Monitoring Production Line Performance to Reduce Failures

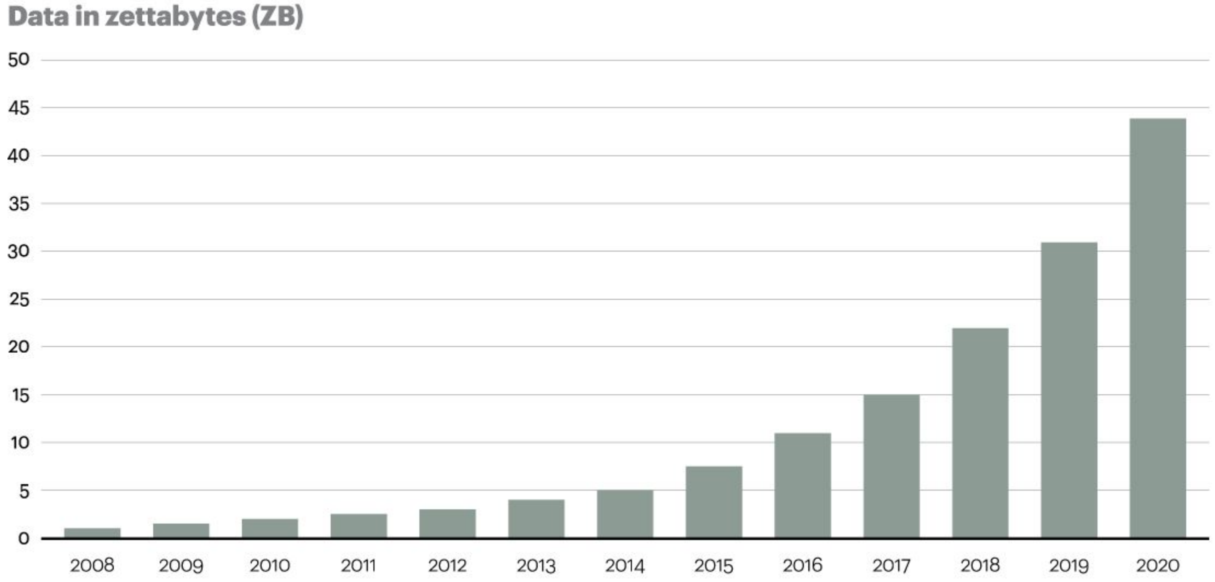
Abhilasha Sancheti, Desh Raj, Kunal Jain, Mrinal Tak
GROUP 18

Dept. of Computer Science and Engineering, IIT Guwahati

Why data science in process monitoring?

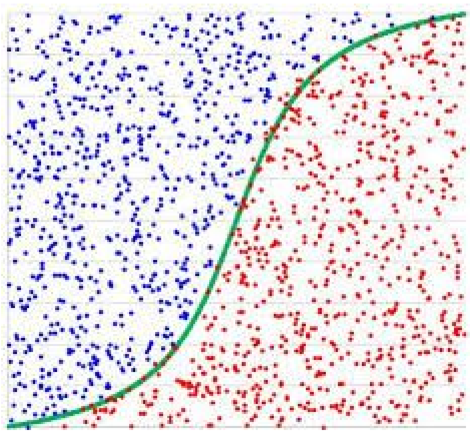
Figure 1

Data is growing at a 40 percent compound annual rate, reaching nearly 45 ZB by 2020

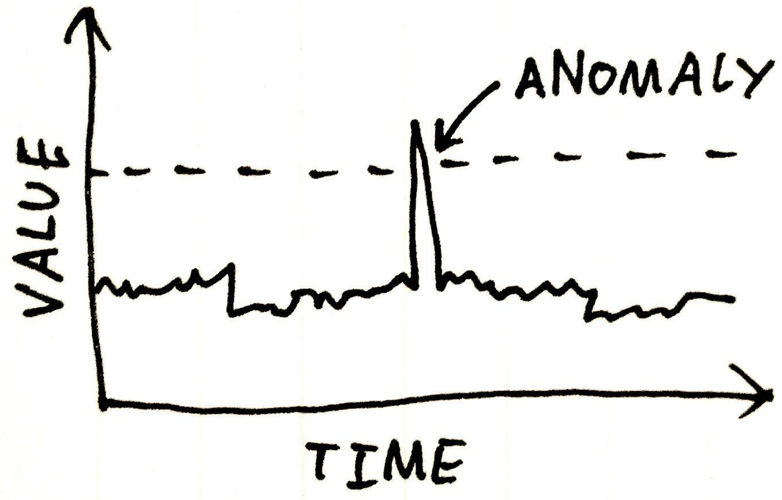


Source: Oracle, 2012

The task of fault analysis



Binary Classification



Anomaly Detection

Dataset Description

We use the Bosch Production Line Performance data set .

Size of dataset: 14.3 Gb

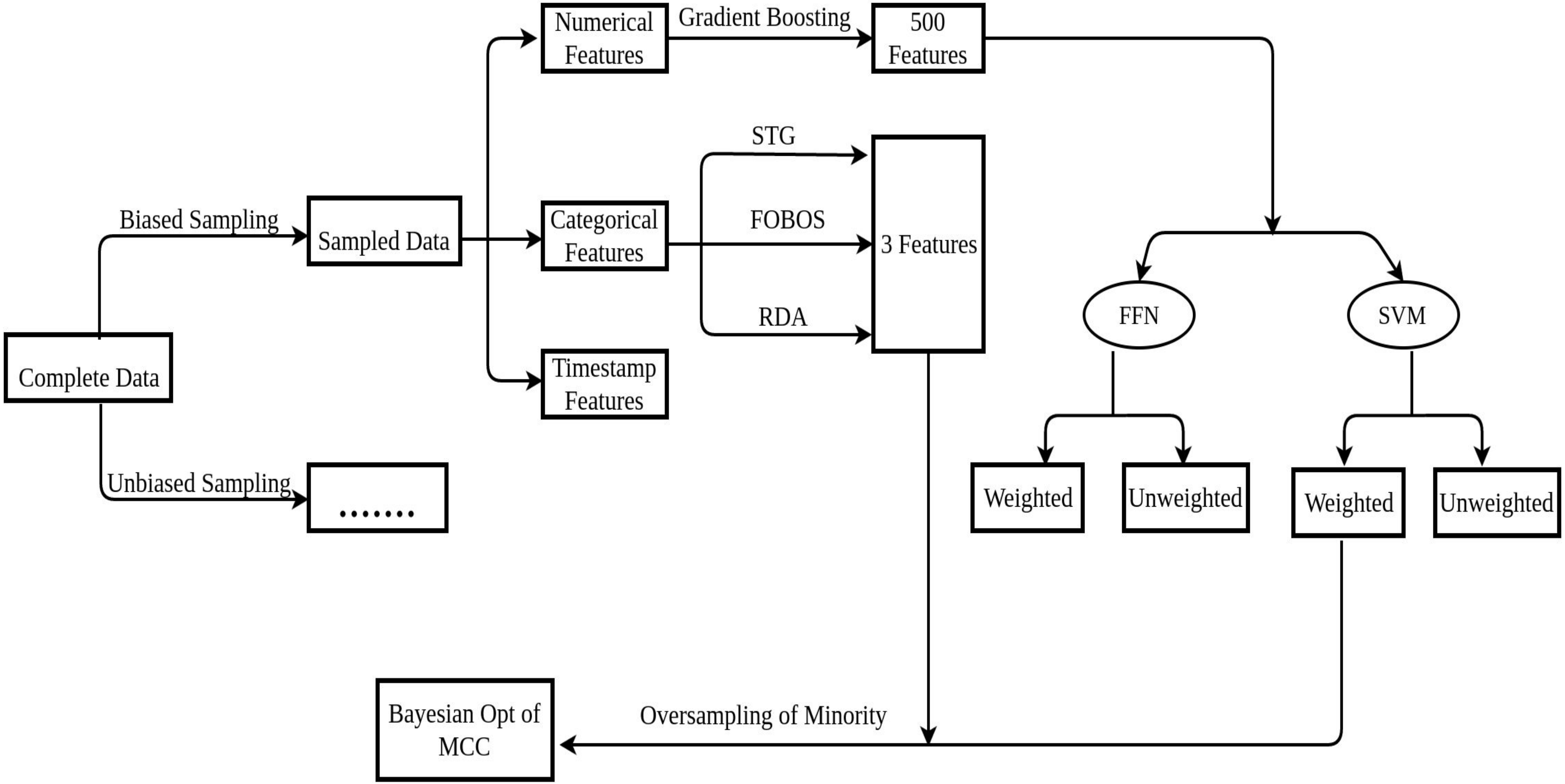
Features: Numerical (968), Categorical (2140) and Date stamps (1156)

Labels: indicating the sample as good or bad.

#samples: 11,84,687

Four stage approach:

1. Undersampling
2. Feature selection
3. Choice of Base classifier
4. SMOTE + BayesOpt

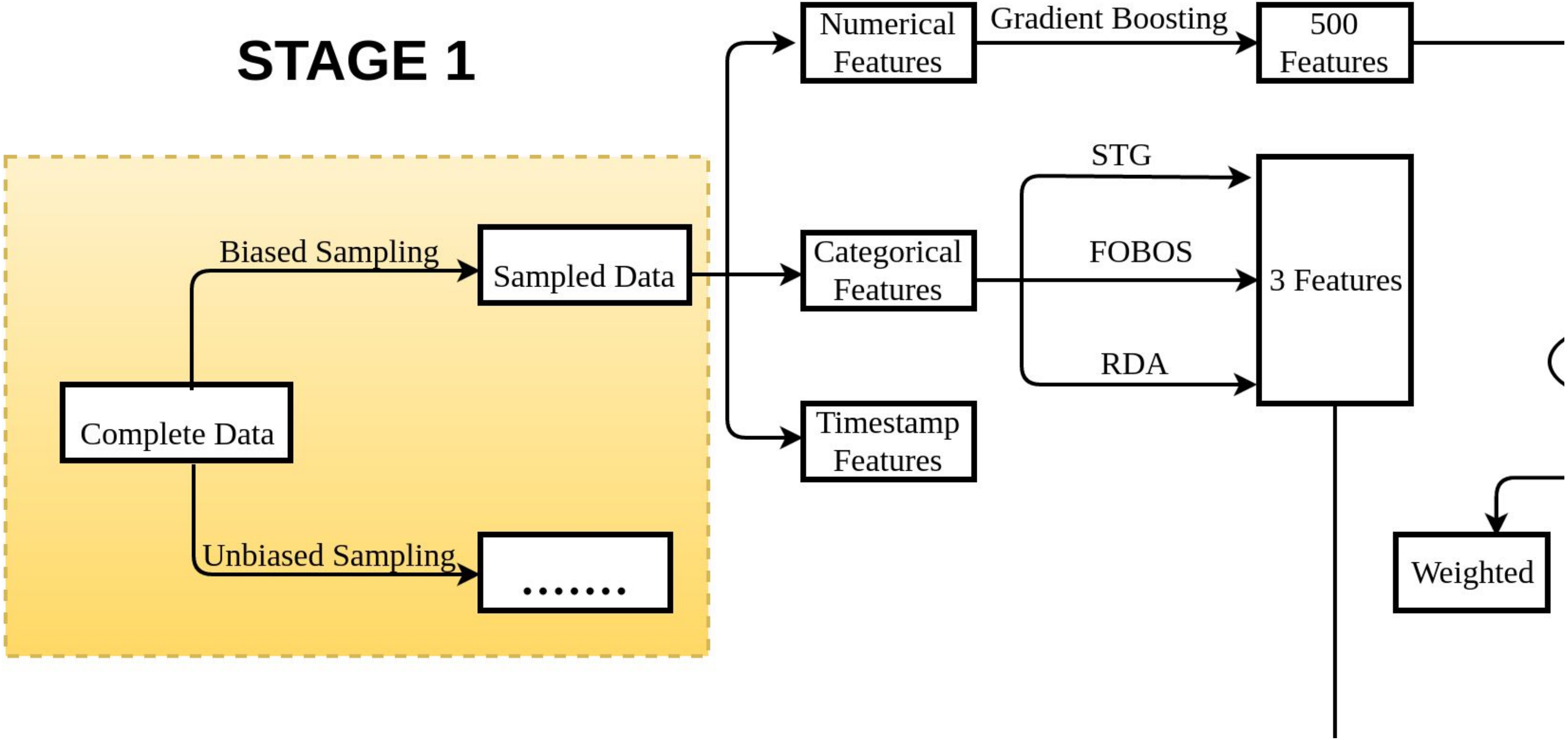

Numerical Features
Gradient Boosting
500 Features
STG
FOBOS
RDA
Biased Sampling
Sampled Data
Categorical Features
3 Features
FFN
SVM
Complete Data
Timestamp Features
Weighted
Unweighted
Weighted
Unweighted
Unbiased Sampling
....
Bayesian Opt of MCC
Oversampling of Minority

STAGE 1
Initial sampling

1) **Unbiased**: Select a subset of the original samples without taking into account the corresponding labels.

2) **Biased**: All the positive instances are retained while performing sampling

STAGE 1

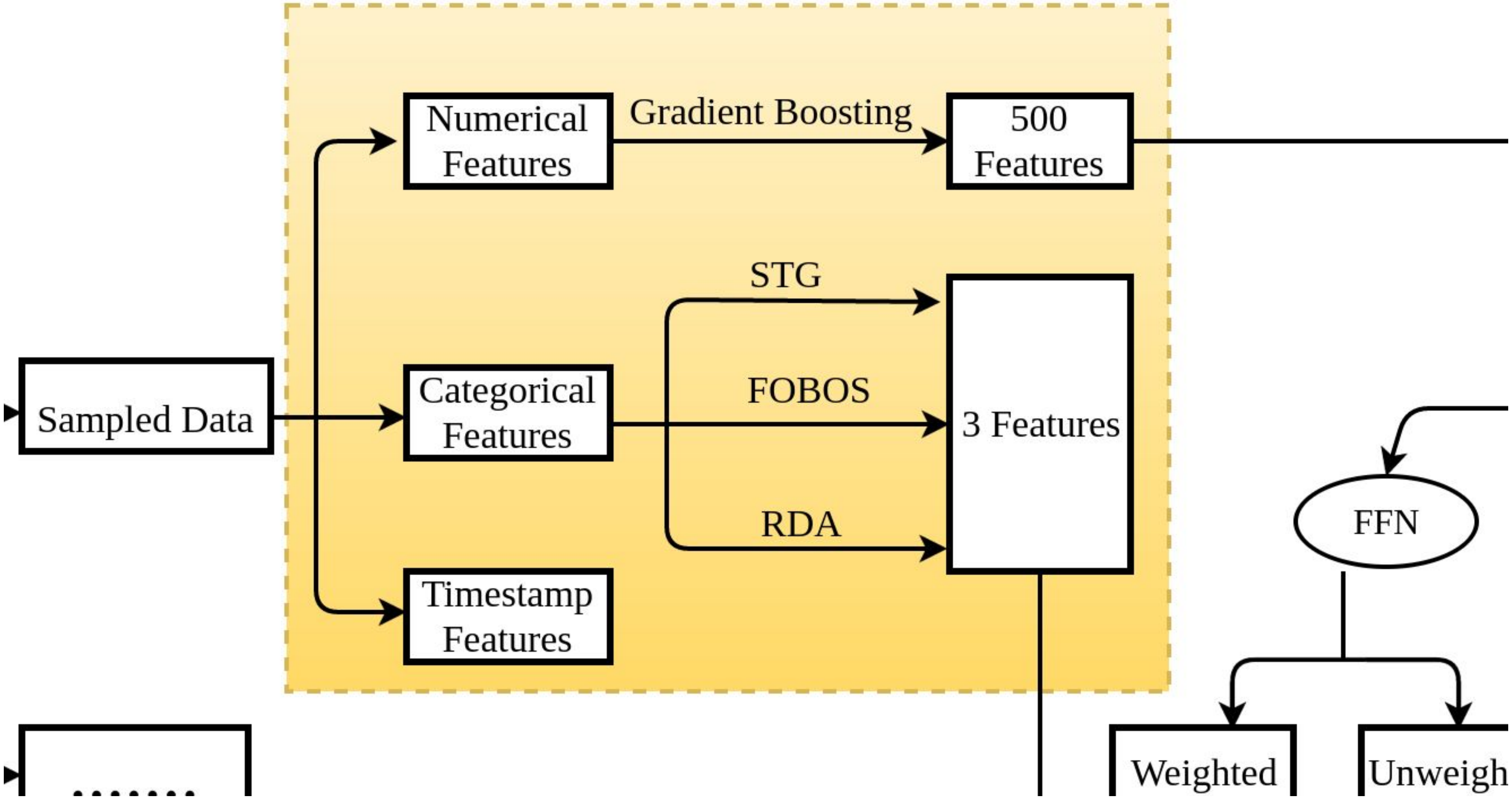

Numerical Features
Gradient Boosting
500 Features
STG
Biased Sampling
Sampled Data
Categorical Features
FOBOS
3 Features
RDA
Complete Data
Timestamp Features
Unbiased Sampling
....
Weighted

STAGE 2
Feature selection

Different methods used for each of:

- Numeric
- Categorical
- Timestamp features

STAGE 2


Sampled Data
Numerical Features
Gradient Boosting
500 Features
Categorical Features
STG
FOBOS
RDA
3 Features
Timestamp Features
FFN
Weighted
Unweigh
......

Gradient Boosting for Numerical Features

$$\min_{\beta} \sum_{(\phi(\mathbf{x}_i),y_i)} \boxed{\ell(\phi(\mathbf{x}_i), y_i, \boldsymbol{\beta})} + \boxed{\lambda|\boldsymbol{\beta}|_1} + \boxed{\mu q_\epsilon(\boldsymbol{\beta})}.$$

Loss function on output of regression trees -> **nonlinearity**

Regularization parameter -> induces **sparsity** for feature selection

Once a feature is extracted, its use is not penalized further.

Xu, Zhixiang, et al. "Gradient boosted feature selection." Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2014.

Sparse Online Learning for Categorical Features

3 features generated using 3 different methods:

1. Stochastic Truncated Gradient (STG)
2. Forward Backward Splitting (FOBOS)
3. Enhanced Regularized Dual Averaging (ERDA)

Each is trained on the train set and used to predict scores for train + test data. This score is used as feature.

Stochastic Truncated Gradient

$$f(w_i) = T_1(w_i - \eta \nabla_1 L(w_i, z_i), \eta g_i, \theta),$$

To control shrinkage since direct rounding to zero is too aggressive.

$$T_1(v_j, \alpha, \theta) = \begin{cases} \max(0, v_j - \alpha) & \text{if } v_j \in [0, \theta] \\ \min(0, v_j + \alpha) & \text{if } v_j \in [-\theta, 0] \\ v_j & \text{otherwise} \end{cases}.$$

Langford, John, Lihong Li, and Tong Zhang. "Sparse online learning via truncated gradient." Journal of Machine Learning Research 10.Mar (2009): 777-801.

Forward Backward Splitting

$$w_{t+\frac{1}{2}} = w_t - \alpha_t g_t$$

$$w_{t+1} = \arg\min_{w} \{ \frac{1}{2} ||w - w_{t+\frac{1}{2}}||^2 + \boxed{\lambda ||w||_1} \}$$

Regularization parameter -> induces **sparsity** for feature selection

Duchi, John, and Yoram Singer. "Efficient online and batch learning using forward backward splitting." Journal of Machine Learning Research 10.Dec (2009): 2899-2934.

Enhanced Regularized Dual Averaging

$$w_{t+1} = \arg\min_{w}\{\boxed{< g'_t, w >} + \boxed{\lambda_t^{RDA}||w||_1} + \boxed{\frac{\gamma}{2\sqrt{t}}||w||_2^2}\}$$

Dual average: obtained by averaging all previous subgradients

$$g'_t = \frac{t-1}{t} g'_{t-1} + \frac{1}{t} g_t$$

Regularization parameter -> induces **sparsity** for feature selection

Additional strongly convex regularization term

Xiao, Lin. "Dual averaging methods for regularized stochastic learning and online optimization." Journal of Machine Learning Research 11.Oct (2010): 2543-2596.

Manual feature engineering for Timestamp features

Following features were extracted:

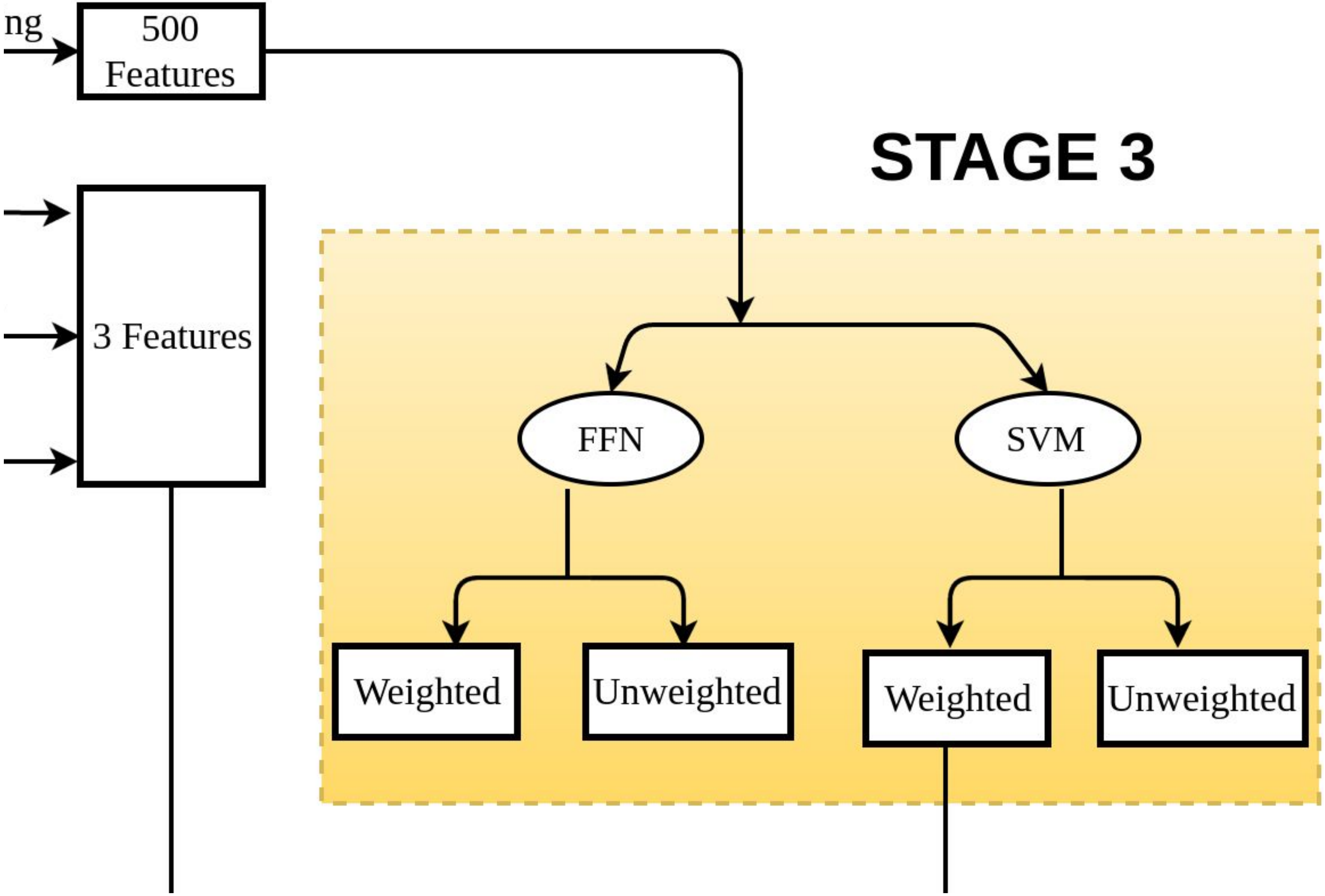
1. Minimum of all timestamps
2. Maximum of all timestamps
3. Mean of all timestamps
4. Duration of sample in production line
5. Number of NA features

No significant improvement in performance!

We ignore timestamp features.

STAGE 3
Base Classifier

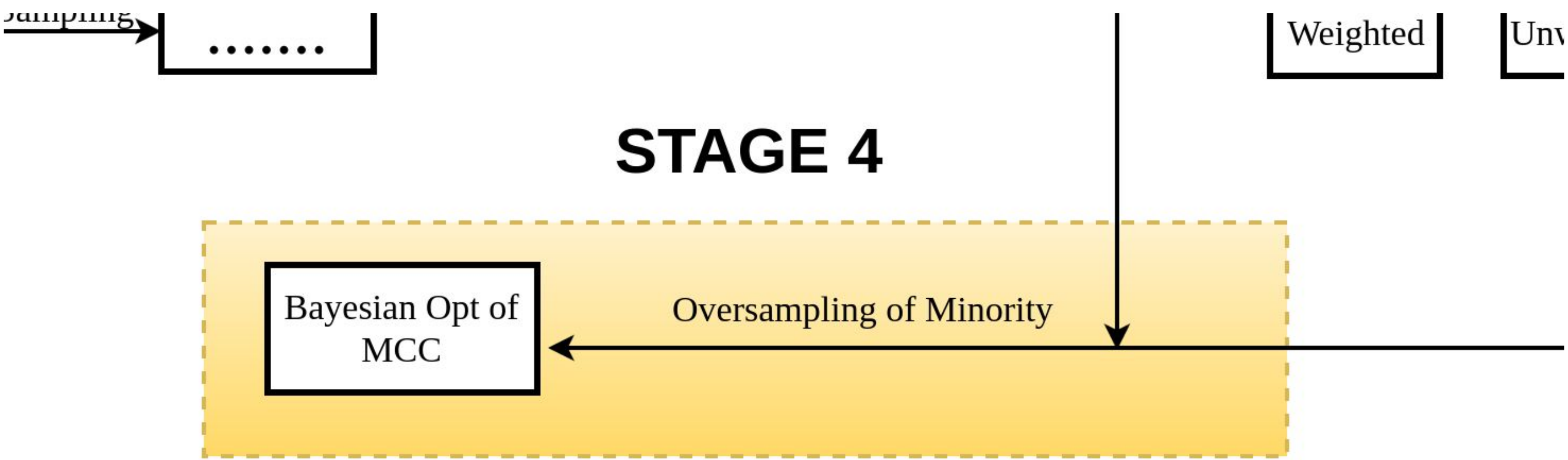
- Numerical features are used to select a base classifier.
- We experiment with feedforward neural networks (FFN) and support vector machines (SVM).
- Weighted and unweighted versions are evaluated.
- Finally the best performing model is chosen for further optimization.


ng
500 Features
3 Features
STAGE 3
FFN
SVM
Weighted
Unweighted
Weighted
Unweighted

STAGE 4
SMOTE + BayesOpt

Two things are done:

1. Synthetic Minority Oversampling Technique (SMOTE)
2. Bayesian Optimization of the evaluation metric


....
Weighted
STAGE 4
Bayesian Opt of MCC
Oversampling of Minority

Synthetic Minority Oversampling Technique (SMOTE)

- Take the difference between the feature vector (sample) under consideration and its nearest neighbor.
- Multiply this difference by a random number between 0 and 1, and add it to the feature vector under consideration.
- This causes the selection of a random point along the line segment between two specific features.

Chawla, Nitesh V., et al. "SMOTE: synthetic minority over-sampling technique." Journal of artificial intelligence research 16 (2002): 321-357.

Bayesian Optimization of MCC

$$
\begin{aligned}
M(w) &= \operatorname{argmin}\ g(w) \\
&= \operatorname{argmin}\ \{w \cdot FP + (1-w) \cdot FN\}
\end{aligned}
$$

Weight parameter

$$M^* = \arg\max MCC(w).$$

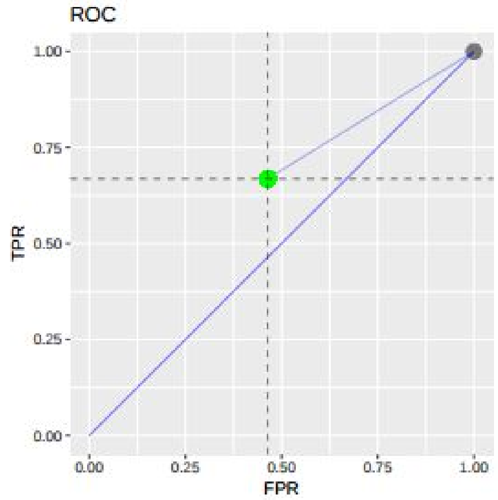
Results

Three observations are made:

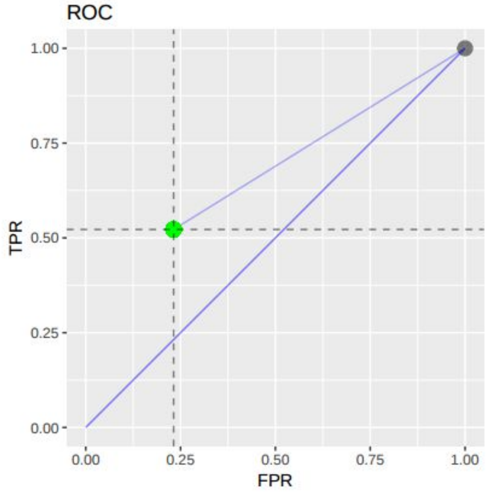
1. Effect of feature types
2. Base classifier performances
3. Effect of class weights

Effect of feature types

Most of the sensitivity of the base classifier was obtained due to the numerical features, and the 3 categorical features only contributed a little in improving performance.



Effect of only numerical features on ROC



Effect of numerical + categorical features on ROC

Base classifier performances

Weighted SVM was found to perform best.

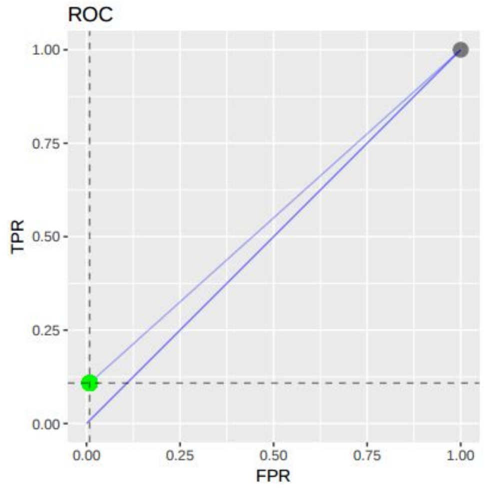
Model	Non-weighted			Weighted		
	Precision	Recall	F1-score	Precision	Recall	F1-score
FFN	**92.53**	4.35	8.32	58.11	10.82	18.24
SVM	84.61	0.77	1.53	13.25	**67.39**	**22.15**

Base classifier performances

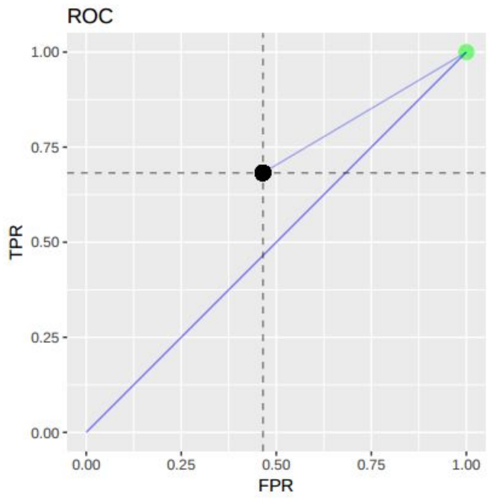
The AUCs for the models are:

FFN = 0.5499

SVM = 0.6014



ROC variation for feedforward network (FFN)



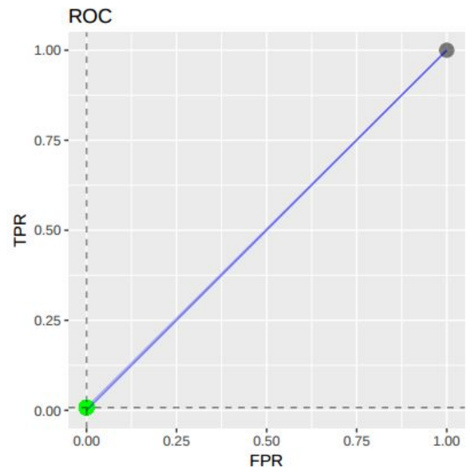
ROC variation for support vector machine (SVM) classifiers.

Effect of class weights

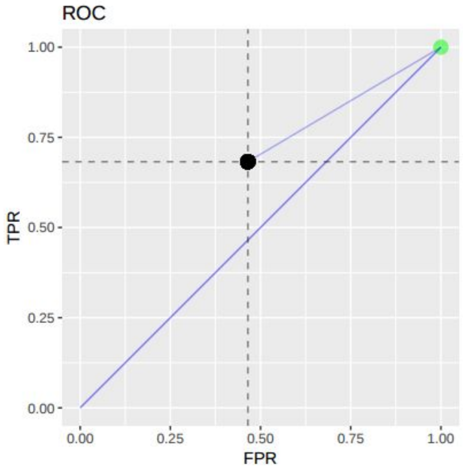
The AUCs for the models are:

Unweighted SVM = 0.5038

Weighted SVM = 0.6014



ROC variation for unweighted SVM



ROC variation for weighted SVM

Bayesian Optimization on evaluation metric can improve performance by as much as **3-4%**.

Conclusion

Simple task of binary classification can be complex in an industrial setting.

Several preprocessing, feature selection, and classifier optimization methods were explored.

Future work: Better base classifiers, extracting more features from categorical and timestamp features.