Enhancing Pet Adoption Predictions: A Novel Supervised Encoding Approach

Nikhil Narayane and Abdullah Salau

12th December 2023

1 Introduction

This paper presents an innovative approach to encoding categorical variables in the context of pet adoption prediction, a field of study with profound implications for animal welfare and shelter management. Originating from a Kaggle competition, the foundational project utilizes a range of features to predict the speed of pet adoption, aiming to identify key factors that influence adoption rates. This endeavor seeks to facilitate strategies that could expedite the adoption process, addressing critical challenges in animal welfare.

Contrasting with the original project's focus, our research emphasizes enhancing the predictive modeling process. We propose a novel method for encoding categorical variables, uniquely integrating adoption speed as a pivotal factor in the vector generation process. This supervised vector generation method diverges significantly from traditional encoding techniques like one-hot encoding, offering a more nuanced approach to data representation.

Our study conducts an exhaustive evaluation of this new encoding strategy, benchmarking it against the conventional one-hot encoding method. Employing the Pet Adoption dataset, we apply widely-used machine learning algorithms—k-Nearest Neighbors (kNN), Random Forest, and XGBoost—to assess the efficacy of our approach. The primary motivation is to develop an encoding method that not only preserves the richness of categorical information but also addresses the challenges of dimensional expansion typical in one-hot encoding.

Additionally, we extend our analysis by generating a simulated dataset, aiming for a more balanced distribution of the target variable. This allows for a comparative analysis of our proposed encoding technique against traditional methods under varied dataset conditions.

2 Data and Preprocessing

The dataset utilized in this study is sourced from the Pet Adoption dataset available on Kaggle[1]. It comprises a comprehensive array of features related to rescued pets, with the target variable indicating the time until adoption, including cases where adoption did not occur. The dataset includes diverse attributes such as pet type (dog or cat), name, age, breed, color, gender, maturity size, fur length, vaccination, deworming, sterilization status, health condition, quantity, adoption fee, state, number of videos and photos uploaded, and a descriptive narrative for each pet.

Preliminary analysis revealed varying predictive powers among these variables. Prior to model implementation, preprocessing was conducted to eliminate anomalous values. The training data consists of 14,993 samples with 23 features, while the testing data comprises 3,972 samples with 22 features. The Breed Labels data, delineating 307 breeds (241 dog breeds and 66 cat breeds), has a shape of (307, 2). The Color Labels data, indicating 7 color categories, has a shape of (7, 1).

In terms of preprocessing, non-essential variables such as 'Name' and 'Description' were excluded from the analysis. Additionally, ID variables were removed. The training dataset predominantly consists of approximately 8,000 dog samples and nearly 7,000 cat samples. Gender distribution within the dataset shows a slight predominance of females over males by approximately 500 in both dogs and cats.

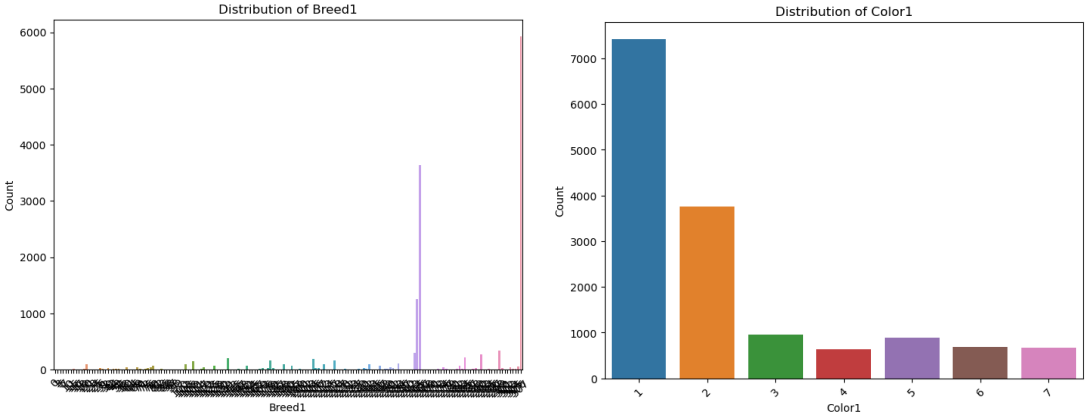
The dataset exhibits a categorical distribution across 'Type' and 'Gender', with each 'Type' category further segmented by 'Gender'. For example, under 'Type' 1, there are 3,005 instances for 'Gender' 1, 4,252 for 'Gender' 2, and 875 for 'Gender' 3. Similarly, 'Type' 2 shows 2,531 instances for 'Gender' 1,

[1] `https://www.kaggle.com/competitions/petfinder-adoption-prediction/data`

3,025 for 'Gender' 2, and 1,305 for 'Gender' 3. This categorization offers insights into the demographic composition within the dataset.

Furthermore, the dataset illustrates the distribution across 'Type' and 'Health' categories. For 'Type' 1, there are 7,845 instances under 'Health' 1, 266 under 'Health' 2, and 21 under 'Health' 3. For 'Type' 2, the counts are 6,633 for 'Health' 1, 215 for 'Health' 2, and 13 for 'Health' 3, indicating a higher frequency in the 'Health' 1 category for both types.

Histogram analysis of each numerical variable confirmed the validity of the data, supported by textual descriptions of the observations. For instance, descriptions like "40 feral cats" or "20 puppies" corroborated the numerical values. The only exception was found in the 'Age' variable, where descriptions did not align with extreme values (e.g., 255 months). Consequently, outliers in the 'Age' column were removed to enhance data integrity



3 New Encoding Procedure

In the encoding process, we adopted a supervised approach by leveraging the labeled samples within the training set. The primary objective was to establish an innovative representation for each categorical variable in the form of a numerical vector. For each distinct value of the categorical variable under consideration, we performed a calculation to determine the percentage of all training set samples sharing the same value that belonged to each specific level of adoption speed.

Let the target variable have p distinct categories/levels to be predicted, named $t_1, t_2, \ldots, t_p$. Therefore, there will be p new variables to replace the categorical variable x, which we can refer to as $t_{1_x}, t_{2_x}, \ldots, t_{p_x}$.

Let our categorical variable under consideration be called x, and let it have different distinct categories: $x_1, x_2, \ldots$. For the n samples in the dataset, let n_c be the subset of samples that have a value of x_c in the categorical variable x. We perform the following calculation to determine the numerical value that will be in t_{i_x}:

$$t_{i_x} = \frac{\text{Total Number of variable } x \text{ samples with target variable } i}{\text{Total Number of variable } x \text{ samples}}$$

This calculation results in a percentage representing the proportion of samples with a specific value in the categorical variable x that belong to each level of the target variable.

Subsequently, these calculated percentages are compiled into a numerical vector, effectively replacing the original categorical variable in the training set:

This approach aimed to transform the inherent categorical information into a more meaningful and continuous representation, capturing the nuanced relationships between the categorical variable and the target variable's adoption speed levels.

Also, it can be noted from the numerical vector representation of the Breed1 variable that this new encoding technique is very robust to categorical variables with very high cardinality, so long as the number of target variable categories is not too high. With ordinal target variables however, we would still be able to divide the target categories into a smaller number of bins to be used for the numerical vector.

Color1	percentage_0	percentage_1	percentage_2	percentage_3	percentage_4
1	0.023293	0.201697	0.274135	0.219873	0.281002
2	0.031733	0.183733	0.254933	0.227733	0.301867
3	0.030623	0.222809	0.300950	0.194298	0.251320
4	0.020505	0.220820	0.228707	0.228707	0.301262
5	0.038462	0.248869	0.269231	0.209276	0.234163
6	0.042398	0.228070	0.285088	0.194444	0.250000
7	0.019490	0.263868	0.272864	0.187406	0.256372

(a) Color1 Numerical Vector Representation

Breed1	percentage_0	percentage_1	percentage_2	percentage_3	percentage_4
0	0.000000	0.000000	0.200000	0.400000	0.400000
1	0.000000	0.000000	0.000000	0.000000	1.000000
3	0.000000	0.000000	0.000000	0.000000	1.000000
5	0.000000	0.500000	0.000000	0.000000	0.500000
7	0.000000	0.000000	0.000000	0.000000	1.000000
...	...	...	...	...	...
303	0.047619	0.190476	0.404762	0.095238	0.261905
304	0.000000	0.000000	0.571429	0.000000	0.428571

(b) Breed1 Numerical Vector Representation

4 Model Comparison and Performance Analysis

4.1 Variable Definitions

In our study, we categorize the variables from the Pet Adoption dataset as follows:

- `cat_vars`: Categorical variables including 'Type', 'Breed1', 'Breed2', 'Gender', 'Color1', 'Color2', 'Color3', 'MaturitySize', 'FurLength', 'Vaccinated', 'Dewormed', 'Sterilized', 'Health', and 'State'.
- `numeric_vars`: Numerical variables such as Age, Quantity, Fee, VideoAmt, and PhotoAmt.
- `cat_to_numeric_vars`: Categorical-to-Numerical variable 'RescuerID'.

The models discussed in Sections 4.2, 4.3, and 4.4 utilize these features in various combinations. Specifically, when referring to one-hot encoding of categorical variables (`OHE`), we imply the transformation of all `cat_vars`. Similarly, the term 'continuous variables' encompasses all `numeric_vars` listed. This segregation is crucial as our method primarily focuses on the encoding of categorical variables. Notably, traditional one-hot encoding combined with continuous variables results in a feature space of 374 dimensions, highlighting the challenge of the curse of dimensionality.

4.2 Performance of kNN Across Encoding Models

Table 1: kNN Performance Across Models

Model	Cross-Validated Accuracy	Test Set Accuracy
Model 1 (OHE with Cont. Vars)	0.36	0.38
Model 2 (OHE without Cont. Vars)	0.34	0.34
Model 3 (Just Cont. Vars)	0.36	0.36

In Model 1, we employ one-hot encoding of categorical variables along with numerical variables. Model 2 utilizes one-hot encoding of categorical variables without continuous variables. Model 3 is based solely on continuous variables. This distinction in variable usage across models is pivotal in assessing the impact of our encoding method.

4.3 Performance of Random Forest Across Encoding Models

Table 2: Random Forest Performance Across Models

Model	Cross-Validated Accuracy	Test Set Accuracy
Model 1 (OHE with Cont. Vars)	0.43	0.43
Model 2 (OHE without Cont. Vars)	0.37	0.36
Model 3 (Just Cont. Vars)	0.40	0.39

4.4 Performance of XGBoost Across Encoding Models

Table 3: XGBoost Performance Across Models

Model	Test Set Accuracy	F1 Score	Precision	Recall
Model 1 (OHE with Cont. Vars)	0.45	0.4064	0.4054	0.4218
Model 2 (OHE without Cont. Vars)	0.37	0.3701	0.3715	0.3848
Model 3 (Just Cont. Vars)	0.386	0.3720	0.3803	0.3982

5 Evaluation of the Novel Encoding Technique

In this section, we rigorously assess the efficacy of our newly developed encoding method. To ensure a consistent basis for comparison, we utilize the same algorithms as in the previous sections. Unlike the prior approach where one-hot encoding was applied to categorical variables, here we implement our innovative algorithm for encoding these variables. Notably, the peak performance achieved by this new method is on par with the one-hot encoding combined with continuous variables, as observed in Model 1 of the XGBoost algorithm in Section 4.4.

5.1 Performance of kNN Using Numerical Vector Encoding

Table 4: kNN Performance Across Models

Model	Cross-Validated Accuracy	Test Set Accuracy
Model 1 (Numerical Vector with Cont. Vars)	0.36	0.36
Model 2 (Numerical Vector without Cont. Vars)	0.36	0.38

5.2 Performance of Random Forest Using Numerical Vector Encoding

Table 5: Random Forest Performance Across Models

Model	Cross-Validated Accuracy	Test Set Accuracy
Model 1 (Numerical Vector with Cont. Vars)	0.43	0.43
Model 2 (Numerical Vector without Cont. Vars)	0.37	0.38

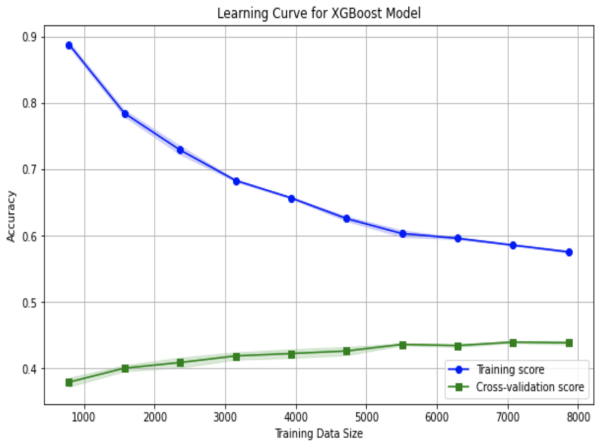


Figure 3: Learning Curve

5.3 Performance of XGBoost Using Numerical Vector Encoding

Table 6: XGBoost Performance Across Models

Model	Test Set Accuracy	F1 Score	Precision	Recall
Model 1 (Numerical Vector with Cont. Vars)	0.44	0.3909	0.3933	0.4116
Model 2 (Numerical Vector without Cont. Vars)	0.38	0.3932	0.4062	0.4064

The learning curve depicted in Figure 3 illustrates a notable trend: as training samples increase, the cross-validation score improves, indicating enhanced model generalization. Conversely, the training accuracy exhibits a decline, suggesting a reduction in overfitting as the model is exposed to more data.

6 Conclusions from Model Comparison

Through our comprehensive analysis of kNN, Random Forest, and XGBoost models, employing various data encoding techniques and variable combinations, we have arrived at several key conclusions, each substantiated by precise metrics:

1. **Comparative Model Efficacy:**
   - XGBoost emerges as the superior model, consistently outperforming kNN and Random Forest, evidenced by its highest Test Set Accuracy of 0.45 (Model 1: OHE with Continuous Variables) and an F1 Score of 0.4064.
   - Random Forest surpasses kNN in performance, achieving a Test Set Accuracy as high as 0.43 (Model 1: OHE with Continuous Variables), in contrast to kNN's peak at 0.38 (Model 2: OHE without Continuous Variables).
   - kNN shows moderate effectiveness, with its best Test Set Accuracy at 0.38 (Model 2: OHE without Continuous Variables).
2. **Role of Continuous Variables:**
   - The inclusion of continuous variables notably enhances model performance, particularly for XGBoost (Test Set Accuracy improvement from 0.37 to 0.45) and Random Forest (Test Set Accuracy improvement from 0.36 to 0.43).
   - kNN's performance exhibits minimal variation with or without continuous variables, maintaining a Test Set Accuracy in the range of 0.36-0.38.
3. **Encoding Methodologies:**

- One-Hot Encoding combined with continuous variables generally yields better results, as demonstrated by XGBoost's Test Set Accuracy of 0.45 (Model 1).
- Our novel numerical vector encoding approach displays competitive performance, especially in Random Forest, where it enhances Test Set Accuracy from 0.36 to 0.38.

4. **Model-Specific Encoding Preferences:**
   - XGBoost significantly benefits from One-Hot Encoding with continuous variables, achieving the highest Test Set Accuracy of 0.45.
   - Random Forest shows consistent performance across various encoding techniques, with a slight preference for One-Hot Encoding with continuous variables (Test Set Accuracy of 0.43).
   - kNN's performance remains relatively stable irrespective of the encoding technique used.
5. **Precision and Recall in XGBoost:**
   - XGBoost exhibits higher precision and recall in the model utilizing One-Hot Encoding with continuous variables, compared to the numerical vector encoding without continuous variables.

Reducing the Curse of Dimensionality: Our new encoding method effectively mitigates the curse of dimensionality, capturing comprehensive information while maintaining performance. This approach signifies a substantial advancement in encoding techniques, offering a more efficient and information-rich alternative to traditional methods.

In conclusion, these findings underscore the importance of careful encoding technique selection and the strategic inclusion of continuous variables in predictive modeling. The superiority of XGBoost in handling diverse data representations, as compared to kNN and Random Forest, is also a significant outcome of this study.

6.1 Evaluation on Simulated Data

In order to validate our conclusions and test the robustness of our models, we extended our analysis to a simulated dataset. This dataset was generated using Python, ensuring that it mirrors the dimensions of the original training data while achieving a more balanced distribution across various features, including the target variable 'Adoption Speed'.

The simulated dataset was designed to address potential biases and imbalances present in the original dataset. By creating a more evenly distributed dataset, particularly in terms of the adoption speed categories, we aimed to provide a stringent test environment for our models. This approach also allowed us to assess the models' performance in scenarios that may not have been adequately represented in the original data.

Upon splitting the simulated data into training and testing sets, the following accuracy metrics were observed:

- F1 Score: 0.2221
- Recall: 0.2652
- Precision: 0.2322
- Test Set Accuracy: 0.2590

These results indicate a moderate level of performance by the models on the simulated data. The F1 Score, a harmonic mean of precision and recall, stands at 0.2221, suggesting a balance between precision and recall, albeit at a lower scale. The recall of 0.2652 indicates the model's ability to correctly identify positive instances, while the precision of 0.2322 reflects the accuracy of these identifications. The Test Set Accuracy of approximately 0.2590, although not high, demonstrates the model's capability to generalize to new, balanced data.

The performance metrics on the simulated dataset underscore the challenges in predictive modeling in balanced datasets, particularly for complex scenarios like pet adoption. These findings highlight the need for further model refinement and exploration of advanced feature engineering techniques to enhance model accuracy and reliability.