Australian Government
Department of Industry, Science, Energy and Resources

Office of the Chief Economist

Australian Government

Economic Data and Analysis Network

Data Integration Partnership for Australia

RESEARCH PAPER 3/2020

Sharpening the BLADE: Missing Data Imputation using Supervised Machine Learning

Marcus Suresh[A & B], Ronnie Taib[B], Yanchang Zhao[B] & Warren Jin[B]

[A] Department of Industry, Science, Energy and Resources

[B] Data61

July 2020

Abstract

Incomplete data are quite common which can deteriorate statistical inference, often affecting evidence-based policymaking. A typical example is the Business Longitudinal Analysis Data Environment (BLADE), an Australian Government's national data asset. In this paper, motivated by helping BLADE practitioners select and implement advanced imputation methods with a solid understanding of the impact different methods will have on data accuracy and reliability, we implement and examine performance of data imputation techniques based on 12 machine learning algorithms. They range from linear regression to neural networks. We compare the performance of these algorithms and assess the impact of various settings, including the number of input features and the length of time spans. To examine generalisability, we also impute two features with distinct characteristics. Experimental results show that three ensemble algorithms: extra trees regressor, bagging regressor and random forest consistently maintain high imputation performance over the benchmark linear regression across a range of performance metrics. Among them, we would recommend the extra trees regressor for its accuracy and computational efficiency.

JEL Codes: C15, C55, C63

Keywords: artificial intelligence, machine learning, data imputation and government administrative data

For further information on this research paper please contact:

Marcus Suresh

Firm Analysis Section

Department of Industry, Science, Energy and Resources

GPO Box 9839

Canberra ACT 2601

Phone : +61 2 6102 8776

Email: marcus.suresh@industry.gov.au

Key points

1. **We employ artificial intelligence to facilitate the generation of synthetic data for the purposes of imputing high-value targets in the Business Longitudinal Analysis Data Environment (BLADE).**
2. **To achieve this we developed PyImpuyte – a Python package which carries out a series of repeated controlled experiments to objectively train and evaluate 12 supervised machine learning algorithms.**
3. **Using PyImpuyte we contribute to the artificial intelligence and applied machine learning literature with the following discoveries:**
    - **experimental results show that without domain-specific knowledge and hyperparameter tuning, three ensemble algorithms - extra trees regressor, bagging regressor and random forest consistently maintain high imputation performance for *Turnover and FTE* over the benchmark linear regression across an exhaustive range of performance metrics;**
    - **random forest and bagging regressor exhibited greater resilience compared to extra trees regressor when we constrain the ingestion of prior knowledge;**
    - **in terms of accuracy we recommend the bagging regressor and random forest regressor for the imputation of missing values; and**
    - **if computational resources are unavailable, we recommend extra trees regressor for its accuracy and computational efficiency.**

1. Introduction

On a daily basis, a multiplicity of important decisions affecting human lives are made. However, in nearly all instances, real-world data are incomplete and suffers from varying degrees of sparsity. This can deteriorate statistical inference and affect evidence-based policymaking. This is traditionally addressed by dropping missing data, but this leads to unreliable outcomes if the residual data is not representative of the whole dataset. A popular and cost-effective remedy is to impute synthetic data, however, the current methods usually remain rudimentary (Bakhtiari, 2019) and inconsistent across agencies and datasets.

The Australian Government's national statistical asset -- the Business Longitudinal Analysis Data Environment (BLADE) (Australian Bureau of Statistics, 2019) is one such example. It combines business tax data and information from the Australian Bureau of Statistics (ABS) surveys with data about the use of government programs from financial years (FY) 2001 to 2016. It is currently being used by various government agencies to study the factors that drive business performance, innovation, job creation, competitiveness and productivity.

In this paper, we explore advanced imputation methods underpinned by machine learning regressors as a way to improve coverage and reliability during imputation and benchmark them using BLADE as our test case. We review, select and compare 12 algorithms, and further examine their benefits and limitations along various dimensions. Our results provide compelling empirical evidence that ensemble algorithms are best suited to generate synthetic data that accurately reflects the ground truth.

2. Related Work

Most statistical and machine learning algorithms cannot handle incomplete data-sets directly (Khan, Ahmad, & Mihailidis, 2019). As such, there have been a plethora of strategies developed to cope with missing values. Some researchers suggest directly modelling datasets with missing values (Bakar & Jin, 2019). However, this means that for every dataset and most statistical inference, we need to build up sophisticated models which are labour-intensive and often computation-intensive. Alternatively, people often use a two-phase procedure -- obtaining a complete dataset (or subset) and then apply conventional methods to analyse the datasets. There are roughly three classes of methods:

1. A commonly used method is dropping instances with missing values (Little & Rubin, 2014). This approach is suitable when there are only a few instances with values missing randomly. For larger instances of missing values, list-wise deletion results in bulk loss of information and smaller, non-representative data leading to biased results.

2. The second class of methods are simple imputation methods, such as mean and median imputation, or the most common, value imputation. However, they often underestimate the variance, ignore the correlation

between the features and lead to poor imputation (Little & Rubin, 2014).

The third class of methods are building statistical or machine learning models based on data or domain knowledge to impute missing values. They usually take into account various covariance structures, such as temporal dependence for time series or longitudinal data, and cross-variable dependence (Jin, Wong, & Leung, 2005 and Little & Rubin, 2014). These methods impute missing values based on a distribution conditional on other features and often have the best performance. In this paper, we focus on these model-based methods.

When imputing missing values, the nature or mechanism of the missingness is important (Rubin, 1976 and Little & Rubin, 2014). Missing data mechanisms could be categorised into three types: missing completely at random (MCAR) where missingness is not related to data observed or missing, missing at random (MAR) where missingness depends only on the observed variables and missing not at random (MNAR) where missingness depends on the missing values themselves. Most imputation methods assume MAR in order to produce unbiased results. However, proving that the pattern of missingness is MAR without prior knowledge of the actual mechanism itself is impossible in a real-world dataset such as BLADE.

Based on the MAR assumption, there are several other more robust statistical imputation methods, ranging from hot/cold deck imputation, maximum likelihood, expectation maximisation (EM) (Jin, Wong, & Leung, 2005 and Rubin, 1976), multivariate imputation by chained equations, to Bayes imputation (Little & Rubin, 2014). These methods are often restricted to relatively small datasets. For example, Khan, et al. (2019) performed an extensive evaluation of ensemble strategies on 8 datasets by varying the missingness ratio. Their results showed that bootstrapping was the most robust method followed by multiple imputation using EM. Bakar and Jin (2019) proposed Bayesian spatial generalised linear models to infill values for all the statistical areas (Level 2) in Australia.

Machine learning and data mining techniques are capable of extracting useful and often previously unknown knowledge from Big Data. Recently, Yoon, et al. (2018) designed a novel method for imputing missing values by adapting the Generative Adversarial Nets (GAN) architecture where they trained two models: a generative model and a discriminative model, and used a two-player minimax game. It is worth noting we cannot evaluate deep learning methods due to security restrictions in the current ABS computing environment, but they remain a possibility in the future.

Surveying the related work reveals that imputation strategies range from simple list-wise deletion to sophisticated neural networks. To date, no study has used the Australian Government's national statistical asset to evaluate supervised machine learning methods for imputation.

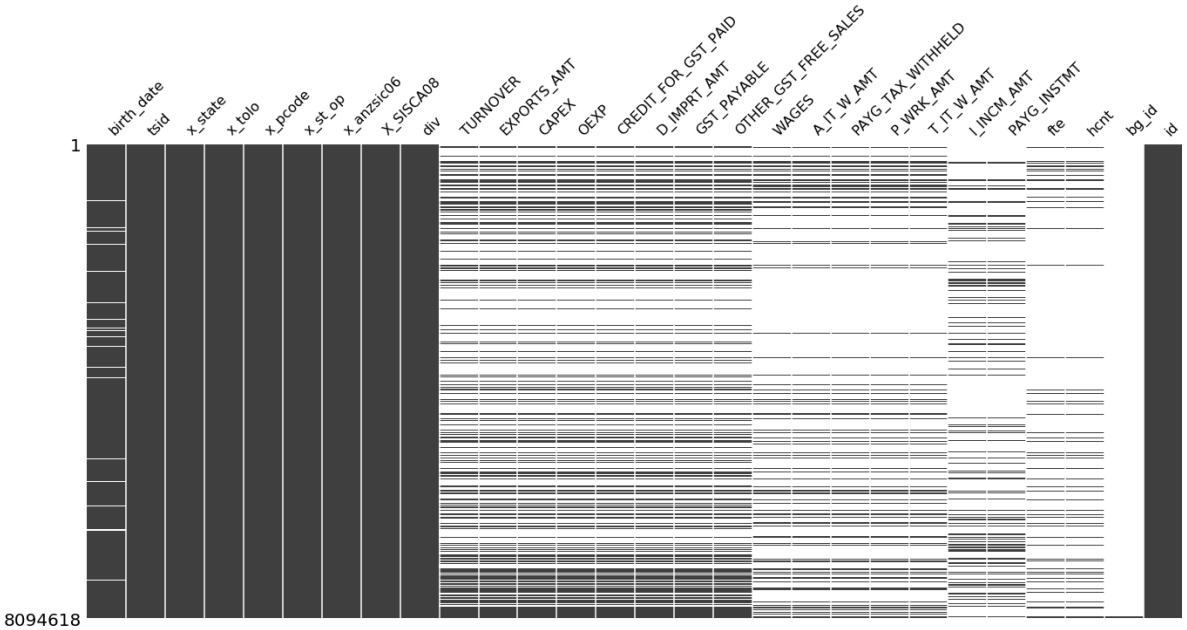
3. The BLADE dataset and Missing Values

BLADE is the Australian Government's national statistical asset which combines business tax data and information from ABS surveys with data about the use of government programs on all active Australian businesses from FY2001-02 to FY2015-16.

A de-identified extract of BLADE is available in the ABS DataLab, a secure virtual environment, for Australian public servants and researchers to undertake complex microdata analysis. The extract spans the full 15 financial years and contains 28 continuous and categorical features. In FY2015-16 there were 8,094,618 rows. The categorical features include *Indicative Data Items* such as the unit and timestamp identifiers, the industry and industrial classifiers, entity type and geo-locational data. The continuous features come from the *Business Activity Statement* (BAS) and *Pay as You Go* (PAYG) *Withholding Tax Statement*. The BAS features include turnover, export sales, capital and non-capital expenditures and total salary, wages and other tax-related payments. The PAYG features include employee headcount and its Full-Time Equivalent (FTE).

Figure 3.1 is a snapshot of the entire BLADE extract for FY2015-16 using a nullity matrix. The nullity matrix converts tabular data matrices into boolean masks based on whether individual entries contain data (which evaluates to true) or left blank (which evaluates to false). The *Indicative Data Items* are observed largely in their entirety because this information is compulsory, as illustrated by the dense vectors. Data sourced from the BAS and PAYG fields appear more sparse given that they only apply to certain types of firms such as those that are employing staff or engaging in exports.

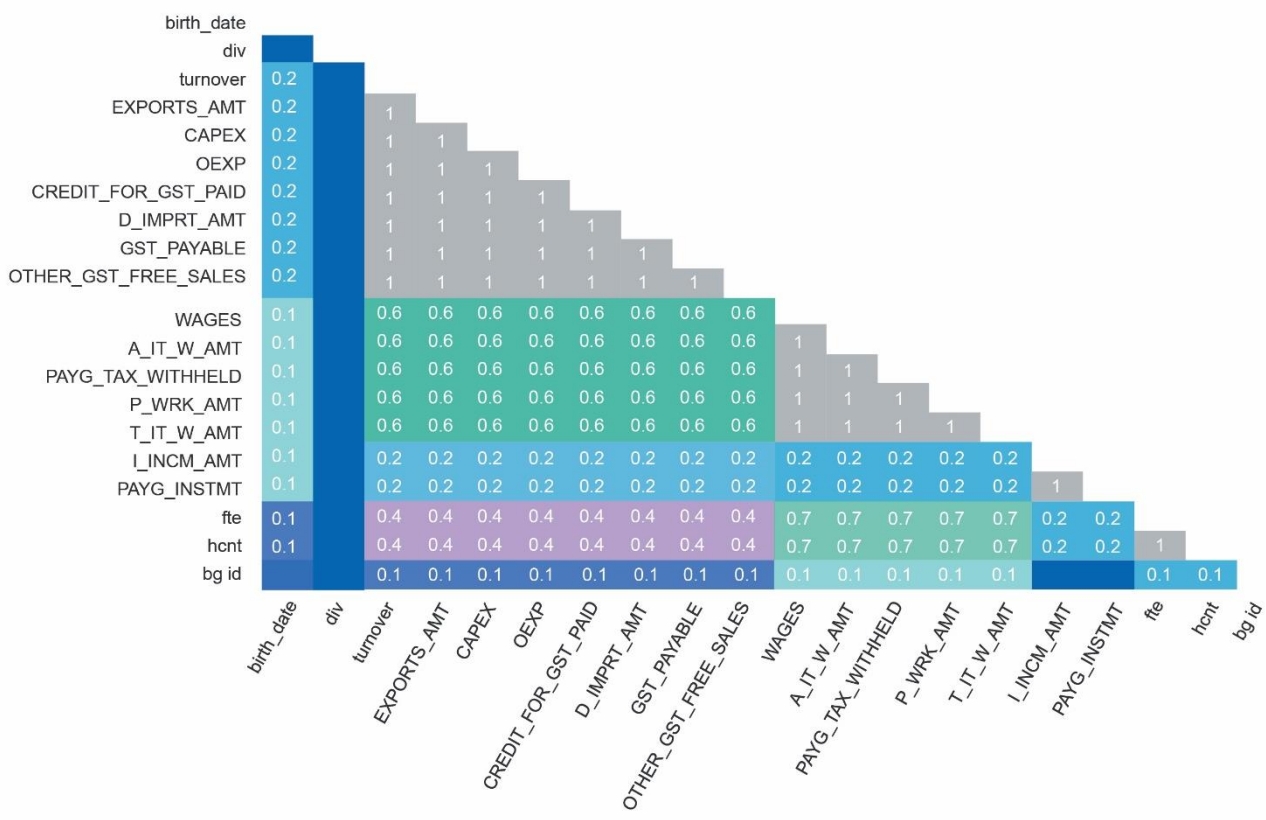
Figure 3.1: Nullity Matrix



Notes: Sparsity denotes the extent of missingness for each vector.

Source: BLADE

Figure 3.2: Correlation Heat map

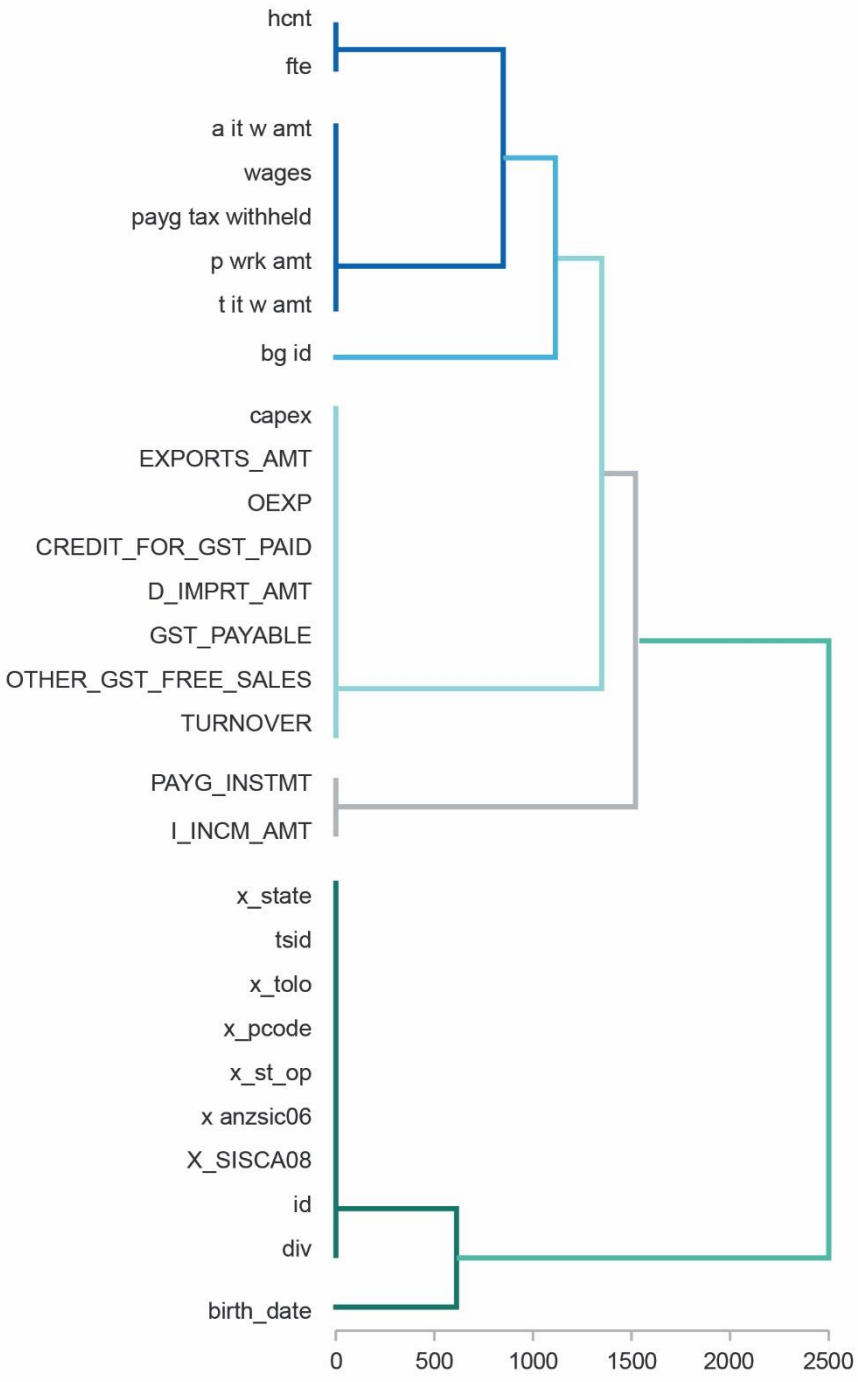


Notes: Correlation between features based on missingness.

Source: BLADE

We probe the underlying structure of missingness across features illustrated by a nullity correlation heatmap in Figure 3.2. The nullity correlation ranges from a value of zero (independent features) to +1 (dependent features). The grey tiles exhibit perfect correlation, meaning that if, for example, *Turnover* is fully observed, then *Capital Expenditure* will exhibit the same properties. Dark blue tiles indicate lower or near-zero correlation -- closer to an assumption of MAR. These features become high-value targets for imputation in Section 5.

Figure 3.3: Data nullity correlations using hierarchical clustering algorithms



Source: BLADE

In Figure 3.3 we also examine higher-cardinality data nullity correlations using hierarchical clustering algorithms to generate and sort each leaf (features) into clusters based on their missingness pattern. The dendrogram uses a hierarchical clustering algorithm to bin features against one another by their nullity correlation (measured in terms of binary distance). At each step of the tree, the features are split up based on which combination minimises the

Euclidean distance of the remaining clusters. The more monotone the set of features, the closer their total distance is to zero, and the closer their average distance (the y-axis) is to zero. Cluster leaves which linked together at a distance of zero fully predict one another's presence. In this specific example the dendrogram glues together the features which are required and therefore present in every record. The 3 broad clusters discovered resemble the underlying structure of Figure 3.2. In the first cluster, features from the *Indicative Data Items* are fully observed, followed by features from *BAS* and the *PAYG Withholding Tax Statement*.

4. Methodology

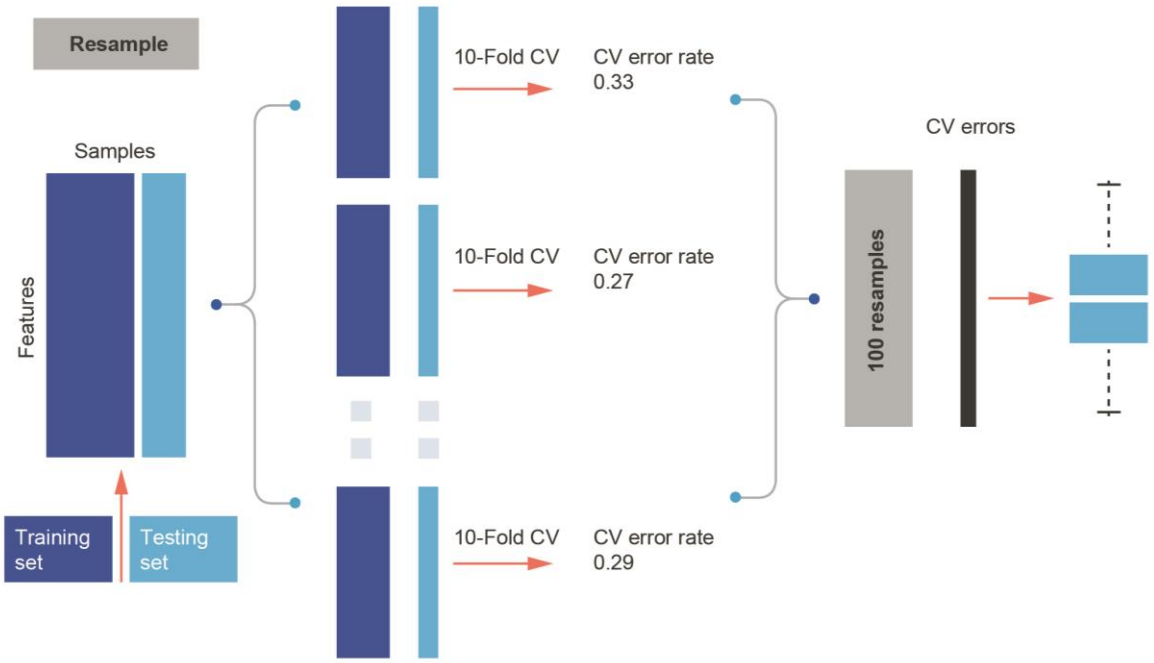
4.1 Process

Data analysis is performed in the ABS DataLab using Python. Based on suggestions by domain experts, we pre-process the data by filtering out businesses with *Turnover, Wages* and *FTE* values that are not positive. This produces a perfectly dense matrix of businesses that are deemed to be actively trading.

All features and targets are scaled using a logarithmic-transformation given by $\log10(x+ \varepsilon)$ where $\varepsilon= 1e^{-6}$ to suppress negative values during the logarithmic-transformation process. Given large corporations exhibit higher *Turnover* and *FTE*, we use this process to reduce long right tail skewness.

The benchmark presented in this paper is performed through a *repeated K-Fold* cross-validation process to train and evaluate our 12 regression algorithms. For each fold, 90% of the data is used for training and the remaining 10% for testing. 10 folds using a different testing set are used to produce performance metrics for each algorithm. Finally, the risk of unbalanced folds is counterbalanced by repeating the entire process 10 times, averaging the performance metrics accordingly. These combined performance metrics are presented in Section 5. Figure 4.2 provides a conceptual overview of the *repeated K-Fold* cross-validation process.

Figure 4.1: Repeated K-Fold cross-validation



Source: Sci-kit Learn

4.2 Regression Algorithms

We brief the 12 learning algorithms (Pedregosa, et al., 2011) below. They were seeded with the *Scikit-learn* v0.20.3 default hyper-parameters.

Linear Regression: A linear modelling technique that seeks to minimise the residual sum of squares between the observed *y* and predicted responses from other features X through linear approximation given by:

$$\min_{w} \; ||Xw - y||_2^2$$

Decision Trees: An estimator that uses a series of boolean functions constructed by if-else conditions which are highly interpretable.

Ridge: A technique that seeks to minimise ridge coefficients through a penalised residual sum of squares given by:

$$\min_{w} \; ||Xw - y||_2^2 + \alpha ||w||_2^2$$

Bayesian Ridge: A ridge regression technique using uninformative priors such as a spherical Gaussian on w like:

$$p(w \mid \lambda) = \mathrm{N}(w \mid 0, \lambda^{-1}\mathbf{I}_p)$$

LassoCV: A linear model trained with *l1* prior as regularisation with the minimisation objective function:

$$\min_{w} \frac{1}{2n_{\text{samples}}} \| Xw - y \|_2^2 + \alpha \| w \|_1$$

Orthogonal Matching PursuitCV: An algorithm for approximating the fit of a linear model with constraints imposed on the number of non-zero coefficients given by:

$$\arg\min_{\gamma} \| \gamma \|_0 \text{ subject to } \| y - X\gamma \|_2^2 \leq \text{tol}$$

Bagging Regressor: An ensemble meta-estimator that fits base regressors each on random subsets of the original dataset and then aggregates their individual predictions to form a final prediction.

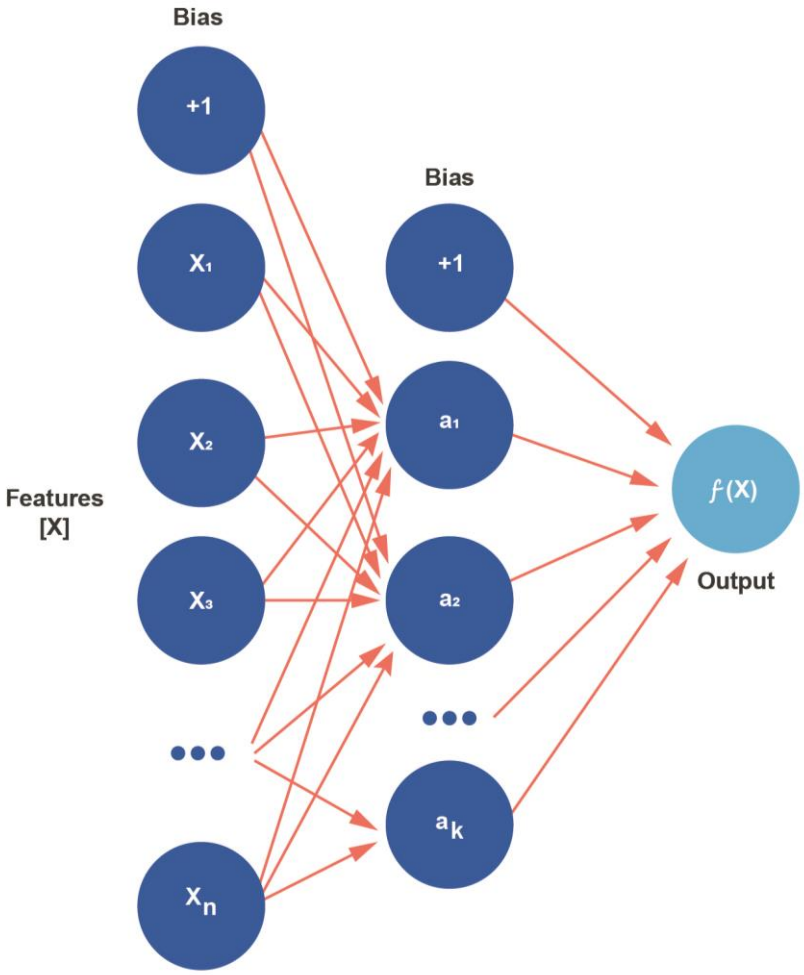
Extra Trees Regressor: An estimator that fits a number of randomised decision trees (extra-trees) on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control for over-fitting.

Gradient Boosting Regressor: An additive model that allows for the optimisation of arbitrary differential loss functions. In each stage, a regression tree is fit on the negative gradient of the given loss function.

Random Forest Regressor: A number of classifying decision trees on various sub-samples of the dataset and use averaging to improve the predictive accuracy and control for over-fitting.

Multi-layer Perceptron: A supervised learning algorithm as shown in Figure 4.2 that learns a function $f(\cdot): R^m \rightarrow R^o$ by training on a dataset, where *m* is the number of dimensions for input and *o* is the number of dimensions for output. Given a set of features $X=x_1,x_2,...,x_m$ and a target *y*, it can learn a non-linear function approximator for either classification or regression. The input layer consists of a set of neurons $\{x_i | x_1, x_2, \ldots, x_m\}$ representing the input features. Each neuron in the hidden later transforms the values form the previous layer with a weighted linear summation $w_1x_1 + w_2x_2 + \ldots + w_mx_m$, followed by a non-linear activation function $g(\cdot): R^m \rightarrow R^o$.

Figure 4.2: Multi-layer Perceptron



Source: Sci-kit Learn

Generalised Additive Models: A non-linear modelling technique where predictor features can be modelled non-parametrically in addition to linear and polynomial terms. GAMs are useful when the relationship between features are expected to be of a more complex form. Its recent variation could include variable interaction (Wood, 2017).

The 12 algorithms are seeded using the default hyper-parameters defined in *Scikit-learn* version 0.20.3.

4.3 Performance Metrics

The experimental results in Section 5 are evaluated through five performance metrics. These are Mean Absolute Error (MAE), symmetric Mean Absolute Percentage Error (sMAPE), Root Mean Squared Error (RMSE), Mean Squared Error (MSE) and R^2, given by:

$$\mathrm{MAE} = \frac{\sum_{i=1}^{n} |\hat{y}_i - y_i|}{n}$$

$$\text{sMAPE} = \frac{100\%}{n} \sum_{i=1}^{n} \frac{|\hat{y}_i - y_i|}{(|\hat{y}_i| + |y_i|)/2}$$

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^{n} (\hat{y}_i - y_i)^2}{n}}$$

$$\text{MSE} = \frac{\sum_{i=1}^{n} (\hat{y}_i - y_i)^2}{n}$$

$$\text{R}^2 = 1 - \frac{\sum_{i=1}^{n} (\hat{y}_i - y_i)^2}{\sum_{i=1}^{n} (y_i - \bar{y})^2}$$

where n is the number of observations, y_i is the i-th observed value, $\hat{y}_i$ is its predicted value and $\bar{y}$ is the mean of y.

4.4 Conditions

Our experiment is a 12 x 3 x 2 x 2 design, described in Table 4.1. [1] To ensure the volume of training data remains equal across conditions, it is run on a 1 million row subset of the original, unfiltered BLADE data. This represents 176,683 rows for 1 financial year (FY) and 579,564 rows for 3FY after pre-processing. The experiments were conducted in the ABS DataLab, providing a shared Intel 10-core 2.2Ghz server with 133Gb of physical RAM.

[1] The 3 input features are Capital Expenditure, Wages and FTE/Turnover (depending on the target feature). The 7 input features include the preceding features in addition to Export Sales, Imported Goods with Deferred GST, Non-Capital Purchases and Headcount. The 14 input features include all preceding features and GST on Purchases, GST on Sales, Other GST-free sales, Amount Withheld from Salary, PAYG Tax Withheld, Amount Withheld from Salary, Amount Withheld from Payments and Amount Withheld from Investments.

Table 4.1: Experiment conditions

# Levels	Conditions	Values
12	Algorithms	See Section 4.2
3	Input features	3, 7, 14 BLADE features
2	Target features	Turnover, FTE
2	Time spans: number of financial years in the data	3FY (2014-16), 1FY(2016)

5. Experiment Evaluation

5.1 Algorithm comparisons for Turnover

We first examine *Turnover* as a target feature, comparing the results of all algorithms, input features and time spans, as shown in Table 5.1. In all cases, the set of 14 features perform better than 7 features, itself performing better than 3 features. This applies to all algorithms and metrics. For this reason, we present results from the 14 feature set and examine the impact of the number of input features on performance.

Using our performance metrics, the ensemble algorithms provide clearly better results than the other types of regressors. In particular, the Bagging Regressor (BR) and Random Forest Regressor (RF) exhibit the lowest MAE at 0.060, closely followed by the Extra Tree Regressor (ETR) at 0.063. The errors are an order of magnitude lower than for most linear methods for which the best MAE is 0.253, for our baseline Linear Regression (LR). The Multi-layer Perceptron's (MLP's) MAE is larger than that of the ensemble methods, yet competitive at 0.078. It is well ahead of the Generalised Additive Models (GAM) at 0.134.

Looking at RMSE, the trends are confirmed and the same three ensemble methods again perform best. This time the ETR exhibits the lowest error at 0.174, but BR and RF are very close with 0.177. Again, the MLP's performance is inferior but reasonably close at 0.185, followed by GAM at 0.244. The linear methods are clearly inferior, and the LR's best RMSE is at 0.381.

As expected, these trends are replicated for sMAPE and MSE, preserving the same rank ordering observed previously. In terms of R^2, the ETR is the best at 93.9%, closely followed by RF and BR, confirming the results from the individual metrics through strong correlation.

Based on these results, the rest of this paper will focus on the top 3 performing algorithms -- BR, RF and ETR -- and refer to LR as a baseline.

Table 5.1: Results for Turnover, 3FY (2014-16)

Algorithm	#Feat	MAE	RMSE	sMAPE	MSE	R^2	Time (s)
Linear Regression	14	0.253	0.381	4.62%	0.145	70.82%	333
Decision Tree	14	0.071	0.236	1.39%	0.056	88.79%	2,003
Ridge Regression	14	0.253	0.381	4.62%	0.145	70.82%	58
Bayesian Ridge	14	0.253	0.381	4.62%	0.145	70.82%	416
LassoCV	14	0.253	0.381	4.62%	0.145	70.82%	1,407
OMPursuit CV	14	0.262	0.392	4.79%	0.154	69.05%	672
Bagging	14	**0.060**	0.177	**1.16%**	0.031	93.69%	18,348
Extra Trees	14	0.063	**0.174**	**1.21%**	**0.030**	**93.90%**	5,709
Gradient Boosting	14	0.074	0.191	1.41%	0.037	92.63%	16,725
Random Forest	14	**0.060**	0.177	**1.16%**	0.031	93.70%	17,527
ML Perceptron	14	0.078	0.185	1.48%	0.034	93.35%	85,805
GAM	14	0.134	0.244	2.47%	0.060	87.98%	9,472

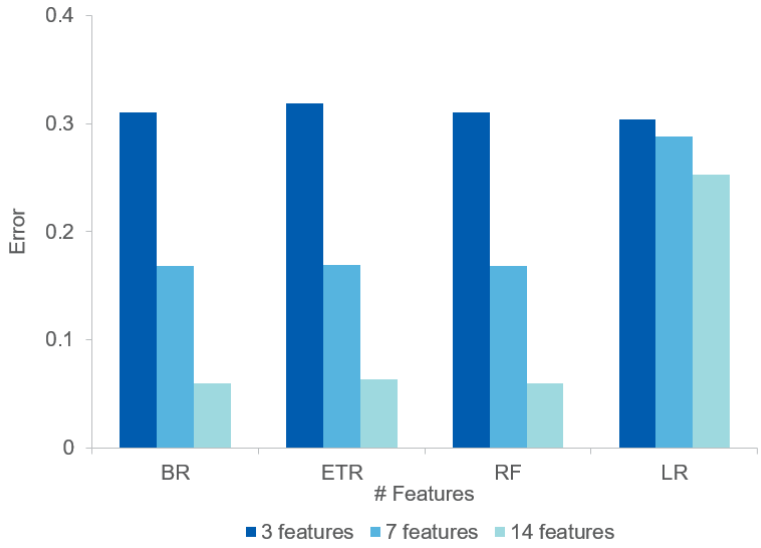
5.2 Impact of Input Features

Focusing on the top 3 algorithms and the LR as the baseline, we now compare the relative performances corresponding to the 3 input feature conditions. In the base condition, we only use 3 features from the dataset, then increase to 7 and finally 14. We use domain knowledge in the selection of features that reflect well-established drivers of productivity growth (Solow, 1956), being capital and labour inputs in the base condition. Similarly, in the second condition, we include the same features in the prior condition and expand it to include imports and exports and other expenditures. In the third condition, we use all continuous features as inputs. While the MAE decreases only slightly for the LR baseline, by 5.3% from 3 to 7 features and 16.8% from 3 to 14 features, the improvements are more dramatic for the ensemble regressors, as shown in Figure 5.1. They register error reductions of 45.8-47.0% when moving from 3 to 7 input features, and 80.3-80.6% when moving from 3 to 14 input features.

As expected, the trends are very similar for RMSE, as shown in Figure 5.2. The improvements for LR are 4.3% from 3 to 7 features, and 12.8% from 3 to 14 features. While more moderate for RMSE than for MAE, the ensemble methods display again a strong improvement as the number of features increases, in the range 32.5-33.7% from 3 to 7 features, and 60.0-61.7% from 3 to 14 features.

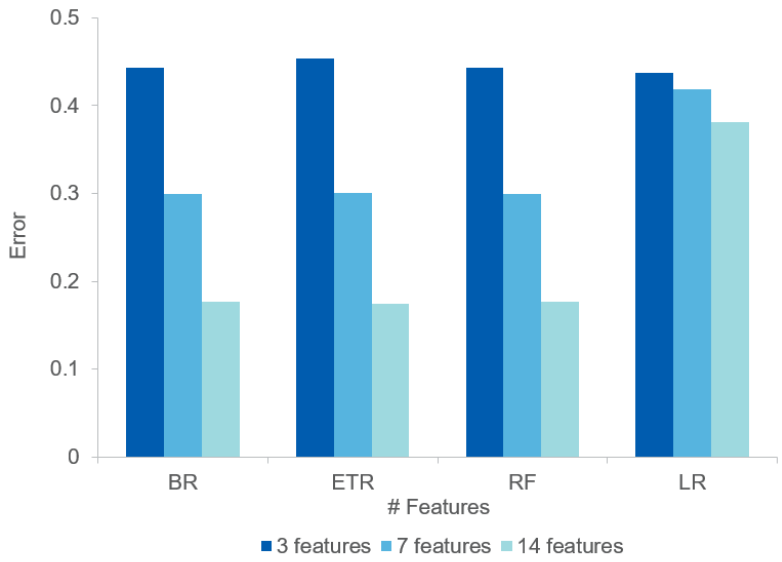
Intuitively, adding more features brings additional prior knowledge correlated to the target feature. However, the correlations are clearly not linear, explaining why the ensemble methods are better suited at capturing complex relationships than LR, hence exhibit much stronger improvement. Based on these findings, we set out to assess the impact of prior knowledge by considering different time spans using only the 14 input features condition.

Figure 5.1: MAE of *Turnover* prediction



Source: BLADE

Figure 5.2: RMSE of *Turnover* prediction



Source: BLADE

5.3 Impact of Time Spans

In some cases, only a single year of data may be available to impute missing data, which precludes algorithms from potentially learning from prior knowledge (time series patterns). We examine this by producing the results of the MAE and RMSE metrics for all algorithms over a single financial year, FY2016 in Table 5.2.

The RF and BR clearly surpass the other algorithms on most performance metrics. In absolute terms, their MAE are 0.062 and 0.063, hence very similar to the 3FY results in Table 5.1 at 0.060. Similarly, their RMSEs are 0.188, slightly worse than the 3FY value of 0.177. Coming third is ETR, but not as close to RF and BR as was the case in the 3FY results. It's MAE and RMSE now stand at 0.070 and 0.191. Our results show that the lack of time-series data affects algorithms to different extents. BR registers a performance drop (accounted for as an increase in error) of -4.80% in MAE and -6.42% in RMSE. RF registers similar drops of -4.69% and -6.17% respectively. For ETR, the drop is the largest of all algorithms, -11.56% in MAE and -9.62% in RMSE, indicating a higher dependence on time-series information. In terms of baseline, LR registered a drop in the MAE from 0.253 to 0.265 (-4.59%). This modest drop is not due to resilience from a lack of time-series data than to the moderate performance it achieves in the first place. These results indicate that all algorithms indeed make use of prior knowledge coded into the time series, with RF and BR demonstrating their resilience even without it.

Table 5.2: Results for Turnover, 1FY (2016)

Algorithms	#Features	MAE	RMSE
Linear Regression	14	0.265	0.389
Decision Tree	14	0.075	0.247
Ridge Regression	14	0.265	0.389
Bayesian Ridge	14	0.265	0.389
LassoCV	14	0.265	0.389
OMPursuitCV	14	0.275	0.402
Bagging	14	0.063	**0.188**
Extra Trees	14	0.070	0.191
Gradient Boosting	14	0.075	0.194
Random Forrest	14	**0.062**	**0.188**
ML Perceptron	14	0.087	0.191
GAM	14	0.136	0.248

5.4 Experimental results for FTE

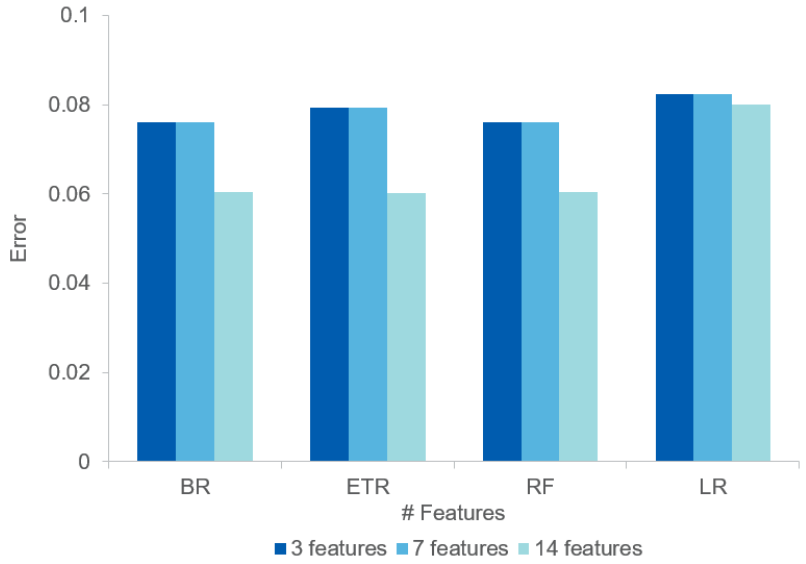
The same experiment was carried out using *FTE* as the target, as it is one of the most sparse vectors in the entire dataset and has a substantially different distribution to *Turnover.*

As illustrated in Figure 5.3, the differences between algorithms are smaller than for *Turnover*. Performance still increases as more input features are used, with the best result achieved by ETR with 14 input features registering a MAE of 0.060. This value is very close to ETR's performance on *Turnover* with 14 input features (0.063). However, using 3 features only, ETR's performance, 0.079, is superior to 0.316 for *Turnover*.

The same pattern applies to most algorithms and can be looked at in terms of improvement as more features are added. For BR, ETR and RF, moving from 3 to 7 features improves MAE by 8.4-11.5%, while from 3 to 14 features improves MAE by 20.5-24.2%. These ranges are much lower than that observed for the same algorithms applied to *Turnover* (45.8-47.0% and 80.3-80.6%) as we have seen earlier. The improvement for LR is also very modest this time, 0.8% from 3 to 7 features, and 3.0% from 3 to 14 features.

The differences in results obtained across the targets with different distribution help us qualify the resilience of the algorithms and hence their potential applicability to other microdata sets. In essence, the best performing algorithms manage to reach similar levels of performance as more features are added, indicating that using more features are indeed useful. However, in some cases, the gain in performance may be modest, in which case fewer features may be used to decrease processing time.

Figure 5.3: MAE of FTE prediction

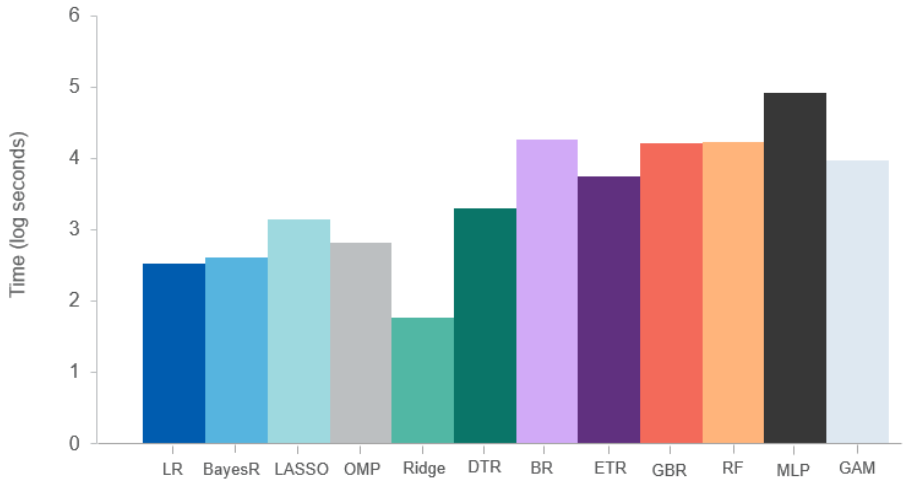


Source: BLADE

5.5 Processing Time

Figure 5.4 shows the elapsed processing time for training and imputation of each algorithm on the 3FY data, testing the 14 input feature condition for the *Turnover* target. The first 5 bars are the linear models which have relatively low processing times ranging from 58 to 1,407s for Ridge Regression and LassoCV. The ensemble family of algorithms are among the highest performers and orders of magnitude more computationally-intensive, up to 55 times longer than LR. Their processing times range from 2,003 to 18,348s for DTR and RF. The clear outlier is the MLP at 85,805s or 4.9 times slower than BR, ETR and RF.

Figure 5.4: Processing time (log-seconds)



Source: BLADE

6. Discussion

The experiment presented in this paper demonstrates the benefits of using machine learning-based imputation algorithms on national microdata sets such as BLADE. The high-performance outcomes achieved should encourage statistical and government agencies to reliably improve their imputation for greater data coverage. Our results help practitioners make the best decisions in terms of algorithms and input features, based on their dataset and analysis needs, while understanding the impact of different imputation methods.

Generally speaking, a single simple model, like a decision tree, is sensitive to training data and the results are likely to be overfitting and unstable. Ensemble algorithms, on the other hand, build multiple sub-models with multiple sub-samples of the dataset and produce a set of simple models that are weakly correlated with high variance, combining their results to make the final prediction. The RF, in particular, introduces additional variance by using a random sample of features for each individual sub-model. However, ensemble algorithms come at the cost of longer processing time.

To maximise the generalisability of our findings, we processed two target values with substantially different characteristics. Cross-validation accuracy results for both *Turnover* and *FTE* are seen as high enough to assist analysts using BLADE. The 1.16% - 1.2% sMAPE for *Turnover* using BR, ETR and RF indicate imputed values are only slightly off from ground truth. Similarly, as indicated by R^2 values, around 94% variability of the true values were captured by the imputed values of these 3 algorithms.

We also quantified how more input features could substantially improve the imputation performance. Interestingly, the benefits were less pronounced for *FTE*, possibly because (i) less training data are available, only about a third of

Turnover; and (ii) *FTE* has a more complicated non-linear relationship to input features because part-time effort may not be reflected linearly to *Turnover.*

The main limitation of our work stems from keeping the process simple to ensure easy adoption and higher generalisability. However, tuning the algorithms' hyper-parameters to each dataset could substantially improve imputation performance. It may also dramatically reduce processing time. Another potential limitation lies in using logarithm transformation to address data skewness. Practitioners will need to adapt scaling techniques to the characteristics of their data.

In the future, we plan to perform feature selection to assess the compared benefits of data-driven feature ranking on imputation performance. This may increase the complexity of the process but improve performance and reduce processing time. Also, it would be useful to validate whether using multiple-year feature values for a single business may lead to more reliable or accurate imputation performance as the temporal dependency could be used explicitly. Finally, we plan to further test these methods on other government datasets.

7. Conclusion

We conducted a comprehensive experimental evaluation of machine learning-based imputation algorithms on the Australian Government's national statistical asset -- BLADE. Using two target features with distinct characteristics, *Turnover* and *FTE*, we compared 12 machine learning-based imputation algorithms and found that the extra trees regressor, bagging regressor and random forest consistently maintain high imputation performance over the benchmark linear regression across the performance metrics outlined at Section 4.3.

We provided detailed results along each algorithm, the number of input features, time spans and processing time conditions. Based on our results, we recommend using extra trees regressor for its overall imputation performance and computational efficiency. This is the most promising algorithm for increasing data coverage within microdata sets containing missing values. This work will help shed some light on novel tools for statistical and government agencies to select and implement supervised machine learning methods for imputation.

8. References

Australian Bureau of Statistics. (2019). The Business Longitudinal Analysis Data Environment (BLADE).

Bakar, K., & Jin, H. (2019). A real prediction of survey data using Bayesian spatial gen-eralised linear models. *Communications in Statistics-Simulation and Computation*, 1-16.

Bakhtiari, S. (2019). Entrepreneurship dynamics in Australia: Lessons from microdata. *Economic Record, 95*, 114-140.

Jin, H., Wong, M., & Leung, K. (2005). Scalable model-based clustering for large databases based on data summarization. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 27*(11), 1710-1719.

Khan, S., Ahmad, A., & Mihailidis, A. (2019). Bootstrapping and multiple imputationensemble approaches for missing data. *Journal of Intelligent and Fuzzy Systems*.

Little, R., & Rubin, D. (2014). *Statistical analysis with missing data.* New York: John Wiley.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., . . . Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 2825-2830.

Rubin, D. (1976). Inference and missing data. *Biometrika*, 581-592.

Solow, R. (1956). A contribution to the theory of economic growth. *Quarterly Journal of Economics, 70*, 65-94.

Wood, S. (2017). *Generalized additive models: An Introduction with R.* New York: Chapman and Hall/CRC.

Yoon, J., Jordon, J., & van der Schaar, M. (2018). Missing data imputation using generative adversarial nets.

ABS Disclaimer

The results of this study is based, in part, on ABR data supplied by the Registrar to the ABS under *A New Tax System (Australian Business Number) Act 1999* and tax data supplied by the ATO to the ABS under the *Taxation Administration Act 1953*. These require that such data is only used for the purpose of carrying out functions of the ABS. No individual information collected under the *Census and Statistics Act 1905* is provided back to the Registrar or ATO for administrative or regulatory purposes. Any discussion of data limitations or weaknesses is in the context of using the data for statistical purposes, and is not related to the ability of the data to support the ABR or ATO's core operational requirements. Legislative requirements to ensure privacy and secrecy of this data have been followed. Only people authorised under the Australian Bureau of Statistics Act 1975 have been allowed to view data about any particular firm in conducting these analyses. In accordance with the Census and Statistics Act 1905, results have been confidentialised to ensure that they are not likely to enable identification of a particular person or organisation.