The Trade-off Between Data Volume and Quality in Predicting User Satisfaction in Software Projects

Łukasz Radliński
Faculty of Computer Science and Information Technology
West Pomeranian University of Technology in Szczecin
Szczecin, Poland
lukasz.radlinski@zut.edu.pl

***Abstract*—Most predictive studies involving the ISBSG dataset used only high-quality cases according to the *Data Quality Rating* and *UFP Rating* and a few predictors with no or very few missing values. This study investigated the trade-off between data volume and quality when predicting user satisfaction in software projects. Specifically, it explored whether machine learning models would perform better when trained using a larger dataset containing some portion of low-quality data, a smaller dataset with only high-quality data, or an intermediate setting. A standardised accuracy, a "win-tie-loss" approach, and a matched-pairs rank biserial correlation coefficient were used to evaluate predictive performance. The rankings of data selection strategies for particular models were created using the Scott-Knott Effect Size Difference test. The robustness of results was assessed using Kendall W. For most models, a higher predictive accuracy was achieved when trained on a larger subset, even though it contained some low-quality data. For most models, data selection strategies were robust to data splits. The ranks of data selection strategies were stable across models. Hence, a practical recommendation for predicting user satisfaction, especially when a dataset is small, is to train predictive models on a relatively high-volume subset despite some low-quality data. Provided rankings may be helpful when setting up future experiments on user satisfaction with the ISBSG dataset.**

***Index Terms*—software projects, user satisfaction, prediction, data quality, data volume, machine learning, ISBSG**

I. Introduction

Training predictive models requires considering two aspects of training data. The first is the data volume, as it is usually desirable to have an extensive training dataset with sufficient data cases to cover as much spectrum of project diversity as possible. The second is data quality, as a training dataset with high-quality data (cases) is usually desirable. Data volume and quality often pose a trade-off for a data scientist. This study explored such a trade-off for user satisfaction prediction in software projects.

A literature and web search revealed that the ISBSG dataset [1] is the only publicly available dataset on software projects containing data on user satisfaction suitable for the present study. Hence, the study involved this single dataset. Earlier studies used it primarily for effort or productivity prediction [2]. The data quality in this dataset varies between projects, i.e., there are projects for which no data quality or integrity issues were found, but there are other projects with several missing values and discrepancies between attributes.

Two attributes in this dataset reflect an expert-based evaluation of project-level data quality, i.e., *Data Quality Rating* and, for projects whose size was expressed using function points, *UFP Rating*. The ISBSG guidelines [3] provided the following recommendation: "The ISBSG considers that projects with a data quality rating of A or B are suitable for statistical analysis. C- and D-rated projects may still provide valuable data, but uncertainty about some of their size or effort values means that it is best not to include them in statistical analyses." Since these guidelines admit a possible value in low-rated projects, this study explored to what degree this is true. This may be especially important because the subset containing values of attributes describing user satisfaction is very small, and such upfront data selection would further reduce it, causing a very low volume of training data.

Another aspect of data quality is related to the fact that most attributes in the ISBSG dataset contain a large proportion of missing values. However, machine learning models usually cannot handle data with missing values, and data preprocessing is necessary. The most straightforward solution is to use only attributes with no missing values. This may lead to discarding potentially valuable attributes. Another way is to fill in the missing values, e.g. by the mean, median or mode across cases with non-missing values. Hence, this approach does not remove any attributes but may potentially decrease the quality of a particular attribute because its several values would be inferred and unreal. An intermediate approach would remove attributes with proportions of missing values exceeding a predefined threshold and impute the missing values in the remaining attributes. The possibility of setting up various values for such a threshold gives data scientists control over the number of discarded attributes.

The particular strategies of training data preprocessing, according to the thresholds of *Data Quality Rating*, *UFP Rating*, and the fraction of valid values, were called in this study "data variants". Based on the above motivations, this paper aimed to explore whether it is better to train predictive models on more data, including low-quality cases and attributes, on a smaller dataset but higher quality or on a dataset with an intermediate volume and quality. In particular, this study investigated the following research questions (RQs):

1) How does the choice of data variant influence the performance of each model?

2) Is the performance of data variants robust to different data splits for each model?
3) Is the performance of data variants stable across models?

RQ1 identified the best- and worst-performing data variants for each model. RQ2 provided justification for the validity of the results by investigating if the data variants performed consistently across different random data splits. The RQ3 investigated whether all models benefited equally from different data variants or if the optimal variant was model-dependent.

The paper's main contributions are the rankings of data variants for particular models that provided the empirical justification for using permissive data variants. Each ranking was based on a different aggregating evaluation measure and prepared using a rigorous methodological approach. These rankings may be used as recommendations for setting up the environment for future predictive studies on user satisfaction.

The rest of this paper is organised as follows: Section II introduces background and related work. Section III explains the experimental setup. Section IV discusses the obtained results. Section V considers the threats to validity. Section VI formulates conclusions and ideas for future work.

II. Background and Related Work

The two quality-related attributes in the ISBSG dataset are defined as follows: *Data Quality Rating* as "*Project Rating. This field contains an ISBSG rating code of A, B, C or D applied to the project data by the ISBSG quality reviewers*", and *UFP Rating* as "*Unadjusted Function Point Rating. This field contains an ISBSG rating code applied to the Functional Size (Unadjusted Function Point count) data by the ISBSG quality reviewers*". Category A implies nothing suspicious in the data, category D indicates little data credibility for a given project, and categories B and C denote intermediate states. As mentioned earlier, ISBSG discourages using projects rated C and D. Hence, most studies using this dataset followed this recommendation [2], [4]–[10]. However, such arbitrary data selection may be too conservative or too optimistic [11]. *Data Quality Rating* mainly indicates data completeness [12] and omissions and inconsistencies related to data reliability [13], while *UFP Rating* mainly reflects data integrity between attributes related to project size [1]. To investigate the possible benefits of keeping low-ranked projects, such upfront data filtering was not applied in the present study.

The most common approach to missing values was excluding projects with missing data in the attributes selected for experimentation [2]. Some studies set up a threshold for a fraction of missing values, e.g. 40% [14], [15], or 25% [16], and attributes not meeting this threshold were removed. Another approach assumed marking missing values in nominal attributes as a separate category, "Unspecified" [13]. The present study investigated the predictive performance with various thresholds for such settings.

Several exploratory studies were focused on identifying relationships between user satisfaction and other attributes describing projects and processes of their development [17]–[19]. Unlike the present study, they were not predictive studies, so they were not explored further here.

Predictive studies on overall user satisfaction were rare. One investigated the predictability of customer rating for mobile apps in two app stores [20]. The results showed that the rating of 89% of apps could be predicted with 100% accuracy. Two other predictive studies used the ISBSG dataset only on projects rated A or B. The first [21] found that the best-performing models were random forest, extreme gradient boosting, and support vector machines. It also compared the predictive performance of different settings for particular models. These results served as the basis for configuring models in the present study (Section III-C). The second [22] identified three versions of random forests that performed significantly better than the other 37 predictive schemes. It also revealed that predictions in some data splits were easier than others because of high data variability across splits.

Another study [23] investigated the performance of data variants in testing effort prediction using the ISBSG dataset. It demonstrated that training most models on data rated A or B (for *Data Quality Rating* and *UFP Rating*) was not justified in terms of predictive performance.

III. Experimental Setup

A. ISBSG dataset

The raw ISBSG dataset [1] contains 9,592 cases described by 253 attributes. Data preprocessing involved several actions to make it usable by the predictive models, such as correcting or removing mistakes and inconsistencies, removing irrelevant or unclear attributes or with extremely low fractions of valid values, converting nominal multi-value attributes to dummy logical, integrating attributes, etc. Because it is beyond the scope of this paper to discuss all the details, the scripts used for such preprocessing were made available online[1].

The ISBSG dataset contains eight attributes reflecting user satisfaction with the ability of a system to meet stated objectives, the ability of a system to meet business requirements, the quality of the functionality provided, the quality of the documentation provided, the ease of use, the training given, the speed of defining solution, the speed of providing solution. Each of them was expressed on a ranked scale with the following values: '1' – user needs met to a limited extent or not at all, '2' – user needs largely met, '3' – user needs fully met, '4' – user expectations exceeded. The outcome attribute for predictions, *Satisfaction Score*, was calculated as the sum of these eight attributes.

The dataset was filtered by including cases with no missing values in these eight satisfaction attributes (134 cases). The preprocessed dataset contained 94 predictors: 14 numeric, 11 nominal, 6 naturally logical, and 63 logical created from 22 multi-value attributes. The projects in the prepared dataset were completed between 1995 and 2013. Figure 1 illustrates the distribution of *Satisfaction Score*. The range of values was 10–31, but most values fell in the narrow range of 18–23.

[1]https://doi.org/10.5281/zenodo.12591070

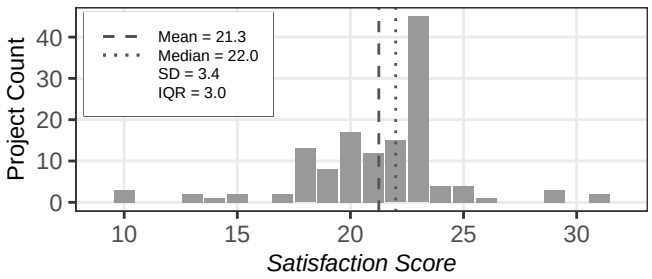

Fig. 1: Distribution of *Satisfaction Score*.

TABLE I: Considered data ratings.

Rating	*Data Quality Rating*	*UFP Rating*	n
AA	$\{A\}$	$\{A\}$	21
AM	$\{A\}$	$\{A, B, C, D, missing\}$	47
BA	$\{A, B\}$	$\{A\}$	33
BB	$\{A, B\}$	$\{A, B\}$	50
BM	$\{A, B\}$	$\{A, B, C, D, missing\}$	121
DM	$\{A, B, C, D\}$	$\{A, B, C, D, missing\}$	134

B. Data Variants

The data variant reflects a data selection strategy for the training subset involving a case selection according to the thresholds of *Data Quality Rating* and *UFP rating* and attribute selection according to a fraction of valid values (TVV). Six data ratings were considered based on the project counts for particular rating attributes (Table I). The reported case counts refer to the whole prepared dataset. Predictive models were trained on fewer cases because some cases were used in the evaluation in the nested cross-validation (Section III-D). The TVV was defined as a set of values in the 0.1–1.0 range with a step of 0.1. Hence, the study investigated sufficiently diverse 60 data variants (6 data ratings $\times$ 10 TVV settings).

The following format for data variants was used throughout the paper: "QUT", where each letter denotes a threshold for a particular setting, i.e., for *Data Quality Rating*, *UFP Rating*, and TVV, respectively. For example, the data variant AM0.3 reflects a strategy with *Data Quality Rating* categorised as A, *UFP Rating* as A, B, C, D or it had a missing value, and only with attributes having at least 30% of valid values.

C. Predictive Models

Table II summarises the predictive models used in this study. MEA and MED were simple baselines that did not use the values of predictors but predicted the outcome based only on their values in the training dataset. All other state-of-the-art models were trained and evaluated in two versions, i.e., with or without the principal component analysis (PCA) attribute synthesis algorithm. Particular settings were based on the requirements for specific models, common knowledge, preliminary experiments, and earlier studies [21]–[23]. For non-baseline models, the skewed numeric predictors were transformed using a $log_{10}(x)$ or $log_{10}(x + 1)$ function as this often improves model performance [24]–[26].

TABLE II: Predictive models and their main settings.

Model[1]	Data[2]	Impute[3]	Normalize[4]	PCA[5]
MEA: baseline predicting $mean(Y_{train})$ (`null`)	R	–	–	–
MED: baseline predicting $median(Y_{train})$ (`null`)	R	–	–	–
EN: elastic net [27] (`glmnet`)	N	+	+	–
	N	+	+	+
GBM: stochastic gradient boosting [28] (`gbm`)	N	+	+	–
	N	+	+	+
KNN: *k*-nearest neighbours [29] (`knn`)	N	+	+	–
	N	+	+	+
LM: linear regression [30] (`lm`)	N	+	+	–
	N	+	+	+
M5: model trees/rules [31], [32] (`M5`)	N	+	+	–
	N	+	+	+
NN: neural net with one hidden layer [33] (`nnet`)	N	+	+	–
	N	+	+	+
RF: random forest [34] (`ranger`)	R	+	–	–
	N	+	+	+
RT: regression tree [35] (`rpart2`)	R	–	–	–
	N	+	+	+
SVM: support vector machines [36] (`svmLinear2`)	N	+	+	–
	N	+	+	+
XT: extreme gradient boosting [37] (`xgbTree`)	N	–	–	–
	N	+	+	+

[1] The base implementation in the `caret` package provided in brackets. [2] R (regular) – dataset with nominal and numeric attributes, N (numeric) – dataset only with numeric attributes, where categorical attributes were converted with dummy encoding. [3] Missing values in categorical and logical predictors filled with their modes and in numeric predictors filled with their medians. [4] Normalization of numeric predictors with the *z*-score. [5] PCA capturing at least 85% variability in the attributes.

D. Model Training and Evaluation

Each model was trained with each data variant using nested cross-validation (CV). The internal 5-fold CV was used to determine each model's best hyperparameters, except MEA, MED and LM, which had no hyperparameters. It involved a stepwise grid search procedure [22], combining grid- and random-search. Initially, a subset of 100 combinations of predefined hyperparameter values were randomly selected and evaluated. Then, the best k sets were adapted by creating new sets with new candidate values of hyperparameters. This process was repeated until no set of hyperparameters achieving better performance could be created. For most models, two versions were trained, with or without PCA, the latter to reduce data dimensionality. Based on the performance in the internal CV, the better was selected for the final training and evaluation.

The external 5-fold CV was applied to train models using the best hyperparameters and perform their final evaluation. To ensure the stability of results, this external CV was repeated 20 times, each time with different random data splits. In terms of experiment complexity 20 repeats of 5-fold CV is an equivalent of 100 repeats of hold-out sampling.

At the level of an individual project, the predictive accuracy

was evaluated using the absolute error, $AE_i = |actual_i - predicted_i|$ as recommended in [38]. To determine the optimum hyperparameters over *n* projects, the study used the root mean squared error, $RMSE = \sqrt{1/n \sum_{i=1}^{n} AE_i{}^2}$.

Three aggregating measures were used to analyse results from the external CV. The first was the standardised accuracy, $SA_{P_k} = (1 - MAE_{P_k}/\overline{MAE_{P_0}}) \times 100$, where $MAE = 1/n \sum_{i=1}^{n} AE_i$. The MAE_{P_k} denotes the MAE for a predictive model *k*, and $\overline{MAE_{P_0}}$ the mean MAE from a high number of runs of random guessing for the outcome attribute among its values in the training dataset [38], [39]. SA tells how much better the predictions were compared to the random guessing strategy, for which $SA = 0$. The maximum value of SA is 100. There is no lower bound for SA, but values below zero indicate predictions worse than from random guessing. Such rare low values were reported in earlier studies [40].

The second was d_{WL}, calculated using a "win-tie-loss" approach that involved comparing the distributions of AE's of a given data variant pairwise with those from other data variants, separately for each model and CV iteration. [41]–[44]. This comparison of distributions of AE's involved the Wilcoxon signed-rank test [45] with a Benjamini-Hochberg adjustment of the *p*-values [46]. If this test did not detect statistically significant differences between a pair of distributions, then the ties count for both data variants increased by one. Otherwise, the count of wins increased by one for a data variant with a lower median AE and the count of losses for a data variant with a higher median AE. Then, it was possible to calculate $d_{WL} = wins - losses$.

The third aggregating measure was r_c, the matched-pairs rank biserial correlation coefficient [47, p. 384]. It was calculated similarly to d_{WL}, but for each pair of distributions of AEs, the r_c was calculated instead of performing the Wilcoxon test. The r_c measures the effect size between a matched pair of AEs. This study used the implementation of r_c provided in the [48] R package. It added a sign to r_c so that a positive value indicated higher values in the first distributions and a negative otherwise. A value of $r_c = 0$ indicates no effect, and $|r_c| = 1$ is a very strong effect.

To answer RQ1, the rankings of data variants were created based on each evaluation measure, mean SA, d_{WL}, and mean r_c, and separately for each model. These rankings involved the Scott-Knott Effect Size Difference (SK-ESD) test (v2.0) [49], [50] to create non-overlapping partitions, i.e., ranks, of data variants. The specific partition groups data variants with similar distributions of the particular evaluation measure across external CV data splits. The SK-ESD internally compared groups of data variants using Cohen's d.

RQ2 and RQ3, related to the robustness and stability of predictions, involved Kendall W statistic [51], [52]. It reflects the extent of agreement among multiple lists of ranks of treatments given by different raters. For RQ2, the ranks were based on the values of SA, d_{WL}, and r_c for particular data variants, separately for each model. Because a specific external CV data split might deliver different ranks, the RQ2 investigated the similarity of data variants' ranks across data splits. Hence, these data splits acted as raters for calculating Kendall W. Because the SK-ESD ranks might differ for each model, the RQ3 investigated the similarity of the data variant's SK-ESD ranks across models. Hence, the models acted as raters. With the ideal agreement between raters, Kendall W equals one, with the perfect disagreement: zero. This study followed the interpretation of the intermediate values proposed in [53], i.e., weak if $W \leq 0.3$, moderate if $0.3 < W \leq 0.6$, and strong if $W > 0.6$. *P*-values were adjusted according to the Benjamini-Hochberg method [46].

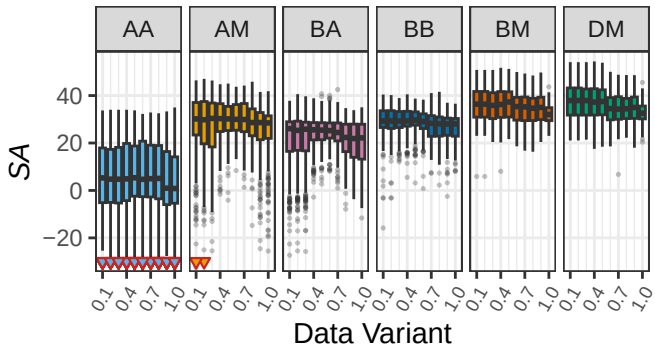

Fig. 2: Overall accuracy of particular data variants across all models. Red-bordered triangles indicate data variants for which lower out-of-scale values of SA were observed.

IV. Results

A. Preliminary observations

Figure 2 illustrates the accuracy of particular data variants aggregated across all models and data splits. On average, SA values were the highest for BM* and DM* data variants and the lowest for AA*. For all AA*, AM0.1, and AM0.2 in some data splits for LM, M5 and NN, the SA values were even lower than bottom limits in the figure, indicating extremely poor predictions caused by the low case count in a training subset. Only for MED, SVM and XT, no negative values of SA were found in any data split. The SA ranges were the shortest for DM*, BM*, and BB*, indicating that the other, more restrictive, data variants performed poorly with some models in some data splits. Within a group of data variants with a common rating, e.g. DM*, the TVV settings usually did not significantly change the accuracy of predictions. However, the values of SA were usually lower for higher values of TVV.

B. (RQ1.) How does the choice of data variant influence the performance of each model?

This RQ was investigated by performing the SK-ESD test separately on each evaluation measure and for each model. Figure 3 illustrates this test's rankings of data variants (the lower, the better). The number of ranks (partitions) varied across models and evaluation measures. For example, for MED on d_{WL}, all data variants were grouped in only three partitions with non-negligible differences. On the other hand, for LM on

SA, the SK-ESD test defined 29 partitions of data variants. Hence, the colour scale in the figure was scaled according to the number of ranks for a particular model.

Among the baseline models, for MEA the best-performing data variants were AM* on SA and BB* on d_{WL} and r_c, and for MED: DM* on SA and r_c, and BM* and DM* on d_{WL} (all data variants in these groups at the same SK-ESD rank). With two exceptions, at least one data variant from the small group of DM0.1–DM0.5 reached the top rank for each non-baseline model on each evaluation measure. These exceptions include NN on SA and r_c, where DM1.0 performed the best. For several models, numerous data variants were tied at the top rank. The most such ties (16) were for LM on d_{WL}. Apart from the group of DM0.1–DM0.5 data variants, the other data variants that appeared tied at the top rank for some models at least on one evaluation measure include DM0.6, DM0.8–DM1.0, BM0.1–BM0.5, BM08–BM1.0, BB0.4, and BB0.5. No AA*, AM*, or BA* data variant reached the top rank for any non-baseline model, even on a single evaluation measure.

With very few exceptions, at least one data variant among AA0.1–AA1.0 reached the lowest rank for each model on each evaluation measure. These exceptions include XT, for which the worst performing data variants were AM0.8 and AM1.0 on SA and AM1.0 on r_c.

Table III provides measures for two groups of data variants for each model: the best overall and the best among a set {AA*, BA*, BB*}. The second group contains data variants based on high data ratings (A or B) following the ISBSG guidelines and used in earlier literature. The best data variants in each group were selected according to the best mean SK-ESD rank across three evaluation measures; in case of ties, the one with higher $\overline{d_{WL}}$, if still tied, the one with higher $\overline{r_c}$. For all models except MEA, the best overall data variants were DM*, usually with low TVV, and the best among {AA*, BA*, BB*} reached the mean SK-ESD ranks only between 2.7 and 13.0. Only for MEA, the best overall data variant was based on a more restrictive rating (BB). The most noticeable differences between the two groups of data variants were for XT, where the best data variants (DM0.1 and DM0.2) had a $\overline{SA}$ higher by 14.3, $\overline{d_{WL}}$ by 40.5, and $\overline{r_c}$ by 0.341 compared to BB0.6. Very high differences were also observed for SVM: 14.2, 20.4, and 0.328, respectively. Hence, the DM* data variants dominated data variants {AA*, BA*, BB*} regarding predictive accuracy.

TABLE III: Measures for the overall best data variants and the best among {AA*, BA*, BB*}, for each model.

Model	Best overall				Best among {AA*, BA*, BB*}			
	Name[1]	$\overline{SA}$	$\overline{d_{WL}}$	$\overline{r_c}$	Name[1]	$\overline{SA}$	$\overline{d_{WL}}$	$\overline{r_c}$
MEA	BB* (1.3)	28.9	12.5	0.199	BB* (1.3)	28.9	12.5	0.199
MED	DM* (1.0)	32.1	0.5	0.150	BB* (2.7)	28.6	0.0	–0.079
EN	DM0.3 (1.0)	42.8	21.0	0.278	BB0.4 (5.3)	35.0	17.4	0.161
GBM	DM0.1 (1.0)	42.1	25.0	0.270	BB0.5 (9.3)	31.3	7.3	0.110
KNN	DM0.5 (1.0)	40.1	19.1	0.241	BB0.2 (4.0)	34.4	12.9	0.186
LM	DM0.1 (1.0)	32.6	17.8	0.311	BB0.4 (5.7)	26.9	18.0	0.233
M5	DM0.4 (1.0)	34.7	14.8	0.262	BB0.3 (5.0)	26.2	12.7	0.128
NN	DM1.0 (1.0)	34.7	14.2	0.246	BA0.5 (7.3)	24.8	8.4	0.106
RF	DM0.3 (1.0)	42.5	28.5	0.282	BB0.1 (9.3)	33.4	11.9	0.110
RT	DM0.5 (1.0)	32.2	14.8	0.209	BB0.1 (4.3)	27.2	9.7	0.115
SVM	DM0.4 (1.0)	49.6	23.2	0.300	BB0.3 (13.0)	35.4	2.8	–0.028
XT	DM0.1 (1.0)	45.2	34.8	0.306	BB0.6 (9.3)	30.9	–5.7	–0.035

[1] For MEA and MED, the TVV setting did not influence predictions inducing ties among data variants. Among the best overall data variants for GBM, LM, and XT, all three evaluation measures for DM0.1 and DM0.2 were equal. Due to space limits, only DM0.1 was reported. Similarly, among {AA*, BA*, BB*} for MED – BA* were tied with BB*. The values in brackets indicate the data variant's mean SK-ESD rank across three evaluation measures.

C. (RQ2.) What is the robustness of data variants to different data splits for each model?

This RQ was investigated using the Kendall W statistic that reflected the agreement of the order of ranks for data variants in particular data splits (Table IV). The highest agreement on SA was 0.97 for MED, and for five other models, it was higher than 0.9, i.e., for GBM, EN, KNN, LM, and RF. The lowest were 0.64 for RT, 0.66 for MEA and 0.67 for XT. On the d_{WL}, the highest agreement was 0.90 for LM, and the lowest was 0.05 for MED. The latter was the only statistically insignificant with the p-value of 0.476. For all other models, W was higher than 0.6. On the r_c, the highest agreement was 0.92 for LM. The lowest were 0.58 for RT and 0.59 for MEA. For all other models, the agreement was higher than 0.6. Based on the evaluation scheme provided in Section III-D, the robustness of data variants to different data splits on SA was strong for all models, on d_{WL} was strong for all models, except MED for which was weak and statistically insignificant, and on r_c was strong for all models, except RT and MEA, for which it was moderate but very close to the strong level.

Hence, most of these results were satisfying as they demonstrated a high level of agreement between ranks of data variants for particular models across data splits. Also, for most models, there were no substantial differences in Kendall W on different evaluation measures.

The results for MED need special attention because its values of Kendall W were highly varying depending on the evaluation measure, i.e. W was the highest on SA but the lowest on d_{WL} among all models. For this model, the TVV setting did not influence performance because this model did not use predictor values when providing predictions. On SA, in 11 of 20 data splits, the DM* performed consistently the best and BM* at rank 2. BM* and DM* were tied in the remaining nine data splits. AA*, AM*, BA*, and BB* performed equally worst on 18 data splits. Hence, the ranks of specific data variants were highly consistent across data splits. In 19 data splits, the d_{WL} was zero for each data variant, indicating no statistically significant differences according to the Wilcoxon

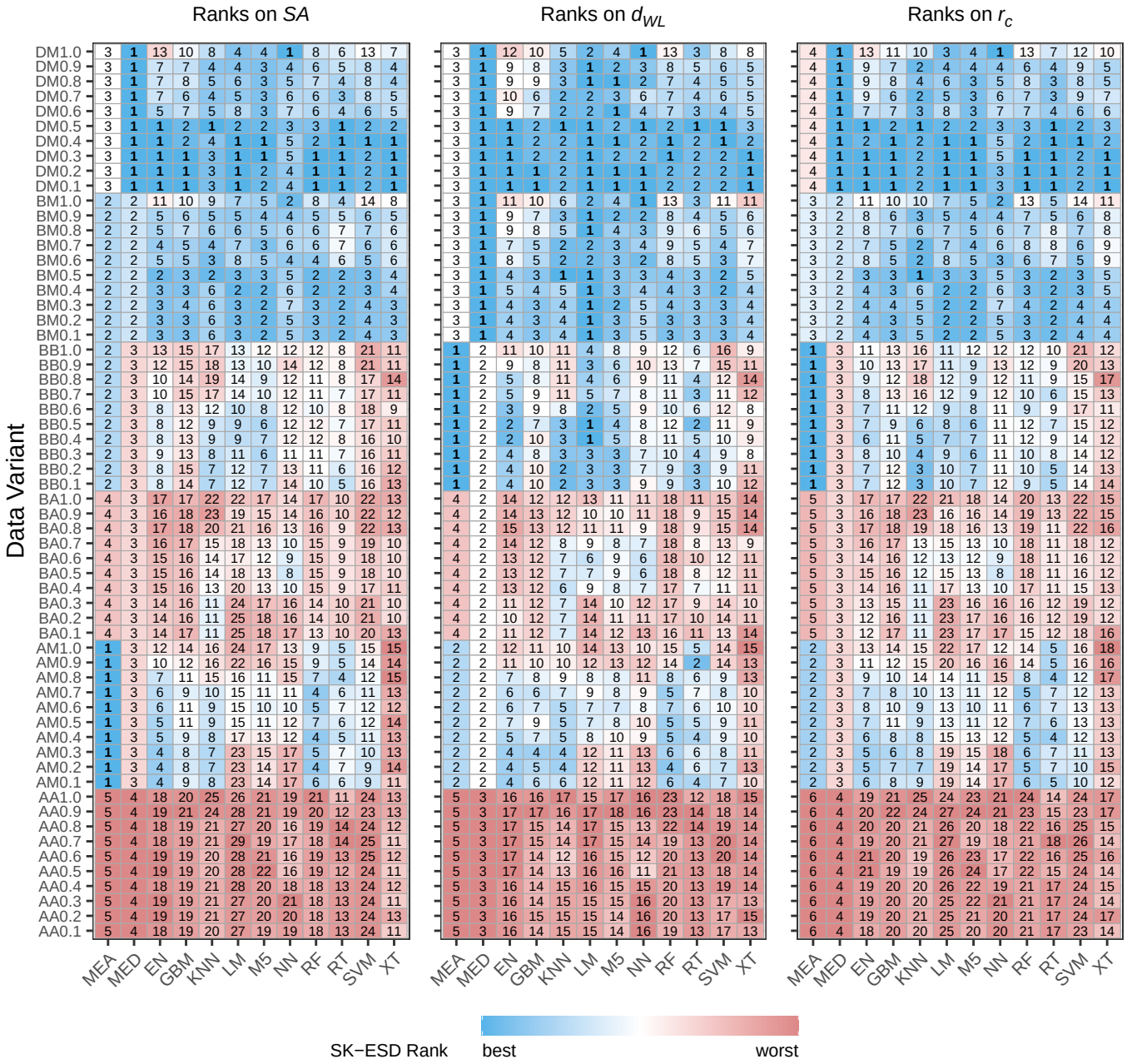

Fig. 3: The rankings of data variants for each model according to the SK-ESD test.

test between each data variant's pair. The BM* and DM* outperformed AA* in only one data split. These circumstances caused the Kendall W to be so low on d_{WL}. Despite MEA being conceptually very similar to MED, such discrepancies in Kendall W were not observed for MEA, for which there was some variability in performance, both on SA and d_{WL} between particular data variants in specific data splits.

D. (RQ3.) Is the performance of data variants stable across models?

This RQ investigated the consistency of data variants' performance across predictive models. Specifically, it explored the agreement between data variants' SK-ESD ranks (from RQ1) across models using the Kendall W coefficient. The SK-ESD ranks for MEA and MED were excluded from this analysis because the TVV did not influence predictions for these models. Therefore, the rankings for these models included significantly fewer groups (partitions) but with many ties that would reduce the values of Kendall W. Besides, investigating if the data variants perform similarly with the baselines and "real" models is not important because these baselines are not supposed to be used in practice.

Obtained values of Kendall W were the following: 0.84 on SA, 0.82 on d_{WL}, and 0.85 on r_c with all adjusted p-

TABLE IV: The values of Kendall W for ranks of data variants across data splits grouped for each model.

Model	SA		d_{WL}		r_c	
	W	adj. p-value	W	adj. p-value	W	adj. p-value
MEA	0.66	< 0.001	0.80	< 0.001	0.59	< 0.001
MED	0.97	< 0.001	0.05	0.476	0.69	< 0.001
EN	0.91	< 0.001	0.74	< 0.001	0.85	< 0.001
GBM	0.90	< 0.001	0.78	< 0.001	0.84	< 0.001
KNN	0.90	< 0.001	0.77	< 0.001	0.86	< 0.001
LM	0.94	< 0.001	0.90	< 0.001	0.92	< 0.001
M5	0.88	< 0.001	0.84	< 0.001	0.85	< 0.001
NN	0.72	< 0.001	0.67	< 0.001	0.70	< 0.001
RF	0.93	< 0.001	0.84	< 0.001	0.87	< 0.001
RT	0.64	< 0.001	0.62	< 0.001	0.58	< 0.001
SVM	0.89	< 0.001	0.78	< 0.001	0.83	< 0.001
XT	0.67	< 0.001	0.65	< 0.001	0.66	< 0.001

values below 0.001. These values are close to each other and very high, indicating strong stability of data variants' SK-ESD ranks across predictive models. Data variants that performed well with some models usually performed well with others. On the contrary, data variants that performed poorly with some models usually performed poorly with others.

V. Threats to Validity

This experiment used a single dataset because there was no other publicly available dataset on user satisfaction in software projects and providing case quality classification. A subset of the ISBSG dataset was selected because only 134 of 9,592 cases contained non-missing values for attributes related to user satisfaction. Since the dataset is relatively small and not a random subset of a population, the results may not generalise outside the experimentation boundaries set up for this study.

The target attribute, *Satisfaction Score*, was calculated as the sum of eight attributes explicitly present in the dataset. With such aggregation, each attribute was treated with an equal weight. In specific projects, particular attributes might have different weights. However, such data was not available.

The data preprocessing involved steps that required making decisions. They were documented in the provided script to partially mitigate these decisions' subjectivity. Such decisions also involved methods and the research process, e.g., the selection of models and their hyperparameters, evaluation measures, statistical tests, the evaluation method, and the parameters for cross-validation. They were made after careful investigation of the alternatives, the ability to reach high accuracy of predictions and stable results, and based on justifications provided in similar studies.

VI. Conclusions and Future Work

Analysts usually face a trade-off between data volume and quality. Existing studies involving the ISBSG dataset usually selected projects rated A or B and attributes with a low fraction of missing values. When predicting user satisfaction, the results from the present study clearly showed that achieving the highest prediction accuracy for user satisfaction using AA* data variants is not reasonable because, with such restrictive data selection, the training dataset was so small that all predictive models struggled to learn the necessary patterns from data and delivered poor predictions. Predictions with BA* data variants were better and even better with AM* and BB*, so some were in the top 5 ranks. However, predictions with BM* were, on average, even better, and with DM* were usually the best. The TVV setting usually had a much lower impact on predictive performance than the data rating. Among the best groups of data variants, i.e., DM*, BM*, and BB*, the best performance for most models was usually achieved for low settings of TVV, i.e., 0.1–0.5.

Provided rankings of data variants for particular models may guide future studies on user satisfaction prediction. Specifically, data variants DM0.1–DM0.5 performed the best for most models. Hence, selecting such data variants is recommended in studies focusing on maximising the predictive accuracy of user satisfaction. Predictive models performed better when trained on a higher volume of data containing some low-quality cases than on a low volume of only high-quality data.

We might consider yet another dimension in the analysed trade-off related to the objective or perceived data reliability or trustfulness, especially in an industrial environment. Despite the results being based on solid methodological grounds and generally stable across data splits and models, some analysts may be concerned about training models on data with low-quality cases classified by ISBSG reviewers. Then, using BB* or AM* data variants with TVV in the range of 0.1–0.7 is recommended, depending on the model.

This study can be extended by identifying the best models for particular data variants. It may be helpful if a data scientist cannot freely select the data variant, e.g., if such selection is enforced for any reason. The extension may also involve identifying the best-performing model and data variant combinations. Although the present study considered several state-of-the-art predictive models, it may be worth analysing other models, especially heterogeneous ensembles. Other possible enhancements involve exploring the impact of feature selection strategies, other hyperparameter optimisation methods, and handling explicit interactions of predictor attributes.

References

[1] ISBSG, *ISBSG Repository Data Release 2020 R1*. International Software Benchmarking Standards Group, 2020.

[2] F. González-Ladrón-de Guevara, M. Fernández-Diego, and C. Lokan, "The usage of ISBSG data fields in software effort estimation: A systematic mapping study," *Journal of Systems and Software*, vol. 113, pp. 188–215, mar 2016.

[3] ISBSG, *Guidelines for use of the ISBSG data*. International Software Benchmarking Standards Group, 2020.

[4] C. Gencel, R. Heldal, and K. Lind, "On the Relationship between Different Size Measures in the Software Life Cycle," in *2009 16th Asia-Pacific Software Engineering Conference*, pp. 19–26, IEEE, dec 2009.

[5] H. Huijgens, A. van Deursen, L. L. Minku, and C. Lokan, "Effort and Cost in Software Engineering: A Comparison of Two Industrial Data Sets," in *Proceedings of the 21st International Conference on Evaluation and Assessment in Software Engineering*, (New York, NY, USA), pp. 51–60, ACM, jun 2017.

[6] C. López-Martín, "Machine learning techniques for software testing effort prediction," *Software Quality Journal*, vol. 30, pp. 65–100, mar 2022.

[7] N. Mittas and L. Angelis, "Ranking and Clustering Software Cost Estimation Models through a Multiple Comparisons Algorithm," *IEEE Transactions on Software Engineering*, vol. 39, pp. 537–551, apr 2013.

[8] J. Murillo-Morera, C. Quesada-López, C. Castro-Herrera, and M. Jenkins, "A genetic algorithm based framework for software effort prediction," *Journal of Software Engineering Research and Development*, vol. 5, p. 4, dec 2017.

[9] K. Ono, M. Tsunoda, A. Monden, and K. Matsumoto, "Influence of Outliers on Estimation Accuracy of Software Development Effort," *IEICE Transactions on Information and Systems*, vol. E104.D, pp. 91–105, jan 2021.

[10] Y.-S. Seo and D.-H. Bae, "On the value of outlier elimination on software effort estimation research," *Empirical Software Engineering*, vol. 18, pp. 659–698, aug 2013.

[11] M. F. Bosu and S. G. Macdonell, "Experience: Quality Benchmarking of Datasets Used in Software Effort Estimation," *Journal of Data and Information Quality*, vol. 11, pp. 1–38, dec 2019.

[12] G. A. Liebchen and M. Shepperd, "Data sets and data quality in software engineering," in *Proceedings of the 4th international workshop on Predictor models in software engineering*, (New York, NY, USA), pp. 39–44, ACM, may 2008.

[13] K. Deng and S. G. MacDonell, "Maximizing data retention from the ISBSG repository," in *12th International Conference on Evaluation and Assessment in Software Engineering*, (Bari, Italy), pp. 21–30, British Computer Society, jun 2008.

[14] R. Jeffery, M. Ruhe, and I. Wieczorek, "Using public domain metrics to estimate software development effort," in *Proceedings Seventh International Software Metrics Symposium*, (Washington, DC), pp. 16–27, IEEE, 2001.

[15] E. Mendes and C. Lokan, "Replicating studies on cross- vs single-company effort models using the ISBSG Database," *Empirical Software Engineering*, vol. 13, pp. 3–37, feb 2008.

[16] K. Dejaeger, W. Verbeke, D. Martens, and B. Baesens, "Data Mining Techniques for Software Effort Estimation: A Comparative Study," *IEEE Transactions on Software Engineering*, vol. 38, pp. 375–397, mar 2012.

[17] M. Bano, D. Zowghi, and F. da Rimini, "User satisfaction and system success: an empirical exploration of user involvement in software development," *Empirical Software Engineering*, vol. 22, pp. 2339–2372, oct 2017.

[18] G. P. Z. Montesdioca and A. C. G. Maçada, "Measuring user satisfaction with information security practices," *Computers & Security*, vol. 48, pp. 267–280, feb 2015.

[19] M. Tarafdar, Q. Tu, and T. S. Ragu-Nathan, "Impact of Technostress on End-User Satisfaction and Performance," *Journal of Management Information Systems*, vol. 27, pp. 303–334, dec 2010.

[20] F. Sarro, M. Harman, Y. Jia, and Y. Zhang, "Customer Rating Reactions Can Be Predicted Purely using App Features," in *2018 IEEE 26th International Requirements Engineering Conference (RE)*, pp. 76–87, IEEE, aug 2018.

[21] Ł. Radliński, "Predicting User Satisfaction in Software Projects using Machine Learning Techniques," in *Proceedings of the 15th International Conference on Evaluation of Novel Approaches to Software Engineering - Volume 1: ENASE* (R. Ali, H. Kaindl, and L. Maciaszek, eds.), pp. 374–381, SciTePress, 2020.

[22] L. Radlinski, "Stability of user satisfaction prediction in software projects," *Procedia Computer Science*, vol. 176, no. Proceedings of the 24th International Conference on Knowledge-Based and Intelligent Information & Engineering Systems, pp. 2394–2403, 2020.

[23] Ł. Radliński, "The Impact of Data Quality on Software Testing Effort Prediction," *Electronics*, vol. 12, p. 1656, mar 2023.

[24] P. A. Whigham, C. A. Owen, and S. G. Macdonell, "A Baseline Model for Software Effort Estimation," *ACM Transactions on Software Engineering and Methodology*, vol. 24, pp. 1–11, may 2015.

[25] R. M. West, "Best practice in statistics: The use of log transformation," *Annals of Clinical Biochemistry: International Journal of Laboratory Medicine*, vol. 59, pp. 162–165, may 2022.

[26] A. B. Nassif, D. Ho, and L. F. Capretz, "Towards an early software estimation using log-linear regression and a multilayer perceptron model," *Journal of Systems and Software*, vol. 86, pp. 144–160, jan 2013.

[27] J. Friedman, T. Hastie, and R. Tibshirani, "Regularization Paths for Generalized Linear Models via Coordinate Descent," *Journal of Statistical Software*, vol. 33, no. 1, 2010.

[28] J. H. Friedman, "Greedy function approximation: A gradient boosting machine.," *The Annals of Statistics*, vol. 29, pp. 1189–1232, oct 2001.

[29] N. S. Altman, "An Introduction to Kernel and Nearest-Neighbor Nonparametric Regression," *The American Statistician*, vol. 46, pp. 175–185, aug 1992.

[30] G. N. Wilkinson and C. E. Rogers, "Symbolic Description of Factorial Models for Analysis of Variance," *Applied Statistics*, vol. 22, no. 3, p. 392, 1973.

[31] Y. Wang and I. H. Witten, "Induction of model trees for predicting continuous classes," in *Proceedings of the Poster Papers of the European Conference on Machine Learning*, (Prague), University of Economics, Faculty of Informatics and Statistics, 1997.

[32] I. H. Witten, E. Frank, M. A. Hall, and C. J. Pal, *Data Mining: Practical Machine Learning Tools and Techniques*. Cambridge, MA: Morgan Kaufmann, fourth ed., 2016.

[33] B. D. Ripley, *Pattern Recognition and Neural Networks*. Cambridge University Press, jan 1996.

[34] L. Breiman, "Random Forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[35] L. Breiman, J. Friedman, C. J. Stone, and R. Olshen, *Classification and Regression Trees*. Boca Raton: Chapman & Hall, 1984.

[36] C.-C. Chang and C.-J. Lin, "LIBSVM," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, pp. 1–27, apr 2011.

[37] T. Chen and C. Guestrin, "XGBoost: A Scalable Tree Boosting System," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '16*, (New York), pp. 785–794, ACM Press, 2016.

[38] M. Shepperd and S. MacDonell, "Evaluating prediction systems in software project estimation," *Information and Software Technology*, vol. 54, pp. 820–827, aug 2012.

[39] W. B. Langdon, J. Dolado, F. Sarro, and M. Harman, "Exact Mean Absolute Error of Baseline Predictor, MARP0," *Information and Software Technology*, vol. 73, pp. 16–18, may 2016.

[40] A. Lustosa and T. Menzies, "Learning from Very Little Data: On the Value of Landscape Analysis for Predicting Software Project Health," *ACM Transactions on Software Engineering and Methodology*, vol. 33, pp. 1–22, mar 2024.

[41] S. Feng, J. Keung, X. Yu, Y. Xiao, and M. Zhang, "Investigation on the stability of SMOTE-based oversampling techniques in software defect prediction," *Information and Software Technology*, vol. 139, p. 106662, nov 2021.

[42] E. Kocaguneli, T. Menzies, A. Bener, and J. W. Keung, "Exploiting the Essential Assumptions of Analogy-Based Effort Estimation," *IEEE Transactions on Software Engineering*, vol. 38, pp. 425–438, mar 2012.

[43] T. Menzies, O. Jalali, J. Hihn, D. Baker, and K. Lum, "Stable rankings for different effort models," *Automated Software Engineering*, vol. 17, pp. 409–437, dec 2010.

[44] F. Sarro and A. Petrozziello, "Linear Programming as a Baseline for Software Effort Estimation," *ACM Transactions on Software Engineering and Methodology*, vol. 27, pp. 1–28, jul 2018.

[45] F. Wilcoxon, "Individual Comparisons by Ranking Methods," *Biometrics Bulletin*, vol. 1, p. 80, dec 1945.

[46] Y. Benjamini and Y. Hochberg, "Controlling the False Discovery Rate: A Practical and Powerful Approach to Multiple Testing," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 57, no. 1, pp. 289–300, 1995.

[47] B. M. King, P. J. Rosopa, and E. W. Minium, *Statistical Reasoning in the Behavioral Sciences*. Hoboken, NJ: John Wiley & Sons, Inc., seventh ed ed., 2018.

[48] S. S. Mangiafico, *rcompanion: Functions to Support Extension Education Program Evaluation*. Rutgers Cooperative Extension, New Brunswick, New Jersey, 2023. version 2.4.34.

[49] C. Tantithamthavorn, S. McIntosh, A. E. Hassan, and K. Matsumoto, "An Empirical Comparison of Model Validation Techniques for Defect Prediction Models," *IEEE Transactions on Software Engineering*, vol. 43, pp. 1–18, jan 2017.

[50] C. Tantithamthavorn, "ScottKnottESD: The Scott-Knott effect size difference (ESD) test," 2023.

[51] M. G. Kendall and B. B. Smith, "The Problem of *m* Rankings," *The Annals of Mathematical Statistics*, vol. 10, pp. 275–287, sep 1939.

[52] M. G. Kendall and J. D. Gibbons, *Rank correlation methods*. London, New York: Oxford University Press, 5th ed., 1990.

[53] G. K. Rajbahadur, S. Wang, G. A. Oliva, Y. Kamei, and A. E. Hassan, "The Impact of Feature Importance Methods on the Interpretation of Defect Classifiers," *IEEE Transactions on Software Engineering*, vol. 48, pp. 2245–2261, jul 2022.