Automating identification of numerical fields that should be treated as nominal

Matan Shamir (ID. 206960239), Osher Elhadad (ID. 318969748)

Final project report for TDS course, BIU, 2023

1 Abstract

Our project aims to enhance the data science pipeline by addressing the common issue of incorrectly missing identification of numeric fields that should be treated as nominal due to their lack of inherent order. Failure to correctly identify these fields can lead to erroneous inferences and false correlations, which can severely impact the process of conclusion making in the data science pipeline.

Another aspect emphasizing the importance of detecting the nature of those fields is the use of machine learning on tabular data science [lectures 10-11]. As suggested in "Applied predictive modeling" by Kuhn and Johnson (2013), Nominal variables may require special methods for modeling, such as decision trees, random forests, or support vector machines, while other numeric variables can be modeled using linear regression, logistic regression, or other methods.

An existing approach for this issue is to manually detect those fields using common sense, testing distribution resemblance to known distributions, and using other manual techniques which will be covered later in this report, but these methods can be tedious in datasets with a large amount of fields, and can cause human errors.

To detect those fields in the beginning of the data science pipeline in a quick efficient way, we propose a supervised learning approach that trains a binary classification model to classify numeric fields correctly. Our approach involves creating a new dataset where the samples are existing fields, and their features are the results of statistical tests. We use various techniques to train the model and demonstrate its effectiveness in distinguishing between nominal and non-nominal fields, reducing the likelihood of incorrect correlations and improving the accuracy of data analysis.

Using an automatic tool which will use existing proven techniques, can give a sense of assurance to a data scientist when trying to figure out whether or not a numeric field has an inherent order, and thus can be tested for correlation with other target features.

2 Problem Description

Our project addresses the issue of incorrectly identifying numeric fields that should be treated as nominal [as discussed in lectures 1 and 2]. Many datasets contain fields that lack inherent order and should be treated as nominal, but are mistakenly identified as numeric due to the type of their value, which is numeric. As shown in lecture 1, Nominal fields require special treatment, such as one-hot encoding, before performing tests on the dataset.

The misidentification of those fields can lead the scientist to perform statistical tests and examine possible correlations between the source and target fields, which might result in false correlations and erroneous inferences during the execution of a data science pipeline. This emphasizes the importance of detecting these fields prior to finding relationships between variables.

As stated in the abstract, selecting an appropriate machine learning model to do different tasks such as predicting missing values in a dataset depends on the nature of the variables. Thus, determining whether the variables are nominal is crucial in this decision.

While some numeric fields such as phone numbers, zip codes, and identification numbers are easily recognized as lacking inherent order, others are more difficult to distinguish. If the dataset pertains to a specific domain in which the scientist is not an expert, it may be challenging to determine whether a field's values have inherent order or not. Our project aims to simplify and automate this process.

3 Solution Overview

In this section we present our solution, which was selected after a process of examining several approaches of machine learning which can solve the problem of numeric nominal field detection in a dataset.

We considered three main ideas for the implementation of this task. Two of them involved creating a supervised-learning classification model, importing existing databases and generating a new dataset for a training phase, and manually labeling the newly created samples. The third solution was simpler and did not require data labeling. We elaborate on each of these solutions in the following section.

We follow Stevens (1946) four scales of measurement, namely nominal, ordinal, interval, and ratio, and note that textual fields are limited to nominal or ordinal.

Our solution suggests supervised learning, which requires labeled data, one that was not available for this task. Therefore, we generated a small labeled dataset to examine the feasibility of the method, which produced suboptimal results due to the limitations of PAC learning. We manually tagged each field in each dataset as nominal or non-nominal. To make the dataset more representative, we converted textual nominal fields to numeric nominal fields through factorization.

3.1 Supervised learning with feature extraction

This approach was constructed from collecting meta-data about the fields distribution and performing several statistical tests that were suggested in other researches, which values might imply the feasibility that the field should be treated as nominal. Then, the samples were the input of a training phase of a classifier, which can be either a type of a decision tree, a linear separator or a multi-layered perceptron. We implement all of the suggested classifiers suggested in this approach and perform comparison between them.

The most successful classifier was a random forest classifier, but we present other classifiers in the experimental evaluation section and compare them.

The selection of parameters which might help a model distinct between nominal and non-nominal fields is not an easy task, as most researches suggest using common sense and self interpretation of the field to be able to distinguish nominal and non-nominal fields, as we did when we labeled the dataset we created.

Despite these, some researches suggest certain parameters that differ between nominal and non-nominal numeric fields in a dataset. Our solution suggests a combination of these factors to feed a machine that might aggregate and make the decision according to what it has learnt based on the training set. We present each feature used for the newly created samples and support its selection according to existing work.

3.1.1 Number of unique values

The number of unique values can be a sign for the type of the field, because typically continues variables (which aren't categorized) have infinite possible values and thus are naturally distributed more evenly across different values, which makes the amount of unique values larger.

On the other hand, certain nominal fields such as ID's have a large amount of unique values (as large as the dataset), unlike typical nominal fields like color of hair which are narrowed to a small amount of possible categories.

Nevertheless, this is a good feature that can help a model distinguish between nominal and non-nominal fields.

For example, consider the field 'Color Intensity'[Figure 1] which is a continues variable. it has many unique values in the dataset, unlike 'Work Class'[Figure 2] which is a nominal variable without many unique values:

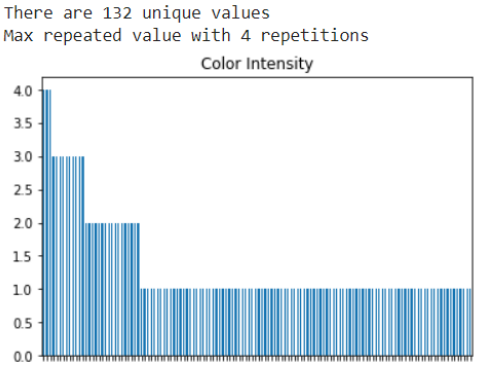


Figure 1: Color Intensity: value-counts()

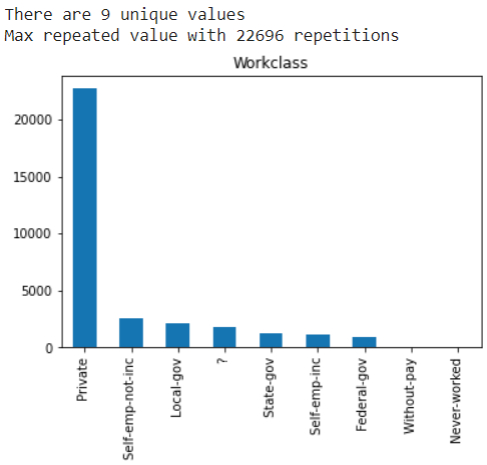


Figure 2: Work Class: value-counts()

3.1.2 Chi-sqaured test

Chi-squared test [presented in lectures 3-4] is a test performed on nominal fields to examine their influence of one another. When performed on a single field's distribution of values, it checks how different its distribution is from the null hypothesis, suggesting that the distribution is uniform. Since nominal variables have no inherent order, each value is considered equally important and meaningful. Therefore, in a dataset with a nominal variable, the frequency distribution of the different values is typically fairly even. This is because there is no reason for any one value to be more or less frequent than another. This is supported by Kuhn and Johnson (2013), which suggest in their paper "Applied predictive modeling" that numeric variables can have different types of distributions, such as normal, skewed, or multimodal, while nominal variables tend to have a fairly even distribution of values. According to this suggestion, measuring the difference from a uniform distribution may be useful when trying to detect nominal fields.

As an example, consider the field 'Alcalinity of Ash' [Figure 3] which is a continues variable. It appears to have an approximately normal distribution, unlike 'Cap Surface'[Figure 4] which is a nominal variable in

which the samples appear to spread evenly between different categories:

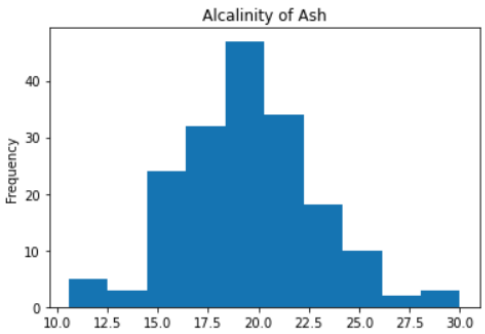


Figure 3: Alcalinity of Ash

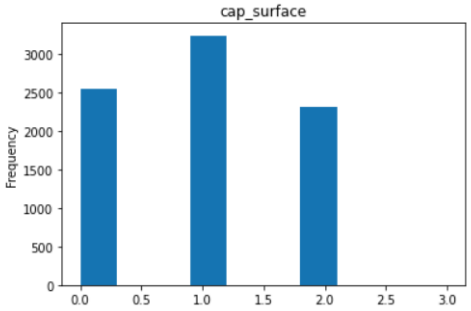


Figure 4: Cap Surface

3.1.3 KS tests

Chi-square test is specifically designed for nominal fields, and may not be appropriate for other types of variables with different distributions, such as continuous or ordinal variables. For those cases, different statistical tests may be needed to evaluate the distribution of the variable. We use both KS-test and chi-squared test as the types of fields our model receives is from both of these types: nominal and non-nominal. The Kolmogorov-Smirnov (KS) test is a non-parametric statistical test used to compare two probability distributions. It is commonly used to determine if a numeric variable follows a particular distribution, such as normal, exponential, logarithmic, or uniform. The KS test can be used to compare the empirical distribution of a numeric variable to a theoretical distribution and determine if the variable is likely to have been drawn from that distribution.

While nominal features have no inherent order and can only take on a limited set of values, some numeric features can take on a continuous range of values. However, not all numeric features have the same distribution. Some numeric features may follow a normal distribution, while others may follow a skewed or multimodal distribution. The KS test can be used to determine if a numeric feature follows a particular distribution, which can be useful in identifying important features for a machine learning model.

For example, consider a dataset with a mix of nominal and numeric features. The KS test can be used to determine if a numeric feature, such as 'Age', follows a particular distribution. If it is found to follow a normal distribution, it may be more useful as a feature in a regression model than a feature with a skewed or multimodal distribution. However, if a nominal feature, such as 'Gender', is mistakenly treated as a non-nominal feature, the KS test can quickly determine that it does not follow a particular distribution and should be treated as a nominal feature.

This lead us to believe that adding the KS-test for several known numeric distribution can help the model distinguish nominal and non-nominal fields.

3.1.4 Skewness and Kurtosis

Skewness and kurtosis [lectures 5-6] are statistical measures used to describe the shape of a distribution. Skewness is an indicator of the extent of asymmetry in the distribution, while kurtosis refers to the degree

of peakedness or flatness of the distribution compared to a normal distribution.

If a nominal variable is coded as numeric, it may exhibit high skewness and kurtosis due to the artificial order introduced by the numerical codes assigned to the categories. As Field (2013) notes, high values of skewness and kurtosis in a nominal variable suggest that it has been incorrectly coded as numeric.

Thus, these measures combined supply a more complete picture as to the distribution of the field, and can be be useful in identifying nominal fields that have been coded as numeric, which would be expected to have high skewness and kurtosis values.

As an example consider the 'Proathocyanins' field. It has very mild positive skewness and relatively small tails, and it is numeric. This is in contrast to the field 'Occupation' which is nominal,It has a much larger skewness, and its kurtosis reflects the large amount of samples that are present in the tails.

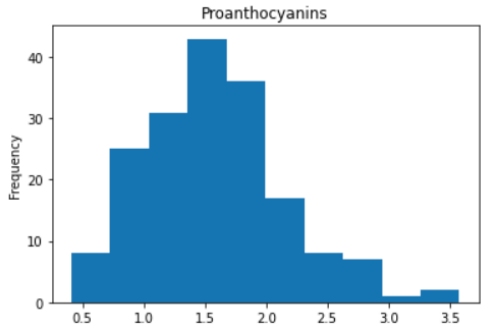


Figure 5: Proathocyanins

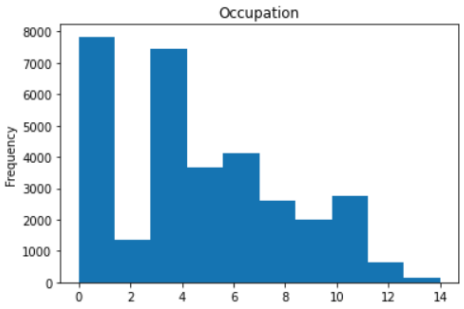


Figure 6: Occupation

3.2 Algorithm presentation

We present a high-level pseudo code to exhibit the main phases of the process executed to train our model.

Algorithm 1 The Elhadad-Shamir method for training a nominal field classification model

Input: A list of datasets $D_1, D_2, ..., D_n$, where each D_i is a dataset containing nominal and numeric fields.
Output: A trained nominal field classification model.

1: Create new empty dataset D_{new} where each sample will be field of D_1 to D_n with results of statistical test
2: Manually label all fields of D_{new} as either nominal or non-nominal
3: **for** each column c in D_{new} **do**
4: **if** column c is textual **then**
5: Factorize c to turn it into a numeric column
6: Compute chi-square, KS-test (uniform and normal), skewness, kurtosis, and number of unique values in c
7: Add the computed values as a sample to D_{new}
8: Train a random forest model M on D_{final} using grid search

4 Experimental Evaluation

In this section, we present other approaches which can be used for comparison to the model we suggested.

Note that an existing automatic tool does not existing as far as we know.

4.1 Supervised Deep Learning Approach

We used a supervised deep learning approach to classify our data, with the though that deep learning architectures can perform feature extraction independently. We used the same labeled dataset preprocessed for the random forest classification model, except this time the features were all of the different values the samples had of this field.

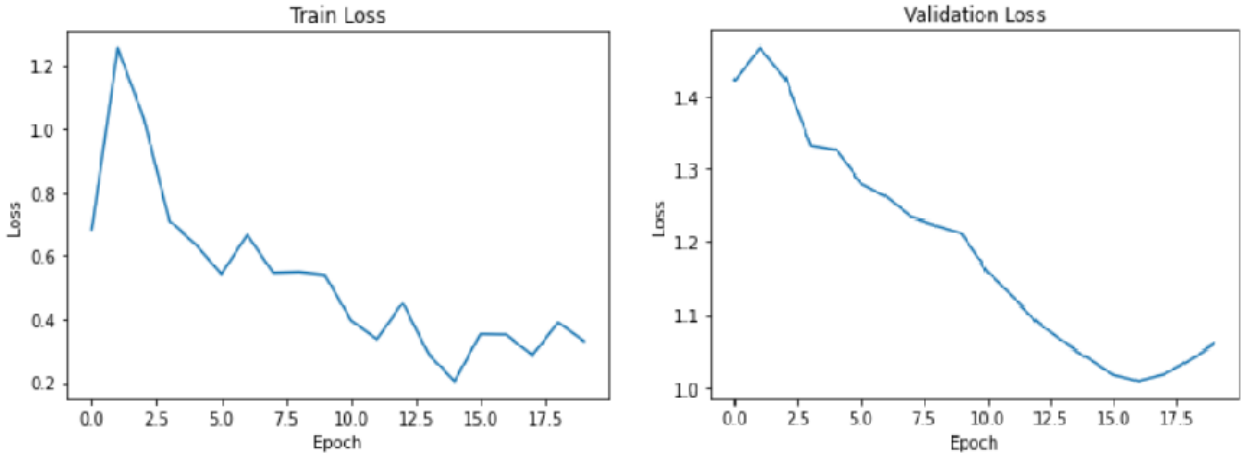
We chose a neural network architecture with three dense layers, batch normalization, and dropout regularization to prevent overfitting. We used the Adam optimizer and binary cross-entropy loss for training, and included early stopping to prevent overfitting.

We used a CNN network, as a fully-connected network gave importance the location of each feature and failed to precept the distribution of the variable.

We used K-cross validation to better validate the performance of our model, which was highly important due to the small scale of our dataset.

However, the results were not as good as expected, mainly due to the fact that the number of labeled samples was extremely low. It was difficult to determine whether the model was too expressive for such a small amount of data, which may have contributed to the lower accuracy. Nonetheless, we believe that this approach could be effective with a larger labeled dataset.

These are the results:




```
Fold 5: Loss: 0.6677100658416748 Accuracy: 0.6666666865348816
Loss: 0.66 (+/- 0.01)
Accuracy: 0.67 (+/- 0.04)
```

4.2 Unsupervised learning approach

This approach did not require labeled data, which makes it easier to implement without the need for preprocess, and also more practical in the near term.

First, we set up labels for each column in the dataset by identifying which columns are nominal and which are non-nominal. Then, we perform Principal Component Analysis (PCA) on the transposed dataset with 10 components.

Next, we apply K-Means clustering on the PCA results with 4 clusters, using k-means++ initialization and running the algorithm 10 times with different random seeds. The resulting cluster labels are then mapped to the column types.

We predict the column type for each cluster based on the proportion of nominal columns in the cluster. If the proportion is greater than 0.5, we predict that all columns in that cluster are nominal. Otherwise, we predict that all columns in that cluster are non-nominal.

Finally, we evaluate the accuracy of the clustering approach by comparing the predicted labels to the true labels. The accuracy is calculated as the proportion of correctly classified columns.

The accuracy of the model was rather low; it received an accuray of 0.688, which led us to presume that the complexity of the data made it hard to separate with a clustering algorithm. Another important reason was the loss of information that PCA results.

This approach can be listed as a plausible alternative for the case of non-existing versatile datasets or low computation ability, due to the lack of need for labeled data.

4.3 Supervised learning approach with different classifiers

This was the method chosen in the previous section. We present alternative classifiers and compare them with the performance of the random forest classifier suggested in our algorithm.

4.3.1 Machine learning linear classifiers

We tried training a logistic regression, SVM and perceptron classifiers and all gave similar results: a rather low accuracy.

We chose SVM as a representor of those family of classifiers. Its accuracy on the validation set was 0.708, which was a bad result.

We assume the low precision was due to the fact that The family of linear separators don't have the capacity for the complexity of the data; meaning it is not exactly linearly seperable.

4.3.2 Random forest classifier

On the contrary, a decision tree is known as a good classifier for small-scale datasets, and as a classifier that can seperate data which is not linearly-seperable.

We chose a random forest classifier as it is considered a state-of-the-art clasifier in the family of decision trees.

It produced an accuracy of 0.9166 on the test set, the highest among other approaches.

5 Related Work

As stated throughout the paper, there are no existing tools that provide an automate way to classify numeric fields of a dataset as either nominal or non-nominal.

We also refer to several existing books and papers which discuss the detection of the nature of variables as ones that have or do not have inherent order, as this is the thing our model aims to detect.

As referenced earlier, Kuhn and Johnson (2013) notes that it is natural to assume that nominal variables' values spread more evenly on the range of possible values as there is no inherent order between them. This makes sense, as many times those variables' null hypothesis is that the distribution is uniform.

Another main citation is that of Field (2013), which mentions skewness as a possible way to detect whether numeric variables are nominal or not. It is explained in detail in that section.

All of these are used as features in the samples crated as inputs for the model we trained.

6 Conclusion

In this study, we explored various approaches to classify nominal and non-nominal numeric fields in a dataset. We started by examining unsupervised learning models, such as KMeans clustering and Principal Component Analysis (PCA), which were not successful in achieving high accuracy due to the lack of importance of the order of the samples provided to the model. Deep learning models were also considered, but the same issue of not perceiving the distribution of the fields made these models unsuitable for our task.

We then moved to supervised learning models, which proved to be the most effective approach. Specifically, using feature extraction techniques, such as the chi-square test, KS-test, skewness, and kurtosis, helped in understanding the distribution of the fields and therefore the classification of nominal and non-nominal numeric fields with high accuracy.

However, we acknowledge that our study was limited by the size of our dataset. With more samples, we believe that higher accuracy would be achieved, as well as more robustness to overfitting.

Finally, we found that the Random Forest classifier was the best choice for our task, as it was able to handle the high-dimensional feature space and had a low risk of overfitting, unlike linear classifiers. Overall, our findings highlight the importance of selecting an appropriate feature extraction technique and supervised learning model when dealing with the classification of nominal and non-nominal numeric fields in a dataset.

References

Field, A. (2013). *Discovering Statistics Using IBM SPSS Statistics*. Sage.

Kuhn, M. and Johnson, K. (2013). *Applied Predictive Modeling*. Springer.

Stevens, S. S. (1946). On the theory of scales of measurement. *Science*, 103(2684):677–680.