Journal of Machine Learning Research 20 (2019) 1-6 Submitted 11/18; Revised 9/19; Published 11/19

# DataWig: Missing Value Imputation for Tables

**Felix Bießmann** FELIX.BIESSMANN@BEUTH-HOCHSCHULE.DE
*Beuth University, Luxemburger Str. 10, 13353 Berlin (work done while at Amazon Research)*

**Tammo Rukat** TAMMRUKA@AMAZON.COM
**Phillipp Schmidt** PHSCHMID@AMAZON.COM
*Amazon Research, Krausenstr. 38, 10117 Berlin, Germany*

**Prathik Naidu** PRATHIKN@STANFORD.EDU
*Department of Computer Science, Stanford University*
*Stanford, CA 94305, USA (work done while at Amazon Research)*

**Sebastian Schelter** SEBASTIAN.SCHELTER@NYU.EDU
*Center for Data Science, New York University, New York, USA (work done while at Amazon Research)*

**Andrey Taptunov** ANDREI.TAPTUNOV@GMAIL.COM
*Snowflake, Stresemannstraße 123, 10963 Berlin (work done while at Amazon Research)*

**Dustin Lange** LANGED@AMAZON.COM
*Amazon Research, Krausenstr. 38, 10117 Berlin, Germany*

**David Salinas** DAVID.SALINAS@NAVERLABS.COM
*Naver Labs, 6 Chemin de Maupertuis, 38240 Meylan, France (work done while at Amazon Research)*

**Editor:** Alexandre Gramfort

## Abstract

With the growing importance of machine learning (ML) algorithms for practical applications, reducing data quality problems in ML pipelines has become a major focus of research. In many cases missing values can break data pipelines which makes completeness one of the most impactful data quality challenges. Current missing value imputation methods are focusing on numerical or categorical data and can be difficult to scale to datasets with millions of rows. We release `DataWig`, a robust and scalable approach for missing value imputation that can be applied to tables with heterogeneous data types, including unstructured text. `DataWig` combines deep learning feature extractors with automatic hyperparameter tuning. This enables users without a machine learning background, such as data engineers, to impute missing values with minimal effort in tables with more heterogeneous data types than supported in existing libraries, while requiring less glue code for feature engineering and offering more flexible modelling options. We demonstrate that `DataWig` compares favourably to existing imputation packages. Source code, documentation, and unit tests for this package are available at: `github.com/awslabs/datawig`

**Keywords:** missing value imputation, deep learning, heterogeneous data

## 1. Introduction

Machine learning (ML) algorithms have become a standard technology in production use cases. One of the main reasons for suboptimal predictive performance of such systems is low data

| Data Type | Featurizers | Loss |
|---|---|---|
| Numerical | Normalization<br>Neural Network | Regression |
| Categorical | Embeddings | Softmax |
| Text | Bag-of-Words<br>LSTM | N/A |

```
table = pandas.read_csv('products.csv')
missing = table[table['color'].isnull()]

# instantiate model and train imputer
model = SimpleImputer(
        input_columns=['description',
                       'product_type',
                       'size'],
        output_columns=['color'])
      .fit(table)

# impute missing values
imputed = model.predict(missing)
```

Figure 1: *Left*: Available featurizers and loss functions for different data types in `DataWig`. *Right:* Application example of `DataWig` API for the use case shown in Figure 2.

quality and one of the most frequent data quality problems are missing values. Imputation of missing values can help to increase data quality by filling gaps in training data. However automated and scalable imputations for tables with heterogeneous data types including free form text fields remains challenging. Here we present `DataWig`, a software package that aims at minimizing the effort required for missing value imputation in heterogeneous data sources. Most research in the field of imputation focuses on imputing missing values in *matrices*, that is imputation of numerical values from other numerical values (Mayer et al., 2019). Popular approaches include *k-nearest neighbors* (KNN) (Batista and Monard, 2003), *multivariate imputation by chained equations* (MICE) (Little and Rubin, 2002), *matrix factorization* (Koren et al., 2009; Mazumder et al., 2010; Troyanskaya et al., 2001) or deep learning methods (Gondara and Wang, 2017; Zhang et al., 2018; Mattei and Frellsen, 2019). While some recent work addresses imputation for more heterogeneous data types (Stekhoven and Bühlmann, 2012; Yoon et al., 2018; Nazabal et al., 2018), *heterogeneous* in those studies refers to binary, ordinal or categorical variables, which can be easily transformed into numerical representations. In practice also these simple transformations require glue code that can be difficult to adapt and maintain in a production setting. Writing such feature extraction code is out of scope for many engineers and can incur considerable technical debt on any data pipeline (Sculley et al., 2015; Schelter et al., 2018). We release `DataWig` to complement existing imputation libraries by an imputation solution for tables that contain not only numerical values or categorical values, but also more generic data types such as unstructured text. Extending the functionality of previous packages, `DataWig`'s imputation automatically selects from a number of feature extractors, including deep learning techniques, and learns all parameters in an end-to-end fashion using the symbolic API of Apache `mxnet` to ensure efficient execution on both CPUs and GPUs.

## 2. Imputation Model

The imputation model in `DataWig` is inspired by established approaches (van Buuren, 2018) and follows the approach of MICE, also referred to as *fully conditional specification*: for each to-be-imputed column (referred to as *output column*), the user can specify the columns which might contain useful information for imputation (referred to as *input columns*).

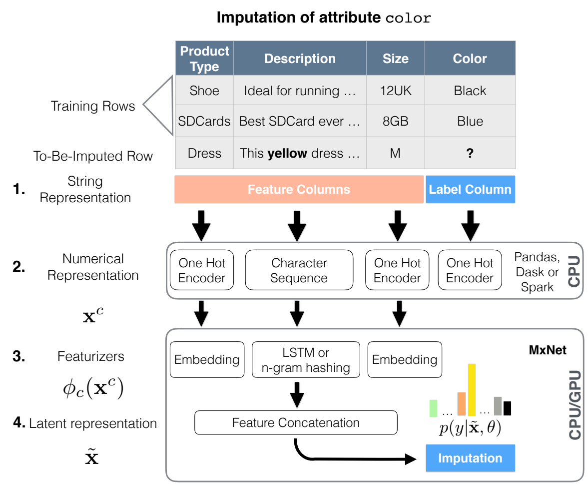


Figure 2: Imputation example on non-numerical data with deep learning.

Depending on its data type, each input column gets a dedicated *featurizer* denoted below as $\phi$. Similarly, depending on the data type for the output column, `DataWig` uses a different loss function. The types of featurizers and loss functions currently available in `DataWig` are listed in Figure 1 (left). The code design enables users to extend these types easily to images or sequences. More formally, `DataWig` imputes values $\hat{y}_o = f(\tilde{\mathbf{x}}_{\mathcal{I}})$ in an output column $o$, where $f$ refers to the imputation model learned on the observed values in column $o$ and $\tilde{\mathbf{x}}_{\mathcal{I}}$ refers to the concatenation of the features extracted from all input columns $\tilde{\mathbf{x}}_{\mathcal{I}} = [\phi_1(\mathbf{x}^1), \phi_2(\mathbf{x}^2), \ldots, \phi_{C_I}(\mathbf{x}^{C_I})]$, see also Figure 2. Depending on the data type in the output column, $f$ is fitted using either a regression or a cross-entropy loss. The API allows imputation of missing values in a table by simply passing in a `pandas` dataframe and specifying the input and output columns, see Figure 1 (right). Alternatively, all missing values in a dataframe can be imputed by calling `SimpleImputer.complete(df)`. Additionally `DataWig` has a number of features that help to automate end-to-end imputation for practitioners: The data types are detected using heuristics and the corresponding features are learned automatically during the training of the imputation model. All hyperparameters and neural architectures are optimized using random search (Bergstra and Bengio, 2012), which can be constrained to a specified time limit. Probabilistic model outputs are automatically calibrated on the validation set (Guo et al., 2017), and if requested explanations for the imputations can be computed for string input columns to better understand the imputations. Moreover, the model is equipped with functionality to compensate for label shift between the training and unlabelled production data using in the approach proposed by Lipton et al. (2018).

## 3. Evaluation

In Figure 3 we compare `DataWig` on *numerical missing value imputation* against three methods from the `fancyimpute` package (mean, KNN and matrix factorization) and two methods from the IterativeImputer of `sklearn` with the estimators RandomForestRegressor and LinearRegression, which are similar to the MissForest approach (Stekhoven and Bühlmann, 2012; van Buuren, 2018), and MICE with a linear model (Little and Rubin, 2002); iterative

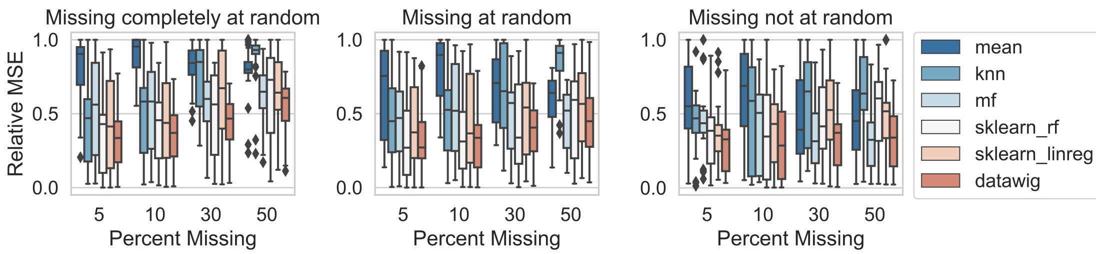


Figure 3: Comparison of imputation performance across several synthetic and real world data sets with varying amounts of missing data and missingness structure. Relative mean squared errors were normalized to the highest error in a condition.

imputation here means that 10 consecutive imputation rounds were performed for replacing the missing values in the input columns. All methods were evaluated on one synthetic linear and one synthetic non-linear problem and five real data sets available in `sklearn`. Values were discarded either completely at random, at random (conditioned on values in another randomly chosen column being in a random interval) or not at random (conditioned on values to be discarded). In Figure 3 the relative mean-squared error is shown, normalized to the highest MSE in a given condition. For `DataWig` the `SimpleImputer.complete` function with random search for hyperparameter tuning was used. For each baseline method, grid search was performed for hyperparameter optimization on a validation set, test errors were obtained on a separate test set, for details and unnormalized results see benchmarks github repository. We observe that `DataWig` compares favourably with other implementations for numeric imputation, even in the difficult missing-not-at-random condition. These experiments allow for a comparison of `DataWig` with existing packages designed for numeric data. For imputation with text data, standard numerical imputation methods cannot be used. When comparing `DataWig` with mode imputation and string matching (Dallachiesa et al., 2013) `DataWig` achieves a median F1-score of 60% across three tasks, imputation of the Wikipedia attributes *birth-place, genre* and *location*, with a simple n-gram model. Mode imputation reached a median F1-score of 0.7% and string matching 7.5% (Biessmann et al., 2018).

## 4. Conclusion

We present `DataWig`, a software package that enables practitioners such as data engineers to achieve state-of-the-art imputation results with minimal set up and maintenance. Our package complements the open source ecosystem by offering deep learning modules combined with neural architecture search and end-to-end optimization of the imputation pipeline, also for data types like free text fields. `DataWig` compares favorably to existing imputation approaches on numeric imputation problems, but also when imputing values in tables containing unstructured text. The software, unit tests, and all experiments are available under `github.com/awslabs/datawig`. While the present version of our software does not impute free form text or images, an interesting topic for future research is using generative models for these types of data building on recent advancements in neural missing value imputation (Zhang et al., 2018; Camino et al., 2019).

## References


Gustavo Batista and Maria Carolina Monard. An analysis of four missing data treatment methods for supervised learning. *Applied Artificial Intelligence*, 17(5-6):519–533, 2003.

benchmarks github repository. `https://github.com/awslabs/datawig/blob/master/experiments/benchmarks.py`.

James Bergstra and Yoshua Bengio. Random Search for Hyper-Parameter Optimization. *Journal of Machine Learning Research*, 13:281–305, 2012. URL `http://dl.acm.org/citation.cfm?id=2188395`.

Felix Biessmann, David Salinas, Sebastian Schelter, Philipp Schmidt, and Dustin Lange. "deep" learning for missing value imputation in tables with non-numerical data. In *International Conference on Information and Knowledge Management (CIKM)*, 2018.

Ramiro D. Camino, Christian A. Hammerschmidt, and Radu State. Improving Missing Data Imputation with Deep Generative Models. feb 2019. URL `http://arxiv.org/abs/1902.10666`.

Michele Dallachiesa, Amr Ebaid, Ahmed Eldawy, Ahmed Elmagarmid, Ihab F Ilyas, Mourad Ouzzani, and Nan Tang. Nadeef: a commodity data cleaning system. In *ACM SIGMOD*, pages 541–552. ACM, 2013.

Lovedeep Gondara and Ke Wang. Multiple imputation using deep denoising autoencoders. *CoRR*, abs/1705.02737, 2017. URL `http://arxiv.org/abs/1705.02737`.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On Calibration of Modern Neural Networks. In *International Conference on Machine Learning (ICML)*, 2017.

Yehuda Koren, Robert M. Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. *IEEE Computer*, 42(8):30–37, 2009.

Zachary C. Lipton, Yu-Xiang Wang, and Alex Smola. Detecting and Correcting for Label Shift with Black Box Predictors. *International Conference on Machine Learning (ICML)*, 2018.

R. J. A. Little and D. B. Rubin. *Statistical analysis with missing data. 2nd ed.* Wiley-Interscience, Hoboken, NJ,, 2002.

Pierre-Alexandre Mattei and Jes Frellsen. MIWAE: Deep generative modelling and imputation of incomplete data sets. In *International Conference on Machine Learning (ICML)*, 2019.

Imke Mayer, Julie Josse, Nicholas Tierney, and Nathalie Vialaneix. R-miss-tastic: a unified platform for missing values methods and workflows. art. arXiv:1908.04822, 2019.

Rahul Mazumder, Trevor Hastie, and Robert Tibshirani. Spectral regularization algorithms for learning large incomplete matrices. *Journal of Machine Learning Research*, 11:2287–2322, 2010. URL `http://portal.acm.org/citation.cfm?id=1859931`.

A Nazabal, Pablo M Olmos, Zoubin Ghahramani, and Isabel Valera. Handling Incomplete Heterogeneous Data using VAEs. 2018. URL `https://arxiv.org/pdf/1807.03653.pdf`.

Sebastian Schelter, Felix Biessmann, Tim Januschowski, David Salinas, Stephan Seufert, and Gyuri Szarvas. On challenges in machine learning model management. *IEEE Data Engineering Bulletin*, 41, 12 2018.

D Sculley, G Holt, D Golovin, E Davydov, T Phillips, D Ebner, V Chaudhary, M Young, and D Dennison. Hidden Technical Debt in Machine Learning Systems. *Neural Information Processing Systems (NeurIPS)*, 2015.

Daniel J Stekhoven and Peter Bühlmann. MissForest—non-parametric missing value imputation for mixed-type data. *Bioinformatics*, 28(1):112–118, 2012.

Olga G. Troyanskaya, Michael N. Cantor, Gavin Sherlock, Patrick O. Brown, Trevor Hastie, Robert Tibshirani, David Botstein, and Russ B. Altman. Missing value estimation methods for DNA microarrays. *Bioinformatics*, 17(6):520–525, 2001.

S. van Buuren. *Flexible Imputation of Missing Data. 2nd ed.* CRC/Chapman & Hall, 2018.

Jinsung Yoon, James Jordon, and Mihaela van der Schaar. GAIN: Missing Data Imputation using Generative Adversarial Nets. *International Conference on Machine Learning (ICML)*, 2018. URL `http://arxiv.org/abs/1806.02920`.

Hongbao Zhang, Pengtao Xie, and Eric P. Xing. Missing value imputation based on deep generative models. *CoRR*, abs/1808.01684, 2018. URL `http://arxiv.org/abs/1808.01684`.