Homework 2 DS GA 1017 Joby George (jg6615) Due 4/14/22

Problem 1: Racial Disparities in predictive Policing

The 2016 study by Lum and Isaac found a disparity between the number of drug arrests in Oakland. The study uses survey data and showed that drug usage does not differ by racial groups, however, drug related arrests were strongly concentrated in a few predominantly non-white counties.

Consider a hypothetical ML system, such as the one scrutinized by Lum and Isaac that uses historical arrest data which optimizes police allocation. The system could complement historical arrest data with other useful datasets (time of year, weather, number of clubs and bars, etc in the neighborhood, etc.)

1A: Give 3 distinct reasons why racial disparities might arise in the predictions of such a system.

1. The historical arrest data suffer from pre-existing bias. The data do not reflect true crime use per-se, but just where arrests were made. This means that the historical policing policies, which have disproportionately impacted minority populations would inform the model to target these groups.

2. Intentional biasing of the model to target minorities. While this is the most nefarious reason for disparities in predictive drug policing, it should not be discounted. If the model's goal is to predict drug arrests using previous drug arrest data, that means the defendant in question has violated a law. John Ehrlichman, the domestic policy chief under Nixon who started the war on drugs by criminalizing marijuana is quoted saying:

   "The Nixon campaign in 1968, and the Nixon White House after that, had two enemies: the antiwar left and black people. You understand what I'm saying? We knew we couldn't make it illegal to be either against the war or black, but by getting the public to associate the hippies with marijuana and blacks with heroin, and then criminalizing both heavily, we could disrupt those communities"[1]

   Given the political and economic power incarceration has over communities, and the stark disparities in drug arrests prior to predictive policing, it is not unfathomable that the model creators (or users) wanted to defend existing policing practices by creating an 'unbiased, evidence-based' model that was just as targeted and nefarious as the

original legislation which criminalized drugs.

3. A perverse model optimization goal. According to the Lum and Isaac's article: predictive policing is defined as “the application of analytical techniques, particularly quantitative techniques, to identify likely targets for police intervention and prevent crime or solve past crimes by making statistical predictions.'

   This definition is widely vague enough to allow for many optimization goals. If this model's goal is to optimize police allocation to maximize drug arrests, volume becomes the primary incentive for the model to optimize for.

In closely examining the graphics in the Lum & Isaac article, I noticed that arrests are most concentrated in West Oakland, between Intestate 580 and Interstate 980. This is not where the reported drug use is highest, which is actually slightly east to where arrests are concentrated. However, this area is conveniently 10 minutes away from the Oakland police department. If there is a volume incentive, or department quota that must be met, the model would prefer likely drug users closer to the department rather than further away, which concentrates arrests and creates the disparities we see. Highlighting this hypothesis, is the original Lum & Isaac graphic, below, showing the difference between the highest use areas and highest arrest areas. Then another visual from Google maps showing an approximate 10 minute drive distance from the Oakland Police department to the area with highest arrests.

John Ehrlichman Quote Source: https://www.vox.com/2016/3/22/11278760/war-on-drugs-racism-nixon

Lum and Isaac Graphic, with circle highlighting most active drug use area on left and drug arrest area on right

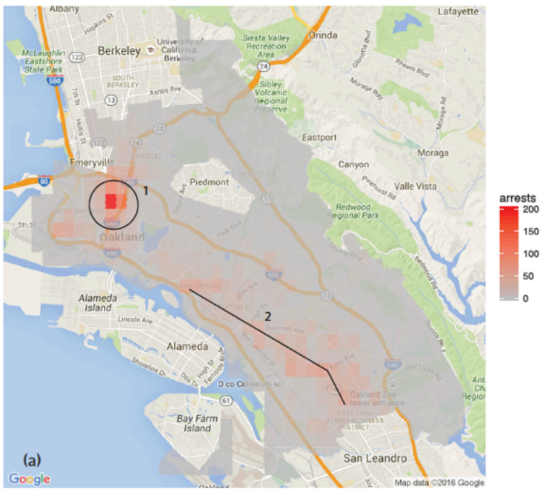


Figure 1(a): Number of drug arrests, 2010.

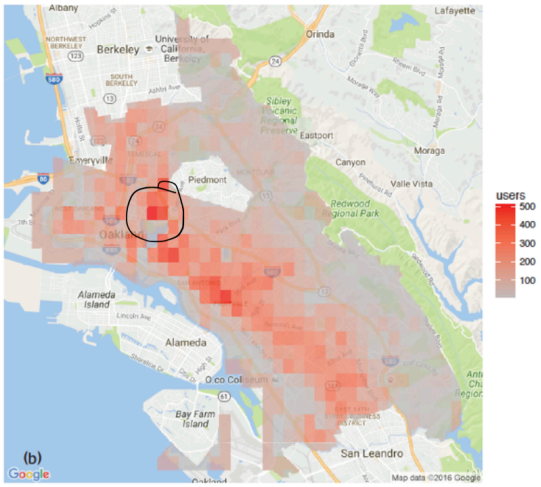
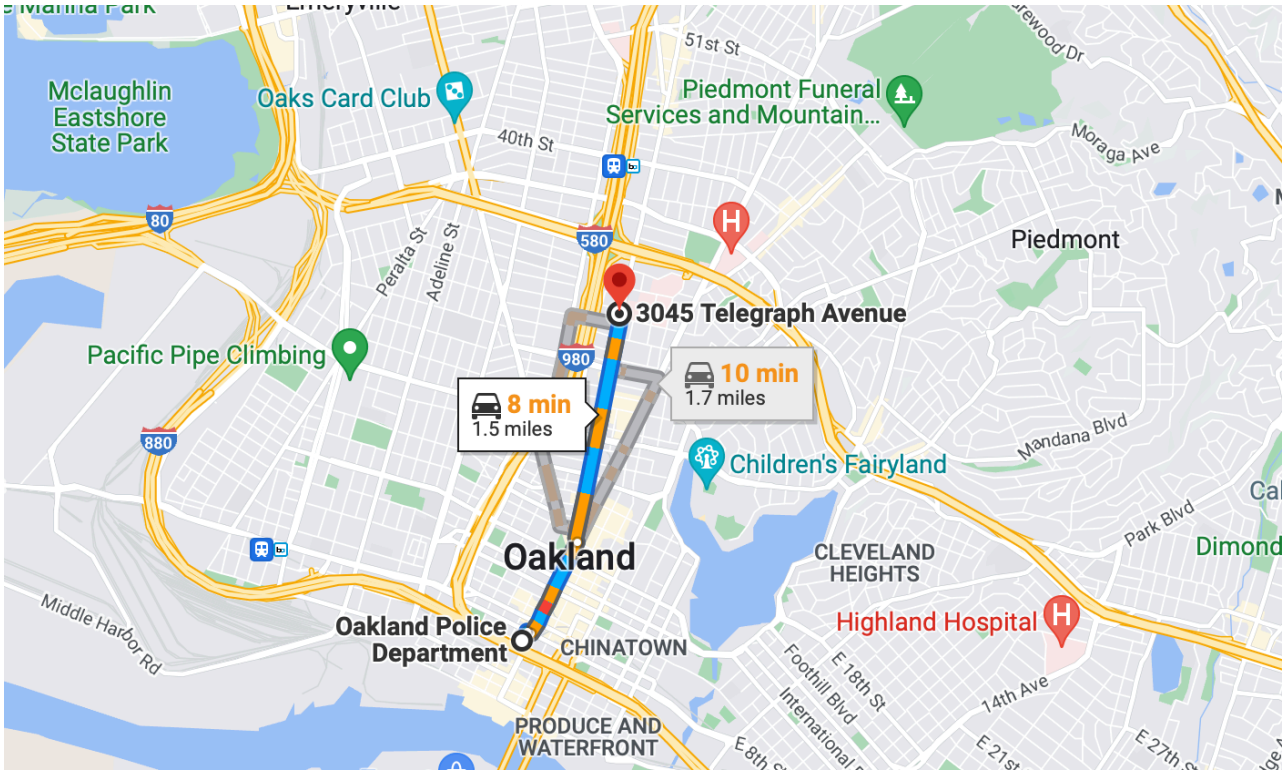


Figure 1(b): Estimated number of drug users, 2011.

Graphic showing distance from most active arrest area to police department



Problem 1B:

Propose two mitigation strategies to counteract racial disparities in the predictions of such a system. Note: It is insufficient to state that we could use a specific pre-, in- or post-processing technique that we covered in class when we discussed fairness in classification. Additional details are needed to demonstrate your understanding of how the ideas from fairness in classification would translate to this scenario.

1. A specific fairness constraint within the model's loss function could mitigate these racial disparities . As mentioned above, if the model is trying to maximize the number of arrests, distance from the police department greatly impacts who will be targeted. However, if there is a fairness constraint, for example number of arrested drug users per capita must be approximately uniformly distributed by zip code, this would prevent the model from exclusively selecting the closest likely targets and ensure likely targets from all of Oakland are included.

1. A second way to mitigate the disparities would be to use Reject Option Post-Processing on the dataset. Our assumption here is that the unprivileged class are People of Color. However, for this model to even be legally admissible, race should not be a feature in the model. Therefore, we would use geography as the variable we would create privileged and unprivileged groups, setting the zip codes with the highest arrest concentration as unprivileged and the zip codes with minimal arrests as the privileged groups.

   It's still worth noting that even this may me be legally inadmissible, as this could be

similar to 'red-lining' but for police data, however, it depends on the level of specificity of geographic location (a 5 digit zip code, vs a 3 digit zip code for example). By doing this, likely targets for drug use within the privileged group would get de-emphasized and likely targets for drug use outside the concentrated areas would be prioritized by the model in order to geographically re-balance arrest concentration.

1. A third remediation would be to use a less accurate model, one that would not incorporate previous crime features at all, but another set of uncorrelated features with drug crimes in predictive policing (height, amount of taxes paid, pet owner, etc.). While this would likely be poorly received by politicians and the police department, a more randomized process would definitively result in police confronting more of the predominant race.

In looking at these three different recommendations, the first proposal seems much more legally fair. fairness constraint that per-capita likely targets by zip code must be approximately uniform enforces that all populations are considered, which is in line with the mission of police to protect and serve the entirety of Oakland, and it also does not try to remediate existing racial bias by specifically calling out populations as privileged and unprivileged. The third option, while it addresses racial disparity is so unexplainable and unfair to the suspected targets that it (hopefully) would never be used by an institution of power.

Problem 2 Randomized response

The simplest version of randomized response involves flipping a single fair coin (50% probability of heads and 50% probability of tails). As in the example we saw in class, an individual is asked a potentially incriminating question, and flips a coin before answering. If the coin comes up tails, he answers truthfully, otherwise he answers "yes". Is this mechanism differentially private? If so, what ϵ value does it achieve? Carefully justify your answer.

Problem 2 answer:

Epsilon, is calculated as the ratio of ln($\frac{P(A|P)}{P(A|-P)}$)

Let:

```
A = the response given by the individual, in our example we set
it to yes
P = True response, which we set to we set it to Yes
```

ϵ, is calculated as the ratio of ln(P(A|P)P(A|–P))

P(A|P) = P(Tails) + P(heads) = 1 P(A|-P) = P(Heads) = 1/2

Therefore this mechanism is differentially private, accomplishing an ϵ of 2 when the true out come is

Let:

```
    A = the response given by the individual, in our example we
set it to No
    P = True outcome of the event, which we set to Yes
```

P(A|P) = 0 (if coin flip is tails, he answers yes instead of no) \newline P(A|-P) = 0 (if coin flip is heads, he answers yes instead of no)

Since one of the combination of coin flips and determined processes cannot happen, the ratio of $\frac{P(A|P)}{P(A|-P)}$ is undefined, meaning this process **is not** differentially private. If the true outcome was yes to the incriminating question, this mechanism will always force the respondent to answer yes.

Problem 3

Problem 3

Consider the dataset below, and assume that sex is one of {M, F}; edu is one of {HS, BS, MS}; and loan is one of {yes, no}. Here, sex is the protected attribute, and loan represents the binary classification outcome (the target variable): loan=yes is the positive outcome, loan=no is the negative outcome.

id	sex	edu	loan
F1	F	HS	no
F2	F	HS	no
F3	F	HS	no
F4	F	HS	no
F5	F	HS	no
F6	F	HS	no
F7	F	BS	yes
F8	F	BS	yes
F9	F	BS	yes
F10	F	BS	no
F11	F	BS	no
F12	F	BS	no
F13	F	MS	yes
F14	F	MS	yes
F15	F	MS	no
F16	F	MS	no

id	sex	edu	loan
M1	M	HS	yes
M2	M	HS	yes
M3	M	HS	yes
M4	M	HS	no
M5	M	HS	no
M6	M	HS	no
M7	M	BS	yes
M8	M	BS	yes
M9	M	BS	yes
M10	M	BS	yes
M11	M	BS	no
M12	M	BS	no
M13	M	MS	yes
M14	M	MS	yes
M15	M	MS	yes
M16	M	MS	yes

A classification association rule (CAR) is a non-trivial association rule of the form X1, ..., Xn →Y, where Y is an assignment of a value to the target variable (loan=yes or loan=no), X1, ..., Xn is an assignment of values to one or several other variables. For example: sex=M, edu=BS → loan=yes is a CAR, while loan=yes → sex=F is not.

To mine CARs from a dataset, you may think of each tuple (row) as a “transaction”, and then apply the Apriori algorithm we covered in class (during the data profiling lecture in Week 6) to find CARs that meet or exceed the specified confidence and support thresholds.

Problem 3.A

List all CARs that relate the likelihood of the classification outcome (loan=yes or loan=no), with the value of the sensitive attribute sex. These CARs should list sex on the left-hand-side, either on its own or in combination with other attributes. List only those CARs that have support >= 3 and confidence >= 0.6. For each CAR you list, state its support and confidence.

Rule	Implies	Loan Status	Support	Confidence

(gender=F)	->	Loan = No	11	.688
(gender=F & edu=HS)	->	Loan = No	6	1
(gender = M)	->	Loan = Yes	11	.688
(gender = M, edu = BS)	->	Loan = Yes	4	.667
(gender = M, edu = MS)	->	Loan = Yes	4	1

Problem 3B

Suppose that you are required to release differentially private versions of the frequent item-sets (the union of their left-hand-side and right-hand-side) that correspond to the CARs you computed in part (a), along with their support. Your overall privacy budget is ε=1. Use sequential and parallel composition to allocate portions of the privacy budget to each frequent item-set you will release. Your goal is to maximize utility of the information you release, while staying within the privacy budget.

Write down a way to allocate portions of the privacy budget to each frequent itemset you will release, to achieve good utility. Be specific, write down an epsilon value for each itemset. Carefully justify your solution using sequential and parallel composition.

Answer:

Query list:

1. SELECT gender, loan_status, count(loan_status) as support FROM T
   GROUPBY gender,loan_status
   (epsilon budget = .6)

2. SELECT gender, education, loan_status, count(loan_status) as support FROM T
   WHERE education = 'HS'
   and gender = 'F'
   and loan_status = 'No'
   GROUPBY gender, education loan_status
   (epsilon budget = .16)

1. SELECT gender, education, loan_status, count(loan_status) as support FROM T
   WHERE education = 'BS'
   AND gender = 'M'
   AND loan_status = 'Yes'
   GROUPBY gender, education loan_status
   epsilon budget = .12)

2. SELECT gender, education, loan_status, count(loan_status) as support FROM T

WHERE education = 'BS'
AND gender = 'M'
AND loan_status = 'Yes'
GROUPBY gender, education loan_status
epsilon budget = .12)

Outputs:

1. (M, Y, 11), (F, N, 11)
2. (F, HS, N, 6)
3. (M, BS, Y, 4)
4. (M, MS, Y, 4)

Explanation:

We can parallelize the largest rules with support, those by gender using a GroupBy. Given these data-points account for 22 people out of the total 36 in our CAR, I decided to assign an epsilon value of .6.

After parallelizing the largest CAR, an additional person added to the dataset would by definition impact any of our other CAR's. For example, if we added a Male, with Bachelors education level and loan status = Y, that would impact the overall number of Males with a Yes loan status, and the number of Males, with BS education getting approved. So we must use sequential queries for the rest of our CARs.

Epsilon values were again, split relatively proportional to the support for each of our rules, with F, HS, N having 6 support and thus getting .16 epsilon, whereas the two last CAR's each had 4 support and were thus assigned .12 epsilon.

Summing our epsilon, we get 1, completely using our budget (.6 + .16, + .12 + .12 = 1).

Problem 4

In this problem you will take on the role of a data owner, who owns tow sensitive data sets, called hw_compas and hw_fake and is preparing to relase differentially private synthetic versions of these datasets.

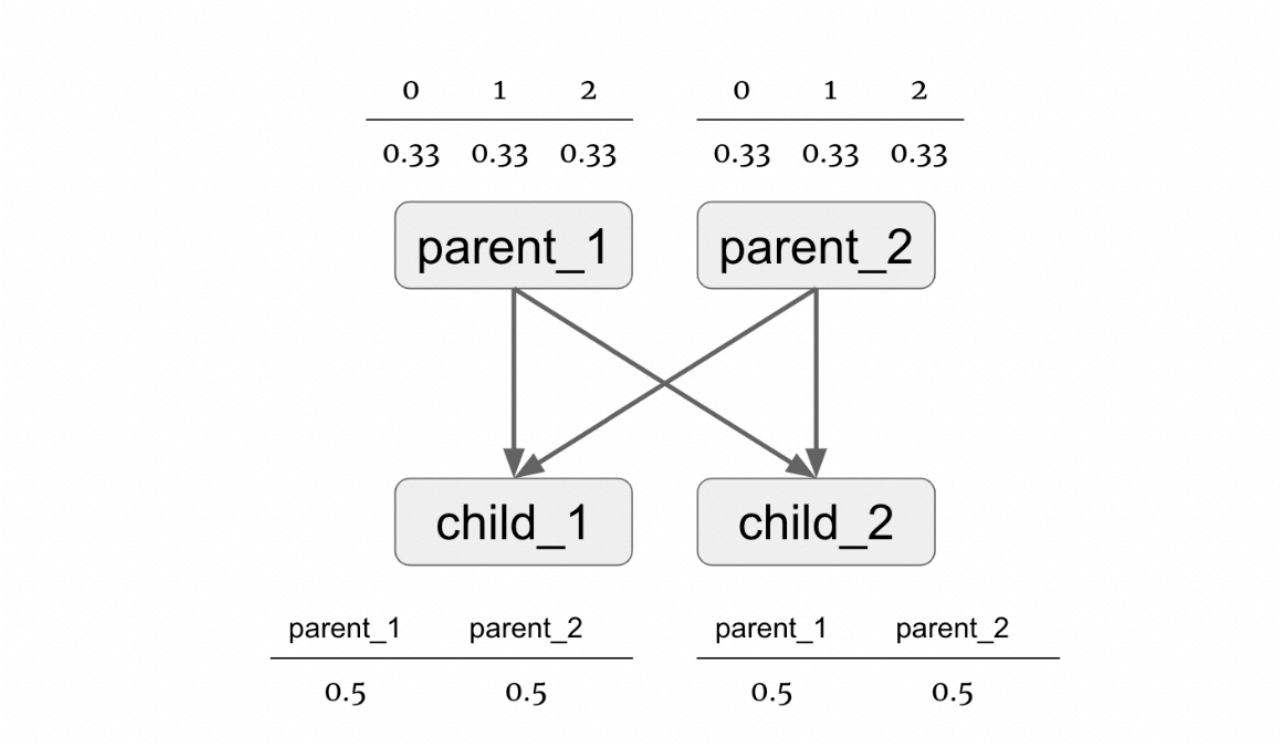
The first dataset, hw_compas is a subset of the datset released by ProPublica as part of their COMPAS investigation The hw_compas dataset has attributes:

1. age

2. sex
3. score
4. race

Note: Sex takes values of 'Male' or 'Femlae', score is an integer between -1 and 10, race is one of 'Other', 'Caucasian', 'African-American', 'Hispanic', 'Asian', 'Native American'.

The second dataset, hw_fake, is a synthetically generated dataset. We call this dataset “fake” rather than “synthetic” because you will be using it as input to a privacy-preserving data generator. We will use the term “synthetic” to refer to privacy-preserving datasets that are produced as output of a data generator. We generated the hw_fake dataset by sampling from the following Bayesian network:



In this Bayesian network, parent_1, parent_2, child_1, and child_2 are random variables. Each of these variables takes on one of three values {0, 1, 2}.

1. Variables parent_1 and parent_2 take on each of the possible values with an equal probability. Values are assigned to these random variables independently.

2. Variables child_1 and child_2 take on the value of one of their parents. Which parent’s value the child takes on is chosen with an equal probability.

To start, use the Data Synthesizer library to generate 4 synthetic datasets for each sensitive dataset hw_compas and hw_fake (8 synthetic datasets in total), each of size N=10,000, using the following settings:

1. A: random mode

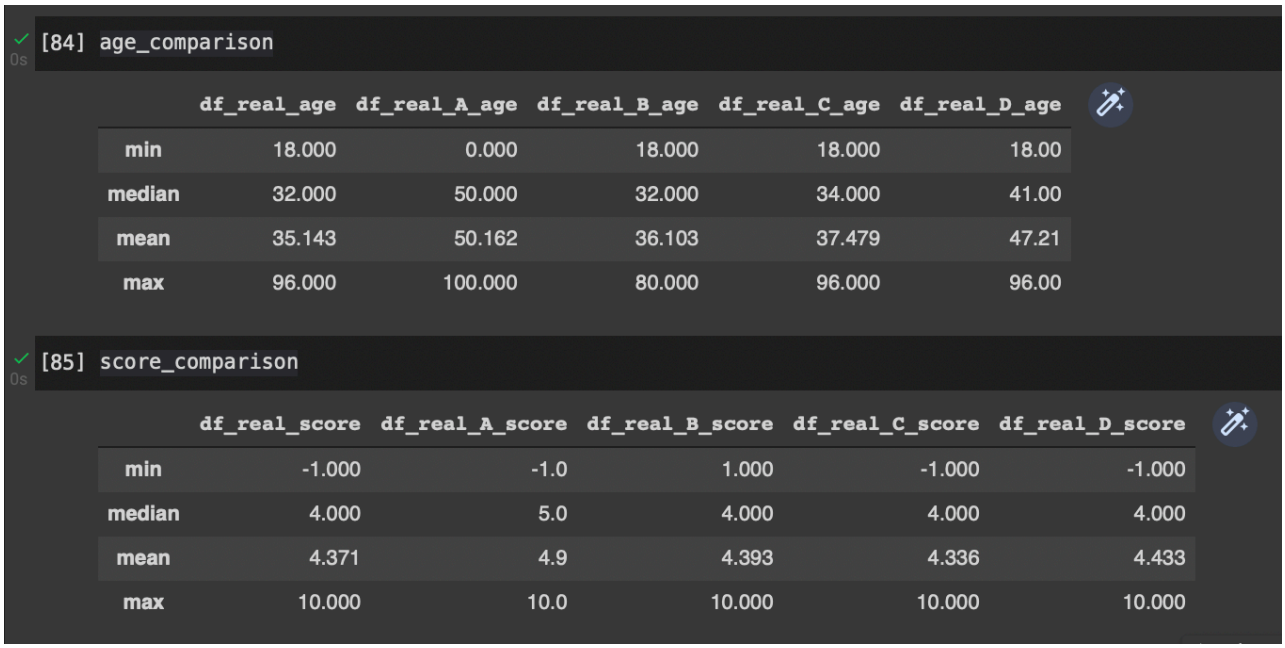
2. B: independent attribute mode with epsilon = 0.1.
3. C: correlated attribute mode with epsilon = 0.1, with Bayesian network degree k=1
4. D: correlated attribute mode with epsilon = 0.1, with Bayesian network degree k=2

4.A

4.A1

(hw_compas only): Execute basic statistical queries over synthetic datasets. The hw_compas has numerical attributes age and score. Calculate Median, Mean, Min, Max of age and score for the synthetic datasets generated with settings A, B, C, and D (described above).

4.A1 Answer



[84] age_comparison

	df_real_age	df_real_A_age	df_real_B_age	df_real_C_age	df_real_D_age
min	18.000	0.000	18.000	18.000	18.00
median	32.000	50.000	32.000	34.000	41.00
mean	35.143	50.162	36.103	37.479	47.21
max	96.000	100.000	80.000	96.000	96.00

[85] score_comparison

	df_real_score	df_real_A_score	df_real_B_score	df_real_C_score	df_real_D_score
min	-1.000	-1.0	1.000	-1.000	-1.000
median	4.000	5.0	4.000	4.000	4.000
mean	4.371	4.9	4.393	4.336	4.433
max	10.000	10.0	10.000	10.000	10.000

We observe that the synthetically generated data that is not purely random (datasets B,C, and D) which use the independent data mode, and bayesian networks are much more accurate at persevering the distributions of our quantitative variables than the random mode.

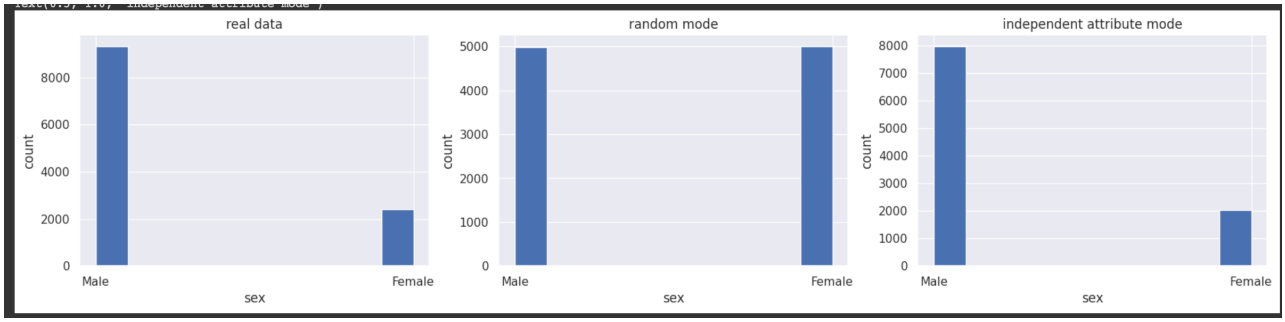
This makes sense, because the random mode data generator, for integer values, takes a random integer value between the minimum and maximum of our variable value, rather than trying to truly replicate the underlying distribution of our data. For the score integer variable, which takes values in a range of -1 to 10, the randomly generated values are actually relatively in line with the true data, although the true data are slightly more concentrated at lower scores.

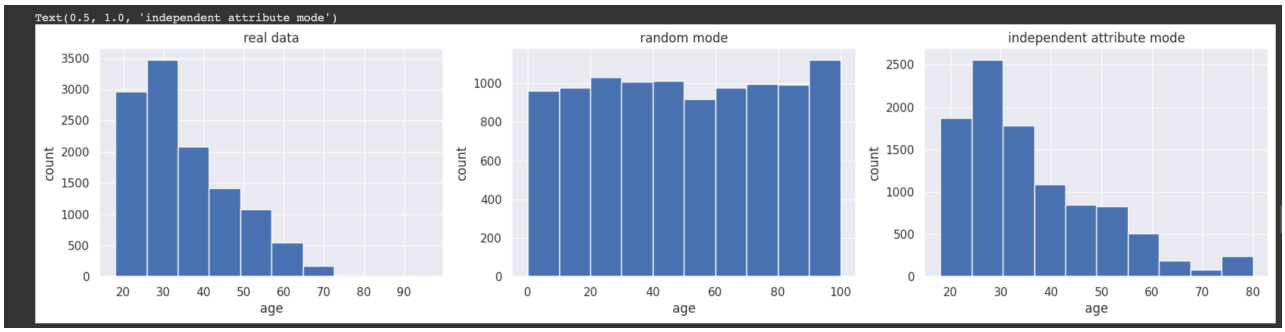
The last call-out to take is that our mean age for a Bayesian network with 2 variables used as parent nodes, the age mean is skewed higher than our real data. In examining the Bayesian Network, the variable age has parents of score and sex, which means that our data generating function is ensuring that the correlation between score and sex with age is persevered.

4.A2

(hw_compas only) Compare how well random mode (A) and independent attribute mode (B) replicate the original distribution.

Answer





As mentioned above, the random mode just samples randomly, or assigns equal probability to every possible value for a given attribute. With age, the data are integers, with values from 100, and we see evenly distributed values there, whereas sex in this case is binary, with a 50/50 split between sexes.

However, in our actual data, we have imbalanced data, with the data containing far fewer women compared to men, and more younger data points compared to people older than 50 years old.

4.A2 Continued

Next, compute cumulative measures that quantify the difference between the probability distributions over age and sex in hw_compas vs. in privacy-preserving synthetic data. To do so, use the Two-sample Kolmogorov-Smirnov test (KS test) for the numerical attribute and Kullback-Leibler divergence (KL-divergence) for the categorical attribute, using provided functions ks_test and kl_test.

The results of the **KS test on age was a** statistic of .3465.** This has a corresponding p value of 0.0, meaning we reject the null hypothesis that the data are originated by the same underlying distribution, which confirms our visual understanding of the data.

When looking at the **KL-divergence, we observe a value of .191.** there is no p-value associated with this metric. Our random data for sex had a 50/50 split for both sexes out of 10,000 samples, while our true data had an 80/20 split with Men being the more frequent sex. The value of .19 is comparatively not too large, given there are only two sexes and seeing an extreme skew is somewhat difficult.

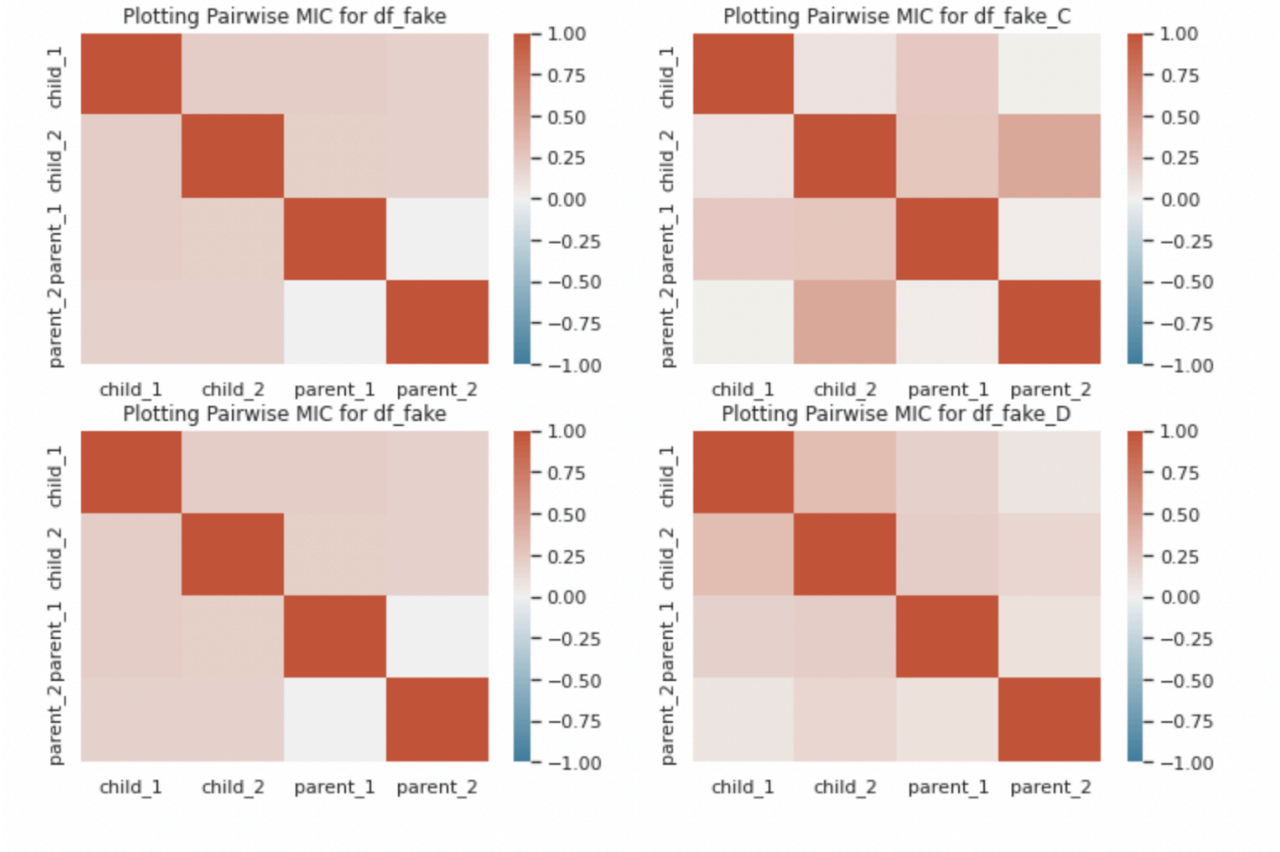
To ground my understanding, I calculated the KL-Divergence using race as the categorical variable. This column had a .587 value, which is magnitudes higher than sex. This is because in our dataset, 90% of the population are contained within the African-American and Caucasian races, compared to the 32% in the randomly generated synthetic dataset.

4.A3

(hw_fake only)

Compare the accuracy of correlated attribute mode with k=1 (C) and with k=2 (D).

Display the pairwise mutual information matrix by heatmaps, showing mutual information between all pairs of attributes, in hw_fake and in two synthetic datasets (generated under C and D). Discuss your observations, noting how well / how badly mutual information is preserved in synthetic data.



When looking at the heatmap, it is worth noting that the **left hand heatmaps** are identical, as they are both for the original fake datset, df_fake.

In examining these heat maps we see that child 1-2 are correlated with each other, as well as both parents. However, the parents are not correlated with each other.

In looking at our synthetically generated dataset, when only using one variable in our marginal (df_fake_C, or the top right graph), we notice that the children variables are not correlated with each other, and that parent 2 is not correlated with child 1.
In looking at the synthetically generated dataset, when using two variables in our marginal, we observe some correlations that are stronger than our original dataset, namely the children are more correlated with each other, and the parents are also more correlated with each other.

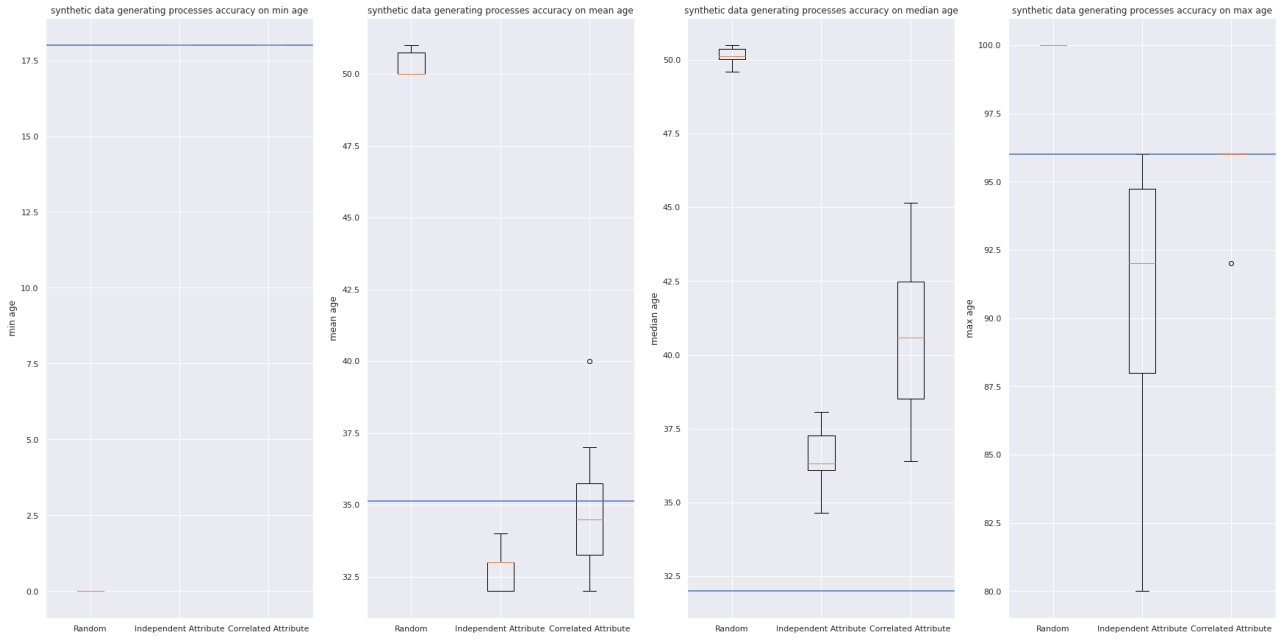
The lack of correlation in df_fake_C between child 1 and parent_2 is important, as it does not align with the true data generating process, and thus mutual information is better preserved with the synthetic data using two variables in its marginal.

4.B

Study the variability in accuracy of answers to Q1 under part (a) for A, B, and C for attribute age.

To do this, fix epsilon = 0.1, generate 10 synthetic databases (by specifying different seeds). Plot median, mean, min, max as a box-and-whiskers plot of the values for all 10 databases, and evaluate the accuracy of the synthetic data by comparing these metrics to the ground truth median, mean, min, and max from the real data.

Carefully explain your observations: which mode gives more accurate results and why? In which cases do we see more or less variability?



The box-plot above shows how minimum, mean, median, and maximum age of synthetic datasets vary for a fixed epsilon of .1 using the Random, Independent and Correlated Attribute mode from the Data Synthesizer package.

In addition, each of subplot has a blue horizontal line that shows the true data's value for that specific aggregate metric. From first observation we can see that the minimum age was constant throughout all ten runs for all of the data generating modes.

Random had a minimum age of 0 every time, while the other two methods perfectly synced up with the true value of 18 years old.

For the mean age, we observe that the random dataset typically choses a value around 50, in line with uniform sampling from a distribution between 0 and 100, whereas our independent and correlated attributes are generally much closer to the true value of 36.

The median is not maintained for any of the data generating methods, with random again,

performing the worst, then correlated attribute mode and independent attribute mode performing the best.

When looking at the maximum age, we observe that the random mode consistently had a maximum age of 100, while independent attribute mode had a lot of variability, and correlated attribute mode took a maximum of 92.5 consistently, compared to the true maximum of 96.

In general, we see the least variability with the random mode, especially with the minimum and maximum aggregate metrics, which consistently took a value at the lowest and highest value of our sampling range. However, the random mode performed the worst in relation to maintaining the aggregate metrics, as it did not take into the account the distribution of age in our dataset.

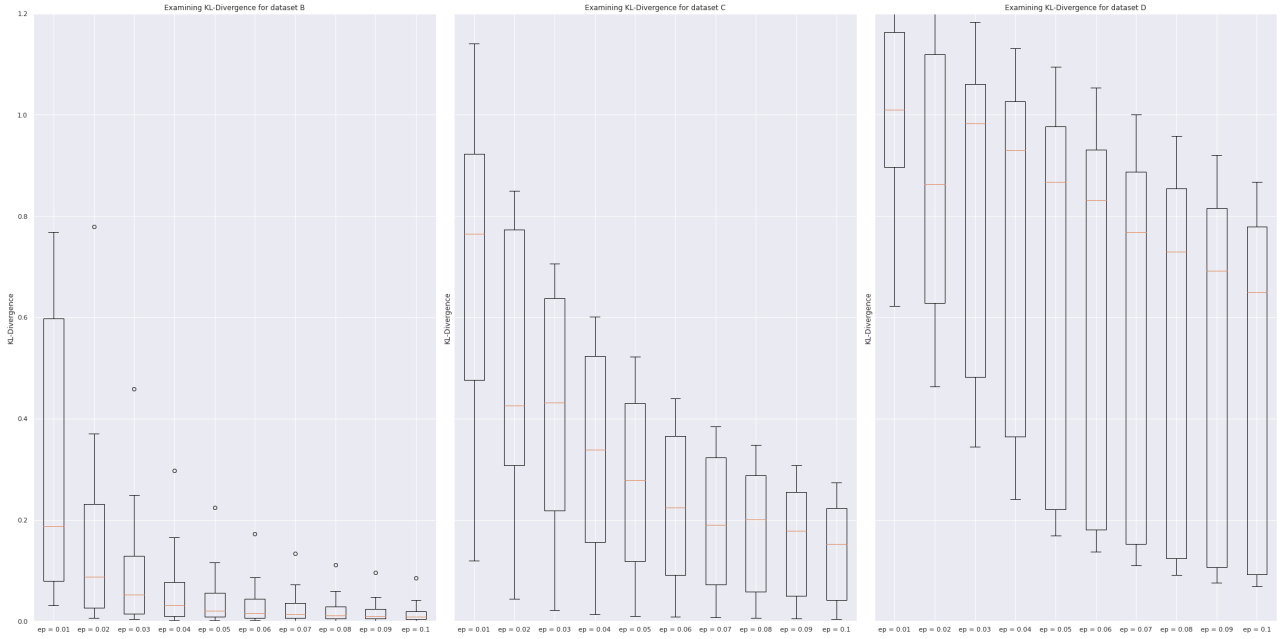
Conversely, the other two modes performed about equally well, with independent mode having less variability on mean and median age, but greater variability on maximum age.

4.C

Study how well statistical properties of the data are preserved as a function of the privacy budget. To see robust results, execute your experiment with 10 different synthetic datasets (with different seeds) for each value of epsilon, for each data generation setting (B, C, and D). Compute the following metrics, visualize results as appropriate with box-and-whiskers plots, and discuss your findings in the report.

KL-divergence over the attribute race in hw_compas. Vary epsilon from 0.01 to 0.1 in increments of 0.01, generating synthetic datasets under B, C, and D. Specifically, the epsilons are [0.01, 0.02, 0.03, 0.04, 0.05, 0.06, 0.07, 0.08, 0.09, 0.1] and in total, you should have 3*10*10 datasets generated. Please plot the distributions of KL-divergence scores (10 samples each) with box-and-whiskers plots where you treat epsilon as the X-axis and generation settings as subplots.

The difference in pairwise mutual information, aggregated (summed up) over all pairs of attributes, for both hw_compas and hw_fake.



Analysis of KL-Divergence variation between different data generating processes

We see the least divergence in the Independent Attribute mode. This makes sense, because when examining the divergence of a singular feature, independent attribute mode should learn the distribution of the feature from our real dataset and faithfully recreate it.

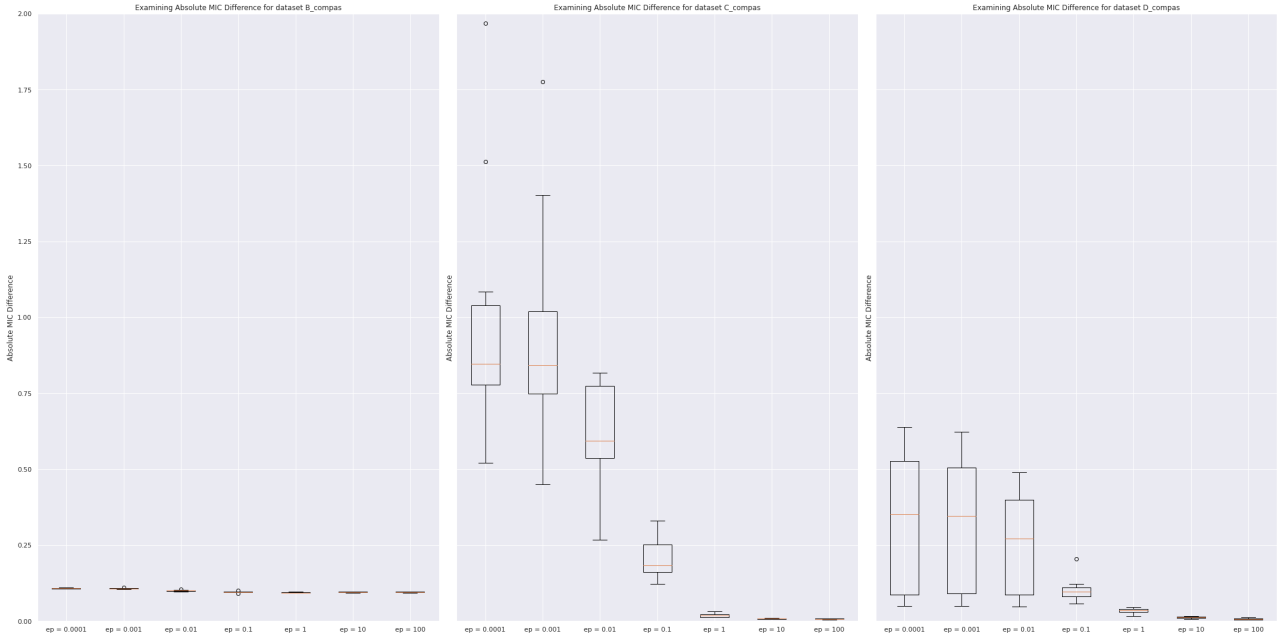
We then see more divergence for the correlated attribute mode with k = 1, and then k =2. The expense of using epsilon to construct the Bayesian network grows as k grows, which contributes to the fact that increasing K increases the overall divergence.

We do observe, that as epsilon increases, across data generating methods, the divergence decreases, albeit it is still less noticeable for dataset D.

Analysis of MIC absolute differentials between different data generating process

Intuitively, if independent attribute mode performed the best on KL-divergence of a single attribute, we would expect our correlated attribute modes to minimize an aggregate measure of divergence, such as pairwise mutual information.

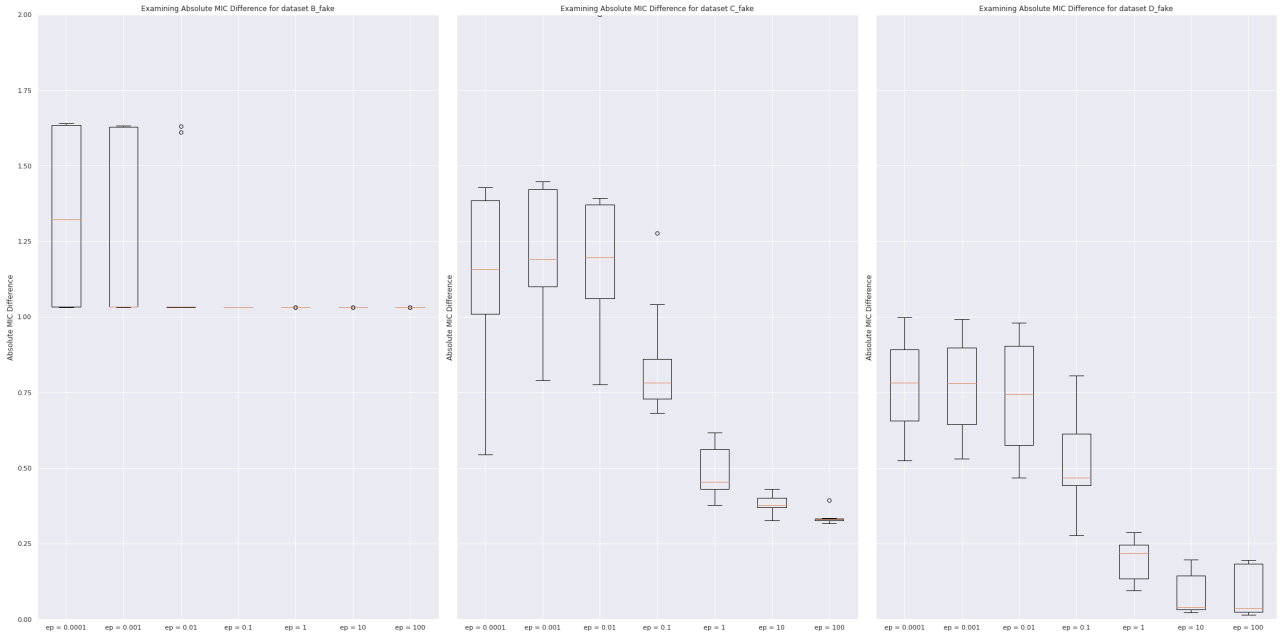
The simulations of our three datasets, B,C,D over 7 different epsilon values and 10 samples each, is plotted below for the Compas Dataset.



We can see that Independent attribute mode is the most consistent and least variable when it comes to MIC differential, however, the correlated attribute modes with an epsilon budget of 1 have a lower absolute difference of pairwise mutual information between the real data.

This makes sense, with a low epsilon budget, the process of creating the Bayesian Network, and generating our data is not done faithfully, but as we expand our budget we begin to create a dataset that captures the underlying distribution of our true dataset, better than assuming our features are independently distributed.

We see this broader trend in our fake data as well, in the box-plot below.



The noticeable difference between the fake dataset and the real dataset is that the correlated attribute modes, even at a low epsilon perform better than our independent

attribute mode. This is most likely due to the assumptions of the data generating process for our fake data, in which the only independent attributes are parent 1 and parent 2. In assuming each of the attributes are independent, the independent attribute mode is less accurate than the correlated attribute mode with a small epsilon.

The other interesting thing about the fake data, is that while the variation decreases as we increase epsilon for Datasets B and C, it does not decrease for dataset D. This is likely because in assuming conditional dependence on two variables, our data generating process creates synthetic noise that does not truly exist and as a result, is more variable across all levels of epsilon.

4.D

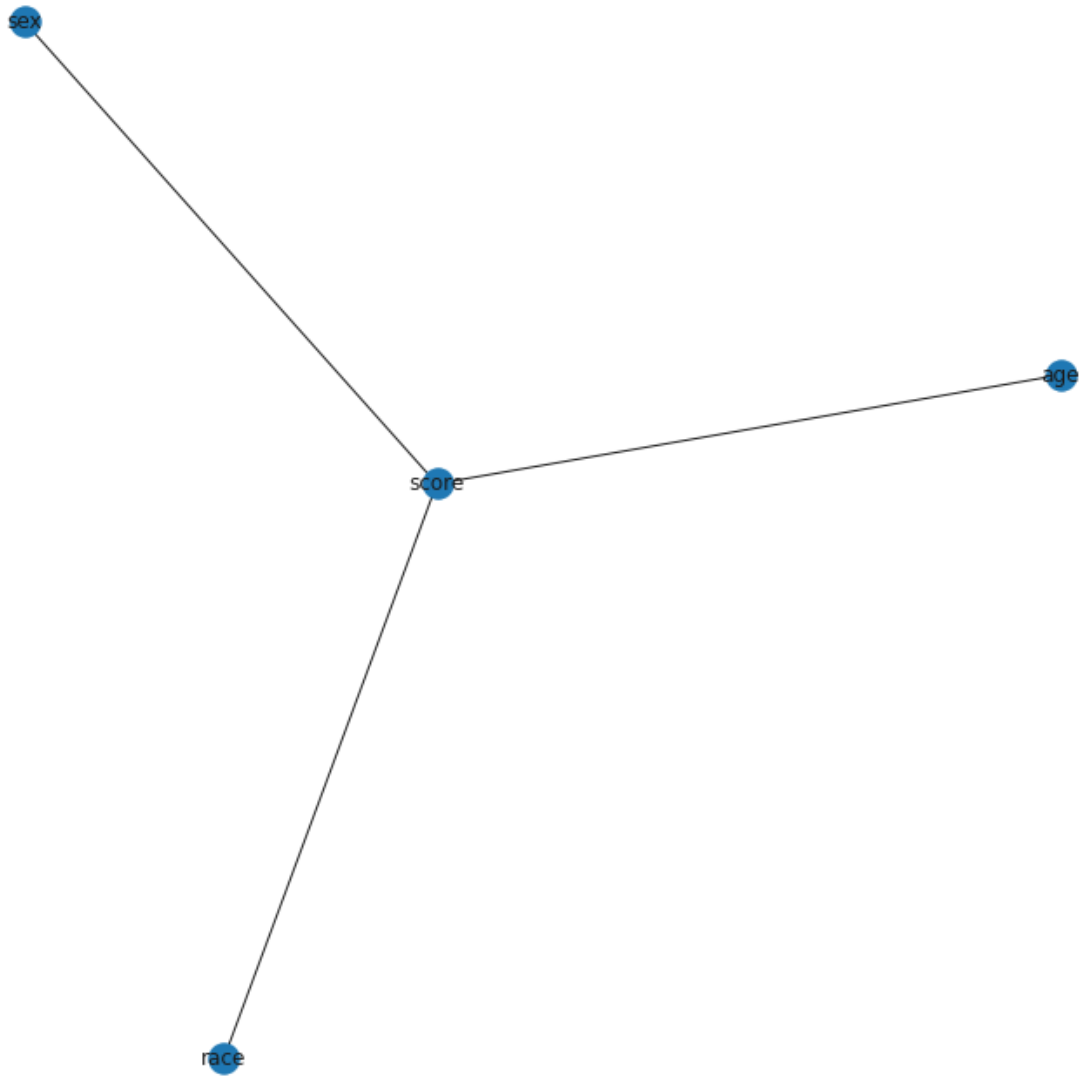
Compare the DP model learned by the DataSynthesizer with the model learned by another synthesizer, MST, from the Private-PGM library.

Recall from our discussion in class that, instead of maintaining an internal Bayesian network, as does DataSynthesizer, MST finds the maximum spanning tree on a graph where nodes are data attributes and edge weights correspond to approximate mutual information between any two attributes. The spanning tree is then used to decide which 2-way marginals to estimate.

To start, use MST to compute the differentially private spanning tree for hw_fake, with epsilon = 0.1. Compare the spanning tree produced by MST with the Bayesian network produced by the Data Synthesizer under condition D (correlated attribute mode with k=2, with epsilon=0.1). Inspect the spanning tree, discuss which marginals were selected, and how the information they capture is similar or different compared to the conditional tables that are estimated by the Bayesian network of the Data Synthesizer in condition D. (Note that there is no easy way to look at the counts inside the marginal, so this question is asking you to discuss the structure of the models.)

The two DAG's for the MST process on our Compas data, and Correlated Attribute Mode with k = 2, epsilon = .1 are shown below.

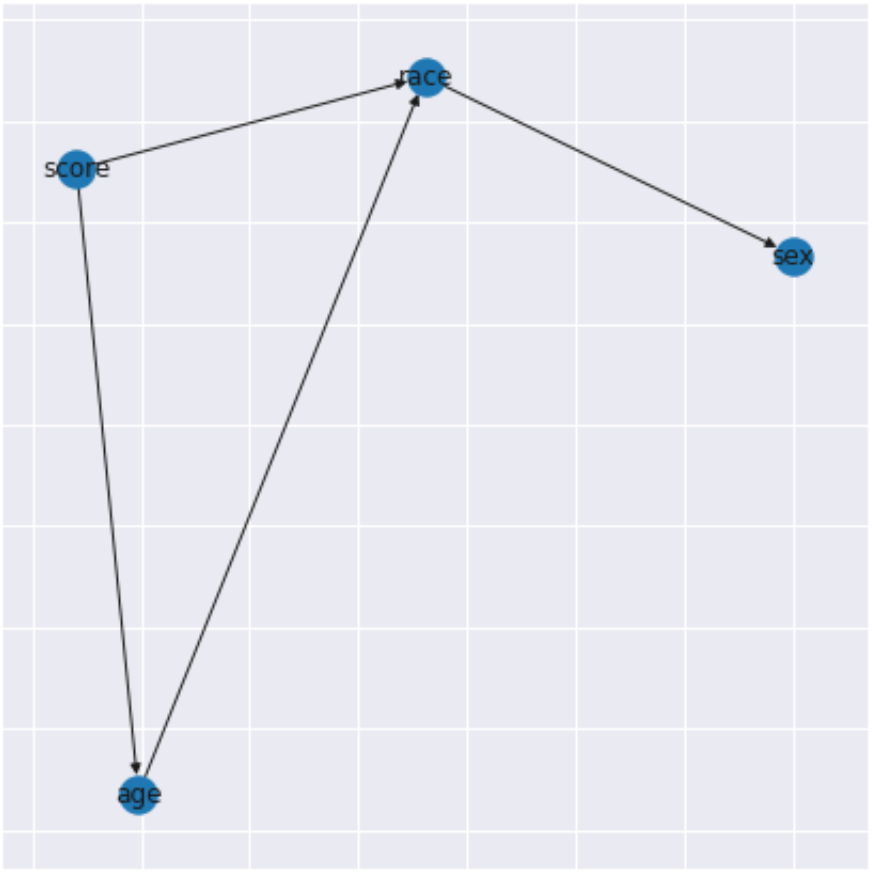
MST DAG



Note: **This is not the same DAG as the one in the template, despite not changing any code**

We can see that in the DAG the MST process produced, score is a child of all of the explanatory variables of interest, race, age, and sex. However, none of the categorical variables have a strong enough pairwise mutual information to be linked to one another.

Correlated Attribute DAG (k=2)



In the Correlated Attribute Mode with K=2, we observe a different structure underlying the features in our dataset. Score is now the parent for most of our features, except sex, which has parents age and race (children of score)

The two way marginal calculation and the heuristic pruning factors in the MST algorithm allow for a simpler structure, with only important feature relationships to be kept in the network that will synthesize synthetic data.

4.E

Repeat part of question (c) above for MST, for both hw_compas and hw_fake. Let's refer to this as condition E: MST with epsilon as specified below, generating synthetic datasets of size N=10,000.

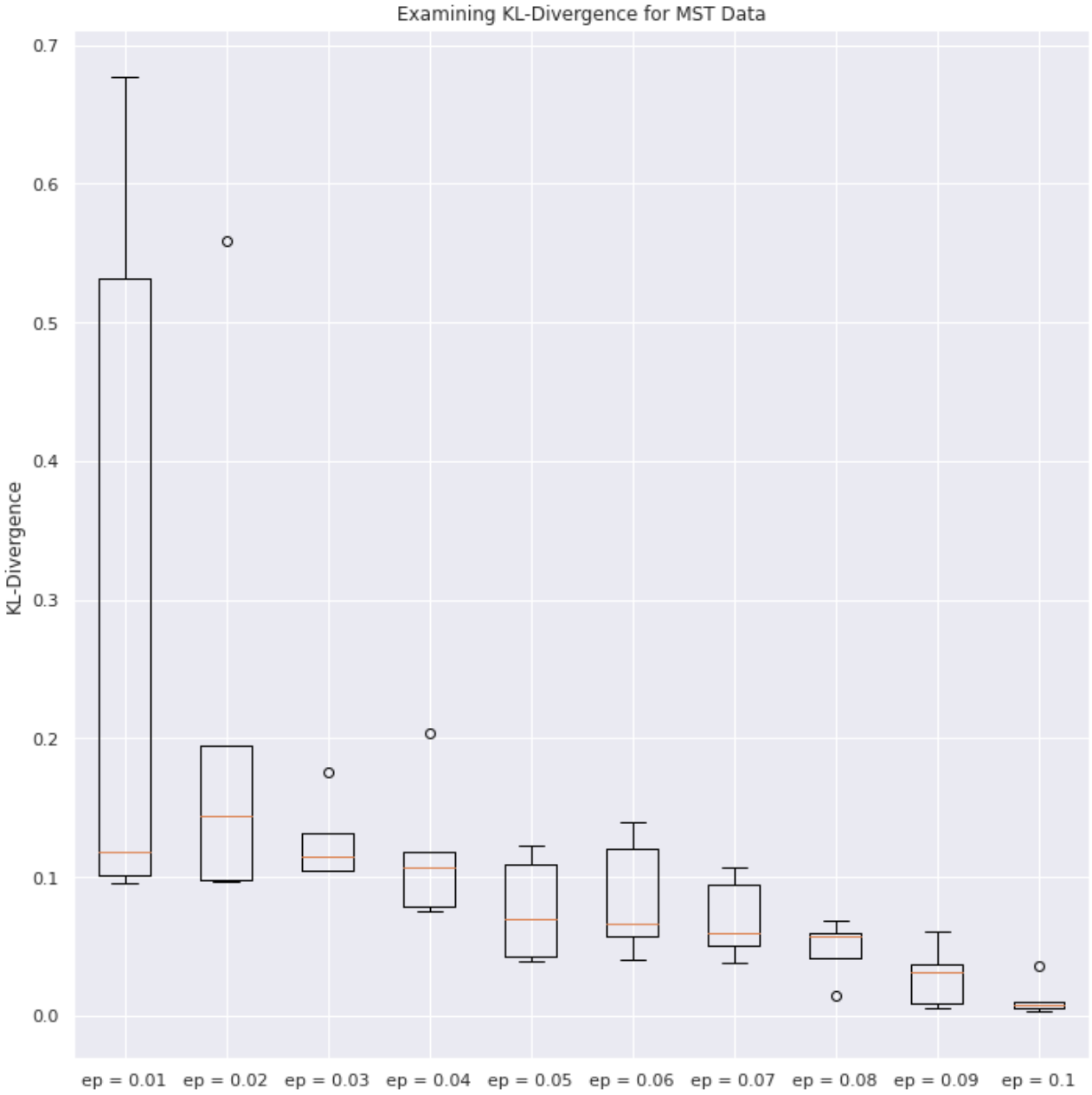
Generate synthetic datasets under condition E as follows: Vary epsilon from 0.01 to 0.1 in increments of 0.01, and generate 5 different datasets for each value of epsilon. (We are reducing the number of datasets to 5, down from 10, because MST takes longer to run.) In total you should have 5*10 datasets generated. Measure KL-divergence over the attribute race in hw_compas. Plot the distribution of KL-divergence scores (10 samples each) with box-and-whiskers, with epsilon on the X-axis.

Generate synthetic datasets under condition E as follows: Use epsilon values 0.0001, 0.001,

0.01, 0.1, 1, 10, and 100, and generate 5 different datasets for each value of epsilon, for each hw_compas and hw_fake. In total you should have 2 *7* 5 datasets generated. Compute and plot the difference in pairwise mutual information, aggregated (summed up) over all pairs of attributes, for both hw_compas and hw_fake, as in part (c) above. The y-axis should start at 0. Ensure that your plot has the same range of y-axis values as the plots in part (c), so that the values are comparable across experiments.

Discuss your findings, comparing performance of MST under condition E and of Data Synthesizer under condition D.

4E1 Plotting KL divergence using MST



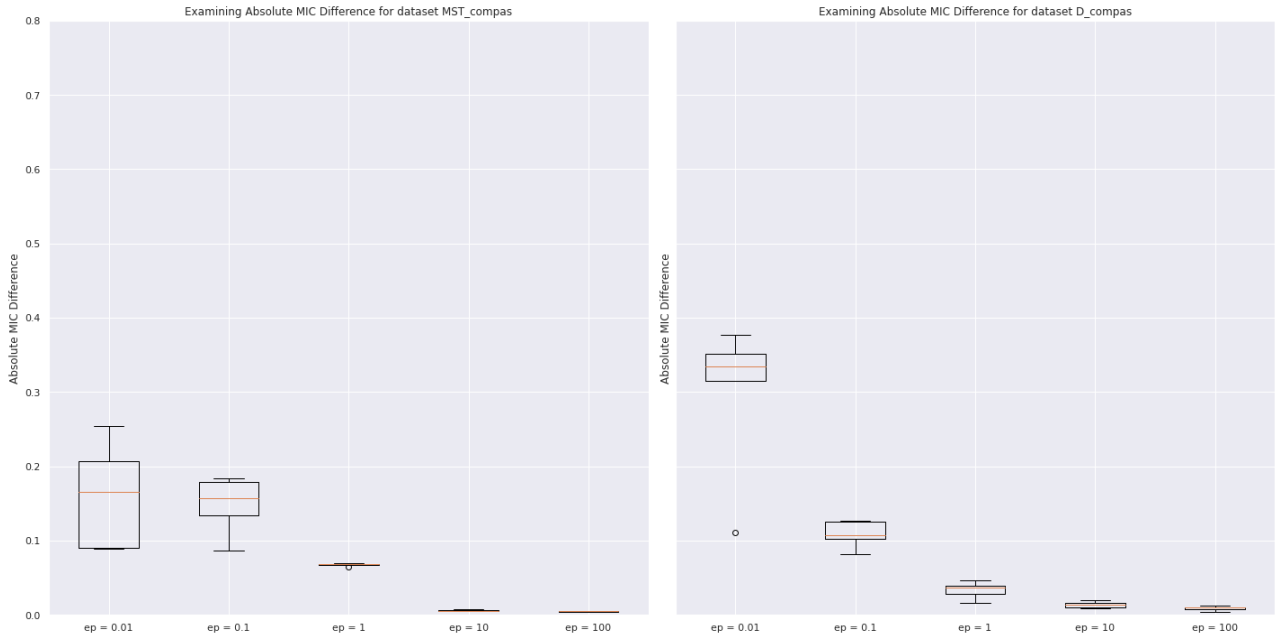
We see that epsilon is an extremely important factor for the MST algorithm, with the algorithm generates results with high variability with a low epsilon and then quickly stabilizes

by epsilon = .03.

The actual KL-divergence levels themselves are lower than our other data generating processes, after epsilon =.02, with a mean value of approximately .01 with an epsilon >=.02. This is much better than our correlated attribute modes for the same epsilon and in line with independent attribute mode's performance on this metric

4E2 Plotting MIC for MST on COMPAS data and Dataset D

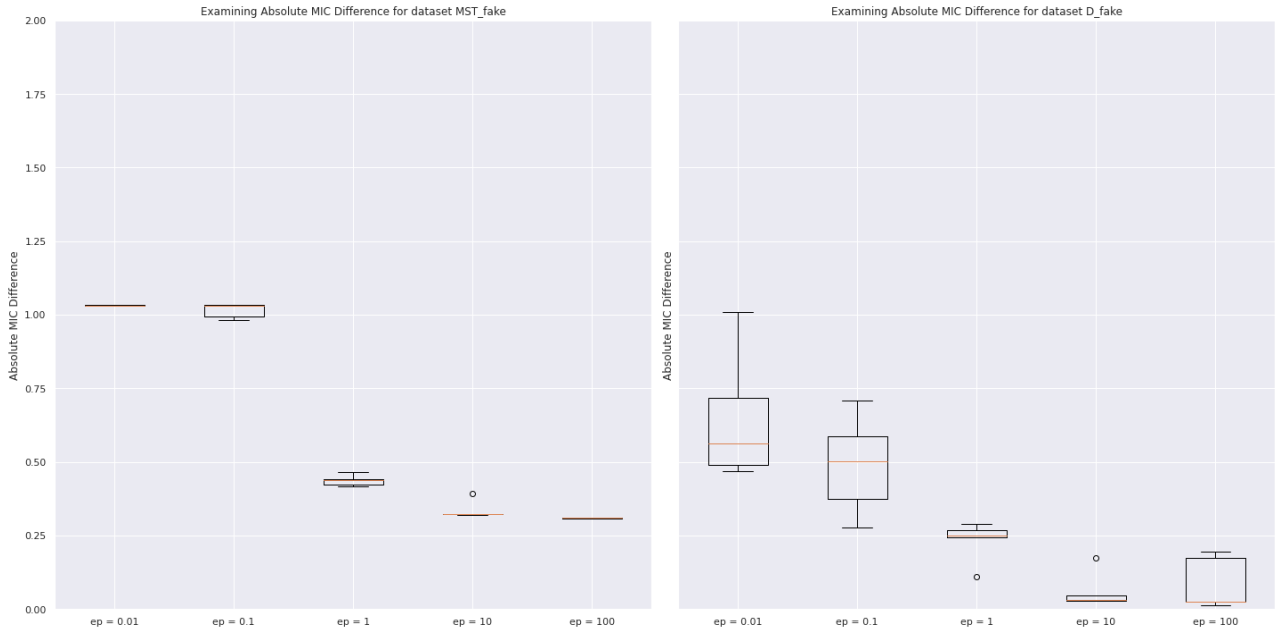
Unfortunately, I was not able to get $\epsilon \in \{.0001, .001\}$ to work, so the box-plots below do not contain those points.



When looking at the MIC for the MST algorithim versus correlated attribute mode with k=2 on the Compas dataset, we observe what Professor Stoyanovich mentioned in class, that the MST algorithm does a much better job compared to Data Synthesizer.

Across all levels of epsilon, we see a lower level of MIC, when comparing the MST algorithm to Correlated attribute mode where k = 2.

4E2 Plotting MIC for MST on fake data and Dataset D



On the fake dataset we observe something different, where the MST has less variation in MIC, but is generally higher than the K=2 correlated attribute mode. This is especially noticeable at higher epsilons. Additionally, at epsilon = 100, the variance for Dataset D was higher. I examined some of the Bayesian networks and MST's for epsilon = .1 and epsilon = 100 for the two methods. At epsilon = .1, both methods had a more interconnected approach, with both parent features being related to children.

My hypothesis as to why Data Synthesizer works better on our fake data is the built in dependence on how the children features take values between the parents. Since the Bayesian Network looks at a simple one way marginal, with up to two parents, this is more reflective of the true underlying data generating distribution for our fake dataset.

In [ ]: