Towards Minimizing Statistical Distortion Introduced by De-Identification Methods

Will Fried, Dimitris Vamvourellis

Abstract

K-anonymity is a popular approach to protecting the privacy of individuals whose personal data is included in a dataset. One method of k-anonymization involves adding synthetic rows that contain the sets of quasi-identifiers (QIs) that are underrepresented in the original dataset, and filling in the non-QI fields for each of these rows. Unfortunately, existing methods for filling in these non-QI fields significantly skew the statistical properties of the original dataset by failing to consider the relationships between the QI and non-QI fields. The objective of our project was to minimize these distortions by using machine learning methods including regression and tree-based models to fill in these fields in such a way that preserves the properties of the original dataset.

1 Introduction

De-identification refers to all algorithms which are used to minimize the probability of an individual being re-identified when a combination of his/her personal information are made public as part of a larger dataset. One of the privacy standards used to quantify this probability is k-anonymity [2], which requires that rare combinations of personally-identifiable information (to which we refer as quasi-identifiers (QIs)) are shared by at least k individuals in a public dataset. In this way, the probability of an individual being re-identified from the given data is reduced to at most $1/k$. To satisfy this privacy standard, every possible combination of quasi-identifiers must be present in at least k different rows of the final public dataset. Any rows which

violate this standard must be altered, thus distorting the original statistical structure of the original dataset. Broadly, there three main ways to achieve k-anonymity before releasing a dataset:

1. Suppression: deleting every row which violates k-anonymity.
2. Generalization followed by suppression: first identify QIs which can take on multiple values and thus contribute to more unique combinations of QIs. Then, generalize such QIs by allowing them to take one value out of a lower number of more broad categories (e.g. generalize city to country). In this way, the number of unique rows is reduced leading to lower suppression rate.
3. Synthetic data generation: for every combination of QIs which exists in the data n times, where $n < k$, generate $k - n$ rows with the exact same combination of QIs.

Each of the methods described above comes with several trade-offs that need to be taken into account. Suppression deletes entire rows from the data which may lead to significant distortion of key statistical properties of multiple columns of the data, such as means and standard deviations. Generalization may reduce the number of rows that need to be suppressed, but at the same time it might significantly distort the correlation between certain columns, thus limiting the ability to do any useful statistical analysis with the final dataset. Finally, the generation of synthetic rows requires us to fill in the values of non-QI columns. This is not a trivial task. Filling in the same values as the ones in original row that violated k-anonymity would result in zero ℓ-diversity, thus increasing the probability of the given individual to be re-identified and defeating the whole purpose of k-anonymity. In this project, we aim to develop methods for synthetically generating new rows to achieve k-anonymity, while preserving the statistical and distributional properties of non-QI fields found in the original dataset. To achieve this, we explore different machine learning methods like regression models and tree-based models in order to predict the values of the non-QI fields using the values of the QIs as features.

2 Experimental Design

2.1 Dataset

To conduct our experiments we used the Massive, Online Open Courses (MOOCs) dataset offered by Harvard University and the Massachusetts Institute of Technology for the academic year 2014-2015.

2.2 Data Preprocessing

For the purpose of this project, we considered six QIs: course ID, level of education (LoE), year of birth (YoB), gender, country and number of forum posts (nforum_posts). Using these six QIs as features, we filled in the value of the grade column. To train our models, we used all the rows in the original dataset for which the grade was not null. An inspection of the nforum_posts column revealed that minimum numerical value is 1. The fact that no records have a value of zero coupled with the fact that a large proportion of records have missing values for these fields means that in all likelihood the missing values correspond to a value of 0. In other words, the default value for these fields is null until a student publishes his or her first post on the forum. Therefore, we replaced any null values with zeros for this column. In addition, we discovered that while there are 254 countries represented in the dataset, the United States is the home of 32% of the students in the dataset. Thus, to reduce the number of one-hot values corresponding to that column, we converted country into a binary column which indicates whether the country is the U.S., corresponding to 1, or not the U.S., corresponding to 0. For all the categorical quasi-identifiers (i.e. gender, LoE, country), we considered Null as another valid category, given that a large percentage of the rows in the original dataset is populated with at least one null value for any of the quasi-identifiers. Finally, we filled in any missing values for year of birth with the mean value and added an indicator variable to the design matrix that denotes whether the corresponding YoB entry was missing in the original dataset.

2.3 Minimizing the number of synthetic records needed

To conduct our analysis, we generated the minimum number of synthetic rows needed to satisfy 5-anonymity, as was performed in [1]. In our first approach, we treated each of the null values just as we would any non-null value by converting each null value to the string 'None'. With this method, we needed to add 6,078,953 rows to the original dataset that contained 6,860,993. As such, the size of the synthetic dataset was roughly twice the size as the original dataset.

Next, we took a more sophisticated approach that accounts for the fact that a null value is not actually equivalent to a non-null value. It's easiest to explain the method with an illustrative example: suppose we're dealing with a quasi-identifier dataset with the following two records: course_id='AC221', Gender='M', YoB=NaN, course_id='AC221', Gender='M', YoB=1997. If we were to simply treat the NaN entry in the first record as a 'null' string value, then the dataset would be 1-anonymous with respect to both records and

in order to achieve 5-anonymity, we'd have to add 4 synthetic records that are identical to the first record (i.e. course_id='AC221', Gender='M', YoB=NaN) and two additional synthetic records that are identical to the second record (i.e. course_id='AC221', Gender='M', YoB=1997). However, if we recognize that the first record is a subset of the second record, then we'd realize that this dataset is actually 2-anonymous with respect to the first record because there are two records in the dataset that have a course_id of 'AC221' and a gender of male. Meanwhile, this dataset is 1-anonymous with respect to the second record because there is only one record in the dataset that has a course_id of 'AC221', a gender of male and a year of birth of 1997. This means that all we need to do to achieve 5-anonymity is add four synthetic records that are identical to the second record (i.e. course_id='AC221', Gender='M', YoB=1997), as this would mean that the dataset is now 5-anonymous with respect to the second record and 6-anonymous with respect to the first record.

Carrying out this approach is a two step process. First, we need to identify which records of the dataset are subsets of other records of the dataset. We can then ignore these records (because achieving k-anonymity with respect to the records that are supersets of these records automatically achieves at least k+1 anonymity with respect to these subset records). Second, for each of the records in the dataset that are not subsets of other rows in the dataset, we need to add the minimum number of synthetic records to the dataset that are needed to achieve k-anonymity with respect to the given record.

With this more advanced method, we had to add 5,319,253 synthetic records to the dataset, which represents a 12.5% decrease in the number of synthetic rows relative to the first method. While this reduction in the number of synthetic rows needed to add to the original is certainly significant, it's important to note that the benefits would be much larger if we all were analyzing the entire dataset with all of the quasi-identifiers included. This is the case because the number of records that are subsets of other records in the dataset is proportional to the number of quasi-identifiers in the dataset and the fraction of entries that are null. For this project, we only considered six quasi-identifiers, two of which (course_id and nforum_posts) have no missing values. This scarcity of null values limits number of records that are subsets of other records and thus don't need to be duplicated.

The Jupyter notebook used to create these synthetic datasets can be found here.

2.4 Measures of evaluation

Ultimately, we want to quantify the amount of statistical bias introduced by adding the synthetic rows with the values being filled in using the predictions of a ML model. To do this, we evaluate the difference between the mean grade in the original dataset and the mean grade in the dataset with the added synthetic rows, a measure to which we refer as *mean bias*. This is a measure of the distortion of the statistical properties of a particular column. In addition, we want to measure how the added synthetic rows impact the relationship between different columns, which is a key element for any downstream statistical analysis conducted based on a given dataset. To do this, we calculate the difference between the correlation of grade and the number of forum posts in the original dataset and on the dataset augmented with the synthetic rows, a measure to which we refer as *correlation bias*.

3 Baseline Methods

Our goal is to achieve k-anonymity by synthetic data generation while preserving the distributional properties of any non-QI fields, such as the grade in this case. To evaluate our methods, we compared the performance of each of the models that we developed against the performance of the baseline methods proposed in [1]. Particularly, in [1], for each synthetic row generated, the value of the grade is drawn from one of the following distributions:

1. Marginal: drawing a random grade value with replacement from the set of grade values in the original dataset.
2. Marginal mean: drawing 5 random samples from the marginal grade distribution and then filling in the average of the samples.
3. Joint: From the set of records with identical QI values, draw a random grade value with replacement.
4. Joint mean: From the set of records with identical QI values, calculate the mean grade and add random noise to mask the synthetic rows.

Drawing values from the marginal distribution is expected to have very low mean bias, since the mean of the random samples approaches the true mean of the column by the law of large numbers. However, this

approach does not take the rest of the QIs or other columns into account. Hence, by definition this approach will introduce bias by distorting the relationships between columns and thus limiting our ability to perform meaningful statistical analysis for downstream tasks. This effect is amplified if we consider that for large datasets the number of synthetic rows can be enormous. For example, in the case of MOOCs dataset, we had to generate more than 5 million synthetic rows in order to satisfy k-anonymity, effectively doubling the number of rows compared to the original dataset. Consequently, filling in grades with random values for half of the dataset would substantially weaken any relationships that might exist between grade and any of the QIs in the original dataset. This is also reflected in Figure 1 (taken from the original paper [1]) where we can see that the most effective baseline method is drawing from the marginal and taking the average of samples. This approach achieved very low mean bias but it resulted in significant correlation bias of about 0.13.

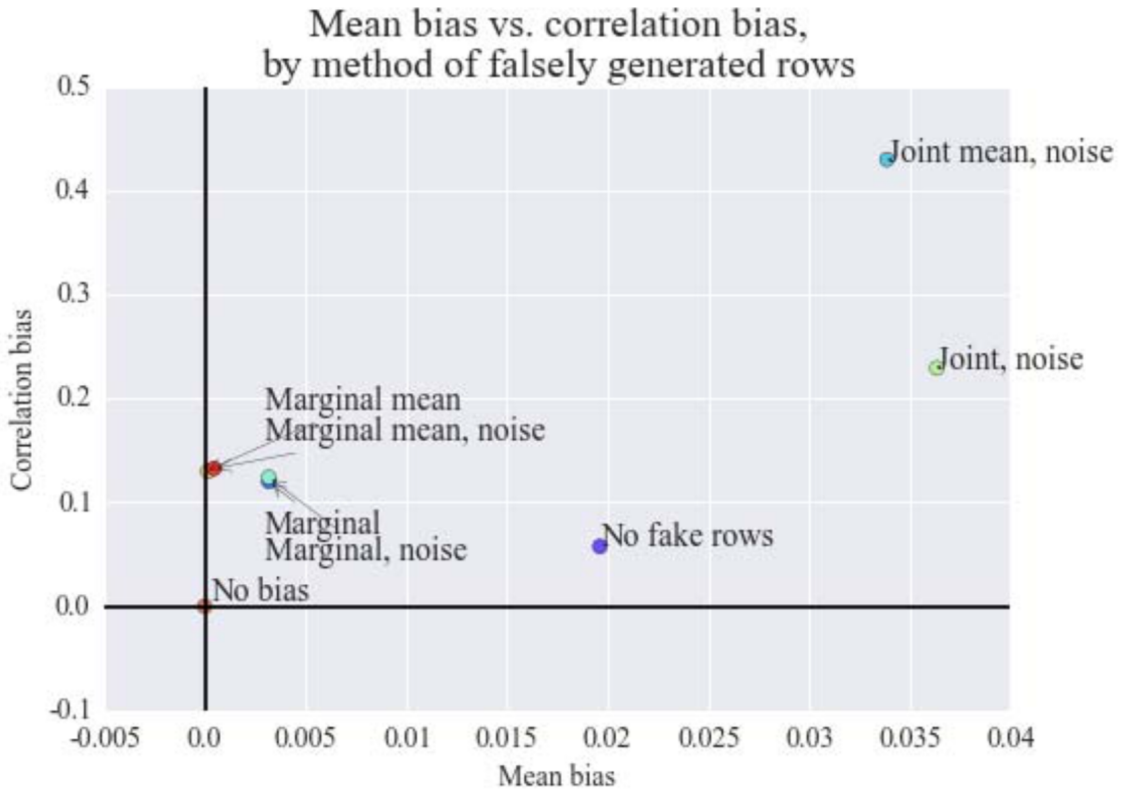

Figure 1: Plot of mean bias versus correlation bias for the four methods discussed in [1]. The marginal mean method performs the best of the four methods with .a mean bias close to zero and a correlation bias of roughly 0.13. The source of the figure is [1].

4 Alternative machine learning approaches

Our goal for this project was to improve on the results of the original paper by achieving a low mean bias, while significantly reducing the correlation bias. To do so, we sought to apply machine learning (ML) techniques to predict the grade values for the synthetic rows from the QIs. This is achieved by training each model on the original dataset, where the predictor variables are the QIs and the response variables are the non-QI fields (in our case, we are only considering grade, so there is only one response variable). Each model is then applied to the synthetic rows by predicting the response variable(s) from the QI predictors. This means that the synthetic rows are analogous to the test set, except in this case there is no ground truth for the response variables of the test set. Instead, the only metrics we can use to evaluate the quality of the model is the mean and correlation bias.

This ML approach is principled for several reasons. First, the marginal distribution of the predicted response variable for well-formulated models should strongly align with the distribution of the response variable in the training set. This means that the mean bias should be relatively low. Furthermore, the core feature of most machine learning approaches is the ability to identify the correlation between each predictor variable and the response variable. As a result, these ML methods should achieve a low correlation bias between the response and each of the predictors.

However, the benefits of ML actually extend beyond just minimizing the mean and correlation bias. Ultimately, we don't really care about just achieving low biases, as it's not as if users of the final dataset will only calculate summary statistics such as means and correlations. Rather, they will perform advanced statistical analysis that take into account more complicated properties of the dataset. Therefore, our true goal is to minimize any distortions to the statistical properties of the original dataset – the mean and correlation biases are simply proxies for this more fundamental objective.

One important manifestation of the statistical properties of a dataset is the p-values and confidence intervals of the parameters associated with a model that is trained on the given dataset. These p-values and confidence intervals reflect the uncertainty in the dataset regarding the relationships between all of the variables. Therefore, an additional goal of the process of filling in the non-QI fields is to maintain the same uncertainty in the synthetic dataset as is present in the original dataset. Fortunately, ML techniques are also adept at capturing this uncertainty, as most models include some sort of error component that represents the fraction of the variance in the response that cannot be explained by the predictors.

The approach we took to fill in the non-QI grade field took advantage of both of these strengths of ML techniques – capturing the relationship between variables, while also capturing the uncertainty of these relationships. Rather than simply filling in each grade value with the mean prediction outputted by the model, we sampled from the distribution of the response variable conditioned on the QI values (i.e. $p(grade|course\_id, YoB, gender, LoE, country, nforum\_posts)$). For example, if we were to fit a linear regression model, we would fill in the response variable by drawing from a normal distribution centered at the mean prediction with a variance equal to the $\hat{\sigma}^2$ estimated by the model. This method ensures that the uncertainty in the original dataset is preserved in the final dataset. Whereas the more rudimentary method of filling in the grade values with the predicted mean would significantly reduce the uncertainty in the final dataset because the mean values would reinforce the same signal in the original dataset and would thus give any model a false sense of confidence about the true relationships between the grade variable and the QIs.

4.1 Regression-based models

We fist applied regression models to fill in the grade value for the synthetic rows. To decide on which regression methods to use, we first examined the distribution of grade values in the original dataset. Of the non-null grade values, almost 89% of grades were zero. A grade of zero indicates that a student enrolled in the given course but failed to complete any of the assignments. Meanwhile, the other 11% of students who received a nonzero grade in a given course completed at least one assignment.

Because it would be extremely difficult to develop one regression model that captures both the zero and nonzero portions of the distribution, we split the process into two separate regression models. The task of the first regression model was to decide if, given a set of QI predictors, the grade should be zero, while the goal of the second regression model was to estimate the grade (i.e. a value between 0.01 and 1.0) for the synthetic rows that the first regression estimated to be nonzero.

Given the binary nature of the first regression task, we constructed a logistic regression model to predict whether a given set of QIs would result in a zero grade. To fit this model using Python's scikit-learn library we first had to one-hot encode each of the unique course_ids, which resulted in an explosion in the number of predictors in the model. We also created a new variable that took on a value of 1 or 0 if the corresponding grade value was zero or nonzero, respectively. The logistic regression model took upwards of an hour to train because of the fact that there were millions of observations and over 300 predictors in the design matrix.

One of the main assumptions of any generalized linear model is that each quantitative predictor is linearly related to the response variable on the scale of the link function. For a logistic regression in particular this means that each quantitative predictor should be linearly correlated with the logit of the response variable. However, because the response variable can only take on two values in the case of a logistic regression, calculating the logit of the response would be meaningless as $log\left(\frac{p}{1-p}\right)$ is undefined when p is 0 or 1. Instead, a more useful diagnostic involves plotting the logit of the predicted probability against each quantitative variable. The resulting plot where nforum_posts was the quantitative predictor revealed a highly nonlinear relationship between the two quantities. This is no surprise because the distribution of nforum_posts is highly rightly skewed and the majority of the students don't post even one time on the forum. To rectify this issue, nforum_posts is shifted up by 1 and then log-transformed. The resulting diagnostic plot indicates a much more linear relationship between the two quantities. At this point, the logistic regression model was refit with the transformed nforum_posts variable.

Having trained the logistic regression model, the next step was to predict whether the synthetic students would earn a grade of zero in their course. Generally this would involve calculating the probability of success (in this case a grade of zero) and then predicting 1 if the probability is above 0.5 and 0 otherwise. However, as explained above, in order to maintain the uncertainty in the original dataset, the grade values were instead classified as zero or nonzero by sampling from a different Bernoulli distribution for each synthetic row, where the probability of success was equal to the probability outputted by the logistic regression model. By applying this methodology, roughly 74% of the synthetic rows were filled in with a grade of zero.

The next step was to fill in the rows of the synthetic dataset where the grade was predicted to be nonzero. This involved training a regression model on the 561,737 rows of the original dataset where the grade value was not null and nonzero. The best way to decide which type of regression is most appropriate is to inspect the distribution of the response variable.

As shown below in Figure 2, the distribution of nonzero grades has most of its mass in the two extremes: most students stopped early on and thus earned very low grades, a significant number of students completed the course and earned high grades, and only few students either stopped in the middle of the course or did very poorly on all the assignments and received grades around 0.5. The shape of the empirical distribution suggests that the distribution is plausibly composed of a mixture of beta distributions, where each grade value is sampled from a beta distribution with parameters that are related to the QI predictors. This observation motivates the use of a beta regression model to predict the nonzero grade values from the QI fields. However,

because beta regression models aren't used very frequently, there are no Python libraries that implement it. Instead, we used the betareg package in R to fit the beta regression model. Unfortunately, this fitting process yielded errors related to the internal implementation of the beta regression model, and occurred even when just 1/50 of the training set was used to fit the model. As a result, we had to devise an alternative model.

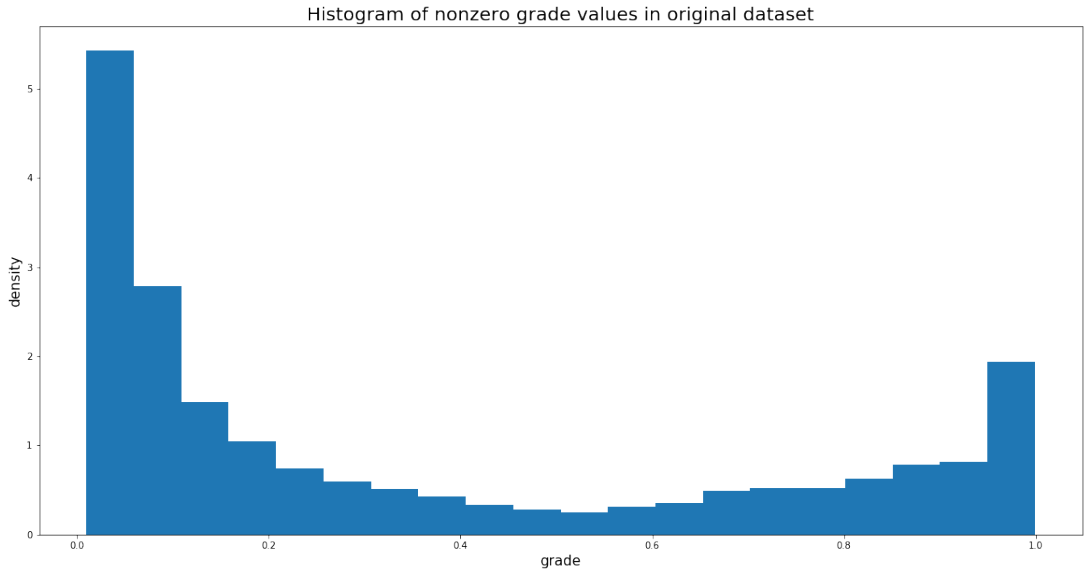

Figure 2: Histogram of nonzero grade values in the original dataset.

Because there aren't any common distributions aside from the beta distribution whose support lies between 0 and 1, we had to transform the grade values so that we could apply a different regression model. Two reasonable transformations were the logit and probit functions, which both map values between 0 and 1 to the entire real line. The probit transformation yielded a more pleasant-looking distribution so we preferred it over the logit transformation. The resulting transformed distribution was right skewed so we decided to use a gamma regression. This first involved having to shift all the grade values such that they were all positive.

We then fit the gamma regression model on the training set using a log link and predicted the grade values on the link scale for the synthetic rows. Next, we applied properties of exponential dispersion models to calculate the rate and shape parameters of the gamma distribution corresponding to each predicted grade. Then, just as we did with the logistic regression model, we sampled each grade value from the corresponding gamma distribution. Finally, we shifted the predictions back and applied the inverse of the probit function (i.e. the cumulative distribution function of a standard normal distribution) to transform the grade values

back to a scale between 0 and 1. Figure 3 below displays the predicted grade distribution and compares it to the grade distribution in the original dataset.

Figure 3 indicates that the distribution of filled-in grade values has the same overall shape as the grade distribution in the original dataset and is able to capture the two peaks in the distribution. However, it places too much mass in the right peak, and not enough mass in the left peak. Nonetheless, it's pretty remarkable that the gamma regression model was able to achieve this degree of accuracy despite the transformations that were applied and the fact that it was only used as a last resort because a beta regression model couldn't be used.

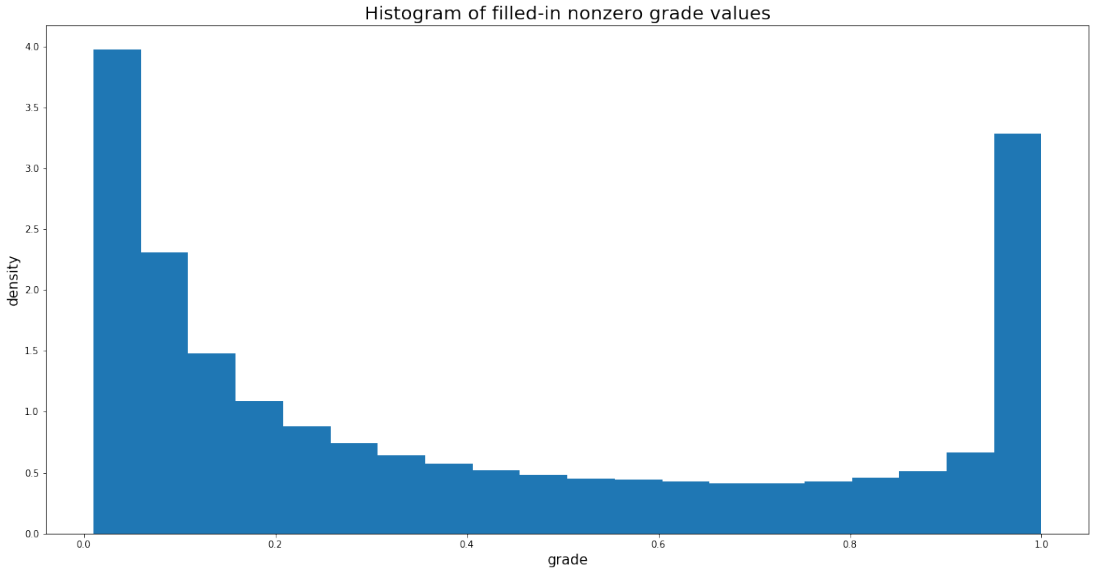

Figure 3: Histogram of filled-in nonzero grade values.

The Jupyter notebook and R Markdown file used to create these regression models can be found here and here.

4.2 Tree-based models

We also fitted tree-based models to fill in the grade values for the synthetic rows generated. Particularly, we fitted simple decision tress as well as ensemble models like random forests and gradient boosting models. Theoretically, tree-based models exhibit some advantages over regression models when it comes to high dimensional, complicated data with a mixture of categorical and numerical predictors. Firstly, tree-based

models are non-parametric and make no assumptions about the underlying distribution of the response variable or the form of the relationship between the predictors and the response variable. Thus, tree-based ensemble models can potentially fit more complex patterns than simple regression models. Secondly, tree-based models are more robust to outliers than regression models, which is especially important when the dataset contains information that was filled in by users and in turn might contain erroneous values (e.g. YoB was 3116 for one of the rows in the MOOCs dataset). Finally, tree-based models can easily be used to fill in both numerical as well as categorical response variables; in other words they can be used both for classification and regression tasks. However, it is worth noting, that tree-based models require a careful tuning of a large number of hyperparameters to balance between low bias and high variance in predictions. In this task, we are less interested in low prediction error and more interested in preserving the distributional properties of non-quasi-identifiers as well as their correlations with the other columns. Hence, the most unbiased model might not always be the best model for this task.

Specifically, we experimented with the following tree-based models:

1. Decision tree with maximum depth of 10 levels (depth was determined based on a simple grid search and cross-validation score).
2. Random forest with trees of maximum depth of 10 levels, 5 features used at each level for doing the split and averaging over 50 trees in total.
3. Bagging algorithm which averages the results of 50 decision trees of maximum depth of 10 levels.
4. Gradient boosting (Adaboost) with 500 steps and learning rate = 0.1.

Tree-based models are deterministic algorithms which means that they will return the same output when presented with the same input. However, we would not like our model to always fill-in the same value of grade for the same combination of quasi-identifiers. This would result in minimal ℓ-diversity and in turn it would be easy to find out which rows are synthetic, defeating the whole purpose k-anonymity. To address this, we added small random noise drawn from $N(0, 0.025^2)$ to the grade returned by any of the above models in order to mask the synthetic rows. In a more sophisticated approach, we could determine the amount of noise based on the standard deviation of the grades that appear in the bottom leaf of the tree based on which the decision is made.

The results are summarized in the next section. The Jupyter notebook with the code used to experiment with these tree-based models can be found here.

5 Results

Before implementing the ML models described above, it's important to first reproduce the results obtained in the original paper. Table 1 below presents the mean and correlation bias that we achieved when we implemented the optimal method presented in [1] of drawing 5 random samples from the marginal mean distribution and filling in the average of these samples. This procedure was carried out for the simple method of creating synthetic rows as well as the more rigorous method based on identifying which sets of QIs are subsets and supersets of each other.

	Original paper	Simple row generation method	Subset row generation method
Mean Bias	0	-0.0000199	0.0000135
Correlation Bias	0.13	0.144	0.137

Table 1: Replication of marginal mean method presented in [1]

As expected, the mean bias is essentially zero for both methods, which concurs with the results of the paper. It's also reassuring that the correlation bias for each methods is also very close (within 0.02) to the correlation bias obtained in the paper. Finally, it makes sense that the correlation bias is slightly lower for the subset method because fewer synthetic rows needed to be added to the dataset, so there should be less distortion to the statistical properties of the dataset.

Table 2 below presents the results of the ML models we used to fill in the grade values of the synthetic rows:

	Regression	Decision Tree	Random Forest	Bagging	Adaboost
Mean Bias	0.0372	0.0387	0.0343	0.0384	0.0721
Correlation Bias	0.0285	0.0578	0.0859	0.0621	0.0547

Table 2: Mean and correlation biases achieved by ML models

Table 2 shows that each of the models achieved a relatively low mean bias and significantly reduced the correlation bias. As expected, the mean bias for any of the ML methods is significantly larger than the mean bias introduced by drawing the grade from its marginal distribution. However, slightly distorting the mean grade should not affect the downstream analysis, as the statistical significance usually depends on the correlations of the response variable with the predictors. Overall, the regression model performed the best

out of all the models considered. This is particularly encouraging for several reasons: first, these low biases indicate that the parametric regression model did a reasonable job capturing the highly complex distribution of grades in the original dataset. This means that the regression model could have presumably performed ever better if the distribution of the response variable were better behaved (e.g. unimodal and resembling a common statistical distribution). Secondly, as explained in the regression methodology section, a beta regression model would have been the most appropriate way to model the nonzero component of the grade distribution; a gamma regression was only out of convenience rather than based on first principles. And as shown in Figure 3, this inferior gamma regression model was unable to precisely capture the nonzero grade distribution in the original dataset. Based on this, we're confident that we would have achieved an even lower mean and correlation biases had we been able to fit a beta regression model.

Taking a step back, it's promising that the relatively straightforward ML models used in this project were able to achieve such solid results. This suggests that more sophisticated methods or more careful tuning of the hyperparameters of algorithms like gradient boosting, may perform even better. For example, given the millions of observations in the original dataset a feedforward neural network could be trained to predict grade from the QI predictors. Such a model would likely be flexible enough to capture the complex distribution of grades in the original dataset. Additionally, one major drawback of the logistic and gamma regression models was that the hundreds of unique courses had to be one-hot encoded into the design matrix. This meant that no information was shared between rows of the dataset with different course_ids, which limited the ability of the models to capture broader relationships in the data and made courses with fewer students more prone to overfitting. One way to overcome these downsides would be to create a Bayesian hierarchical regression model where the information from all the courses would be pooled together and courses with fewer students would be shrunk toward the overall average, thus safeguarding against overfitting.

6 Conclusion

Overall, this project demonstrates the potential of ML techniques to fill in the non-QI fields of synthetic rows in such a way that minimizes the distortion to the statistical properties of the original dataset. However, it's important to caution that these results are only preliminary and that more work needs to be done to confirm the advantages of these ML models. For example, we only focused on filling in the grade variable and ignored the dozens of other non-QIs in the dataset. Therefore, the methodology described in this report should be

repeated to fill in these other fields and the overall mean bias and correlation bias should be averaged across all of these variables. Moreover, as explained in the report, evaluating the statistical distortion to a dataset extends beyond measuring the mean and correlation bias. Hence, more sophisticated metrics should be used to gauge how the synthetic dataset differs from the original dataset. One possible approach would be to fit the same regression model on the original dataset and the synthetic dataset, and then calculate the differences in the p-values and confidence intervals of the parameters associated with the two models. It is only after performing this more comprehensive analysis that we can conclude that these ML methods are in fact a more principled way to fill in the non-QI fields.

References

[1] O. Angiuli and J. Waldo. Statistical tradeoffs between generalization and suppression in the de-identification of large-scale data sets. In *2016 IEEE 40th Annual Computer Software and Applications Conference (COMPSAC)*, volume 2, pages 589–593. IEEE, 2016.

[2] L. Sweeney. k-anonymity: A model for protecting privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 10(05):557–570, 2002.