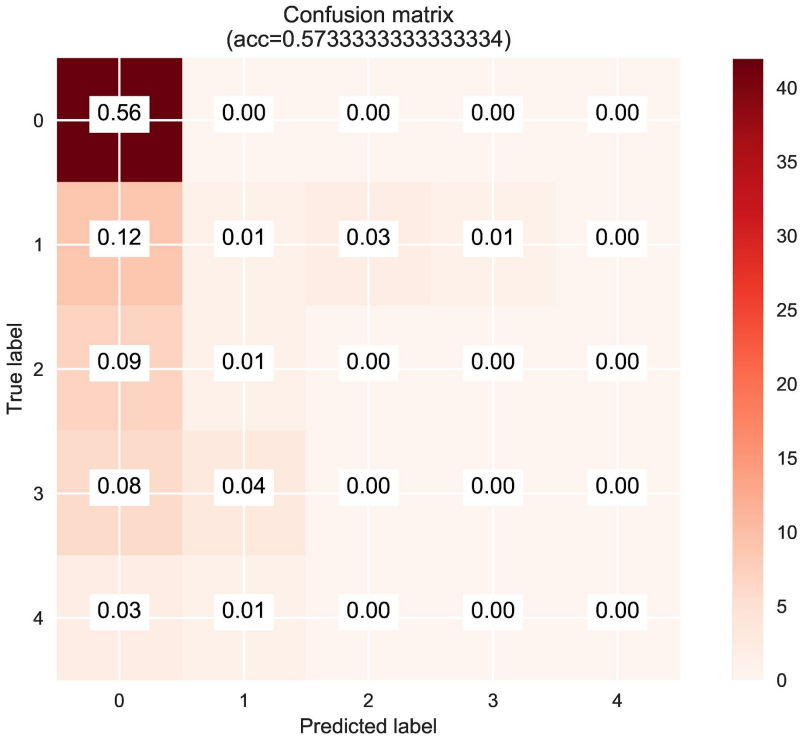
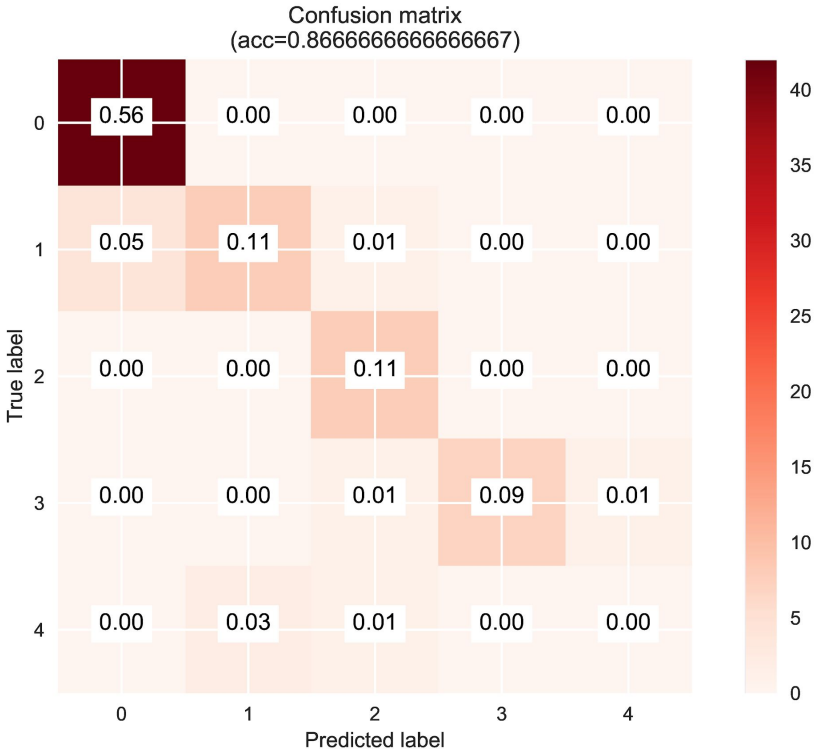
- Unscaled: BAD
  - CM: only guesses 0
- Scaled: OKAY
- norm:  
Manhattan ( $p=1$ ) > Euclidean ( $p=2$ )
- weights:  
distance > uniform



# KNN Classifier - Confusion Matrices (BONUS)



Scaled vs. unscaled

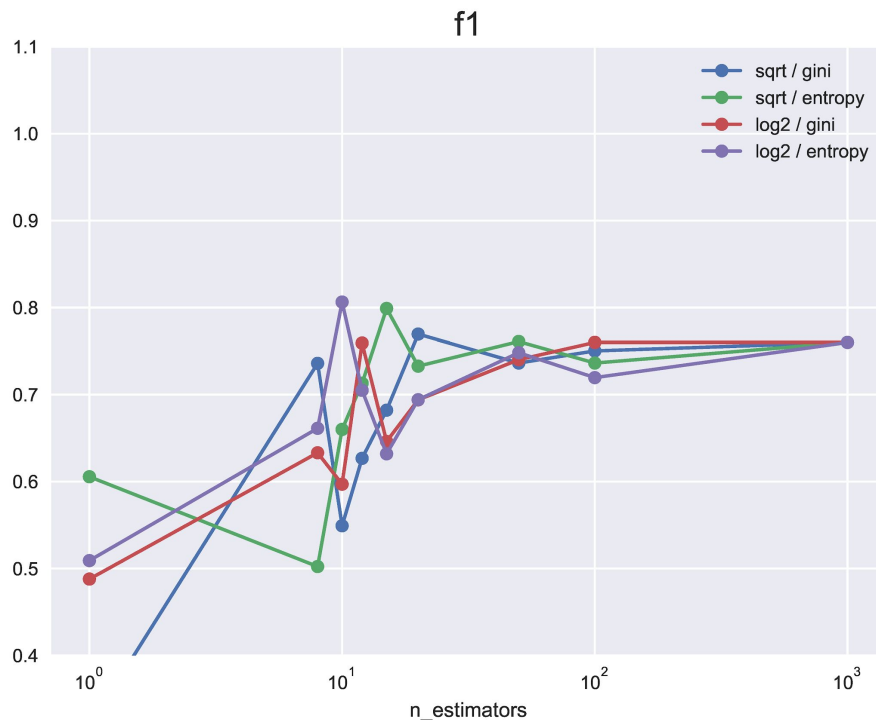
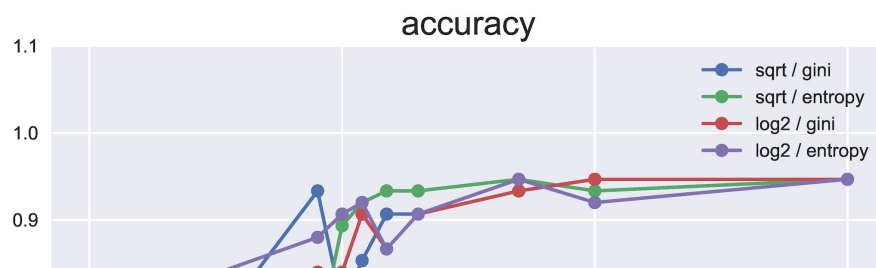


# RF Classifier (BONUS)



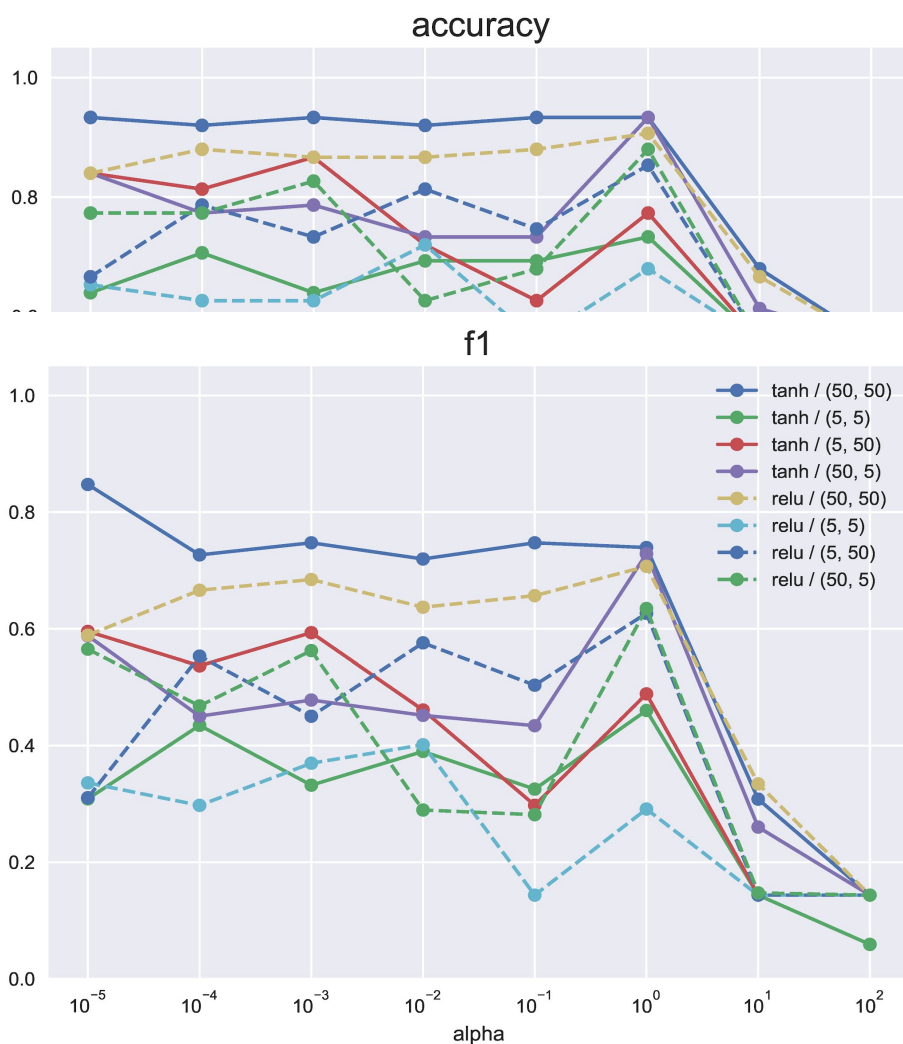
## Results:

- scaled, unscaled: no difference
- n\_estimators:
  - 10 - 20 best
- criterion:
  - entropy > gini
- max features (per split):
  - no difference





- scaled > unscaled data
  - right side: scaled
- Regularization alpha:
  - $1e-5$  best (even smaller?)
- activation:
  - mostly relu > tanh
  - BUT: tanh has best
- hidden layers:
  - bigger = better



# Final decision



**RF**

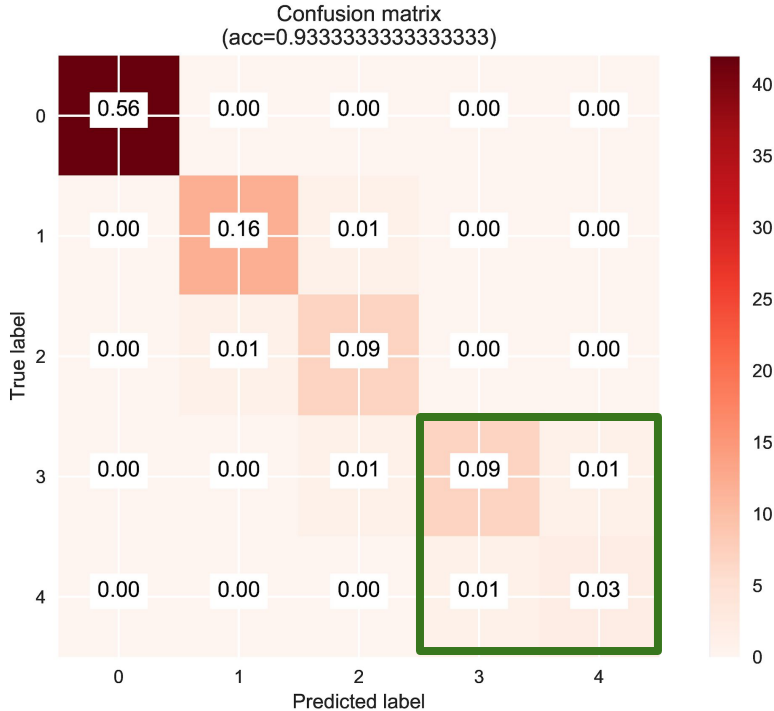
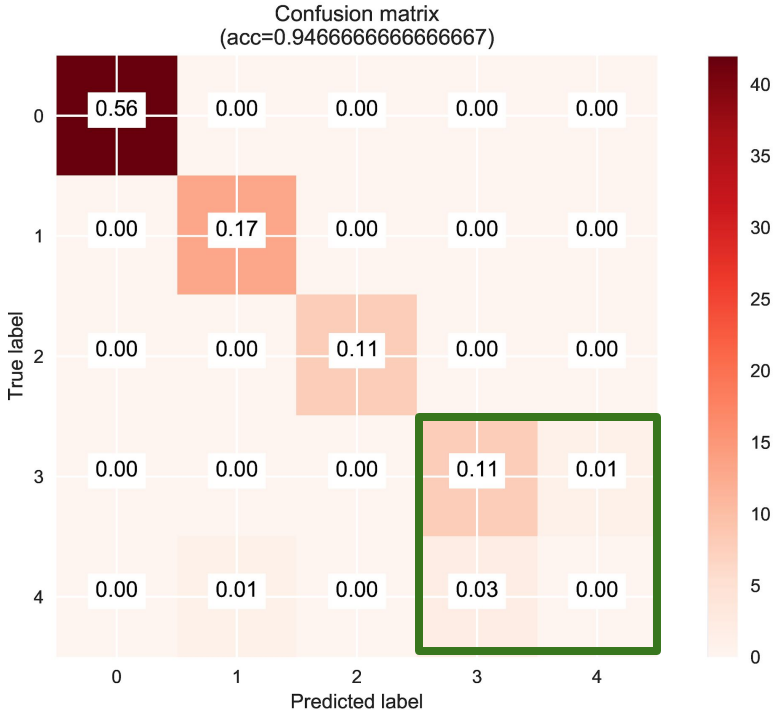
F1=0.801

VS

**WINNER**

**MLP**

F1=0.847





# Amazon

# Amazon



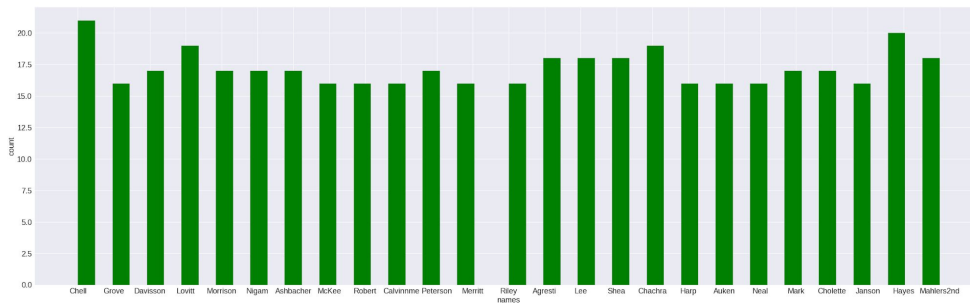
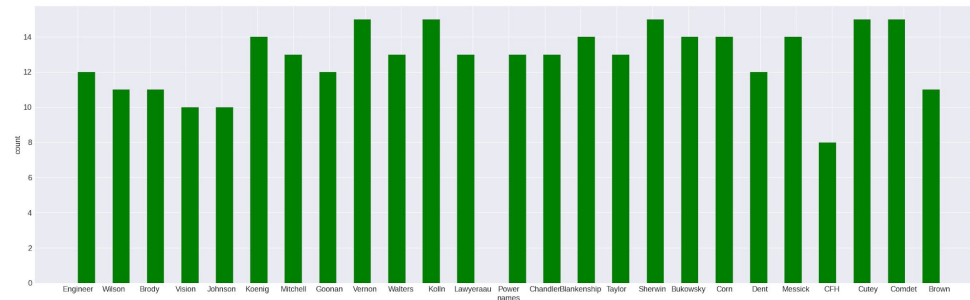
## Description:

Class: 50 different names

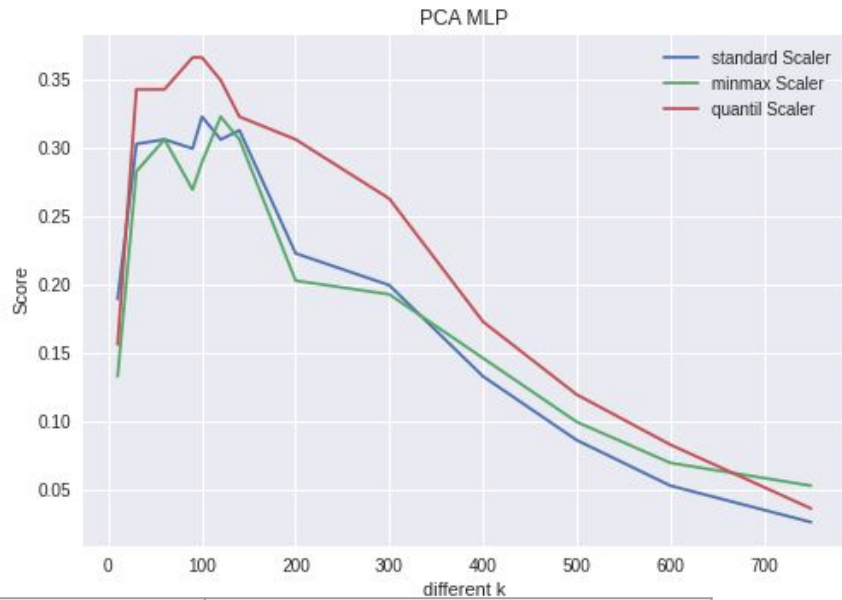
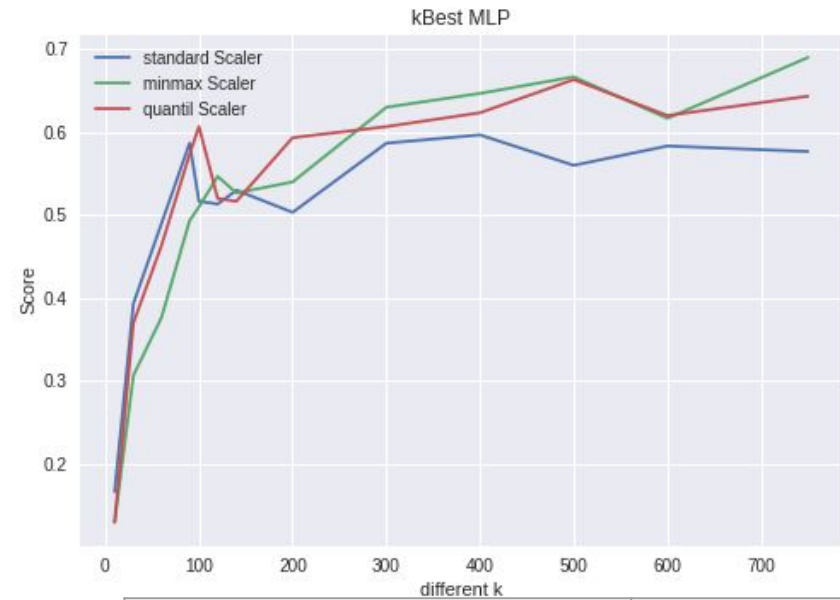
features: numeric values

## Preparation:

feature selection: kBest or PCA



# Feature Selection



Classifier	k-value	scaler
Knn	500	quantil
RFC	300	quantil
MLP	750	MinMax

# Results



## parameter:

KNN n\_neighbors = 8; weights = distance; algorithm = auto

RFC n\_estimators = 1000; max\_features = sqrt; criterion = gini

MLP alpha = 0.01; layers = (100,100); solver = lbfgs; activation = tanh

Classifier	score	F1 score	recall	precision	train-time
Knn	0.477	0.453	0.489	0.537	0.059s
RFC	0.593	0.572	0.612	0.627	6.558s
MLP	0.616	0.608	0.636	0.622	10.969s



# Congressional Voting



# Congressional Voting



## Description:

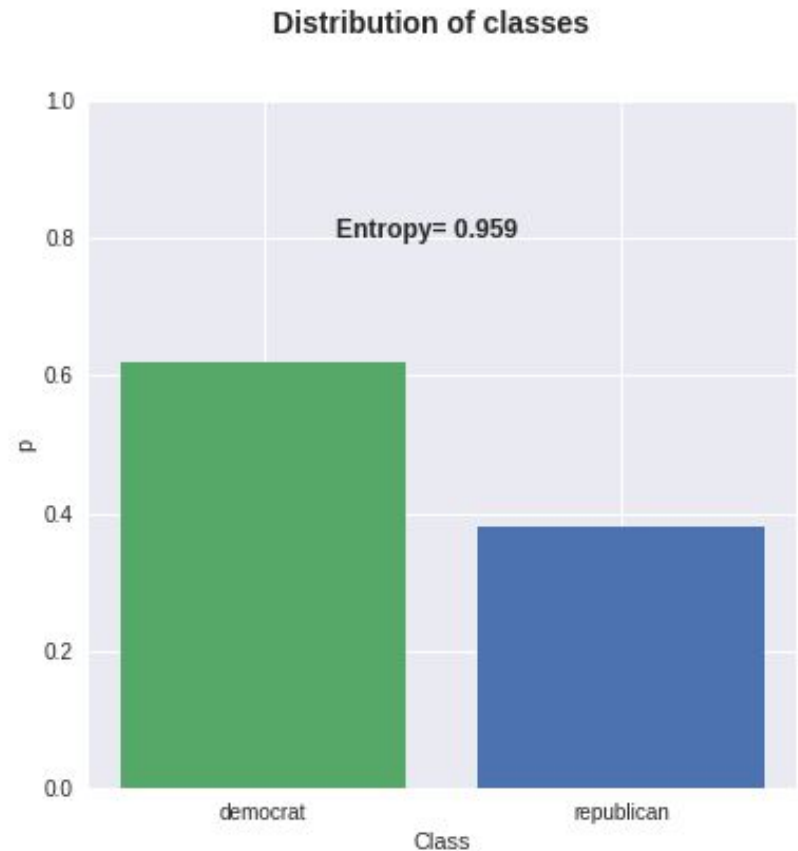
Class: Republicans or Democrats

features: nominal values “y”, “n” and “unknown”

## Preparation:

write “n” -> -1; “y” -> 1; “unknown” -> 0

even distributed -> prevent scaling





# Results KNN



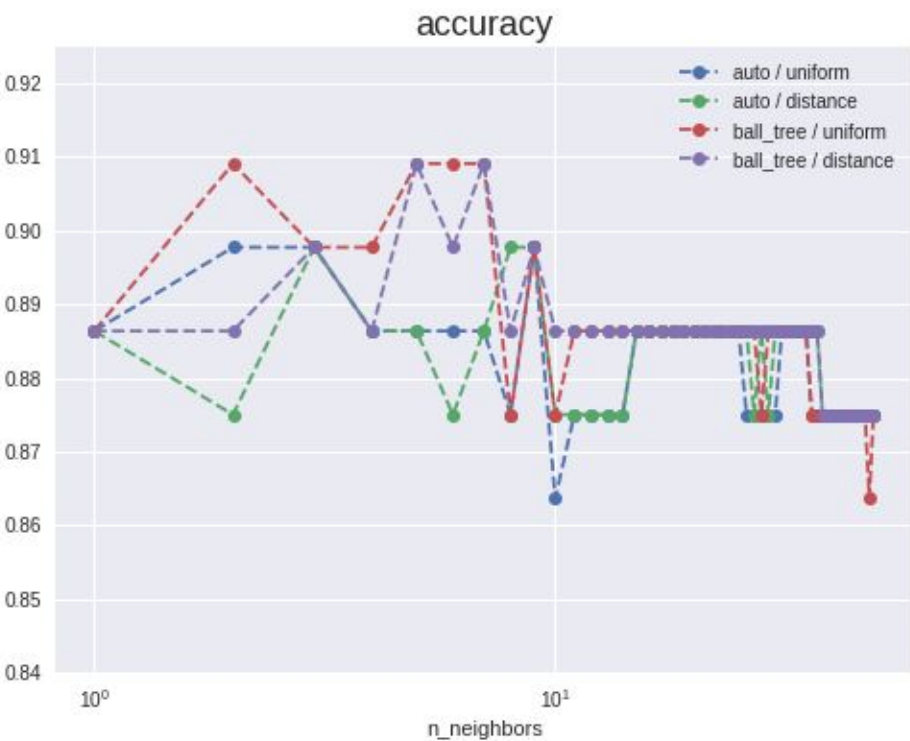
**parameter:**

auto/ball\_free or uniform/distance

**results:**

very short time

a good score



score	F1 score	precision	recall	runtime
0.915909	0.915368	0.916842	0.914155	2.082ms

# Results RM

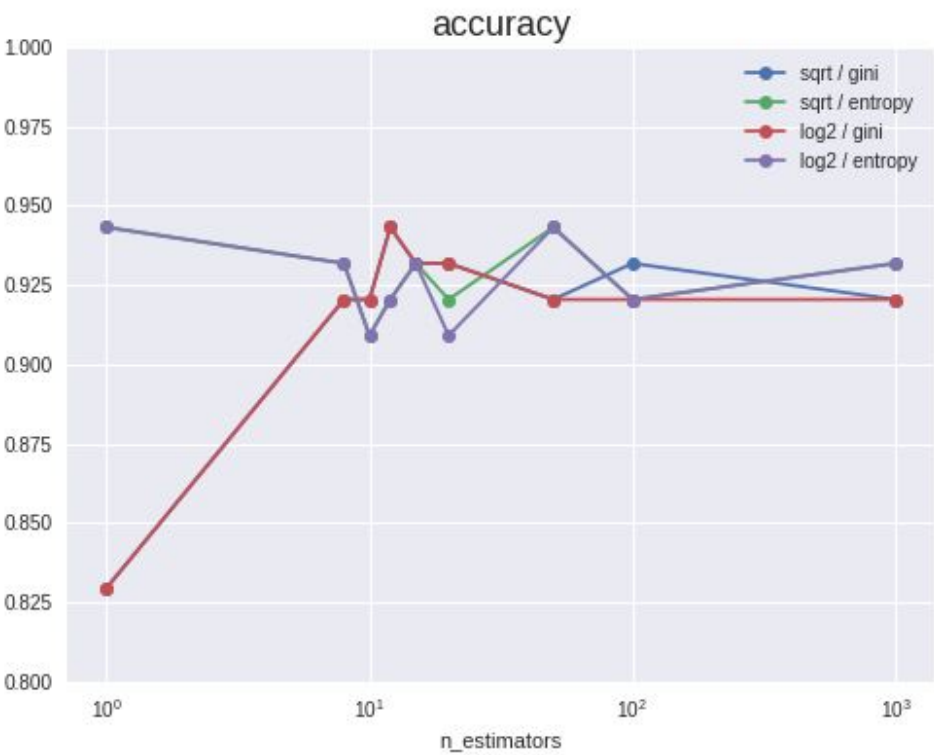


**parameter:**

sqrt/log2 or gini/entropy

**results:**

better score than the KNN



score	F1 score	precision	recall	runtime
0.9476	0.9296	0.9276	0.92445	4.402ms

# Results MLP



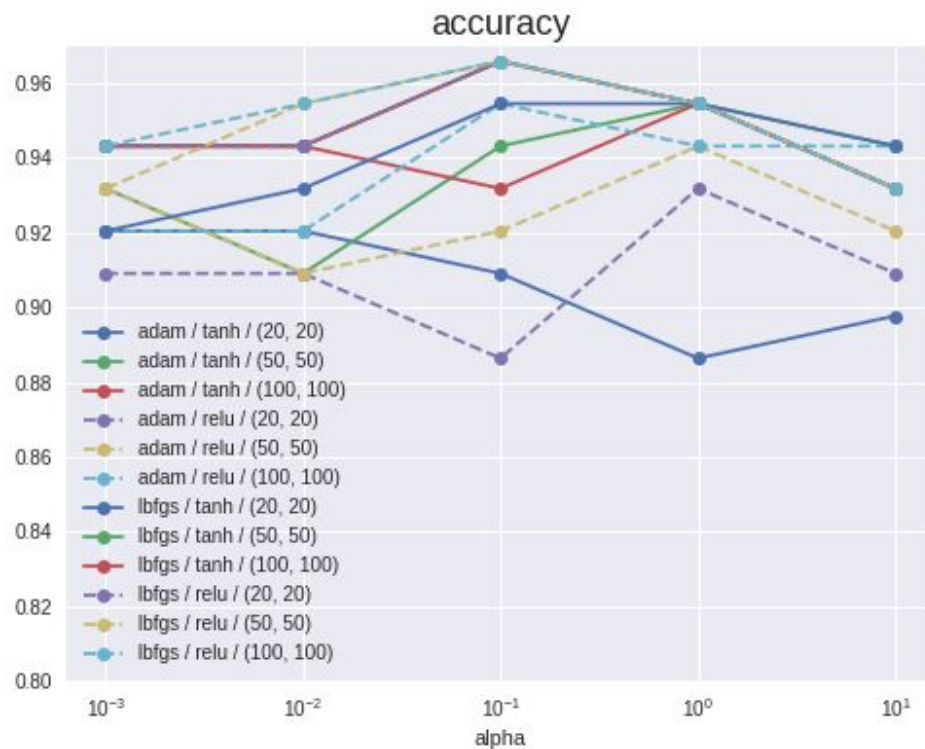
**parameter:**

sqrt/log2 or gini/entropy

**results:**

best score

200 times more intensive



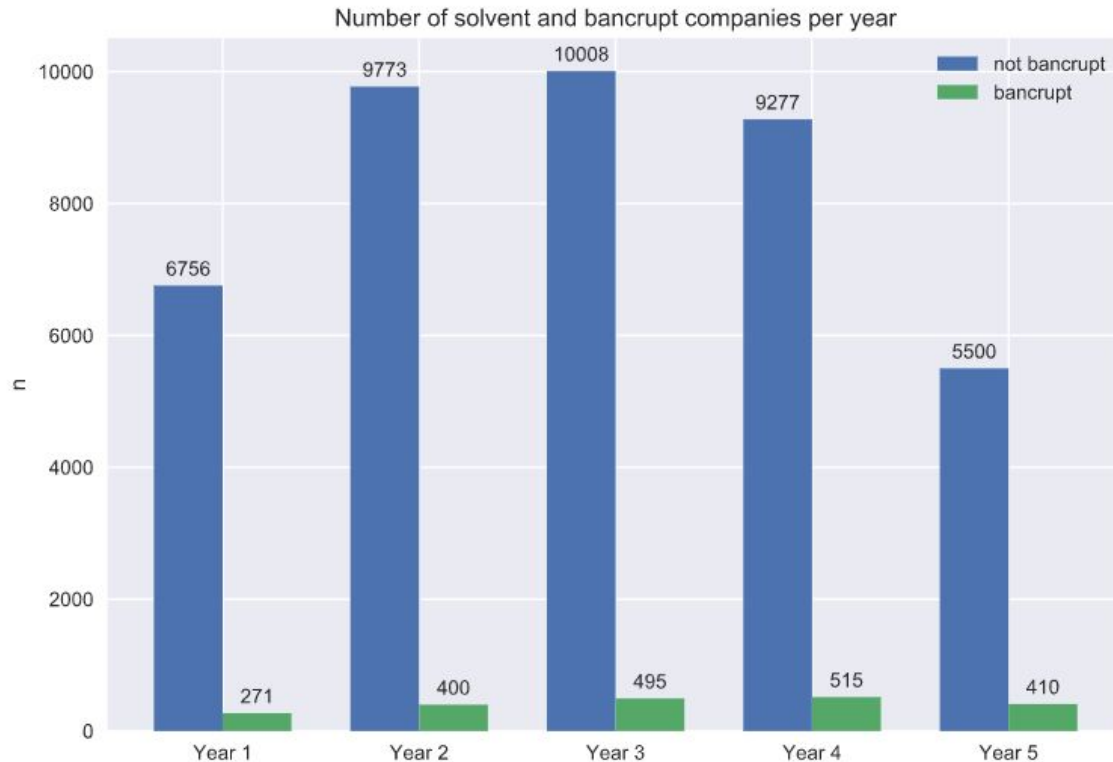
score	F1 score	precision	recall	runtime
0.965	0.9653	0.9641	0.9668	487.08ms



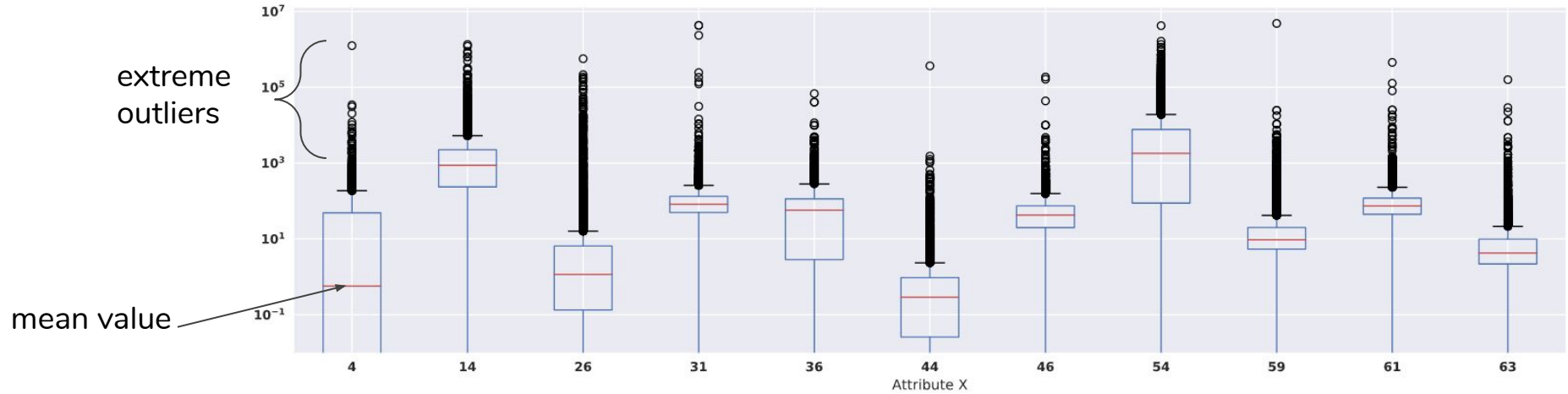
# **Polish Company Bankruptcy**

# Polish company bankruptcy

- 5 data sets
- 5k-10k samples / year
- 64 dimensions
- 2 different classes
- high imbalance of class distribution ~15:1



# Preprocessing



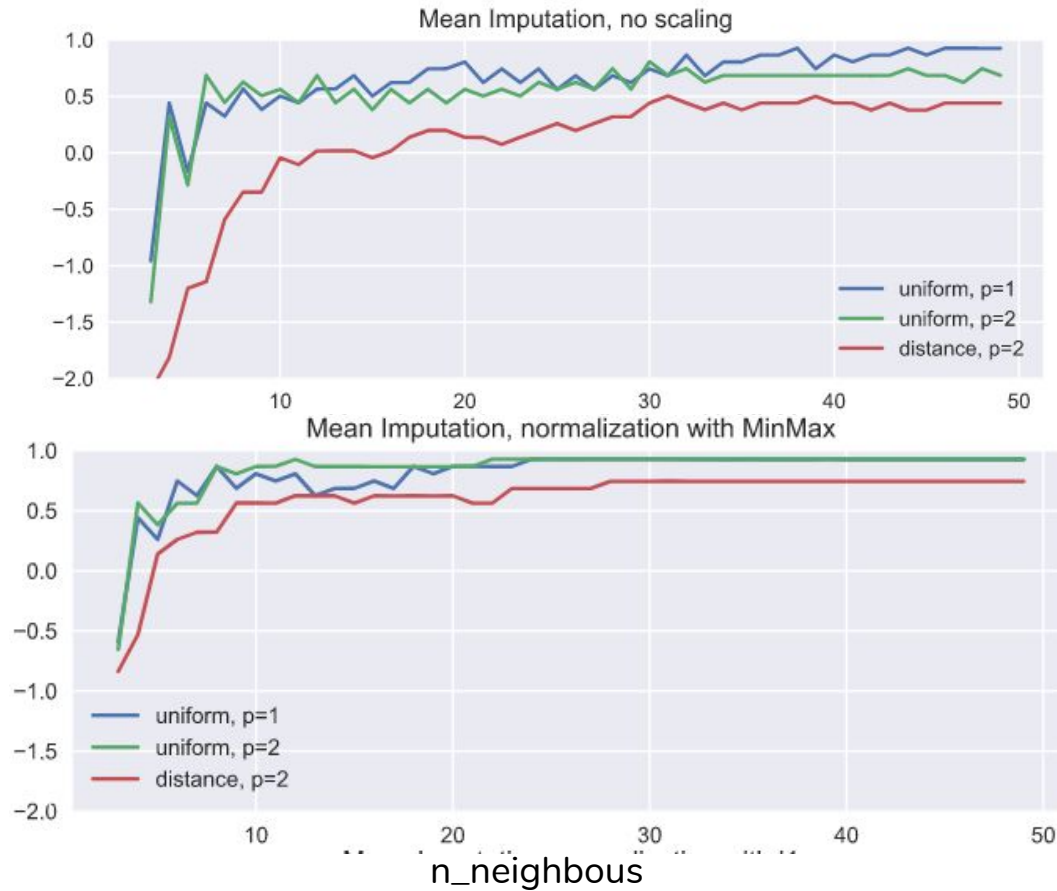
-> Normalization expected to have a positive impact

# Rating classifiers / Cost Matrix

- FP is much worse than FN
- Rating based on normalized cost
  - 1: perfect prediction
  - 0 -1: good model
  - < 0: bad model

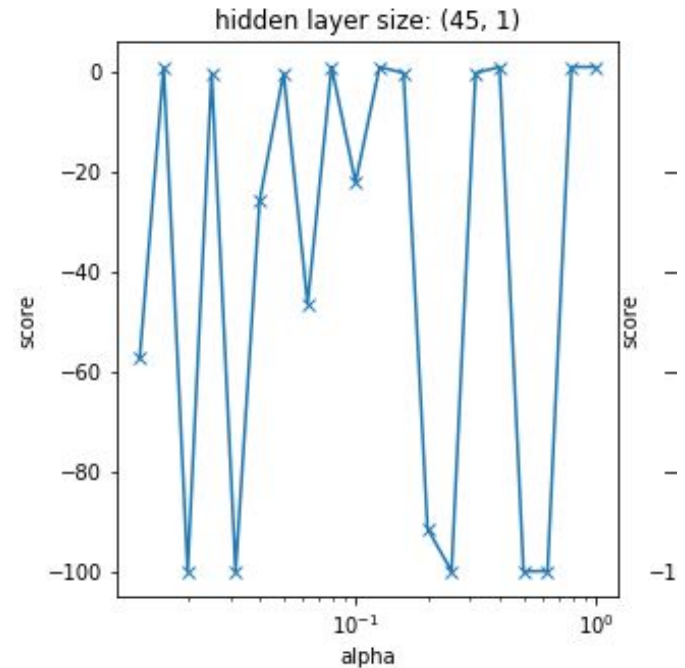
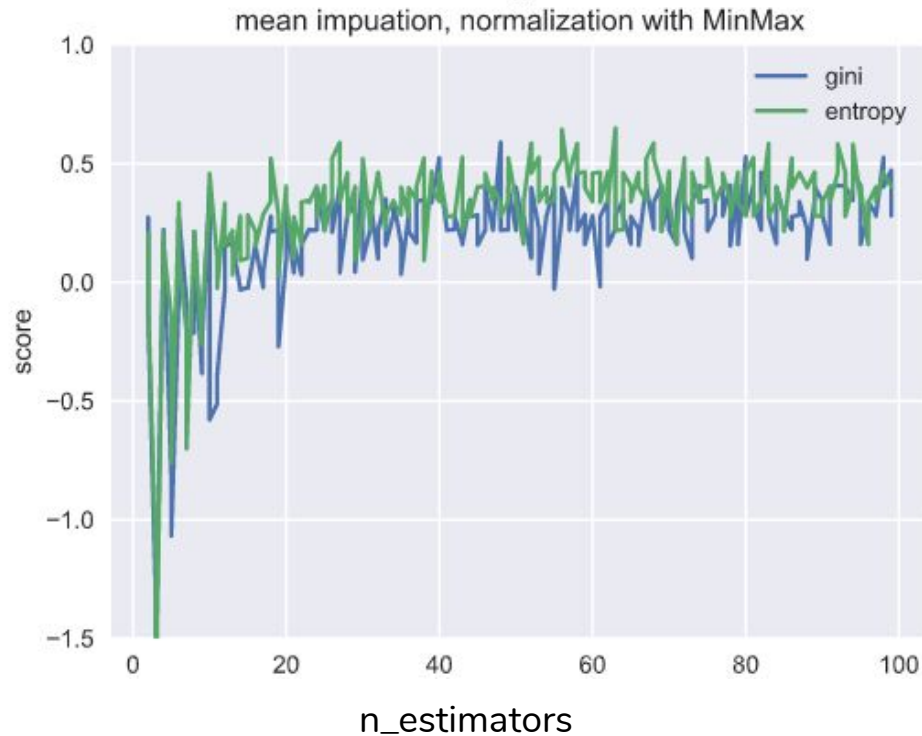
Cost Matrix		actual	
		solvent	bankrupt
pred.	solvent	-1 (TP)	100 (FP)
	bankrupt	1 (FN)	0 (TN)

# Findings - KNN





# Findings - RFC and MLP



# Findings - Conclusion

- KNN reached the highest score consistently
- Classifiers with few FP are favored due to cost matrix
- No model with high recall AND precision was found
- => best models have low recall / high precision

Cost Matrix		actual	
		solvent	bankrupt
pred.	solvent	-1 (TP)	100 (FP)
	bankrupt	1 (FN)	0 (TN)

	accuracy	score	F1 score	precision	recall	runtime
k-nearest neighbors	0.9362	0,9303	0,1374	0.9000	0.0743	0,142s
random forest	0.9334	0.6864	0.1060	0.6363	0.0578	0,664s
multi-layer perceptron	0.9328	0.8674	0.0480	0.7500	0.0247	3,811s



# Scores

$$\text{ACC} := \frac{\text{TP} + \text{TN}}{\text{n\_samples}}$$

$$\text{Recall} := \frac{\text{TP}}{\text{TP} + \text{FN}}$$

punishes false negatives  
ability to find all positive samples

$$\text{F1} := \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

$$\text{Precision} := \frac{\text{TP}}{\text{TP} + \text{FP}}$$

punishes false positives  
ability to not mislabel negative samples