# The Confusion Matrix Evaluation Metric

Author: Sowmya Srinivasan

## Executive Summary

### Metric Overview

The objective of the confusion matrix metric is to measure the quality of the privatized dataset in comparison to the ground truth dataset with respect to both map and time segments. This metric has been adapted from the pie chart metric created by Christine Task, Knexus Research Corporation Issac Slavitt, DrivenData Inc. The original pie chart metric quantizes the effects of the privatization algorithm by looking at the Jensen-Shannon distance between the privatized dataset and the ground truth dataset in addition to the number of false presences as well as the overall increase of the values in the privatized dataset. The confusion matrix metric seeks to cover more of the holes in data quality the privatized dataset may contain, by introducing two additional components: the rank change penalty (RCP) as well as the trend penalty (TP).

In the pie chart metric, all of the data is normalized and records that contain less than k% (in this case we will be using 5%) are dropped before calculating the Jensen-Shannon distance (JSD) between the privatized and ground truth datasets. This value is then added to the Misleading Presence Penalty (MPP) which is a tunable parameter that adds a penalty (in this example we use 0.2) to the JSD every time there is a false positive in the privatized dataset (essentially when a record is 0 in the ground truth dataset and non-zero in the privatized dataset). Prior to normalization and dropping of insignificant counts, the Bias Penalty (BP) is calculated, which is another tunable parameter that is added to the JSD and MPP if the total record count in the privatized data is more than the Bias Penalty Threshold (in this example we use BPT = 500). The resulting metric uses the sum of these three parameters in the following expression: 1 - min(JSD + MPP + BP, 1) to get the score that describes the quality of the privatized dataset.

In the confusion matrix metric, the pie chart metric is altered to include the RCP along with the addition of a parameter separate to the categorical pie chart and RCP metrics known as the trend penalty. The TP looks at whether or not the time-series pattern present in the ground truth data is preserved in the privatized data. The ability to make decisions based on how data changes over time is extremely important, so this penalty is to be looked at separately from the evaluations done on the spatial data. The RCP penalizes the dataset for large changes in categorical rankings after privatization. Decision-making relies on prioritization of issues based on their importance, which can be expressed by ranking issues based on record count. Therefore, it is important that the metric penalize large ranking changes that render the dataset unusable for these purposes. An example of the metric utility is given below:

Ground truth: [36, 41, 0, 0, 0, 58, 0, 0, 33, 0] Privatized: [40, 43, 6, 7, 1, 58, 0, 11, 34, 7]

In this example, we assume each index in the array represents a unique category, and the values at each index represent the number of records per category. For these two arrays, JSD = 0.60 (for base = 2), MPP = 1.0, BP

= 0. To calculate RCP, we first create 10 bins (tunable parameter) of width max(array)/9 (also a tunable parameter) and then assign each value in the arrays to a bin. The bins for these two examples are:

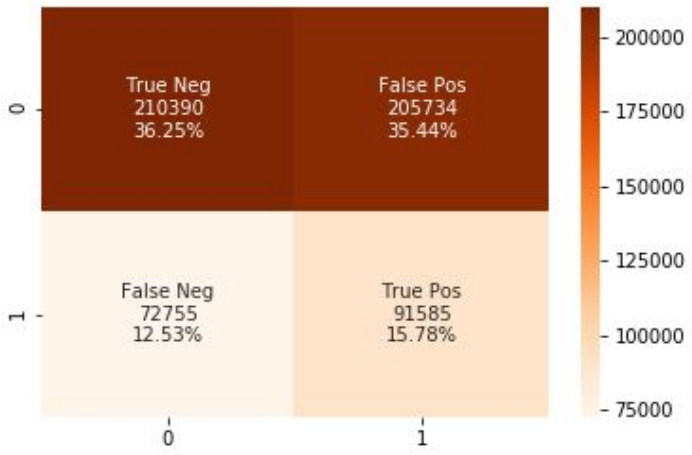
[ 0,  6.44, 12.88, 19.33, 25.77, 32.22, 38.66, 45.11, 51.55, 58, 64.44]

The numpy.digitize function assigns each value in the arrays to one of the bins in the array above and outputs an array of bin indices that each value in the arrays corresponds to. The output for this example was:

Ground truth: [ 6,  7,  1,  1,  1, 10,  1,  1,  6,  1]  Privatized: [ 7,  7,  1,  1,  1, 10,  1,  2,  6,  1]

From the output above, it is evident that the rankings changed for two values with the privatization algorithm. To penalize this, we multiply a tunable penalty (in this case, we use 0.1) by the number of values whose rankings changed and add this to the JSD, MPP, and BP and evaluate the metric using the expression 1 - min(JSD + MPP + BP + RCP, 1). In this case, RCP = 0.2, so the total score for this data comes out to be 0.43. This metric goes from 0 to 1, with 0 being unusable to 1 being a perfect match to the original dataset. To get a score for a dataset with multiple rows, this score is evaluated per row and then averaged over all rows.

To explore the quality in time-series patterns, we turn to a simple curve-fitting score, the most well known of which is r_squared. In particular, for a dataset with both map and time segments, we group by the map segments and sum the values of the rows to get one value per time segment per map segment. Doing this for both the privatized and ground truth datasets yields a total count per time segment for multiple time segments for a particular map point. To demonstrate this, we created a random 10x12 matrix that we then added a 10x12 matrix of a small amount of noise to, in order to get a pseudo-privatized matrix. This represents a set of 10 categories over a span of 12 points in time for one particular map point. Summing along the x-axis for these matrices yields two 1x12 matrices that in theory could be graphed to get a time-series look at the counts. For this particular example, the r_squared value turned out to be 0.95. For a dataset with many map points, this can be done for each and then averaged to get an average r_squared that can be used to determine the reliability of the time-series patterns in the privatized dataset.




The accompanying visualization for this metric, a confusion matrix, shows the categorical reliability of the privatized dataset. Essentially, it reveals the number of misleading presences caused by the privatization algorithm. For the poor quality data provided for the challenge, the matrix looks like the figure shown above. 0 stands for a category with a count of 0, while 1 stands for a category with a nonzero count. As shown, there are many false positives and false negatives in that dataset.

## Real World Use Case

Confusion matrices are often used in machine learning to quickly visualize the quality of the data output by an algorithm. That is very relevant to the current challenge, as it is important to be able to see how many misleading presences and misleading lacks in presence there are in the dataset. Having false positives and false negatives can greatly impact decision making in any sector so it is vital to be able to judge these aspects of a dataset.

# Metric Definition

## Technical Background

**Jensen-Shannon Divergence**
The Jensen Shannon Divergence is a way of measuring the distance between two probability distributions. The Jensen Shannon Divergence is a symmetric and finite version of the Kullback-Leibler divergence. In this metric, it will be used as a baseline distance between the ground truth and noisy datasets. This distance could be used on its own, however in this metric it is used in conjunction with a variety of other components to give a more comprehensive score to the data. This component was used in the pie chart metric created by Christine Task, Knexus Research Corporation Issac Slavitt, DrivenData Inc. More information about this and its implementation can be found here:

- Scipy Jensen-Shannon documentation: https://docs.scipy.org/doc/scipy/reference/generated/scipy.spatial.distance.jensenshannon.html
- Statistical explanation of KL Divergence and Jensen-Shannon Divergence: https://medium.com/datalab-log/measuring-the-statistical-similarity-between-two-samples-using-jensen-shannon-and-kullback-leibler-8d05af514b15

**Confusion Matrix**
The confusion matrix is a very useful visualization for comparison of two datasets. Often used in machine learning to compare the true data with the model-predicted data, it provides an intuitive way of looking at errors in a binary classification. In the case of the data provided for this challenge, one thing we want to look at during the privatization process is the number of false positives and false negatives in the noisy dataset, as these errors can significantly change any inherent patterns in the dataset. In this metric, we use the confusion matrix as a way to visualize these errors simply by denoting nonzero values with 1 and creating a confusion matrix from the resulting data.  More information on confusion matrices and their implementation can be found in the following:

- Seaborn heatmap documentation: https://seaborn.pydata.org/generated/seaborn.heatmap.html
- Explanation of confusion matrices for binary classification: https://medium.com/towards-artificial-intelligence/confusion-matrix-what-is-it-e859e1bbecdc

# Formal Metric Definition

In this section we provide a detailed definition of the confusion matrix metric in addition to steps for computation. This metric takes in two datasets: the ground truth dataset and the differentially private dataset. The structure of these two datasets should be that each row in the ground truth dataset should correspond to the same row in the privatized datasets. In addition, this metric is formulated to work only with datasets with counts of many events, as the rank change penalty and the misleading presence penalty would not work with datasets that have one event category and one count. In the toy example provided below, we show the structure of one record of the dataset that would work with this metric.

Ground truth: [36, 41, 0, 0, 0, 58, 0, 0, 33, 0]

The above shows the ground truth dataset. In the case of this challenge, it can be representative of the counts of 10 incident types in one particular neighborhood for one month in one year. The incident type is represented by the index of the count in the array. Upon privatization of this dataset, the resultant record for the same neighborhood in the same month in the same year is

Privatized: [40, 43, 6, 7, 1, 58, 0, 11, 34, 7]

Each index in the array above still corresponds to the same incident type as in the ground truth data. In the case of this example, a very small amount of positive noise has been added to the ground truth array to shift the values in order to approximate what a real privatization algorithm would do. As can be seen, the values are only slightly different from the ground truth data, and patterns appear to be somewhat consistent. However to measure their similarity, we introduce a few components: the Jensen-Shannon Distance (JSD), the Misleading Presence Penalty (MPP), the Bias Penalty (BP), and the Rank Change Penalty (RCP). The first three components are taken from the pie chart metric created by Christine Task, Knexus Research Corporation Issac Slavitt, DrivenData Inc. These components are then used to evaluate the following expression: 1 - min(JSD + MPP + BP + RCP, 1), which will give a score for how well the category-based patterns in the data have been preserved. In addition to this, the data will also be scored on how well patterns over time have been preserved, however we will discuss this later on.

The following details how to calculate the category-based score for the toy example given above:

1. **Zero insignificant counts and normalize the row:** for each record, set any count less than some frequency threshold (FT) k% of the sum of all counts in that record (in our case we used 5%) to zero and normalize the record.

   The resulting records after this are:

   Ground truth: [ 0.21428571, 0.24404762, 0., 0., 0., 0.3452381, 0. , 0., 0.19642857, 0.]

   Privatized: [ 0.21505376, 0.2311828 , 0., 0., 0., 0.31182796, 0., 0.05913978, 0.1827957, 0.]

2. **Calculate the Jensen-Shannon Distance:** for computation of this, we used the Jensen-Shannon function in the Scipy package with the ground truth record, privatized record, and base = 2 as parameters. The output of this came out to be 0.60.

3. **Calculate the Misleading Presence Penalty:** add a penalty to the Jensen-Shannon distance every time there is a nonzero count in the privatized data that is zero in the ground truth data. This is essentially penalizing for false positives. In our example, there is only one category with a false positive after privatization. We set our penalty (MPP) to be 0.2*number of false positives, so this comes out to be 0.2.

   Ground truth: [ 0.21428571, 0.24404762, 0., 0., 0., 0.3452381, 0. , 0., 0.19642857, 0.]

   Privatized: [ 0.21505376, 0.2311828 , 0., 0., 0., 0.31182796, 0., 0.05913978, 0.1827957, 0.]

   The highlighted values show the false positive.

4. **Calculate the Bias Penalty:** this is essentially adding a penalty (BP) (in our case 0.25) if the sum of the counts in the privatized dataset is more than a threshold value (BPT) (in our case we used 500) larger than the sum of counts in the ground truth data. In our example, there was no such difference so this value comes out to be 0.

5. **Calculate the Rank Change Penalty:** this component penalizes a change in ranking between categories. Essentially if a certain incident type had the largest count in the ground truth record but was overtaken by another incident in the privatized data, we add in a penalty of 0.1 every time this occurs. The way to accomplish this is to create a certain number of bins (B) that will encompass all of the count values (in our case we created 10) and multiplied the penalty by the number of values that changed bins. In the case of our example above, there are 2 bin changes so the RCP comes out to be 0.2.

   These are the bins we created for this example:
   Ground truth bins:
   [0.0, 0.04], [0.04, 0.08], [0.08, 0.12], [0.12, 0.15], [0.15, 0.19], [0.19, 0.23], [0.23, 0.27], [0.27, 0.31], [0.31, 0.35], [0.35, 0.38]

   Privatized bins:
   [0.0, 0.03], [0.03, 0.07], [0.07, 0.1], [0.1, 0.14], [0.14, 0.17], [0.17, 0.21], [0.21, 0.24], [0.24, 0.28], [0.28, 0.31], [0.31, 0.35]

   The indices of the bins each count is in:
   Ground truth: [6, 7, 1, 1, 1, 10, 1, 1, 6, 1]
   Privatized: [7, 7, 1, 1, 1, 10, 1, 2, 6, 1]

   The highlighted values show that two of the counts changed bins, which shows that their values changed by quite a bit.

6. **Evaluate the final expression:** we plug in all of our values calculated above into the expression 1 - min(JSD + MPP + BP + RCP, 1). This expression outputs values between 0 and 1, with 0 indicating a completely different dataset and 1 indicating a perfect match. The score for our example is 0.43, indicating that while many patterns have been preserved, some have been lost.

In the real use-case where we have two large datasets with multiple rows of similar nature to the example above, the expression is to be calculated for each row and then averaged to give an overall score for the entire dataset.

The second portion of this metric is the time-series evaluation, which is contingent upon a continuous time component being present in the dataset. In the case of this challenge, the dataset has the record counts and the neighborhood as mentioned earlier along with a month and a year. In our example, we look to preserve the pattern over the course of a year, so we sum the counts for 12 months and fit the privatized curve to the ground truth curve and calculate the r-squared value. In the case of our example, we provide 12 records to show how things look for one neighborhood over the course of a year:

Ground truth:

```
[[51., 18., 80., 40., 42., 34., 49.,  8., 24.,  5.],
 [74., 28., 26.,  1., 14., 32., 20., 90., 87., 66.],
 [13., 89., 34., 39., 24., 70.,  7., 22., 36., 57.],
 [33., 10., 44.,  4., 98.,  5., 50., 41., 36., 90.],
 [74., 80., 49., 96.,  8., 44., 95., 50., 91., 52.],
 [ 6., 36., 46., 94., 89., 94., 30., 57., 73., 82.],
 [49., 58., 22., 75., 75., 53., 95., 12., 14., 78.],
 [64., 20., 50., 35., 64., 85., 57., 71., 81.,  6.],
 [36., 41.,  6.,  7.,  0., 58.,  0.,  9., 33.,  0.],
 [91., 49., 68., 81., 78., 74., 36.,  8., 96., 22.],
 [99., 38., 83., 98., 96., 76., 85., 45., 74., 96.],
 [82., 23., 77., 75., 49., 76., 29., 30., 34.,  8.]]
```

Privatized:

```
[[ 53.,  24.,  83.,  44.,  48.,  37.,  51.,  11.,  26.,   7.],
 [ 77.,  28.,  28.,   5.,  21.,  33.,  23.,  92.,  88.,  68.],
 [ 16.,  93.,  41.,  43.,  27.,  70.,   8.,  24.,  36.,  57.],
 [ 35.,  16.,  45.,   6., 107.,   6.,  51.,  54.,  37.,  91.],
 [ 76.,  80.,  49.,  97.,  15.,  46., 101.,  51., 101.,  54.],
 [  6.,  36.,  48.,  98.,  92.,  94.,  39.,  57.,  80.,  85.],
 [ 55.,  59.,  23.,  81.,  81.,  57., 104.,  21.,  20.,  80.],
 [ 70.,  26.,  59.,  45.,  70.,  86.,  62.,  74.,  84.,  11.],
 [ 40.,  43.,   0.,   0.,   0.,  58.,   0.,  11.,  34.,   0.],
 [ 96.,  53.,  70.,  81.,  83.,  76.,  38.,  11.,  98.,  26.],
 [102.,  42.,  85., 103.,  96.,  78.,  85.,  46.,  76.,  98.],
 [ 83.,  23.,  82.,  76.,  60.,  76.,  34.,  35.,  34.,   9.]]
```

In order to get an r squared value, we first sum each vector in the matrix and plot the resulting curve for both sets. Then we fit the privatized data to the ground truth data and calculate r squared. In the example above, r squared is 0.95. This value is generally between 0 and 1 with 0 being a horizontal line and 1 being a perfect fit, however it can in some cases be negative if the fit is bad enough. Our value shows that the privatized data maintains the patterns of the ground truth dataset well. For a dataset with multiple neighborhoods and years, the r squared value will be averaged across all to get one score.

# Explanation of Metric Parameters

In this section, we go over all of the parameters involved in this metric. The first subsection details the parameters required for the metric to work, and the second subsection details the parameters that can be adjusted to change the output of the metric accordingly.

## Data Configuration Parameters

**Record Configuration**
This metric takes in a dataset with the structure of a set of counts across multiple categories for each map and time segment. This has to be the configuration in order for this metric to be effective in the spatial component. The main reason for this is that the Misleading Presence Penalty and Rank Change Penalty do not work unless each record corresponding to a particular map and time segment has more than one value, otherwise the penalties become extremely large. However, in the case of the time segment, the only two things necessary are any time component and any other value. In the case of datasets with demographic, neighborhood, and other factors for which the record type is not numerical, penalties will have to be adjusted based on the "difference" between each value type in the particular category.

## Tunable Parameters

The default values for the components taken from the pie chart metric created by Christine Task, Knexus Research Corporation Issac Slavitt, DrivenData Inc are also taken from this metric.

**Frequency Threshold (FT)**
In order to minimize unnecessary penalty, this metric first zeroes values that are "insignificant" in the record. The FT basically zeroes all values in the record that do not make up at least k% of the record total. The smaller k gets, the more the metric discriminates. However, this may not always be a good thing, as the larger the penalty gets the more large patterns that may have been preserved during privatization get overshadowed. Therefore, the penalty may not be indicative of how useful the dataset could be. In the above walkthrough, k=5.

**Misleading Presence Penalty (MPP)**
This component of the metric penalizes the data for false positives, which occur when a category has a value of zero in the ground truth data but is nonzero after privatization. The amount it penalizes can be changed. In the above walkthrough, it has been set to 0.2*the number of false positives in the record. A more detailed penalty that penalizes extra based on how much larger than zero the false positive is will be discussed below as the Rank Change Penalty.

**Bias Penalty Threshold (BPT)**
In addition to false positives, the addition of too much noise can alter the sum of all values in the record by so much that the data could be rendered useless. For this case, we introduce the bias penalty. This component penalizes the data if the record count for the ground truth data is BPT (bias penalty threshold) away from the record count in the privatized data. In the walkthrough above, we use BPT=500. The lower the value, the more harshly it penalizes an epsilon value that is too small.

**Bias Penalty (BP)**

The amount that the metric penalizes for going above the BPT is the BP. The larger BP gets, the more harshly the metric penalizes for the epsilon value being small. In addition, this will penalize very harshly if there are many zero values and man categories in the ground truth dataset. In the walkthrough above, we set BP=0.25.

**Bins (B)**
The component that penalizes the data for significant changes in ranking is known as the Rank Change Penalty (RCP). In this component, we create bins that encompass all of the values in the record, and keep track of which category value falls into what bin. If the bin for a category changes after privatization, we add on a penalty. The number of bins for this is a tunable parameter. In the walkthrough above, we use B=10. As the number of bins increases, the more harshly the component penalizes changes, as the bin sizes become smaller and therefore more sensitive to small changes in value with noise.

**Rank Change Penalty (RCP)**
This component utilizes (B) above in order to penalize significant changes in the ranking of the category counts in a record. For example, if a certain category is the most frequent, and another is far less frequent, the RCP will penalize if the two are swapped. As mentioned above, the more bins are used the more harshly this component penalizes for changes in value. In order to compute the penalty, we multiply the number of values whose bins changed by 0.1. This is also a tunable value, and the larger it gets the more sensitive this component gets to value changes by noise.

**Time Segment**
The only tunable parameter that the time-series component of this metric takes in is the number of months/time segments to sum over. In the above walkthrough, for one neighborhood we sum values for each month and then plot those values for 12 months and fit the curves. The number of time segments is a tunable parameter, and the more points that get plotted and fit the more accurate the fit is likely to be, as r squared generally tends to be larger as the curve gets smoother which occurs with more points.

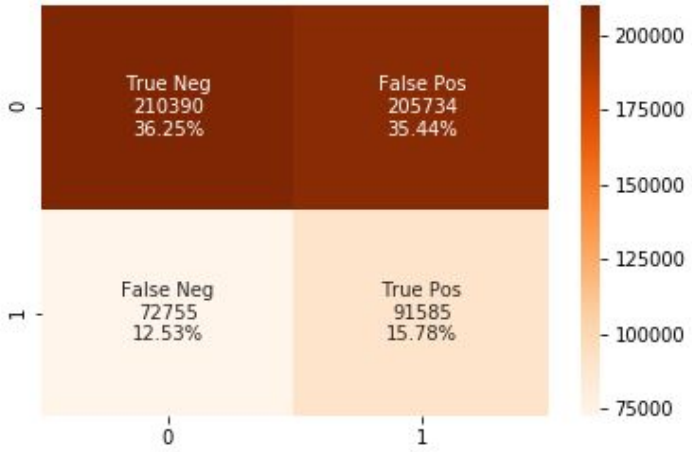
## Snapshot and Deep Dive

This section details how the metric can be used either to give one overall score or to create visualizations that show details of the patterns in the data.

### Snapshot
This metric has two components: a categorical or spatial component, and a time component. The categorical/spatial component is reliant on the data having multiple categories with counts in each and gives a score between 0 and 1, with 0 being an unusable dataset and 1 being a perfect match. The time component is essentially a best fit analysis, and gives an r squared score for the data over a segment of time. This part is usually between 0 and 1 but in cases of really bad fitting can be negative. To get an overall score for a complete dataset, an average of the scores can be taken. This can be done for both the categorical and time components.

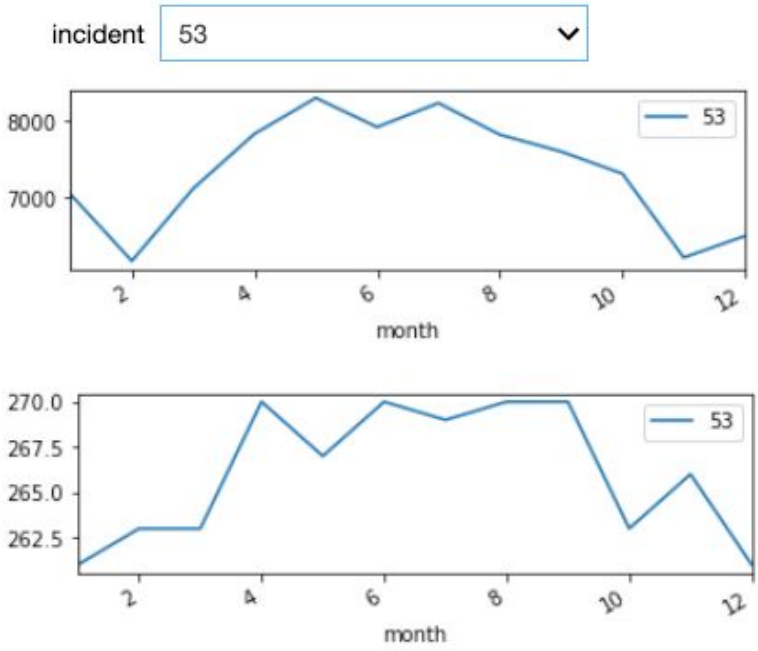
### Deep Dive

This metric can also be used to look at specific parts of the data in order to get a more specific comparison. The below visualizations show some examples of how the data can be looked at in order to get a deeper view.




One good way to get a gauge on how well the privatization algorithm preserved patterns in the data is to use a confusion matrix. For a deeper dive into the data, a confusion matrix can be made for each neighborhood to show if certain neighborhoods tended to have more false positives and false negatives than others. This could show a pattern between what neighborhoods tend to be more harshly altered by the privatization algorithm than other neighborhoods.

Another visualization we can use to take a deeper look into the time aspect of the data is the interactive fitting visualization. This particular visualization makes use of the time component in the records and shows what kind of fit there is between the incidents over the course of a year. This shows another way to calculate the time component of this metric: to calculate the r squared over a year for each incident and then average all of the incidents. As can be seen from this particular visualization, some fits are decent, while others are quite different. In addition as the amount of noise varies, the shift in the incident counts becomes extremely large, however looking at the overall pattern in this way can help with discerning whether or not some pattern was preserved after privatization. From this it can also be seen that depending on the amount of noise, in order to get an r squared value that measures solely



the pattern fit it may be necessary to add an amount to the ground truth data in order to get it close to the privatization data.

# Metric Defense

## Exploration of Parameter Tuning

### Data

The dataset used for this metric is the 2019 Baltimore police incident dataset provided by the challenge. This data has two levels of noise added to it: epsilon = 0.5 for the poor quality dataset and epsilon = 4 for the mediocre quality dataset. Smaller epsilon indicates more noise. The dataset is of the structure of a few time components, at least one spatial component, and multiple categories with counts.

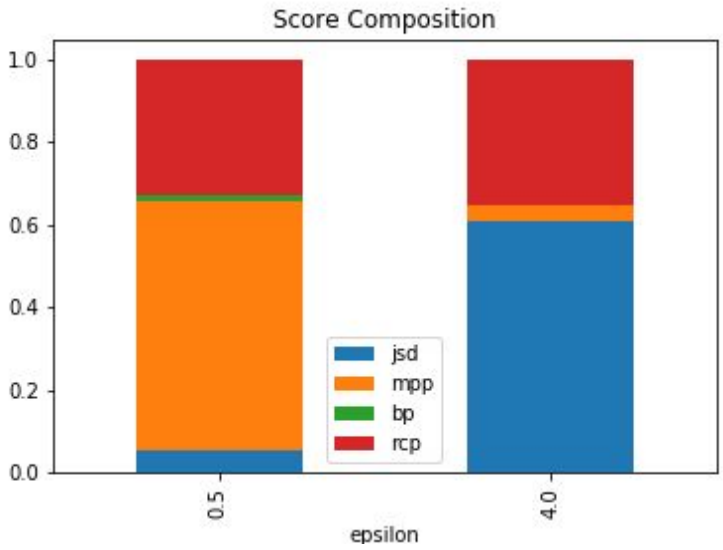
### Metric Parameters

All metric parameters in the above walkthrough use the default values mentioned in the explanation of metric parameters section above.

### Score Composition

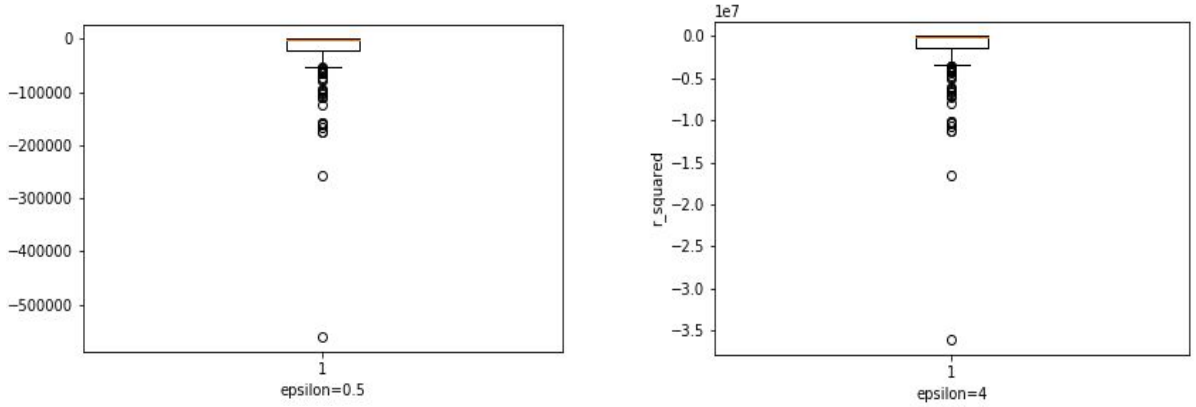
This metric is split into two large components: the categorical/spatial component and the time series component. The first component is made up of the JSD, MPP, BP, RCP. The second component is made up of only r squared.

In the case of the categorical/spatial component, the score composition changes quite a bit depending on the level of noise added to the data. For epsilon=0.5, which has more noise added to it, we can see that the majority of the score is determined by the misleading presence penalty. Essentially this shows that there are lots of false positives and false negatives (instances of a category having a nonzero value after privatization despite having a zero value before) in the privatized dataset. In addition, we are also seeing some occurrence of bias penalty being added, which shows that there are a few instances wherein the record total after privatization is 500 more than the ground truth dataset. The default value of 500 is already extremely large and it shows that there is an almost unusably large amount of noise added to this data. The rank change penalty makes up the next largest portion, which is also the case in the mediocre quality data.



The score composition for the mediocre quality data (epsilon=0.5) is largely composed of the Jensen-Shannon Distance. In addition, there are definitely far fewer false positives in the mediocre quality dataset in comparison to other differences, so the misleading presence penalty makes up a very small portion of the score. The rank change penalty however has stayed the same and makes up a decent portion of both scores, which could be a sign that given the default number of bins used (10), there is a large amount of value change going on regardless of noise. This could also be because the penalty does not scale with how much the value changed, which can be adjusted later on.

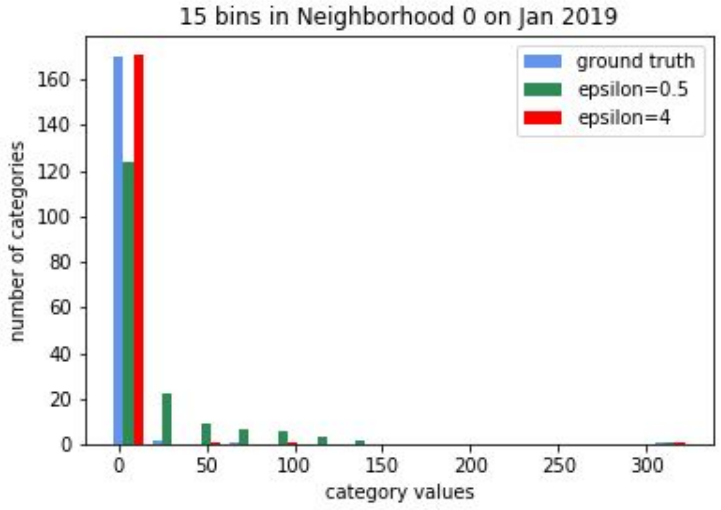
In the case of the time component, it can be seen by the plots below that the added noise changes the pattern across time quite significantly. If two graphs have any amount of correlation, usually the r squared value is going to be between 0 and 1. However, in this image it can be seen that there is a large number of negative values that are present, which indicates that the fits are even worse than a fitting with a horizontal line. The method of averaging done for this example is done by grouping the dataset by

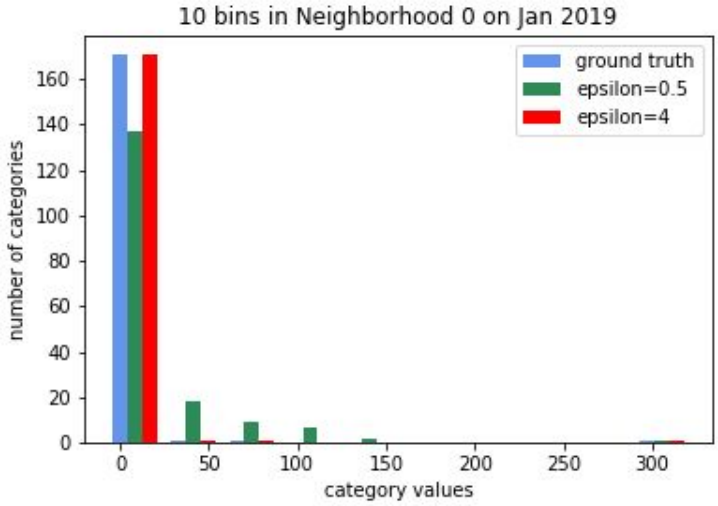


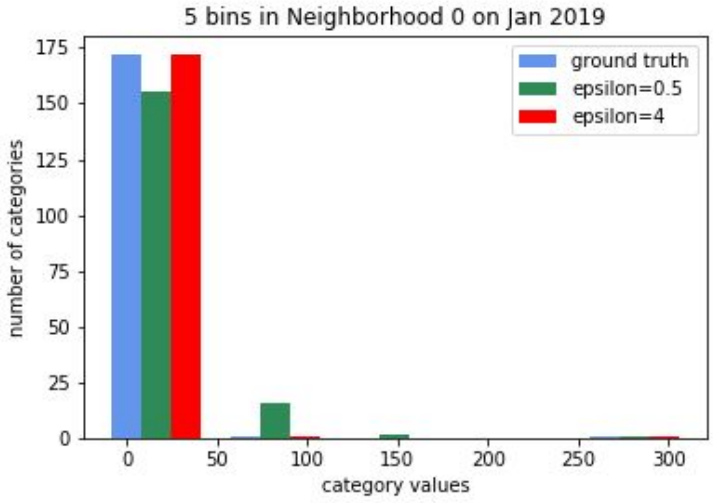

months, fitting each incident, and averaging the r squared values over all of the incidents. As can be seen from these images, the privatization algorithm alters the fitting by quite a bit, as there are quite a few r squared values below zero. In the case of epsilon=0.5, the smallest r squared value is -500000 which is already quite bad, however in the poor quality dataset (epsilon=4) the lowest value is -3.5e7 which is extremely low. However, it is evident that these scores are not necessarily reflective of the overall preservation of the pattern as can be seen by the second figure in the deep dive section. This figure shows the plot for incident 53 from both the mediocre quality and ground truth datasets and it appears as though they both have a somewhat similar shape. However upon looking at the y-axis, it is obvious that the bad r squared scores are due to the grossly large shifts in values, as the maximum value in the ground truth dataset is 270 while in the mediocre quality dataset it is 8000. In order to get the r squared value to weight pattern more than shift, it may be necessary to shift the values of one of the datasets to somewhat match the other.

## Tuning the Bins

We now explore the effects on the rank change penalty of altering the bins from their default amount (10 bins). In addition, we will discuss two ways of binning that may impact scoring of the data as well.




The method of binning used in the walkthrough above involves first zeroing “insignificant” values and normalizing the record, which leads the maximum possible value of the data to be 1.0. Our default number of bins is 10, however we get very different results if that number is changed. From the images to the left, it is evident that the number of differences in values increases as the number of bins is increased.

All of the images depict the incident landscape in neighborhood 0 on January 2019. The topmost image shows a histogram of incident values with 15 bins. Looking at the image, we can see that the poor quality dataset’s bars (green) vary significantly from the ground truth dataset (blue) while the mediocre quality dataset’s bars (red) are much closer to the ground truth data. There are 7 different bars for the poor quality dataset that differ from the ground truth, thus there is more penalty to be added to that dataset. In contrast, in the middle image, which uses 10 bins, there are only 5 poor quality dataset bars that differ from the ground truth dataset, which will decrease the penalty significantly. In a similar fashion, for the histogram with only 5 bins, it can be seen that the number of poor quality dataset bars that differ from the ground truth is only 3, and so the penalty is decreased there as well. Essentially, as the width of the bins increases, the chances for noise addition to cause values to

change bins become smaller. From these images, we can see that depending on the use case, the bins can be adjusted according to necessity. For instance, if only extremely large changes in value are to be penalized, then the number of bins can be decreased in order for the metric to be of better utility for that situation.

## Time Series Averaging Method

One additional interesting property of this metric is that it can change utility depending on the method of aggregation. Our default is to average the results across various segments, however the method of averaging can also play a role in how the metric scores the data. The component that is most significantly affected by this is the time-series component.

There are two different methods with which we can average across fitted data. One method is to group the data by month and sum all of the incidents individually across the months, fit each incident column to the ground truth data and average the r squared across all incidents. This method seems to produce the worst r squared values. This is also the method used in the walkthrough section of this paper. Using this method, the average r squared value of the poor dataset is -1,604,000, which is extremely low. The average r squared value of the mediocre dataset is -24,742, which is also extremely low but not anywhere near the poor quality dataset. These extreme values could be attributed to the fact that they are reliant on how the values of each incident have been changed, as we plot the behavior of the total number of a certain type of incident over the course of a year. It could be said that the privatization algorithm harshly affects the pattern of the total count per incident.

By contrast, the method that averages total counts by neighborhood produces results that are not nearly as low as the method above. This method sums all incidents per neighborhood and calculates the r squared value. The r squared is then averaged across all neighborhoods. With this method, the r squared value for the poor dataset is -53,142, which is not nearly as low as in the previous method. Similarly, the r squared value for the mediocre dataset is -790 which is a large improvement on the score from the previous method. It suffices to say that all of these r squared values are quite bad, however as discussed earlier, a lot of this is attributed to the large shift in values and may not be reflective of whether or not the overall pattern was preserved.

## Further Questions

In addition to what was addressed, there are many other aspects to this metric that could be analyzed. Some of these are:

- How does the metric penalize large changes in value as the RCP number changes? How can the penalty be altered to reflect the size of the change?
- Is averaging all of the results the best way to go about getting a unified score for the metric? Is there a better way to aggregate the results?
- Is there a way to combine the time series portion and the spatial portion of this metric in a way that effectively reflects the quality of the data?
- How can this metric be altered to better serve a specific purpose? For instance, if a user values the overall pattern in the time series more than scale, how can the metric give a score meaningful for this case?
- How does the metric score datasets that have extremely sparse values? Can it still give a meaningful score despite the high likelihood of false positives?

- How does this metric score data with very few categories? Currently the dataset it was tested on had 173 individual categories, does it get more or less accurate as this number decreases?

## Description of Discriminative Power

How well does this metric distinguish between the ground truth dataset and the privatized dataset?

### Capabilities

- This metric penalizes large value changes of one category in comparison to other categories in the same map-time segment (it prioritizes ranking).
- This metric has the capability to score datasets in how well they preserve time-series patterns.
- This metric penalizes false positives in the privatized data, so in the case that it is necessary to have data with mostly true values this metric will be very useful.
- This metric captures the level of added noise well in its scoring.
- This metric penalizes shifts in values that are large enough to render the dataset useless if the use case is reliant on specific values in the dataset.

### Limitations

- This metric currently does not have a way of comparing only the shapes of the time-series curves while ignoring large shifts in value.
- The metric may penalize false positives too harshly depending on how small the value change is in comparison to the rest of the record.
- This metric has two separate components: the map segment and the time segment. They are not combined to give one score.
- The metric's rank change penalty gives one penalty regardless of how large the value change is, which could be detrimental when there are not very many bins.
- This metric does not currently have a way to score for machine learning and other types of processes that are affected by noise in different ways.

## Description of Discriminative Power

How well does this metric cover a variety of possible use cases?

### Capabilities

- The confusion matrix is an intuitive way to quickly visualize the quality of the data from a binary classification perspective. It allows the user to quickly see what proportion of the data after privatization resulted in false positives and false negatives and what proportion remained the same.
- The metric gives a somewhat accurate reflection on how well the overall spatial/categorical patterns have been preserved after privatization due to its use of the various components.
- The time component in the metric is highly discriminative, so in the event that the metric needs to reflect how well the privatization algorithm preserves both scale and overall shape of the curve, the score will be accurate.

- For the case that the metric be used for only one of the components (either spatial/categorical or time), the scores will reflect according to necessity since the two components are not combined.

### Limitations

- In the case that the metric be used for only detecting overall time-series pattern preservation, in its current state it will come up short as r squared penalizes heavily for large changes in scale. In this case, a correlation coefficient may be of more use.
- In the case that a user may want one score for each dataset, the metric currently does not allow for that as it does not combine the spatial and time components into one score
- This metric currently does not accommodate datasets with non-numerical values very well, as the rank change penalty is heavily reliant on numerical binning. In addition, r squared cannot directly be computed on non-numerical data.
- For datasets with sparse values and/or very few categories, the metric may penalize too harshly for inconsistencies as a lot of the metric is reliant on relative changes within each record

## Scalability

This metric was evaluated on multiple 3336 x 178 datasets on an average computer and took a few seconds to run. The time component ran in very few seconds, while the categorical component took a couple of seconds longer likely due to the binning process.

## Generalizability

This metric can also be applied to the following cases:

- The categorical component of the metric can be used to evaluate patterns in datasets outside of the police data, including record types such as poverty, demographic, sex, and education.
- The time component of the metric can be used to score financial data as well, including income earned and income total.
- By adjusting the bins in the rank change penalty, the metric can be used to score value changes in all of the categories mentioned above.