PetFinder: Predicting Adoption Speed of Pets

Jinyi Liu

Haslam College of Business, University of Tennessee, Knoxville, jinyi.liu@utk.edu, https://orcid.org/0009-0009-8600-9965

In this project, we use Xgboost to predict the adoption speed of pets. We do a feature engineering and use a Xgboost regressor to train the model. This gives us an average of $\kappa = 0.4694$ with a standard error 0.01058 for a 5-fold crossvalidation.

Key words: Classification, Supervised Learning, Xgboost, Feature Engineering

1. Background and Motivation

Millions of stray animals suffer on the streets or are euthanized in shelters every day around the world. If homes can be found for them, many precious lives can be saved — and more happy families created.

PetFinder.my has been Malaysia's leading animal welfare platform since 2008, with a database of more than 150,000 animals. PetFinder collaborates closely with animal lovers, media, corporations, and global organizations to improve animal welfare.

In this project, we use PetFinder.my's database to create an algorithm that predicts how quickly a pet will be adopted. From this, PetFinder can improve their pet's profile to attract more potential adopters.

2. Problem Statement

2.1. Data

The training data set consists of 14993 pet profiles. Each profile contains the following features:

- **AdoptionSpeed** - Categorical speed of adoption. Lower is faster. **This is the value to predict.**
- **PetID** - Unique hash ID of pet profile
- **Type** - Type of animal
- **Name** - Name of pet
- **Age** - Age of pet when listed, in months
- **Breed1, Breed2** - Breeds of pet

- **Gender** - Gender of pet
- **Color1, Color2, Color3** - Colors of pet
- **MaturitySize** - Size at maturity
- **FurLength** - Fur length
- **Vaccinated** - Pet has been vaccinated
- **Dewormed** - Pet has been dewormed
- **Sterilized** - Pet has been spayed / neutered
- **Health** - Health Condition
- **Quantity** - Number of pets represented in profile
- **Fee** - Adoption fee
- **State** - State location in Malaysia
- **RescuerID** - Unique hash ID of rescuer
- **VideoAmt** - Total uploaded videos for this pet
- **PhotoAmt** - Total uploaded photos for this pet

2.2. Aim

Animal adoption rates are strongly correlated to the metadata associated with their online profiles. In this project, we use this dataset to create an algorithm that predicts how quickly a pet will be adopted. The adoption speed is determined by how quickly, if at all, a pet is adopted. The values are determined in the following way:

0 – Pet was adopted on the same day as it was listed.

1 – Pet was adopted between 1 and 7 days (1st week) after being listed.

2 – Pet was adopted between 8 and 30 days (1st month) after being listed.

3 – Pet was adopted between 31 and 90 days (2nd & 3rd month) after being listed.

4 – No adoption after 100 days of being listed. (There are no pets in this dataset that waited between 90 and 100 days).

2.3. Model

Since the adoption speed to be predicted is categorical, we adopt a xgboost model. This model is a gradient boosting algorithm that is optimized for speed and performance. It is based on decision trees and is a popular choice for machine learning competitions.

2.4. Evaluation

We use quadratic weighted kappa to evaluate the performance of our model. Predicted results have 5 possible ratings, $0,1,2,3,4$. The quadratic weighted kappa is calculated as follows. First, an $\mathrm{N} \times \mathrm{N}$ histogram matrix O is constructed, such that $O_{i,j}$ corresponds to the number of adoption records that have a rating of i (actual) and received a predicted rating j. An $\mathrm{N} \times \mathrm{N}$ matrix of weights, w, is calculated based on the difference between actual and predicted rating scores:

$$w_{i,j} = \frac{(i-j)^2}{(N-1)^2}$$

An $\mathrm{N} \times \mathrm{N}$ histogram matrix of expected ratings, E, is calculated, assuming that there is no correlation between rating scores. This is calculated as the outer product between the actual rating's histogram vector of ratings and the predicted rating's histogram vector of ratings, normalized such that E and O have the same sum. From these three matrices, the quadratic weighted kappa is calculated as:

$$\kappa = 1 - \frac{\sum_{i,j} w_{i,j} O_{i,j}}{\sum_{i,j} w_{i,j} E_{i,j}}.$$

The metric κ typically varies in $[0,1]$. The higher the κ, the better the agreement between the raters. A value of 0 indicates agreement equivalent to chance, and a value of 1 indicates perfect agreement. In the event that there is less agreement between the raters than expected by chance, κ can be negative.

3. Experiment and Results

There are 14993 pet profiles in the training data set and contains 23 features.

3.1. Data Preprocessing

Since we adopt the Xgboost method, the missing value is not a problem. We combine the breed and color features into one feature. For example, if the pet has two breeds, we combine the two breeds into one feature. If the pet has three colors, we combine the three colors into one feature. We also add one more column named hard-interaction which conbines pet's type, gender, vaccinated, dewormed and sterilized status. Some profiles contain multiple pets so we calculate the average fees and photos to create a new column. Furthermore, there are some pets without a name, we think this matters for the adoption speed so we add one more column to indicate whether the pet has a name and also the name length and strangeness of the name.[1] The breed of the pet is also an important factor. Thus, we add features to consider the breed name, breed number, mixed breed, domestic breed and whether it's pure.

3.2. Feature Engineering

To better train the model, we do a feature engineering. We briefly introduce the features we add in this section and leave the details in the appendix code. We add the following features: Color, Breed, State, State-Breed1-Color1, State-BreedFull-ColorFull, Name, RescuerID, Hard-Interaction. For each one, we calculate the mean, minimum, maximum and standard deviation. After that we drop the mentioned features and only keep the new features since they are highly correlated. After this, we get a new 136 numerical features.

3.3. Model

3.3.1. Xgboost Classifier Intuitively, we think the Xgboost classifier is a good choice for this problem since the predicted variable has 5 values. We use the Xgboost classifier to train the model. To find the best hyperparameters, we do a grid search and find the best hyperparameters.[2] This model gives us an average of $\kappa = 0.40$ which is not good enough.

3.3.2. Xgboost Regressor Since the Xgboost classifier does not give us a good result, we try the Xgboost regressor. We do a new grid search for the best hyperparameters. In this case, we choose the best hyperparameters as follows:

- objective: reg:squarederror
- eval_metric: rmse
- max_depth: 7
- subsample: 0.8
- colsample_bytree: 0.8
- α: 0.05
- η: 0.01

To be clear, since we use a regression model, the predicted value is not an integer. We use the following method to convert the predicted value to an integer:

1. Calculate the c.d.f. of the AdoptionSpeed in the training set and store the result in a list $c = [c_0, c_1, c_2, c_3]$.

[1] We assume the name is strange if it contains number and special characters.

[2] It costs us 4 hours. The hyperparameters are shown in the appendix for brevity.

2. Predict the values for the training set and have the minimum $p_{\min}$ and the gap between maximum and minimum as δ. We then have the cutoff value as cutoff values$= p_{\min} + \delta c = [d_0, d_1, d_2, d_3]$.

3. Predict the values for the validation set, and for each value v, we find the smallest i such that $v \leq d_i$. Then we set the predicted value as i. If no such i exists, we set the predicted value as 4.

This gives us an average of $\kappa = 0.4694$ with a standard error 0.01058 for a 5-fold crossvalidation, which is better than the Xgboost classifier.

4. Discussion and Implications

We do not use the image and the metadata in this project. We think the image and the metadata are important for the adoption speed. However, we do not have enough time to do the image analysis and the metadata analysis. We think the image analysis is a good direction for future research. Also, there is also a lot of information in the description of the pet. We think the description is also a good direction for future research. Though we don't use the image and the metadata, we think the result is good enough. A κ of 0.4694 is not bad since we only use 176 features to predict the adoption speed.

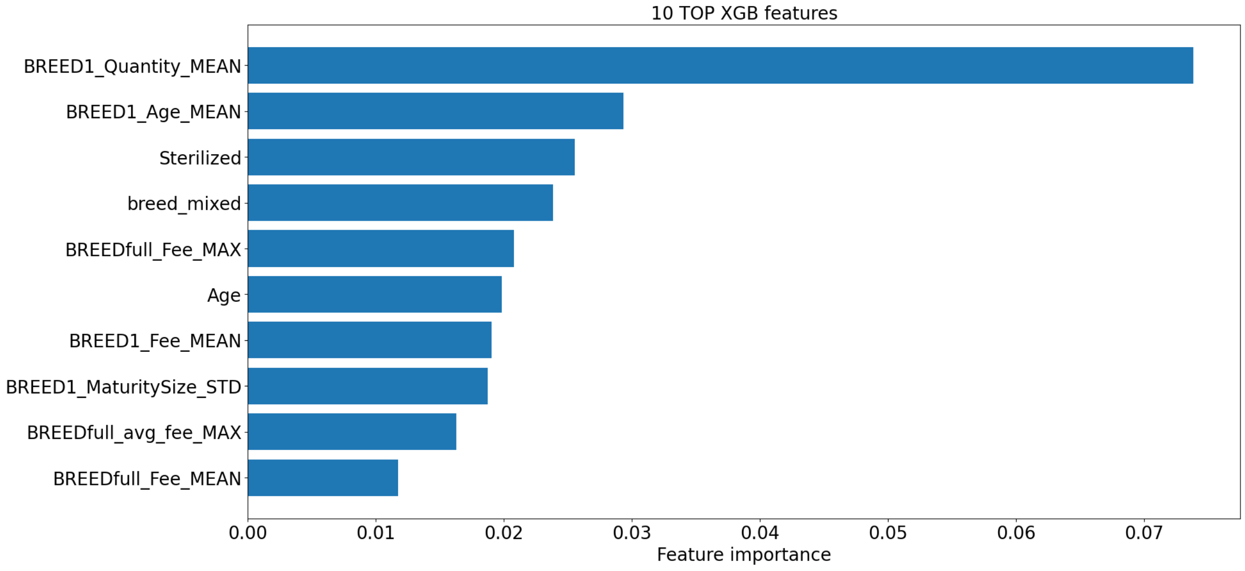

Figure 1 Feature importance

From Figure 1 we can see that, the main breed, i.e., Breed1, is the most important factor affecting the adoption speed. Also, whether is sterilized, whether is breed is mixed and the age of the pets are also important factors. This is intuitive since in reality, people tend to adopt pets with a younger age. Also, the sterilized status is also important since people tend to adopt pets that are sterilized. The color of the pet is not important. This is reasonable since the color of the pet is not a good indicator of the adoption speed. We can also see that, the adoption fee also matters.

Thus, to improve the adoption speed, we can do the following things:

- Highlight the main breed of the pet.
- Show mixed breed.
- Sterilize the pet.
- Highlight the age of the pet.
- Set a reasonable adoption fee.

Appendix

The code and the data is available at `https://github.com/Jinyi-Liu/BZAN645-Midterm`.