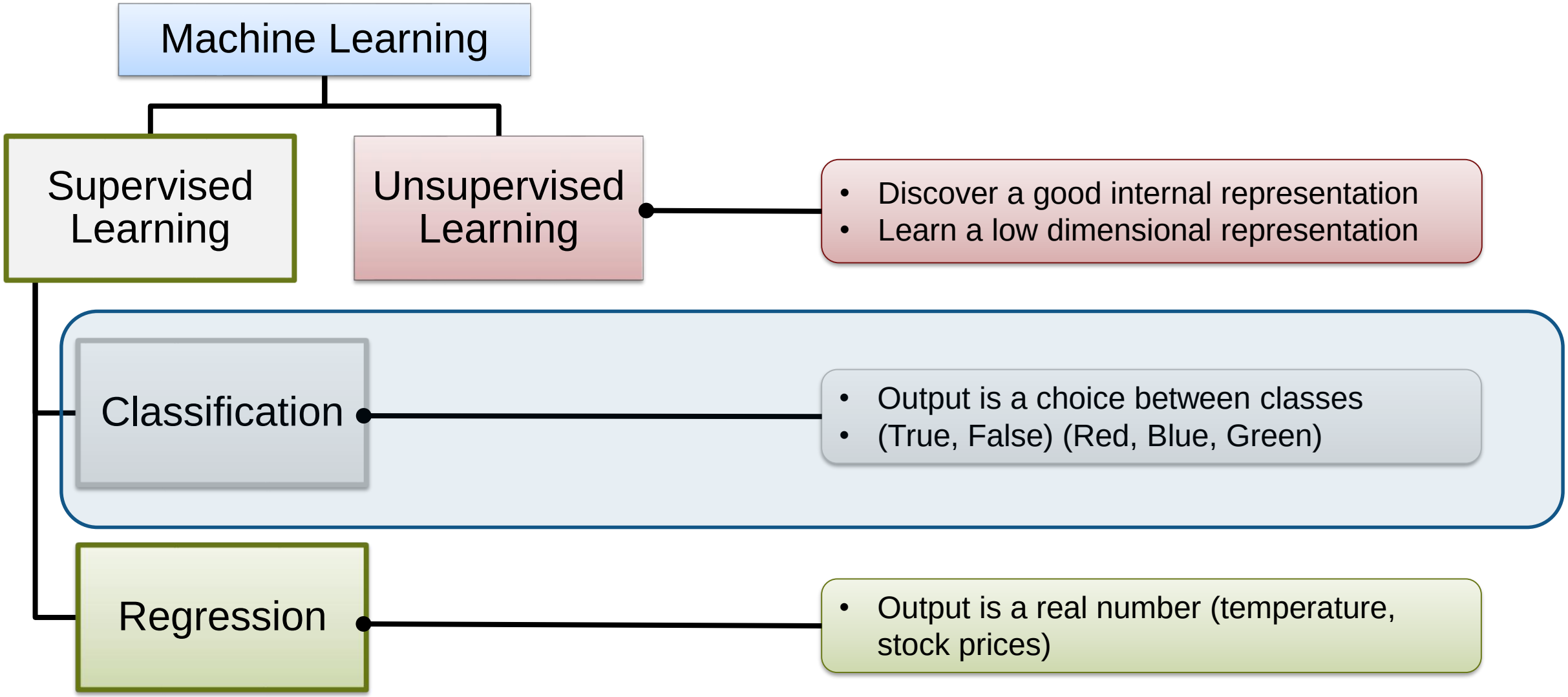
MATLAB EXPO 2016

Develop Predictive Maintenance Algorithms using MATLAB

Dr. Sarah Drewes, MathWorks Consulting Services

Different Types of Learning



Classification in Predictive Maintenance

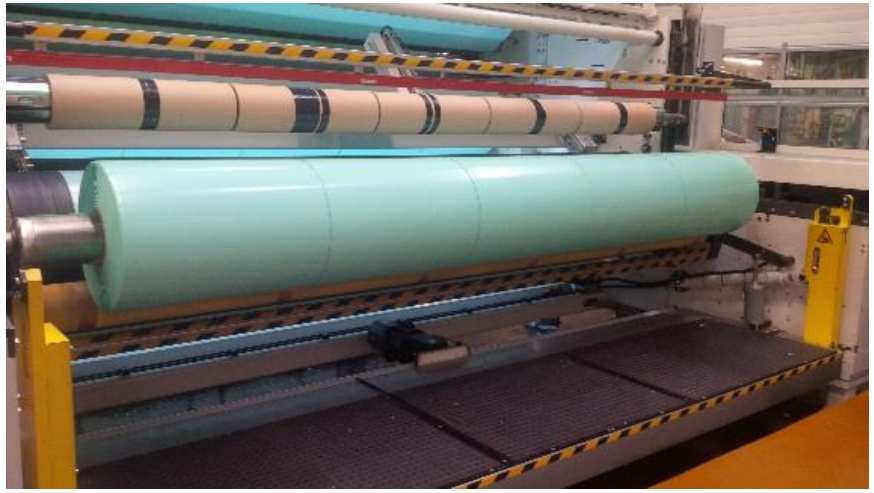
- Parameters/Predictors: Sensor data, control settings
- Classes/States: Failure states, time horizon until failure/ material fatigue

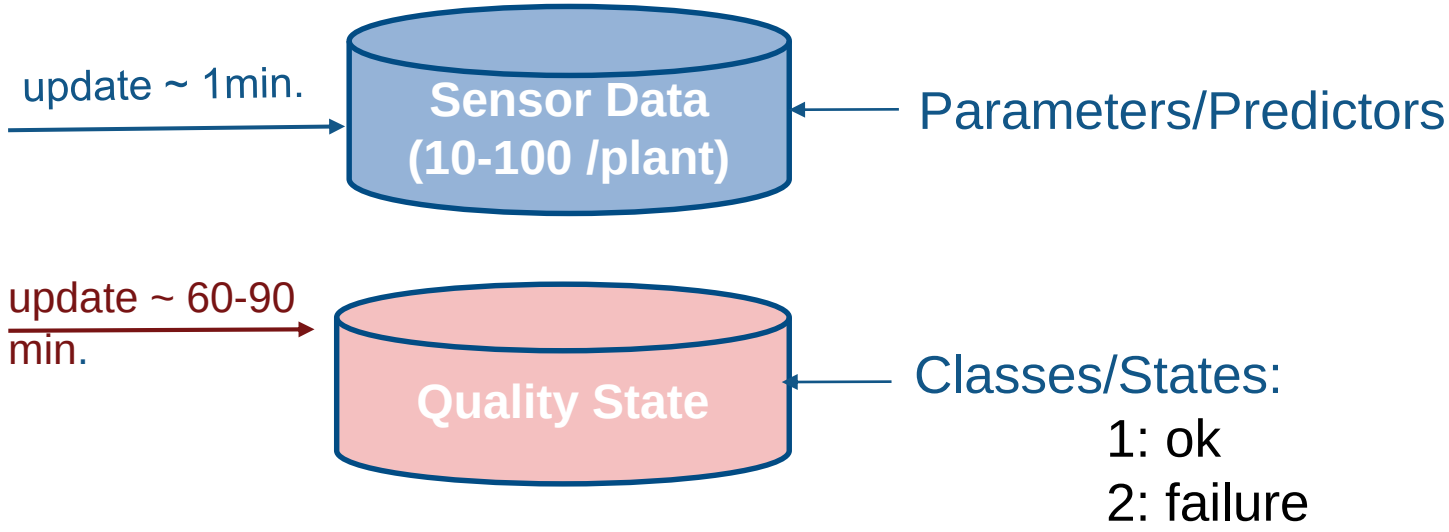
Goal: Predict failure from sensor data

Prerequisites:

- Machine-readable data format
- Sufficient historical data containing meaningful information

Classification model generation @MONDI Gronau

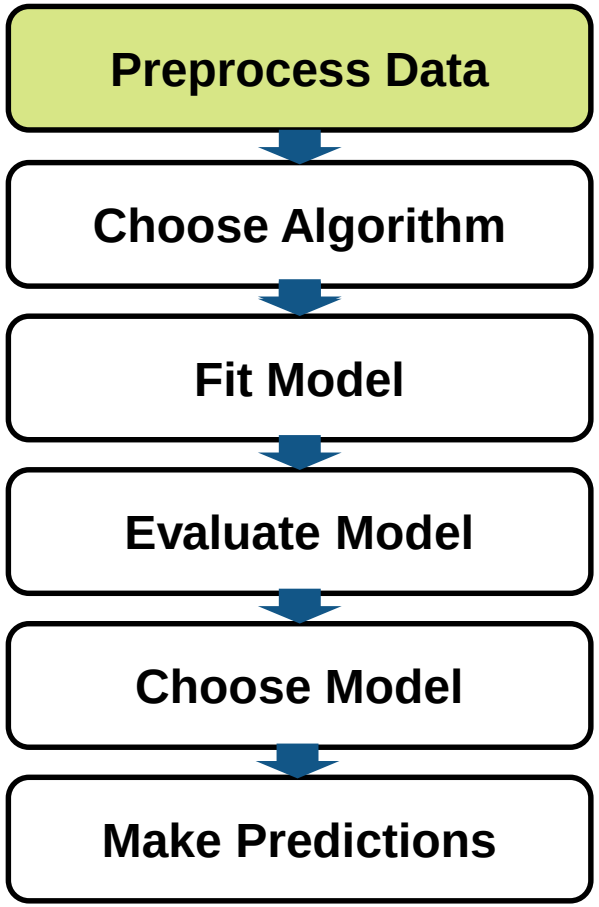




Which sensor measurements indicate machine failure?

Classification model generation

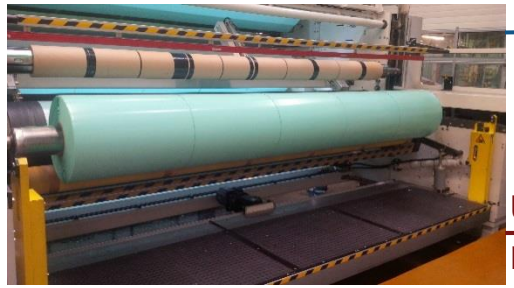
Basic Workflow



Classification model generation- Prepare data

- Preprocess sensor data: clean invalid data, disregard constant values, identify data types

- Aggregate per time stamp



update ~ 1min.

update ~ 60-90 min.

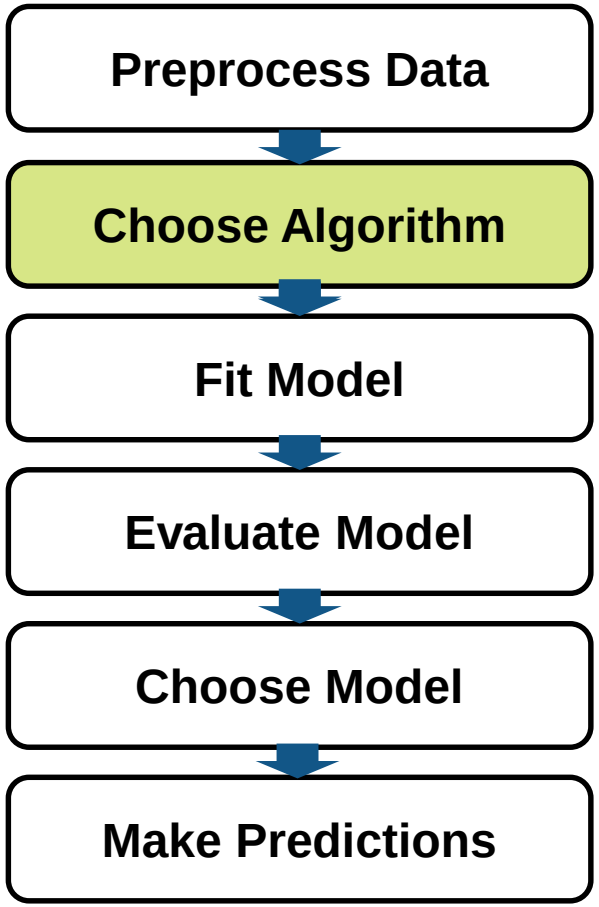
Sensor Data (10-100 /plant)

Quality State

1 TIMESTAMP	2 PARAMETER										3 STATE
'2015-07-14 00:49:12.0'	160	160	160	160	1000	7	1000	9	33	32	1
'2015-07-14 00:50:12.0'	160	160	160	160	1000	8	1000	10	33	32	1
'2015-07-14 00:51:13.0'	160	160	160	160	1000	8	1000	10	33	32	1
'2015-07-14 00:52:12.0'	160	160	160	160	1000	8	1000	10	33	32	1
'2015-07-14 00:53:12.0'	160	160	160	160	1000	8	1000	11	33	32	2
'2015-07-14 00:54:12.0'	160	160	160	160	1000	8	1000	12	33	32	2
'2015-07-14 00:55:12.0'	160	160	160	160	1000	8	1000	10	33	32	2

Classification model generation

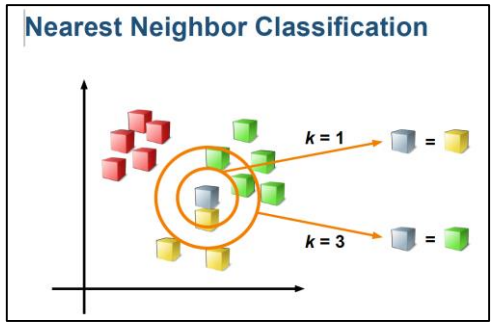
Basic Workflow

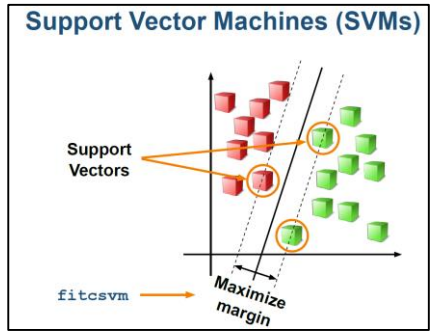


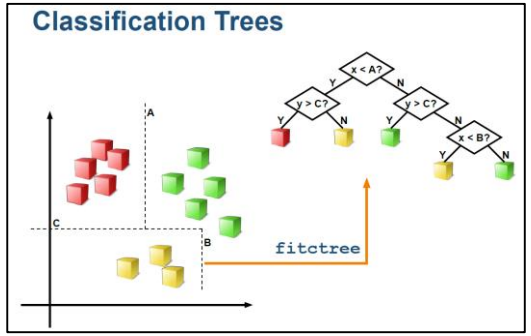
Classification model generation
Choose algorithms

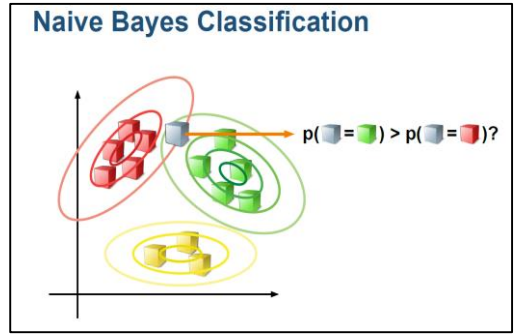
Possible Classification Methods

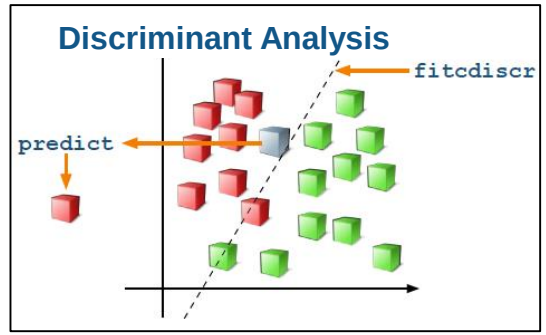
Statistics and Machine Learning



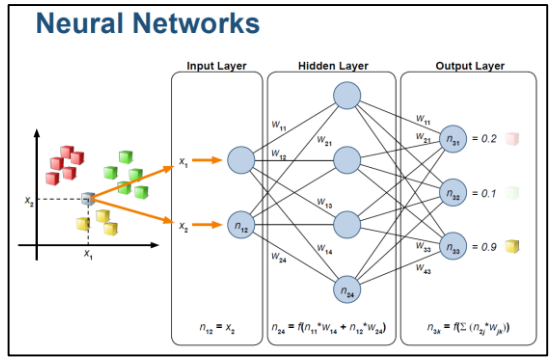








Neural Network



Classification model generation
Choose an algorithm

- Distinguish 'categorical' (= discrete) and other (= continuous) predictors
- A priori analysis of data, e.g., test for normal distribution
- Reduce dimension of predictor variables, e.g., principal component analysis (PCA)
- Use ensemble learning to reduce sensitivity of learning algorithms, e.g. TreeBagger for classification trees

Classification model generation
Choose an algorithm

Algorithm	Function	Categorical Predictors?	Data	Functions to Examine Data	Notes
Nearest Neighbor	`fitcknn`	Y (but not both)	Normalize (distance-based calculation).	`pdist` `pdist2`	Better results in lower dimensions. High memory usage.

Naive Bayes	`fitcnb`	Y	Assumes normal distributions (can specify kernel for nonnormal).	`probplot`
				`jbtest`
				`ksdensity`
					Popular for high dimensional problems. Computationally efficient. Widely used for text classification.

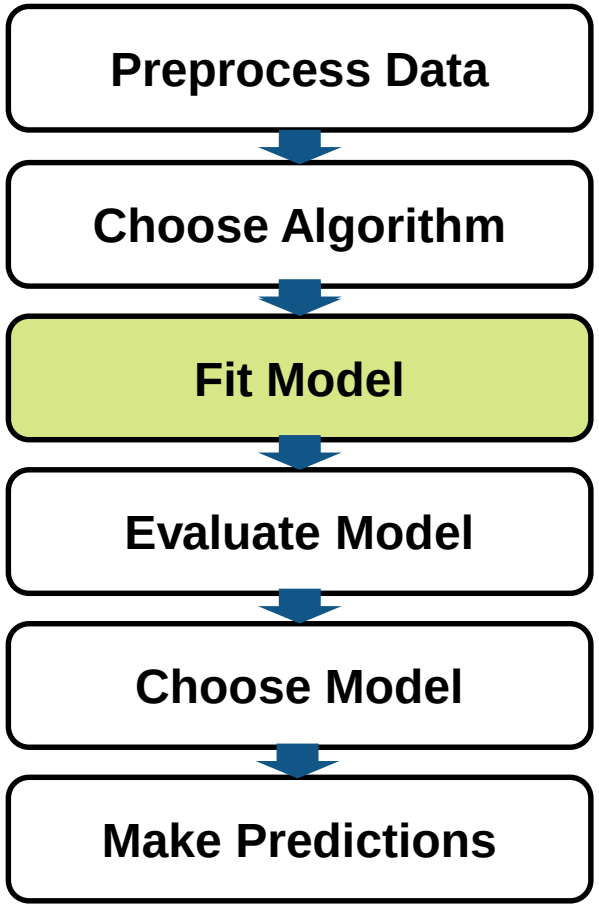
Discriminant Analysis	`fitcdiscr`	N	Multivariate normal distribution by class.	`cov`
				`vartestn`
				`anova1`
				`kruskalwallis`
					Determines mean and covariance for each class. Can specify linear or quadratic discriminant type.

Trees	`fitctree` `fitrtree`	Y	Any arrangement. Binary comparisons and structure of tree can be examined/adjusted.	`view`	Computationally efficient. Highly sensitive to training data.
SVM	`fitcsvm` `fitcecoc`	N	Linearly separable hyperplane (can specify nonlinear kernel).	`ksdensity`	Can specify nonlinear kernel distributions. Can adjust optimization parameters.

Neural Network	`patternnet`	N	Transpose (columns are observations). All data must be numeric.	`dummyvar`
				`plotconfusion`
				`plotroc`
					Use `dummyvar` for categorical classes. Models are available as Simulink® blocks.

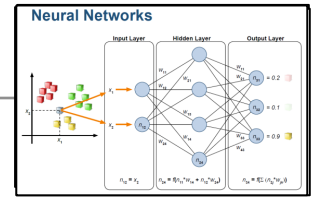
Classification model generation

Basic Workflow



Classification model generation
Fit model

Fit model based on historic data

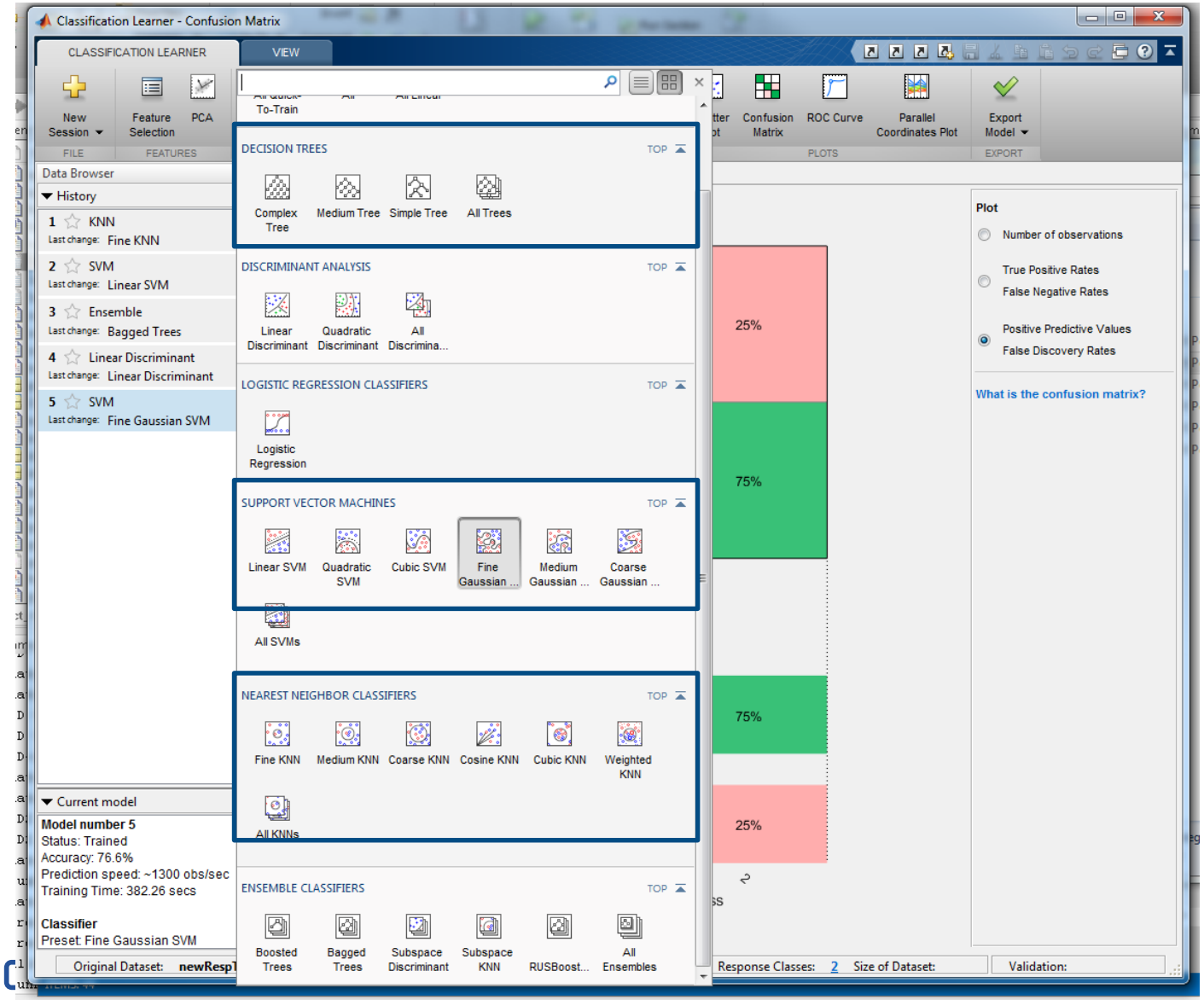



```
PredictionModel = fitctree(PARAMETER, STATE)
```

Training Data, e.g. 70% of historic data

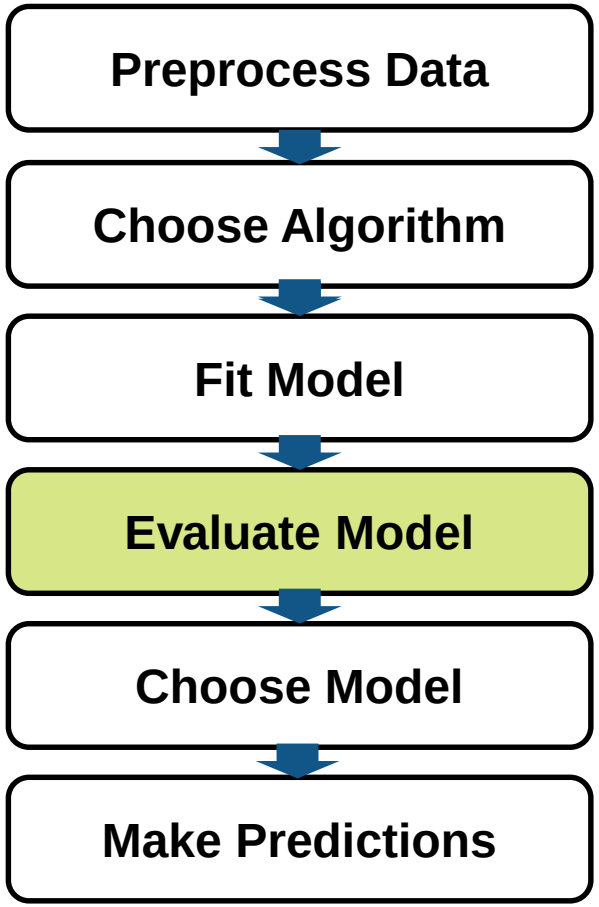
1 TIMESTAMP	2 PARAMETER										3 STATE
'2015-07-14 00:49:12.0'	160	160	160	160	1000	7	1000	9	33	32	1
'2015-07-14 00:50:12.0'	160	160	160	160	1000	8	1000	10	33	32	1
'2015-07-14 00:51:13.0'	160	160	160	160	1000	8	1000	10	33	32	1
'2015-07-14 00:52:12.0'	160	160	160	160	1000	8	1000	10	33	32	1
'2015-07-14 00:53:12.0'	160	160	160	160	1000	8	1000	11	33	32	2
'2015-07-14 00:54:12.0'	160	160	160	160	1000	8	1000	12	33	32	2
'2015-07-14 00:55:12.0'	160	160	160	160	1000	8	1000	10	33	32	2

Classification model generation
Fit model



Classification model generation

Basic Workflow



Classification model generation
Evaluate model

predictedState = PredictionModel(Parameter)

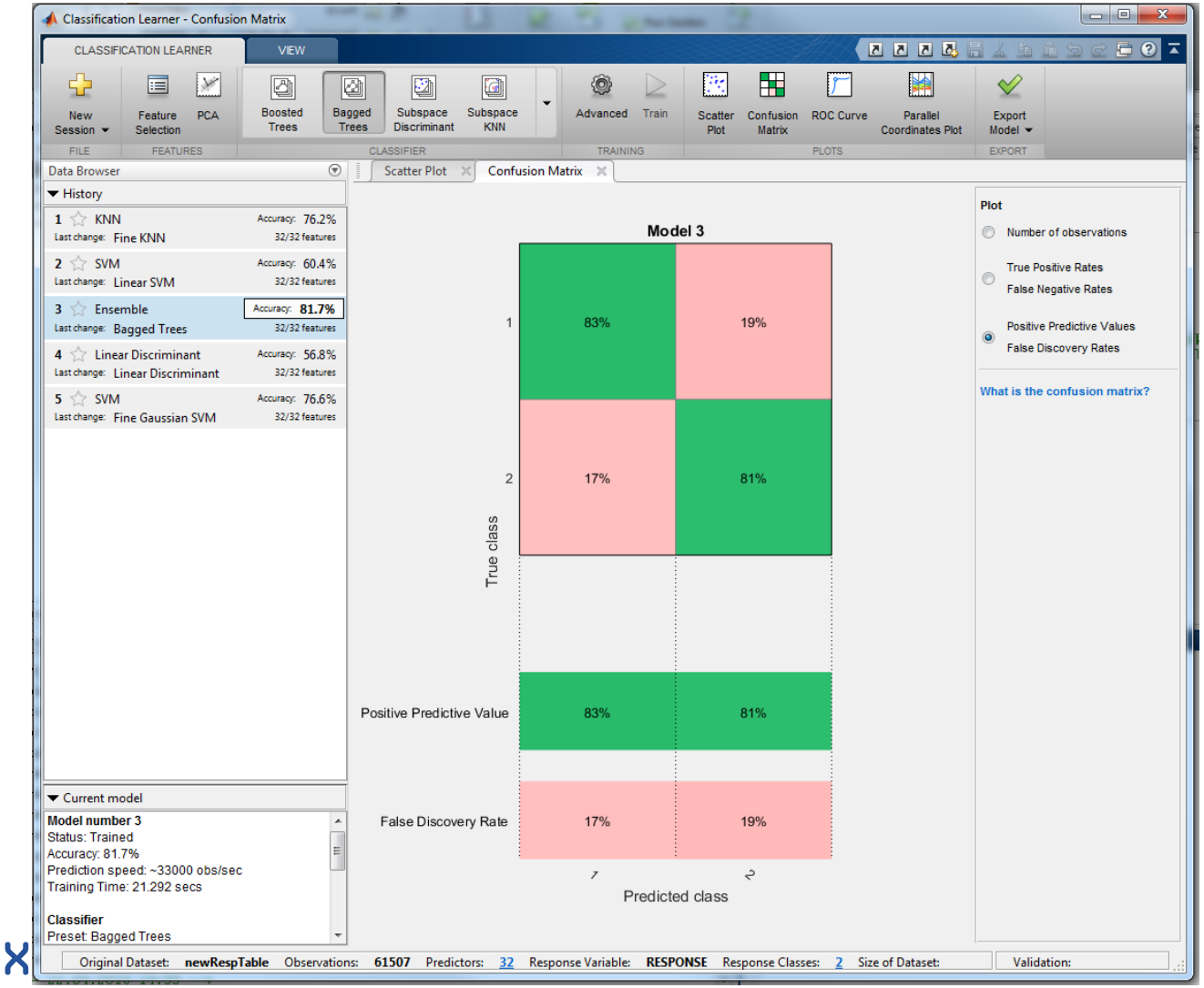
PredictionModel

Validation Data, e.g. 30% of historic data

1 TIMESTAMP	2 PARAMETER										3 STATE	predictedState
'2015-07-14 00:49:12.0'	160	160	160	160	1000	7	1000	9	33	32	1	1
'2015-07-14 00:50:12.0'	160	160	160	160	1000	8	1000	10	33	32	1	1
'2015-07-14 00:51:13.0'	160	160	160	160	1000	8	1000	10	33	32	1	1
'2015-07-14 00:52:12.0'	160	160	160	160	1000	8	1000	10	33	32	1	1
'2015-07-14 00:53:12.0'	160	160	160	160	1000	8	1000	11	33	32	2	2
'2015-07-14 00:54:12.0'	160	160	160	160	1000	8	1000	12	33	32	2	2
'2015-07-14 00:55:12.0'	160	160	160	160	1000	8	1000	10	33	32	2	1

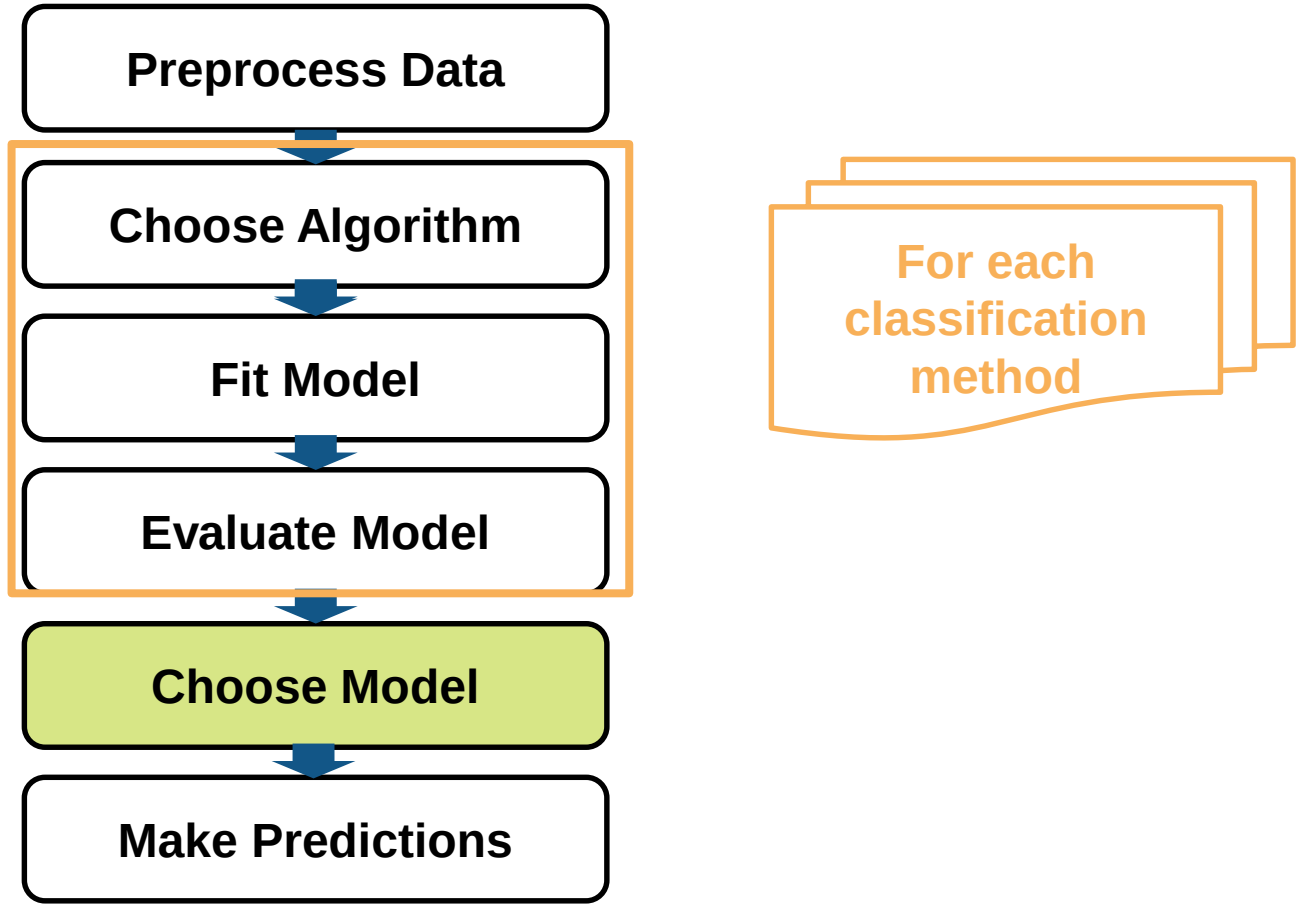
Misclassification rate 1 of 7: 14.28 %
Accuracy: 85.72 %

Classification model generation
Evaluate model - using Classification Learner App



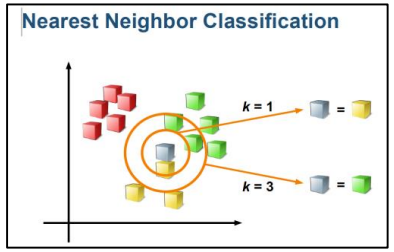
Classification model generation

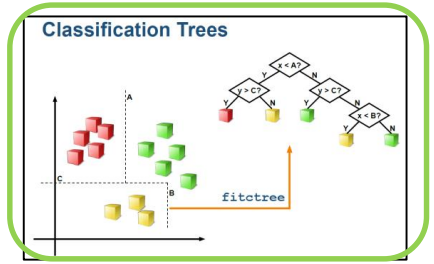
Basic Workflow

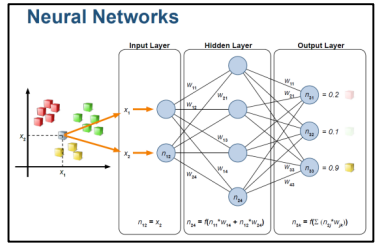


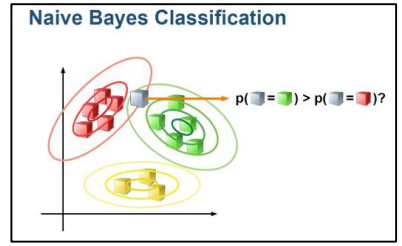
Classification model generation
Choose model

Choose Model with best misclassification rate



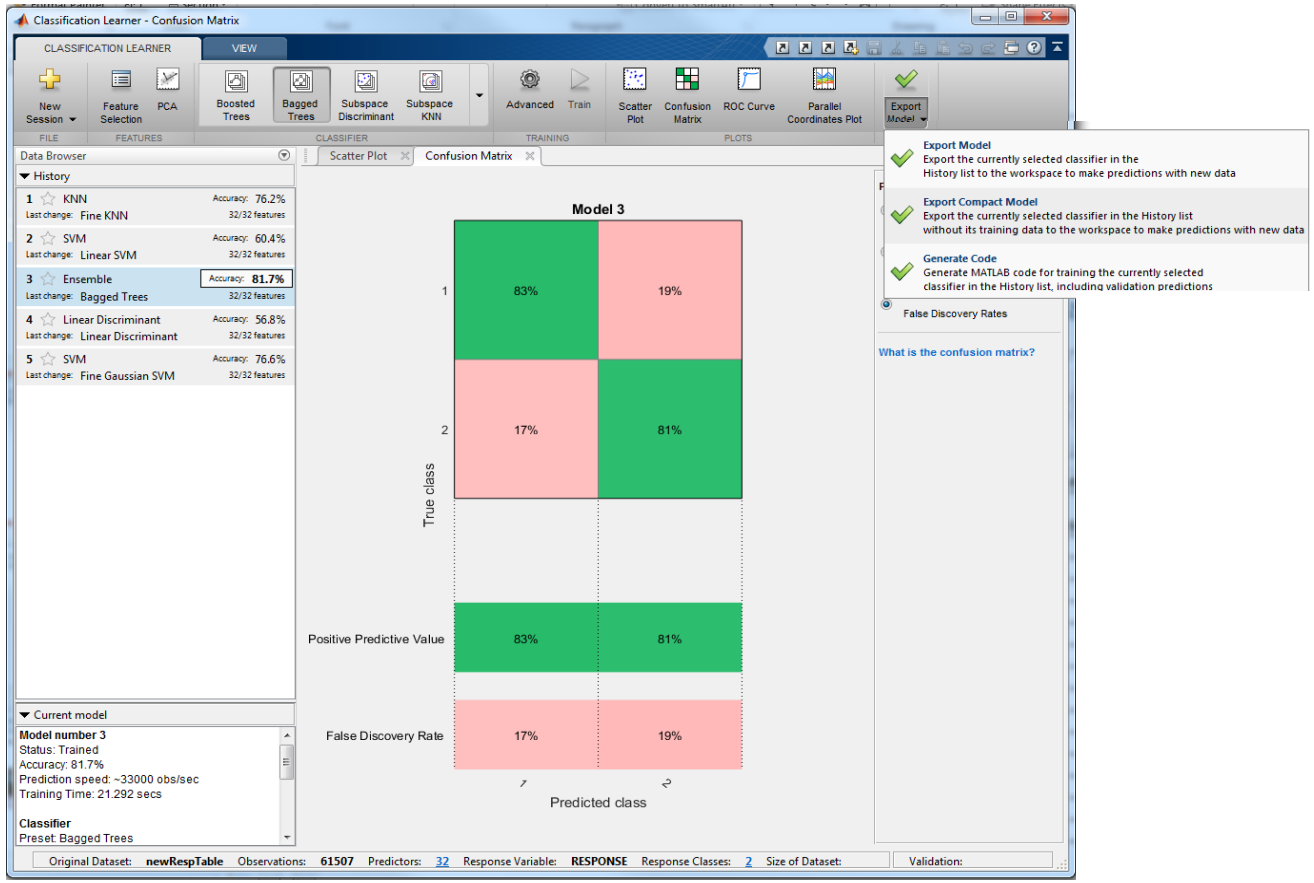






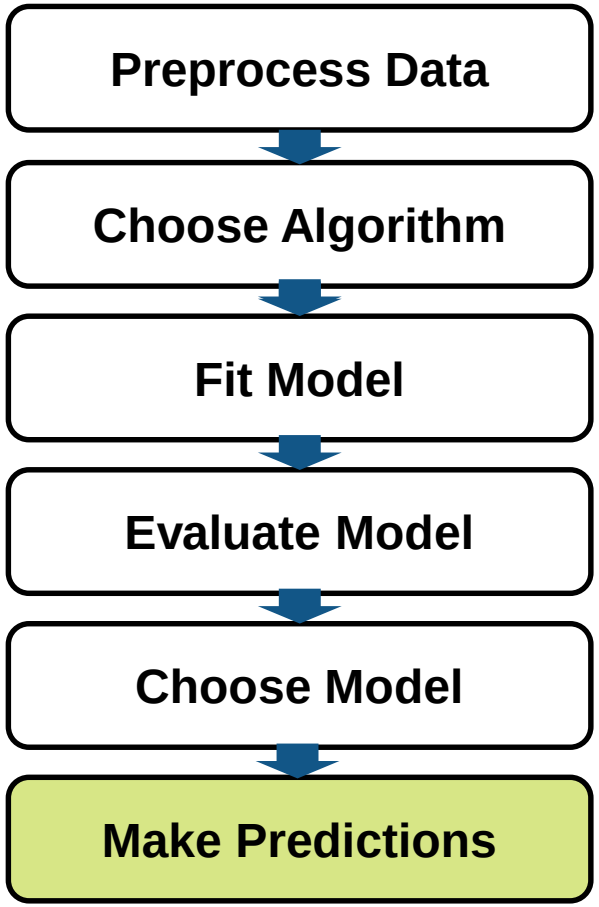
	NearestNeighbour	TreeBagger	NeuralNetwork	NaiveBayes
	Misclassification % (Mean)	*Misclassification % (Mean)*	*Misclassification % (Mean)*	*Misclassification % (Mean)*
M151	24%	2%	8%	10%
M152	44%	5%	23%	13%
M153	23%	2%	13%	13%
M156	12%	2%	3%	9%
M157	11%	1%	10%	8%
M158	29%	2%	14%	17%
M159	21%	0%	3%	2%
M181	1%	0%	1%	2%

Classification model generation
Choose model



Classification model generation

Basic Workflow



Predictive monitoring at MONDI Gronau - Use the predictive model

Predict current machine states during operation.

update ~ 1min.

Sensor Data (10-100 /plant)

update ~ 60-90 min.

Quality State

1 TIMESTAMP	2 PARAMETER										3 STATE
'2015-07-14 00:49:12.0'	160	160	160	160	1000	7	1000	9	33	32	1
'2015-07-14 00:50:12.0'	160	160	160	160	1000	8	1000	10	33	32	1
'2015-07-14 00:51:13.0'	160	160	160	160	1000	8	1000	10	33	32	1
'2015-07-14 00:52:12.0'	160	160	160	160	1000	8	1000	10	33	32	1
'2015-07-14 00:53:12.0'	160	160	160	160	1000	8	1000	11	33	32	2
'2015-07-14 00:54:12.0'	160	160	160	160	1000	8	1000	12	33	32	2
'2015-07-14 00:55:12.0'	160	160	160	160	1000	8	1000	10	33	32	2

State is: ok

Sensor data (now)

Prediction Model

Update Prediction Model (historic data)

Predicted State (now)

Process monitoring at MONDI Gronau – Domain knowledge and tools

Tools:

- MATLAB
- Database Toolbox
- Statistics and Machine Learning Toolbox
- Neural Network Toolbox
- MATLAB Compiler