# Automatic Categorical Feature Binning

Yarin Shaked and Eden Shuker

February 2023

**Abstract**

Categorical feature binning is a data preprocessing technique used to transform a categorical feature into a numerical feature by dividing the original feature into several bins or categories. Categorical feature binning can help improve model performance by reducing the complexity of the feature space and capturing meaningful patterns in the data. Manually binning all the categorical features in your dataset can take a long time, which could be invested in other areas of the data science pipeline. We developed an automatic tool that finds binning for nominal categorical features for classification tasks that aim to maximize the correlation of the categorical feature with the target class. We illustrate the empirical effectiveness of our tool on multiple classification tasks in terms of the accuracy metric.

## 1 Problem Description

Binning is a technique used to group a set of continuous or numerical data into a smaller number of discrete "bins" or intervals. This technique is often used with categorical data, which can be difficult to work with in machine learning and statistical models due to its non-numerical nature. Binning a categorical feature involves dividing the data into a specific number of groups or bins and assigning each data point to a bin based on its value. This can help to simplify the data and make it more manageable for analysis, while also reducing the dimensionality of the data and making it easier to visualize.

There are two types of categorical variables - nominal and ordinal. Nominal features are a data type that is purely descriptive, they don't have any quantitative or numeric value, and the various categories cannot be placed into any kind of meaningful order or hierarchy. For example, a variable indicating the

color of a car (red, blue, green, etc.) is a nominal variable because the different colors do not have a natural order or ranking. Ordinal features are a data type where the variables have natural, ordered categories variable. For example, a variable indicating the level of education (high school, associate degree, bachelor's degree, etc.) is an ordinal variable because the different levels of education have a natural order or ranking. The categories can be ranked, but not always the distance between them is equal.

The process of binning categorical data requires an acquaintance with the categorical variable, deciding on the number of bins, and assigning each data point to the appropriate bin. This can be a tedious and repetitive routine, especially when the dataset contains multiple categorical variables. This draws a lot of time from the data scientist which could be invested in more difficult parts of the problem. In our project, we developed an automatic tool to find the best binning for your specific classification problem.

## 2 Solution Overview

We have focused on binning nominal categorical features for classification problems. Our approach is to recursively try to split the categories into two bins if it improves the binning score. The binning score is a measurement we defined that represents the correlation between the categorical variable after the binning and the target variable. Correlation is a statistical measure that describes the relationship between two variables. A high correlation between a feature and the target variable can be an indicator of its importance in predicting the target variable.

There are different types of correlation measures suitable for different types of data: continuous, categorical, binary, etc. We chose the uncertainty coefficient (also called entropy coefficient) as the correlation measure which represents the binning score in our project. The uncertainty coefficient is a measure of the association between two categorical variables in a contingency table. It ranges between 0 and 1, and it measures the proportion of the total association between the two variables that cannot be explained by the marginal frequencies of the two variables. A value of 0 indicates that the two variables are independent, meaning that the value of one variable does not provide any information about the value of the other variable. A value of 1 indicates that the two variables are perfectly associated, meaning that the value of one variable completely deter-

mines the value of the other variable. The Uncertainty coefficient is calculated using formula: U = (H(X,Y) - H(X—Y) - H(Y—X)) / H(X,Y), Where H(X,Y) is the joint entropy, H(X—Y) is the conditional entropy of X given Y, and H(Y—X) is the conditional entropy of Y given X.

We developed a program that receives a train data set, the target feature, and the categorical feature to calculate the binning for. First, the program calculates the contingency table of the categorical variable and the target (also categorical) variable, where the rows of the tables are the values of the categorical feature, and the columns are the values of the target class. As mentioned before, the uncertainty coefficient calculates the correlation between the two variables in the contingency table. Binning multiple categories into one is taking their corresponding rows in the contingency table and totaling them into one row.

Our algorithm 1 starts with creating one bin for all categories and calculates its binning score. It iterates over all possible splits of that one bin into two bins. If there is a split that gives a higher binning score than that one bin containing all categories, we recursively try to split these two bins into smaller bins. We recursively compare the binning score after applying the binning of the two smaller bins with the binning score of the bin before the split. Our stop condition is when there is no split of the one bin which will result in a higher binning score.

**Algorithm 1** FindOptimalBins

**Require:** categories, contingencyTable
**Ensure:** optimalBins

```
oneBin ← [categories]
oneBinScore ← GetBinningScore(oneBin, contingencyTable)
maxTwoBinScore ← 0
bestBins ← null
for twoBins in getAllPossibleSplits()
    twoBinScore ← GetBinningScore(twoBins, contingencyTable)
    if twoBinScore > maxTwoBinScore
        maxTwoBinScore ← twoBinScore
        bestBins ← twoBins
    end if
end for
if maxTwoBinScore < oneBinScore
    return oneBin
else
    firstBin, secondBin ← bestBins
    firstBinOptBin ← FindOptimalBins(firstBin, contingencyTableAfterApplyingSecondBin)
    secondBinOptBin ← FindOptimalBins(secondBin, contingencyTableAfterApplyingFirstBin)
    return [firstBinOptBinn, secondBinOptBin]
end if
```

# 3 Experimental Evaluation

## 3.1 Datasets

We tested our automatic binner on the following datasets:

**Churn Modelling** [2] This data set contains details of a bank's customers and the target variable is a binary variable reflecting the fact whether the customer left the bank (closed his account) or he continues to be a customer. We tested our binner in the categorical feature "Tenure".

**Titanic** [4] This data set contains details of the passengers in the Titanic disaster. The target variable is the outcome of each passenger. Here we tested our binner on two categorical features: "SibSp" the number of siblings/spouses aboard the Titanic, and "Parch" the number of parents/children aboard the Titanic.

**Home Credit Risk** [3] This data set contains details of people's applications for loans. The target variable is binary, whether the applicant is capable of repaying a loan. We tested our binner on the following categorical variables:

"NAME_INCOME_TYPE", "NAME_EDUCATION_TYPE", "NAME_FAMILY_STATUS" and "WALLSMATERIAL_MODE".

**Banking** [1] This data set contains bank clients' data. The target variable is binary, and the mission is to predict if the client will convert or not. We tested our binner with the "Education" variable.

### 3.2 Evaluation Method

The end goal of the feature engineering process and specifically binning is to improve the accuracy of a given model. In order to evaluate our automatic binner, we check for an improvement in the accuracy of a model as a result of the binning application.

We wrote a generic classification model which gets the data set and the target class and trains on these. Our program split the data set into three sets: train, evaluation, and test. The training and evaluation sets are for the training phase, the test set is for the accuracy evaluation.

For each database and categorical feature, we mentioned before we took the following steps:

1. Trained a classification model and calculated its accuracy.
2. Found a binning for the categorical feature with our automatic binner program.
3. Applied the binning on the categorical feature, trained a classification model on this new data set, and calculated its accuracy.
4. Compare the accuracy of the model before and after the binning.

### 3.3 Results

Table 1 summarizes the results of applying our automated binner on various datasets. As can be seen, our automatic binner resulted in an improvement in the accuracy of the classification model in three out of four of the reviewed databases, while in the fourth it saved it as is. For the "Churn Modeling" data set we got an improvement of 0.52% thanks to the binning of the "Tenure" column. For the "Home Credit Risk" data set we got an improvement of only 0.03% for the income type feature. However, if we delve into the partition to bins it seems very intuitive. The possible categories are Businessman, Commercial

| Dataset | Categorical Feature | Score without Binning | Optimal Binning Model Score | # Categories | # Bins | Total Time (seconds) |
|---|---|---|---|---|---|---|
| Churn Modeling | Tenure | 83.76 | 84.28 | 11 | 6 | 7.62 |
| Titanic | SibSp | 80.717 | 81.614 | 7 | 4 | 0.73 |
| Titanic | Parch | 80.717 | 81.166 | 7 | 4 | 0.77 |
| Home Credit Risk | NAME_INCOME_TYPE | 91.869 | 91.908 | 8 | 5 | 2.87 |
| Home Credit Risk | NAME_EDUCATION_TYPE | 91.869 | 91.888 | 5 | 3 | 0.91 |
| Home Credit Risk | NAME_FAMILY_STATUS | 91.869 | 91.878 | 6 | 4 | 0.657 |
| Home Credit Risk | WALLSMATERIAL_MODE | 91.869 | 91.894 | 7 | 4 | 1.78 |
| Banking | Education | 89.512 | 89.512 | 8 | 5 | 0.83 |

Table 1: Results of applying our automated binner on various data sets

associate, Maternity leave, Pensioner, State servant, Student, Unemployed, and Working. The resulting bins are:

1. Student, State servant
2. Pensioner
3. Businessman, Commercial associate
4. Maternity leave, Unemployed
5. Working

The grouping of *Maternity leave* and *Unemployed* together, and *Businessman* and *Commercial associate* together, makes a lot of sense in the context of credit risk.

However, there are tests on other data sets and categorical variables we did not necessarily reach an improvement in.

Regarding the time complexity of our algorithm, though in the worst case, it is exponential, in practice we converge to good results very fast (as can be seen in the table of the results). Our algorithm only calculates the contingency table at the start of the algorithm, and then does operations on it (e.g. total rows). This way we are only affected by the number of categories, and not by the size of the data set. From the experiments we ran, our program started to be stuck in 15 categories and above.

# 4 Related Work

Feature Binning is not an active subject area in the formal literature. However, it is an important part of the feature engineering process and there are many blog posts in this field reviewing the different methods and best practices.

We took an approach of supervised feature binning, meaning that the relationship between the feature and target variable is considered, and the binning process is guided by the aim of improving the performance of the predictive model.

There are many works in the field of feature selection that are based on the hypothesis that good feature sets contain features that are highly correlated with the target class [5]. Li Zahng and Xiaobo Chen [6] explored new methods for feature selection in machine learning. These methods incorporate both symmetric uncertainty coefficients and independent classification information to identify the most relevant features in a dataset.

Guided by these works we gave a score to our feature, based on its entropy with the target feature. Our assumption was that the better the binning is the better the feature is, so the score for the binning is the score of the feature after the binning application.

# 5 Conclusion

Our goal in this project was to automate the process of binning for nominal categorical variables in classification tasks. We developed an automated software that outputs the best grouping to bins found according to the binning score and the partition method of recursively splitting into two bins. We have tested our tool on multiple databases and categorical variables and measured our success by the increment in the accuracy of the model as a result of the binning. As shown in the previous sections we have reached good results in many of our tests, but in others, we failed to reach an improvement.

# References

[1] Bnaking dataset. https://www.kaggle.com/datasets/prakharrathi25/banking-dataset-marketing-targets.

[2] churn-modelling. https://www.kaggle.com/datasets/shrutimechlearn/churn-modelling.

[3] Home Credit Default Risk. https://www.kaggle.com/competitions/home-credit-default-risk/data.

[4] titanic. https://www.kaggle.com/competitions/titanic/data.

[5] Mark A Hall. *Correlation-based feature selection for machine learning*. PhD thesis, The University of Waikato, 1999.

[6] Li Zhang and Xiaobo Chen. Feature selection methods based on symmetric uncertainty coefficients and independent classification information. *IEEE Access*, 9:13845–13856, 2021.