The Twelfth International Conference on Advanced Collaborative Networks, Systems and Application
COLLA 2022

Automated Data Preprocessing for Machine Learning Based Analyses

Authors: Akshay Paranjape, Praneeth Katta, Markus Ohlenforst

Presenter: Praneeth Katta, Machine Learning Engineer

IconPro GmbH, Aachen, Germany

Akshay Paranjape

Team Lead - ARES

Akshay Paranjape has a master's degree in Simulation Sciences from RWTH Aachen with a focus on Machine Learning. During his studies, he worked at Zeiss in the field of Computer Vision for classification problems. His thesis work on "Open Set Classification using Deep Learning" is recognized as IP of Zeiss. Prior to his thesis, he worked at the Informatics Department at RWTH Aachen and obtained his bachelor's degree in Physics from IIT Delhi, India.

Praneeth Katta

Machine Learning Engineer

Praneeth obtained his M.Sc degree in Simulation Science at the RWTH Aachen University with a background in Mechanical Engineering (B.Sc) at the Ramaiah Institute of Technology, in India. At IconPro, he is part of the Ares team and assists developing and deploying this software for Predictive Quality and Process Optimization.

Contents

- Introduction and Motivation
- Related work
- Methodology
  - Feature Selection
  - Sampling – Bin-based
  - Target Discretization
  - Hybrid Feature Engineering
- Experiments and Results
- Outlook
- References
- About us

Introduction

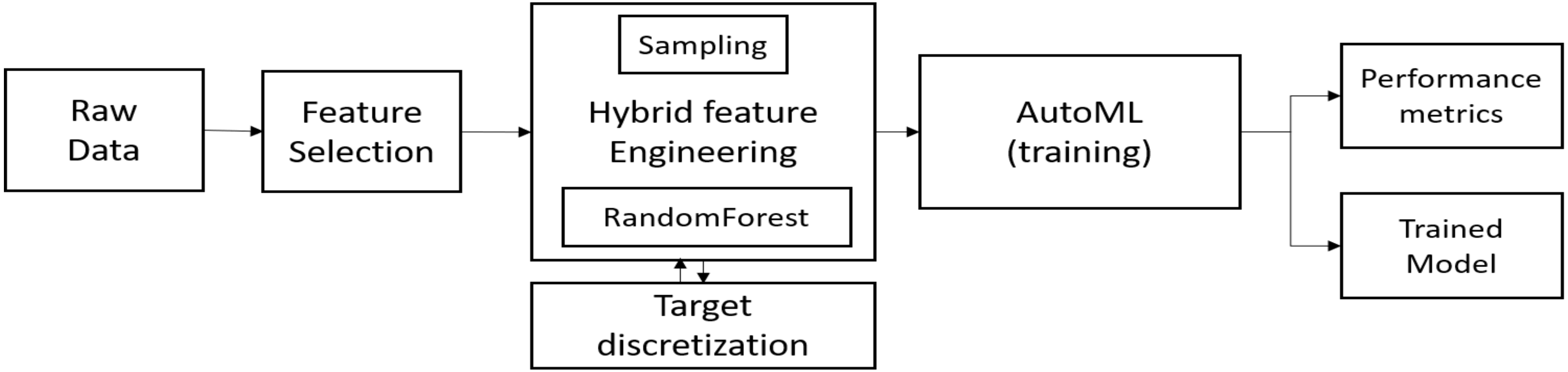
- Data preprocessing is a crucial step in Machine Learning
- Preprocessing is performed to prepare the compatible dataset for analysis
- Preprocessing is mainly categorized into two types:
  - Type1: model compatible preprocessing
  - Type 2: quality enhancement preprocessing
- AutoML Libraries focus on Type 1 preprocessing
- This paper is focused on Type 2 preprocessing which have been not implemented in AutoML Libraries

TABLE I
PREPROCESSING STEPS INCLUDED IN DIFFERENT AUTOML SOLUTIONS

Name	AutoSklearn	AutoKeras	TPOT	AutoGluon	H2O
Balancing	yes	no	no	yes	yes
Categorical encoding	yes	yes	yes	yes	yes
Imputation	yes	yes	no	yes	yes
Standardization/Normalization	yes	yes	no	yes	yes
Others	Densifier, PCA, minority coalescence, select percentile	Data augmentation	Feature selector	Introduce ”unknown category”	None

Motivation

- Extraction of features (Feature Engineering), selecting the most important features among a big list is a time-consuming step/process to do manually
- This paper is aimed to automate the above-mentioned manual process
- Production dataset can be huge and can take few hours to many days to train a model
- This paper also introduces a Sampling method to sample the dataset statistically in a better way compared to mostly used random sampling which inturn reduces the computation time to train a model but retains the efficiency

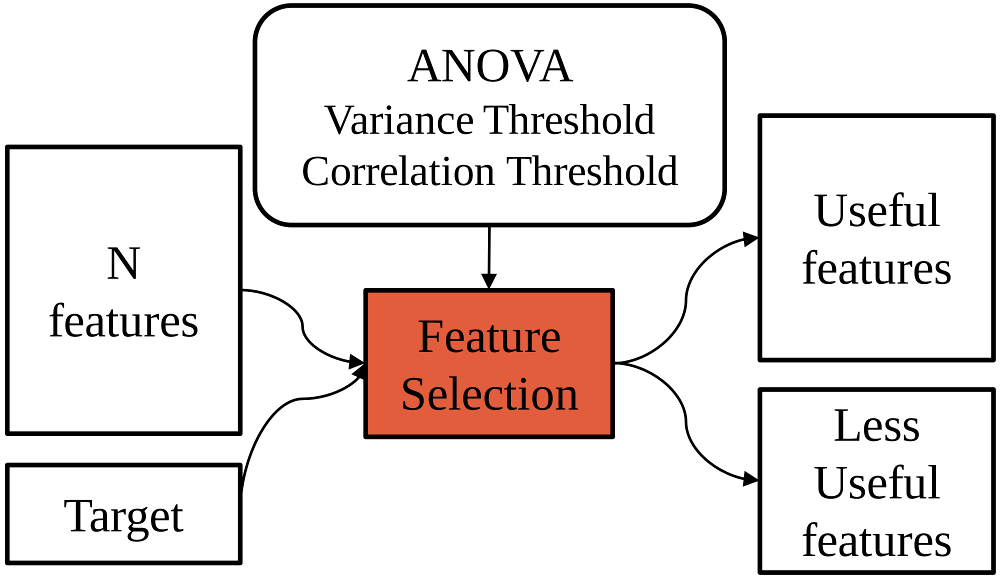


Related Work

- Cognito: Automated Feature Engineering for Supervised Learning [1]
  - Generating new features by transforming existing features
- Explore Kit: Automatic Feature Generation and Selection [2]
  - Trimming down the generated features with help of Ranking Classifier
- Cochran, W. G. Sampling techniques [3]
  - Stratified Sampling is the well-known sampling, but it is mathematically complex to sample
  - This sampling technique closely resembles the original distribution statistics
- Efficient Sampling Methods for Discrete Distributions [6]
  - Adjusting the Sampling methods to make it faster to compute
- Analysis of variance (ANOVA) comparing means of more than two groups [4]
  - ANOVA gives a preliminary score of correlation of each feature with Target feature in a dataset

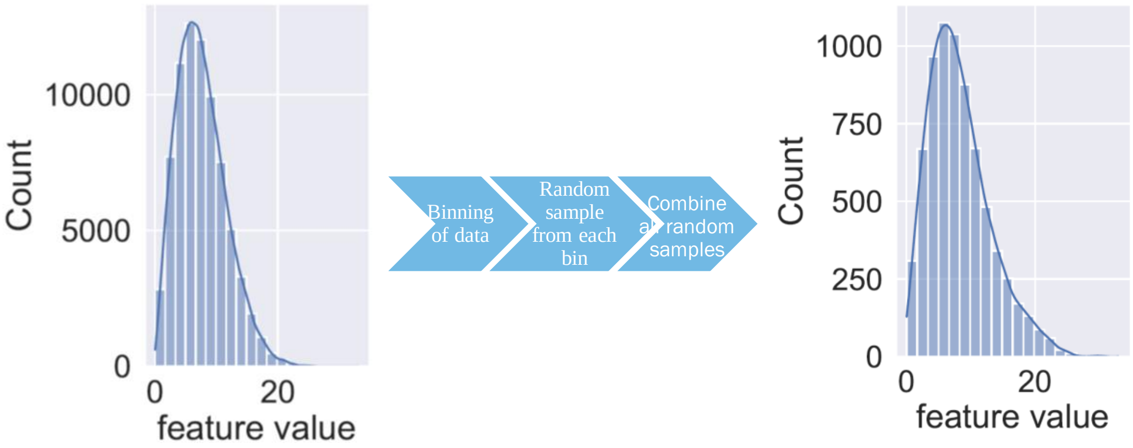
Methodology – Feature Selection

- ANOVA:
  - F-test ANOVA is the ratio of variability between groups to variability within group
  - $F = \frac{Variability\ between\ groups}{Variability\ within\ group}$
  - $F = \frac{S_b^2}{S_w^2} = \frac{\sum_{j=1}^{m} n_j\left(\overline{x_j}-\bar{X}\right)^2 / m-1}{\sum_{j=1}^{m}\sum_{i=1}^{n_j}\left(x_{ij}-\overline{x_j}\right)^2 / n-m}$
- Variance Threshold:
  - Features with 0 variance,
  - Categorical features with 100 variance are removed (Name, Machine ID etc.,)
- Correlation Threshold:
  - Features with high correlation of 95% are removed



Methodology – Bin-based Sampling

- $P(b_i)_{BS} = 1,\ P(s)_{BS} = P\left({}^{s}/_{b_i}\right)_{BS} = \frac{1}{size(b_i)}$,
  BS = bin-based sampling
- Population size = N, Sample size = S, features = M, sample = s, feature = f
- Number of bins = n $(b_1, b_2, b_3, \ldots, b_n)$



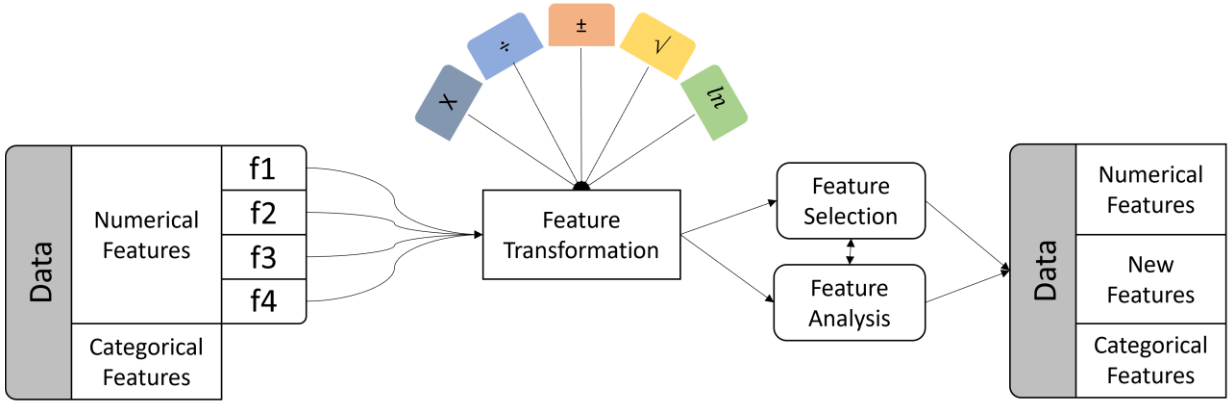
Algorithm 2: Binbased Sampling

for $Feature_i$ *in* $Features$ **do**
  if $Feature_i$ *is* $Categorical$ **then**
    for $Category$ *in* $Categories_{Feature_i}$ **do**
      Sample $\leftarrow$ RandomSample(Category)
    end
    featureSample $\leftarrow$ Concat(Sample)
  else
    Bins $\leftarrow$ Discretize($Feature_i$)
    for Bin *in* $Bins$ **do**
      Sample $\leftarrow$ RandomSample(Bin)
    end
    featureSample $\leftarrow$ Concat(Sample)
  end
end
BinbasedSample $\leftarrow$ Concat(featureSample)
Result: Binbased Sample

Methodology – Target Discretization

- Target Discretization transforms Regression task to Classification task by converting numerical output feature to categorical values
- This can be used for the datasets where regression R2-score is significantly low or unacceptable
- The prediction of categorical values has less degree of freedom than the prediction of numerical values
- In this, each data point in the continuous domain is converted into a discrete class domain
- Different types of target discretization methods can be considered based on domain expertise
- As an automated solution, we have considered the discretization of the target variable based on its z-score values

Methodology – Hybrid Feature Engineering



Inspired from two research papers – Cognito[1] and Explore Kit[2]

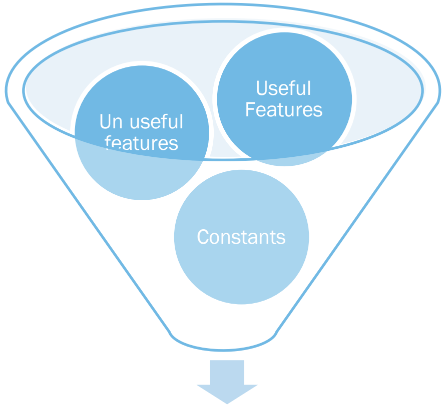
Algorithm 3: Hybrid Feature Engineering

```
for Operator in Unary/Binary Operators do
    for Feature in Numerical-Features do
        NewFeatures ← Operator(Feature)
    end
    allNewFeatures_operator ←
      FeatureSelection(NewFeatures)
end
allNewFeatures = RankingModel(allNewFeatures)
Result: Generated Features
```

Algorithm 4: Ranking Model

```
thresholdf = baseModel(Dataset)
for i = 0 to allNewFeatures do
    Dataset.append(allNewFeatures[i])
    RankedFeatures = []
    featureScore = baseModel(Dataset)
    if featureScore ≤ thresholdf then
        continue
    else
        RankedFeatures.append(allNewFeatures[i])
    end
    return RankedFeatures
end
Result: Ranked features
```

Experiments and Results – Feature Selection



Useful Features

Using Adaboost classifier/regressor

TABLE II
VALIDATION OF FEATURE SELECTION TECHNIQUE FOR CLASSIFICATION TASK

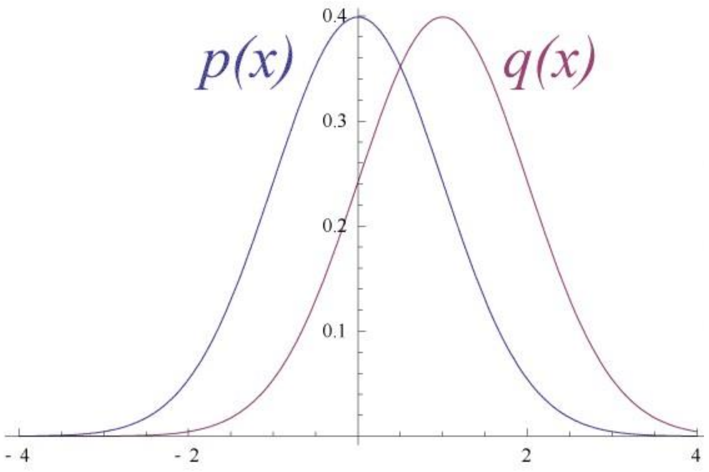
OpenML dataset	Accuracy w/o feature selection	Accuracy with feature selection	Number of features removed	Difference in accuracy
11	0.607	0.607	3	0
54	0.753	0.753	0	0
188	0.579	0.579	0	0
333	0.908	0.908	3	0
335	0.977	0.977	2	0
470	0.661	0.661	4	0
1459	0.588	0.588	0	0
1461	0.692	0.692	2	0
23381	0.560	0.560	5	0
amazon-employee-access	0.943	0.943	3	0
australian	0.857	0.857	2	0
bank-marketing	0.692	0.692	2	0
credit-g	0.761	0.761	2	0
sylvine	0.941	0.941	7	0

TABLE III
VALIDATION OF FEATURE SELECTION TECHNIQUE FOR REGRESSION TASK

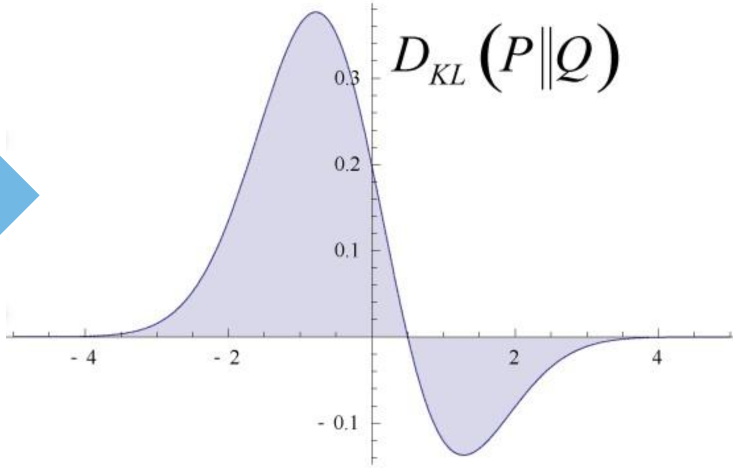
OpenML dataset	r2-score w/o feature selection	r2-score with feature selection	Number of features removed	Difference in r2-score
537	0.484	0.484	0	0
495	0.616	0.616	5	0
344	0.999	0.999	2	0
215	0.948	0.948	1	0
189	0.579	0.579	0	0
507	0.391	0.390	0	0

Experiments and Results – Bin-based Sampling

- Kullback Leibler divergence compares multi variate distribution of population and sample
- $D_{KL}(P \,||Q\,) = \sum_{x\in\chi} P(x).\ln(\frac{P(x)}{Q(x)})$ [5]



KL-
Divergence



[7]

Experiments and Results – Bin-based Sampling

TABLE IV
SAMPLING COMPARISON ON OPENML DATASETS CALCULATED OVER 100 TRIALS

OpenML dataset	Mean of KL-divergence Bin-Based sampling	Mean of KL-divergence Stratified sampling	Mean of KL-divergence Random sampling	Time (in sec) Bin-Based sampling	Time (in sec) Stratified sampling
183	0.017	0.173	0.057	0.359	5.230
223	0.067	0.079	0	0.273	7.600
287	0.076	0.356	0.027	0.399	4.807
307	0.0	0.097	0.006	0.214	7.572
528	0.0	0.0215	0.0	0.054	0.489
537	0.190	0.886	1.160	2.052	133.939
550	0.0	0.011	0.004	0.302	0.738
Amazon-employee-access	0.019	0.460	0.753	0.466	2.112
Blood-transfusion	0.065	0.002	0.001	0.062	0.069
Phoneme	0.0	0.168	0.143	0.580	1.383

- Sample Size: Cochran's formula is used to calculate sample size [3]

Experiments and Results – Hybrid Feature Engineering

TABLE V
HYBRID FEATURE ENGINEERING FOR CLASSIFICATION DATASETS WITH BASELINE MODEL

OpenML datasets	Number of features	Number of classes	Accuracy before	Accuracy after	Percentage Gain	New features
188	14	5	0.466	**0.506**	8.386	2
1461	7	2	0.692	**0.718**	4.748	2
1459	7	10	0.588	**0.635**	7.952	1
54	18	4	0.753	**0.759**	0.786	2

TABLE VI
HYBRID FEATURE ENGINEERING FOR REGRESSION DATASETS WITH BASELINE MODEL

OpenML datasets	Number of features	r2-score before	r2-score after	Percentage Gain	New features
189	8	0.579	**0.615**	6.227	1
507	6	0.390	**0.411**	5.361	1
537	8	0.484	**0.494**	2.000	1
495	13	0.616	**0.632**	2.700	2

Experiments and Results – Overall Preprocessing pipeline

TABLE VII
OVERALL PREPROCESSING PIPELINE PERFORMANCE COMPARISON WITH AUTOML LIBRARIES (CLASSIFICATION - ACCURACY)

OpenML datasets	AutoGluon		AutoSklearn		H2O		RandomForest	
	w/o	w	w/o	w	w/o	w	w/o	w
188	0.728	0.726	0.674	0.696	0.717	0.739	0.466	0.506
1461	0.914	0.914	0.906	0.906	0.907	0.907	0.692	0.718
1459	0.815	0.82	0.919	0.919	0.922	0.927	0.588	0.635
54	0.858	0.857	0.839	0.839	0.707	0.708	0.753	0.759

TABLE VIII
OVERALL PREPROCESSING PIPELINE PERFORMANCE COMPARISON WITH AUTOML LIBRARIES (REGRESSION - R2-SCORE)

OpenML datasets	AutoGluon		AutoSklearn		H2O		RandomForest	
	w/o	w	w/o	w	w/o	w	w/o	w
189	0.913	0.913	0.902	0.903	0.913	0.918	0.579	0.615
507	0.731	0.741	0.753	0.753	0.762	0.761	0.390	0.411
537	0.815	0.821	0.862	0.865	0.861	0.869	0.484	0.494
495	0.496	0.495	0.494	0.494	0.441	0.442	0.616	0.632

Outlook

- This paper reviews and suggests some advanced preprocessing steps that can either be used individually or combined as a pipeline
- Datasets that have inter-feature dependency can be observed to perform better
- The proposed method preprocess the data without domain knowledge in an automated manner
- This paper also introduces a new sampling method that can be used for general application as well as for ML-based modeling
- A significant performance improvement of around 4-7% is observed for the analysis conducted with the baseline model on OpenML datasets
- For the same set of datasets, a marginal improvement was observed for analysis with the AutoML libraries
- The proposed pipeline is currently not parallelized. Parallelization can significantly reduce the time for feature engineering and this we would like to focus on in our future work

References

[1] U. Khurana, D. Turaga, H. Samulowitz and S. Parthasrathy, "Cognito: Automated Feature Engineering for Supervised Learning," 2016 IEEE (ICDMW), Barcelona.

[2] G. Katz, & E. Shin & D. Song, 2016. ExploreKit: Automatic Feature Generation and Selection.

[3] W. G. Cochran, APA (6th ed.), 1977. Sampling techniques. New York: Wiley.

[4] H. Y. Kim. Analysis of variance (ANOVA) comparing means of more than two groups.

[5] S. Kullback, R. A. Leibler, On information and sufficiency, 1951. The Annals of Mathematical Statistics.

[6] K. Bringmann and K. Panagiotou , Efficient Sampling Methods for Discrete Distributions.

[7] kav.lbp2ampcoil.fun/kl-divergence-between-two-gaussians.html (image).

Spin-off Award
Eine Auszeichnung der RWTH Aachen

Our team is...

YOUNG MOTIVATED DIVERSE SMART EFFICIENT HARD-WORKING CAPABLE QUALIFIED PROFESSIONAL PASSIONATE

Managing Director

Dr. Markus Ohlenforst

Computational Engineer,
PHD in Production Technology,
Team Lead at WZL

Managing Director

Dr. Martin Peterek

Industrial Engineer,
PHD in Production Technology,
Managing Director at WZL

Advisory Board

Dr. Edgar Dietrich

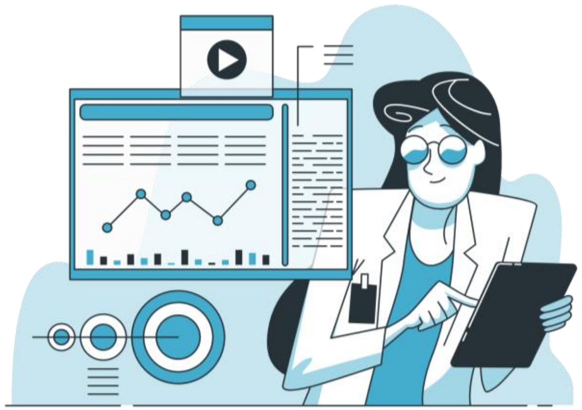
Reknown Industrial Statistician,
Founded & sold Q-DAS,
Entrepreneur

Advisory Board

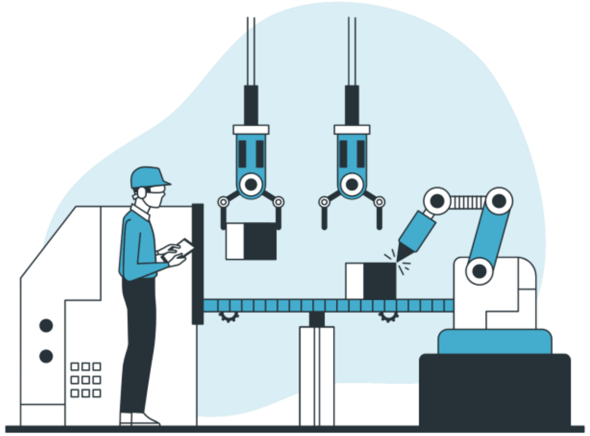
Prof. Dr. Robert Schmitt

Electrical Engineer,
Reknown Production Researcher,
Director at WZL

AI IN PRODUCTION



[6]



[7]

Intuitive

Automated

Platform - agnostic

solutions for the actual Stakeholders

Production and Quality Managers & Process Engineers.

[8]

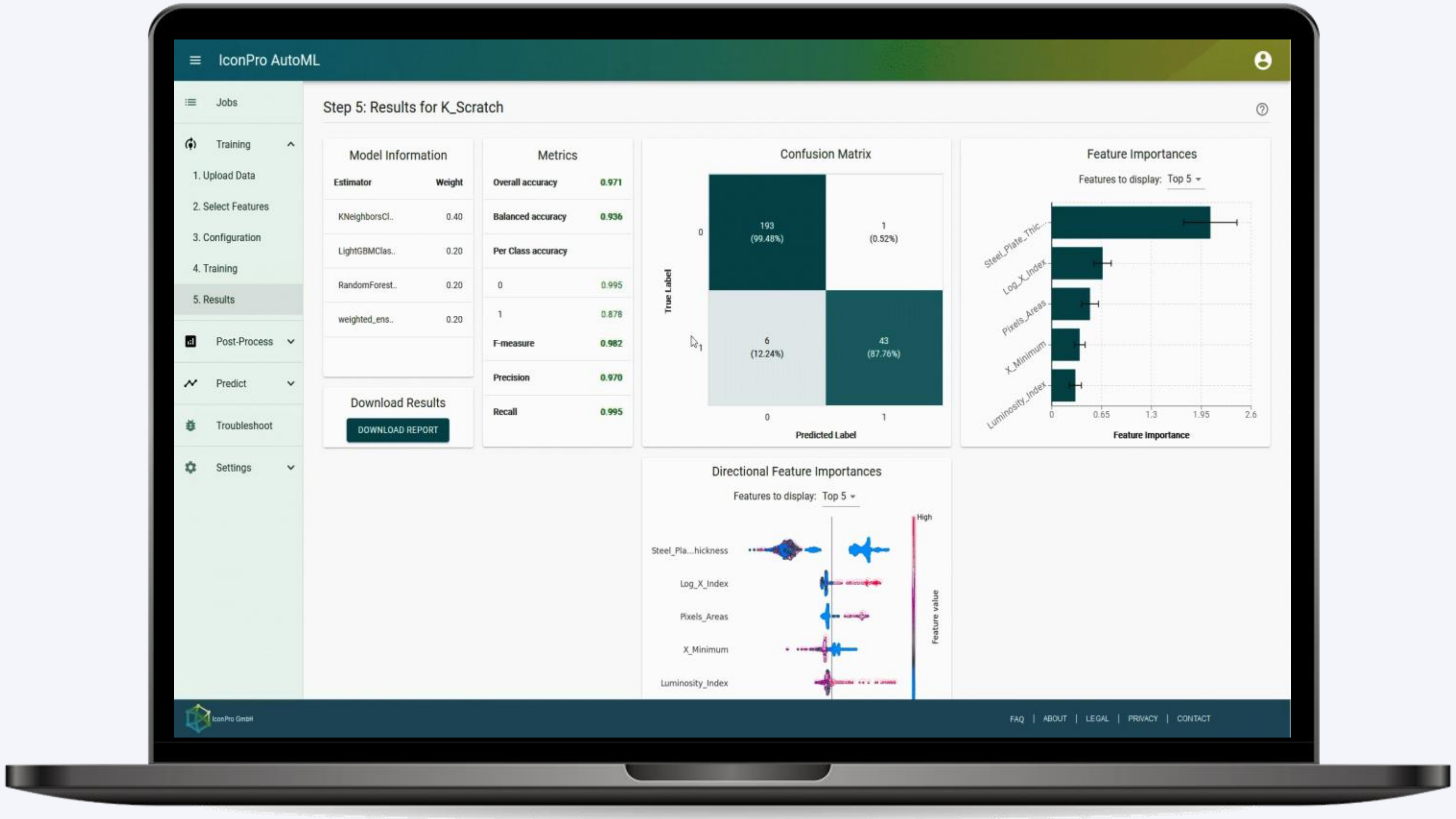
[9]

ICON PRO

Image references: [6],[7],[8],[9]

PREDICTIVE QUALITY SOLUTION

ARES



Load in Any Data Source

Auto Find & Utilize all Relations in Data

Easily put predictions and optimization into operation

No-Code + Platform-Agnostic

ICONPRO

THANKYOU

If any questions,

you can contact us.

- Akshay Paranjape
  - akshay.paranjape@iconpro.com
- Praneeth Katta
  - praneeth.katta@iconpro.com