Tabular Data Science course

Final Semester Project report

Eva Hallermeier (ID. 337914121), Amebet Belachew (ID. 324488287)

March 12, 2023

1 Abstract

As part of the data analysis process, the data analyst needs to analyze his dataset and in particularly all his features. A feature can have many characteristics and if the feature is numerical, it has a distribution: how the data is distributed in a specific range of values. A way to visualize this is using an histogram.

The project's goal is to automate and improve data visualization of histogram, in order to assist the user in performing a better and easier data science process by developing a method that find the optimal number of bins for a given numerical feature.

We first observed the results of existing methods, some of them work pretty well but they often overestimate the number of bins and propose a too noisy histogram. In addition, any of the existing method use the number of unique values in a feature which is a significant information that can be used.

The method we develop select the minimal value proposed by the different existed methods and take also in consideration the number of unique values. After defining the method, we evaluated it by applying it on different datasets. We got nice results, better or equivalent to the default method of ten bins.

2 Problem description

2.1 Subject of the project:

We focused in the Data Science pipeline on improving the visualization of distributions of features in Exploratory Data Analysis (EDA).

In class, we learned about Exploratory Data Analysis and in particularly about distributions and we saw histograms as a tool for visualizing data and understanding the probability distribution of a feature and identifying clusters or trends within the data.

2.2 Problem:

However, a good visualization relies heavily on the parameters, such as the number of bins. Today, it's hard to get a valuable histogram because the user doesn't know which parameter to choose to visualize his histogram. In fact, the number of bins in the histogram is a critical parameter that needs to be considered. This parameter is either determined by default or left up to the user to choose, but there is no guarantee that the user will make the optimal decision for the value of this parameter. This can lead to histograms that are either too noisy or too simplistic to accurately represent the distribution of the data. In fact, if the number of bins in the histogram is too small, there may not be enough detail, while too many bins can lead to noisy or confusing results. Few methods already exist but they not always give good results and who really know which method to choose?

So we can see here that there is a problem by deciding on parameters of visualization in distribution of data. Our idea is to apply a method that will return a number of bins for a specific feature and that will work for all numerical features type and distribution and size without specific information.

2.3 Important points:

- In this research, we focused only on equal bin width and numerical features not binary.

- In matplotlib library, the default value for the number of bin is 10 : we will see that 10 is not all the time the best option as number of bins.

- Numpy also propose "auto" value as methods which is the maximum of the 'sturges' and 'fd' estimators. Numpy describe this method by the following expression: "provides good all around performance."

2.4 About existed methods:

- Details about existed methods can be found here: `https://numpy.org/doc/stable/reference/generated/numpy.histogram_bin_edges.html#numpy.histogram_bin_edges`

- Sturges is the most common method used

- We observed that even the 'auto' method is not so good so our goal is to elaborate a method better than 'auto'.

3 Solution overview

3.1 Observations

We wrote one notebook that shows and explains the research phase where we observed few features of different datasets with existed methods that compute number of bins for the histogram and we criticized the results.

We took only numerical features - integers and float values, from different distributions, from small and large data sets in order to over as many types of data as possible. We applied few existing methods to see if they are good and checked which value seems optimal. In addition, we checked if 10 bins - the default value of matplotlib library is really a viable option.

During the research phase, we wanted analyze features with different distributions. In order to check it, we used the qq-plot that we learned in class, this tool permit to see the level of fit between our feature distribution and the theoretical distribution which helps us verify if distribution we thought was the correct one. As a start, in the research phase we used human eyes to examine whether a number of bins creates a good histogram or not. Later, we validated it using a metric which use a train and test datasets.

Symbols used to describe methods:

- a for 'auto'
- f for 'fd'
- d for 'doane'
- sc for 'scott'
- sq for 'sqrt'
- sto for 'stone'
- r for 'rice'
- stu for 'sturges'

Findings of observations:

feature	distribution	data set size	good bins	best methods	worst methods
battery power	uniform	2000	13-19	a,f,d,sc,stu	sto,r,sq
fc	exponential	2000	13-21	a,f,d,sc,stu	sto,sq
int memory	uniform	2000	13-20	a,f,d,sc,stu	sto,sq,r
m dep	uniform	2000	10-20	a,f,d,sc,stu	sto,rice,sq
mobile wt	uniform	2000	13-14	a,f,d,sc,stu	sto
housing median age	normal	17000	15-20	d,stu	sto,sq
latitude	bimodal	17000	17-33	a,f,d,sc,stu	sto,sq

The problem we tried to solve is selecting the proper number of bins of histogram in order to get a good visualization of the distribution of the data. The project also asked how do you know that a histogram represents the data well?

Nowadays, it is still difficult to define the quality or 'goodness' of a histogram. In our experiment we evaluate the quality of histogram by how much it allows us to infer the general distribution of the data.

3.2 Solution

Based on our observations, we develop a method that automatically select the number of bins for the histogram for a feature in a dataset.

The number of bins need to be smaller than the number of different values in the feature. Otherwise we will get really thin bins and many holes which create a useless histogram.

Code of the solution: receive a dataset and a name of feature and return a number of bins with the formula chosen:

```
method_list =['auto', 'fd', 'doane', 'scott', 'sturges']

def getNbBins(df, feature):
  unique_values=df[feature].nunique() #nb of different values
  bins_list = []
  for m in method_list:
    nb_bins = len(np.histogram_bin_edges(df[feature], bins=m))
    bins_list.append(nb_bins)
  min_value = min(bins_list)
  if min_value < unique_values:
    i = bins_list.index(min_value)
    min_method = method_list[i]
    return min_value, min_method
  else:
    return unique_values, "unique_values_amount"
```

Based on the observations, we saw that almost always, the bins size that was a good choice is the smallest number among the numbers of the various existing methods.

Explanation of the method:

First, we defined that the initial number of bins is the number of unique values in the feature received. The reason is that during the research phase we noticed that there were methods whose number of bins is sometimes more than the amount of the unique values, which does not make sense. Then we selected the most successful methods from the observations: 'auto', 'fd', 'doane', 'scott', 'sturges' and then we select the method that return the smallest value.

If this value is bigger than the number of unique values, so we returned the number of unique values and otherwise we return the smallest values from the existing methods we selected.

4 Experimental evaluation

The idea of the evaluation is to see if the histogram created by the method in the train set will be similar (in term of partition) to the histogram of the test set. To evaluate effectiveness of our methods, we will compare histograms of train and test set (with same number of bins returned by our method) by comparing the partition by bins in percentage and we will use a metric that compute the average of the differences between the two histogram in term of percentage of data in each bin. This value will be compared to a threshold we define (set to 0.01). If the average of differences is too high: this means that the value for bins number is not good.

We evaluate our method on four different datasets and few features in each one. We applied the method on different features, and evaluate the results using the metric. In addition we compared our results to the default method of always using 10 bins.

For each feature we did the following steps:

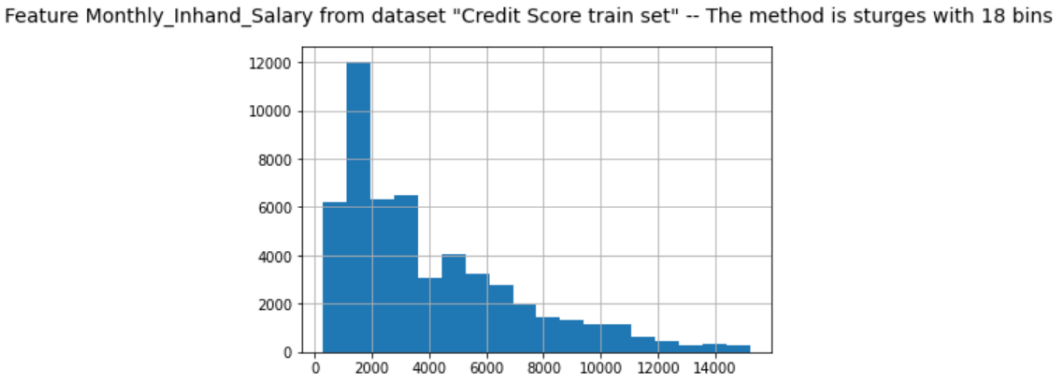
1.First, we divided the data into train and test.

2.We invoke our method on the train data and get number of bins.

3.We created histogram for the test set with the same number of bins and check if we have indeed a similar behavior/distribution.

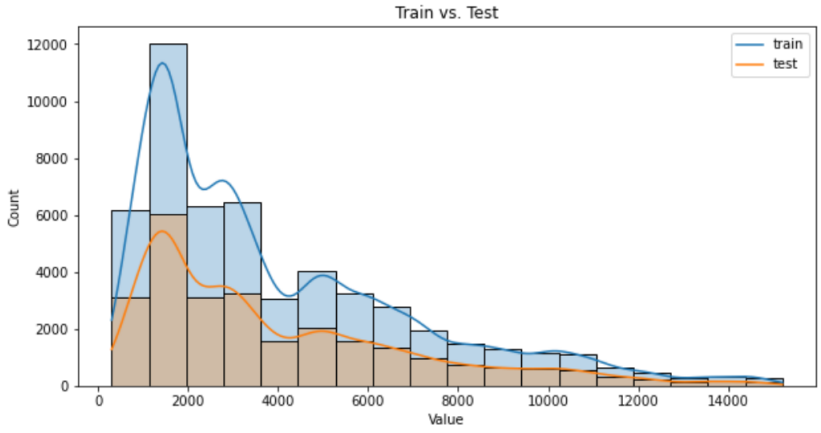
3.1.We compute the value of our metric and compare it to the threshold. If the value is smaller than threshold: the value return by the method is good otherwise there is too much difference between the histogram of train and test set and the method is not good.

Example of a complete evaluation with the feature “montly inhand salary” from the dataset Credit Score 1- invocation of the method and show histogram with the number of bins received for the train set



The bins number according to our method is 18.

Let’s see the histograms of train and test together.

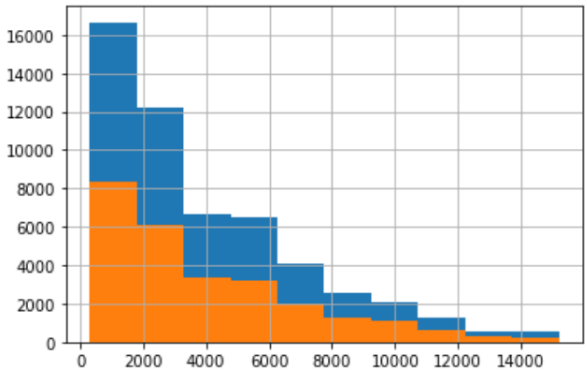


They are really similar. Now, let’s compute the mean difference between train and test:

Difference between histograms: 0.00083. The difference is below the threshold 0.01 that we set

So the method is effective for this feature.

We also did the same evaluation with 10 bins (default method) to check if our method give better result than the default method.



We resume the entire evaluation with four datasets in this table: comparison between our method and the default method:

dataset	feature	bins	mean difference our method	mean difference bins=10
Credit score	Monthly Inhand Salary	18	0.00083	0.00079
Credit score	Credit Utilization Ratio	18	0.00853	0.01488
Loan prediction	Funded Amount Investor	19	0.0063	0.00993
Loan prediction	Debit to Income	19	0.00535	0.00961
Iris flower	Sepal width	9	0.03	0.03
Wine dataset	alcohol	9	0.04	0.05

In the majority, our method produce a histogram that show a distribution closer to the test distribution. So we can say that our method is pretty successful.

5 Related work, discussion

We observed that existing methods based their computation on the number of the samples of the dataset ('sturges': log(n), 'sqrt': square root of n, etc). While our metric focused on the unique values size. In case of large dataset, those methods will give histogram with too much bins.

In addition, our method is similar in its technique to the existing method 'auto' which compute the number bins as the maximum of the 'sturges' and 'fd' estimators while we use minimum and with more different methods.

Some methods build histogram with not equal bin width, it can improve the visualization but we didn't experiment this type of histograms.

6 Conclusion

We observed several things in this project

- In general, a histogram will show understandable distribution with number of bins between 5 and 20.

- Choose 10 as number of bins is not always the best option, in particularly when there are many different values

- The best existing methods based on our obeservations are: auto, doane and sturges.

- Our method frequently chose the value from the 'sturges' method which is the most popular method.

- Method sqrt and Stone overestimate the number of bins and give too noisy histogram.

7 Github repository

link to the github repository:

`https://github.com/evaHallermeier/TabularDSProject_improve_histogram_visualisation`

Notebook 1 - Observations on different data (optional)

`https://colab.research.google.com/drive/1OTBbwNK72qAlZzHjouNPjd2iwRHjqYTU?usp=sharing`

Notebook 2 - Our method, application and evaluation (main one)

`https://colab.research.google.com/drive/1oKpaSYUkLNx1hXJHwsrKjabDS5xCW3mp?usp=sharing`