**Abstract**

The data generated by online advertising industry for digital ads contains many categorical variables. Predicting click through rate (CTR) for digital ads is vital for efficiency of ads marketplace (real time bidding for digital ads) and often involves dealing with these categorical variables. Since, most of machine learning algorithms are designed to work with numerical data; we need efficient numerical encoding of these categorical variables. A common challenge in online advertising is the presence of high cardinality in these categorical variables. This report examines various encoding techniques for categorical variables and their impact on predictive ability of classification algorithms for CTR. The experiment involves seven encoding techniques (one hot, ordinal, binary, frequency, target, hash, quantile) and two classification algorithms (logistic regression, gradient boosted trees). The results from the analysis yield that one hot encoding yields the best roc-auc for predicting CTR. However, it leads to high dimensional space for input data in presence of high cardinality categorical variables, which further poses computation and memory challenges for the data. Binary and Target encoding yield slightly less roc-auc than one hot encoding for predicting CTR, without transforming data into high dimensional space.

**1. Introduction**

This project aims to tackle practical challenges associated with analyzing and predicting click through rate (CTR) for digital advertisements. CTR is an industry-wide used metric in online advertising to assess performance of digital ads. In online advertising industry, a digital ad is a transaction between two parties; advertisers (who want to promote/sell a product) and publishers (who allow their website, mobile app, search engine etc. to publish the advertisement). The advertisers have the option to pay fee to the publishers either per impression (every time an ad is shown to the user) or per click (every time a user clicks on the ad). Majority of advertisers prefer to pay per click basis and bid for the ads on real time auction bidding platform provided by publishers. Thus, it is critically important for involved parties (publishers and all advertisers) to accurately estimate probability of an ad being clicked (CTR) for a fair price transaction on the bidding platform.

One of the key challenges in predicting CTR is the presence of high cardinality categorical variables. A conventional CTR dataset for digital ads contains attributes for users, ads and context. Very often these attributes are categorical in nature and possess high cardinality. For instance, identification variables for website or mobile app, user device id/model may exhibit multitude of values leading to issues of high cardinality. Also, the user attributes such as location, country etc. and ad attributes such as banner position etc. are categorical in nature. Since most machine learning algorithms are designed to work with numerical data, it’s crucial to find appropriate numerical representations of categorical variables. To address this, the report examines various encoding techniques. The effect of these encoding techniques on predictive ability of various machine learning classification models is studied to determine optimal encoding technique.

**2. Encoding Techniques**

This report considers 7 encoding techniques to encode multiple levels of a categorical variable as a way to reduce dimensionality:

**2.1 One Hot Encoding**

One hot encoding, dummy encoding or indicator encoding is the most commonly used encoding technique for categorical variables. It expands a categorical variable into dummy columns taking values in {0, 1} to indicate whether or not that dummy column is true for that observation. Each dummy column corresponds to one of levels of the categorical variable. If the cardinality of categorical variable is high, it leads to large sparse datasets.

**2.2 Ordinal Encoding**

This replaces the levels of categorical variable by ordinal number. It’s easy to implement and doesn’t lead to sparse datasets unlike one hot encoding. However, it introduces arbitrary order among levels of categorical variable which doesn’t make sense practically.

**2.3 Binary Encoding**

First, the levels of categorical variable are converted to ordinal numbers. Then the ordinal numbers are converted to their binary representations which are then split into different columns. This can be particularly helpful where we don’t want to use one hot encoding on high cardinality categorical variable as it leads to fewer dimensions than one hot encoding.

**2.4 Frequency Encoding**

Frequency encoding maps each level of categorical variable to its relative frequency in the dataset. The inherent assumption being that high frequency of a level is associated with target. This approach is similar to n-grams approach in Natural Language Processing to encode tokens based on relative frequency.

**2.5 Target Encoding**

This encodes each level of categorical variable by its conditional target mean (if target is continuous) or conditional target probability (is target is binary).

**2.6 Hash Encoding**

The hash encoding first converts each level of a categorical variable into an integer using a hash function. This integer is transformed to indicator representation, similar to binary encoding, using a given hash size. This can lead to collision where multiple levels of categorical variable can map to one indicator representation. Smaller hash size leads to more collision unlike larger hash size.

**2.7 Quantile (50%) Encoding**

This approach is similar to target encoding except that instead of target mean, target median is encoded for each level of categorical variable.

**3. Classification Models**

This report considers the effect of various encoding techniques on predictive ability of following 2 classification models:

**3.1 Regularized Logistic Regression**

The logistic regression with L1 penalty is used with all the above encodings to predict CTR. It maps linear combination in encoded categorical variables to CTR probability through logit link function. The L1 regularization helps feature selection and model fit simultaneously.

**3.2 Gradient Boosted Trees**

The Gradient boosted trees are state of the art ensemble methods which combine several weak learners into an ensemble for better prediction. It iteratively fits many trees and up-weights/down-weights incorrectly/correctly predicted observations from previous iteration. The gradient boosted trees are fitted with each ensemble to assess the effect of encoding on its performance.

**4. Experiment Setup**

The dataset used in this analysis has been made available on Kaggle by online advertising company Avazu. The dataset contains 10 days history of CTR on digital ads along with associated categorical variables. The categorical variables are first converted to a numerical representation using an encoding technique. The classification model is then fitted, tuning various hyper-parameters using 5 fold cross validation. The best fitted model is selected based on scoring function: AUC-ROC. The best fitted model is then tested on test set for ROC-AUC. The aim is to find the encoding which achieves high CTR prediction AUC-ROC across most of classification models.

**5. Results**

The following table shows average cross validation AUC-ROC for training set and test set for different combinations of encoding and classification models.

|  |  |  |  |
| --- | --- | --- | --- |
| **Encoding** | **Model** | **Avg CV score** | **Test score** |
| One Hot Encoding | Logistic Regression | 0.730193 | 0.727923 |
| One Hot Encoding | Gradient Boosting | 0.704358 | 0.697265 |
| Ordinal Encoding | Logistic Regression | 0.576326 | 0.569783 |
| Ordinal Encoding | Gradient Boosting | 0.689237 | 0.690202 |
| Binary Encoding | Logistic Regression | 0.686627 | 0.687019 |
| Binary Encoding | Gradient Boosting | 0.719450 | 0.717643 |
| Frequency Encoding | Logistic Regression | 0.628360 | 0.621864 |
| Frequency Encoding | Gradient Boosting | 0.702945 | 0.700258 |
| Target Encoding | Logistic Regression | 0.708825 | 0.703415 |
| Target Encoding | Gradient Boosting | 0.693570 | 0.691067 |
| Hashing Encoding | Logistic Regression | 0.635071 | 0.626784 |
| Hashing Encoding | Gradient Boosting | 0.693055 | 0.692240 |
| Quantile Encoding | Logistic Regression | 0.547550 | 0.545026 |
| Quantile Encoding | Gradient Boosting | 0.547534 | 0.545017 |

The one hot encoding yields the best result. However, one hot encoding leads to large sparse datasets leading to high dimensionality especially in presence of high cardinality categorical variables. Binary Encoding and Target Encoding transform categorical data into numerical data without significant loss in test performance.

**6. Limitations**

The optimal encodings are determined with respect to two classification models and CTR dataset. Ideally, the optimal encoding should be model agnostic. So practically accommodate this, this study needs to expand to other classification techniques and datasets

**7. Conclusion and Future Scope of Work**

Binary encoding and target encoding achieve almost equivalent performance as one hot encoding without transforming data into very high dimensions unlike one hot encoding. However, it’s generally mentioned in the literature that one hot encoding works better for low cardinality categorical variables. Possible future direction could include using one hot encoding for low cardinality features and other encoding techniques for high cardinality features. This solution may achieve better prediction accuracy like one hot encoding without transforming dataset to high dimensions unlike one hot encoding.

**8. References**

1) Regularized target encoding outperforms traditional methods in supervised machine learning with high cardinality features: [Link](https://arxiv.org/abs/2104.00629)

2) Modern models for learning large-scale highly skewed online advertising data: [Link](https://escholarship.org/content/qt7mc0k1v8/qt7mc0k1v8_noSplash_68fe9a60ebefb849e7eb26b341d072b0.pdf)