PetFinder.my Adoption Prediction

MLiP Erasmus Team

March 2019

1 Introduction

1.1 Petfinder.my Competition

This paper reports on the participation in the ongoing competition on the Kaggle platform, called PetFinder.my Adoption Prediction [1], based on the Malaysia's leading animal welfare platform PetFinder. The aim of the competition is the prediction of the adoption rate of rescued animals, in order to improve work of animal shelters and, consequently, to increase the adoption rate of pets.

1.2 Report Organization

This report is divided into two main sections. The first one describes the provided dataset, including the analysis, pre-processing and engineering of the data. The second section focuses on the models tested to predict the outcome, their performance analysis and the techniques we used to improve the score. Finally, a thorough analysis on the model that retrieved the best results is made.

2 Data Overview

The available dataset contains two subsets - train and test data, presenting information about 14994, respectively 3949 pets. Information about each pet is stored in different formats: tabular files, JSON format (for image metadata and sentiment analysis) and image data.

2.1 Data Analysis

The main tabular file presents a multitude of diverse features (22), which can be categorical (nominal, ordinal, ratio) and continuous (age, quantity, fee, video and photos amount). They belong to either dogs or cats. The features expressing color, breed and state of the rescued pet are encoded in 3 optional tabular files. The training set also presents the target label, which is the 'AdoptionSpeed' (a value between 0 and 4), as the submission file requires this value as indicator of the speed at which one pet is adopted.

Also, sentiment metadata was extrapolated from the available descriptions of the pets using the Google Natural Language API [2]. It reveals information such as: sentiment score (value between -1 and 1) and the sentiment magnitude - the power of sentiment - (value between 0 and infinity), for both the description overall and each sentence inside the description. //TODO: why the overall description data is sufficient - no need to consider each sentence? There is an interesting explanation on linearity in the google doc

Moreover, the image data, extremely varied, is present in two forms. On the one hand, there is image metadata, extracted from the images using the Google Vision API [3]. It includes dominant colors or details about objects recognized in the images. One pet can

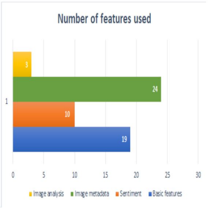

Figure 1: Available features

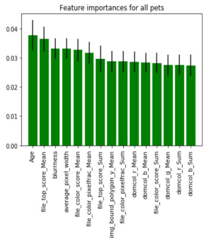

Figure 2: Feature importance using random forest algorithm

have numerous photos available (up to 30 in the training set). On the other hand, multiple images per pet are available in different sizes. In order to process them - extract additional features and feed them to convolutional neural networks -, pre-processing was performed to ensure uniform size and aspect ratio of the images.

Image number 1 presents the entire amount of features used for training the models using different subsets.

2.2 Feature Engineering

An essential step in data preparation before feeding it to different models was feature engineering. It turned out that this helped improving the score of the models significantly, as a large majority of them performed better when using the additional extracted data. This involved features extracted from different sources:

- the given tabular data - using composition (pure breed)
- the sentiment data
- the metadata
- the image dataset - using image processing techniques, the image quality was assessed and features such as blurriness, dullness, uniformity and whiteness were extracted [4]

//TODO: give examples of extracted features for each category and additional explanations

2.3 Feature Importance

During data analysis, feature importance has been analysed using different techniques, such as random decision forest, LGB or XGB. This way, the models were tested and analysed using different datasets containing selected features. Using the most relevant features visibly increased the accuracy of the models, which will be proved later in the model analysis section. Image number 2 reveals the importance determined by using the random forest algorithm.

//TODO: replace plot with the most recent importance analysis

3 Models

3.1 Models Created

Several models using different machine learning algorithms have been implemented using libraries such as scikit-learn [5], Tensorflow [6] and Keras [7], trained and tested, both locally and on Kaggle, using different feature subsets in order to optimize them (according to the feature importance analysis). Below there is an overview of the models used and the obtained results:

- Multilayer Perceptron Classifier
- Convolutional Neural Networks (performing both classification and regression)
- Random Forest Classifier, best accuracy 40%
- Support Vector Machines, best accuracy 38%
- Ordinal Logistic Regression, best accuracy 32%
- Gradient Boost Classifier, best accuracy 42%
- Ada Boost Classifier alone, best accuracy 38%
- Decision Tree Classifier + Grid search for optimal parameter choice and Ada Boost Classifier. best accuracy 34%
- Random Forest with Grid Search for optimal parameter choice, and Ada Boost as improvement of the Random Forest, best accuracy still 40%
- LGB
- XGB
- CAT

3.2 Combining Models

Combining several models that performed well outperformed each of the individual models tested, substantially improving the predictions. The following techniques were used to achieve this: //TODO: what models were combined + data set used (for the best score)

3.2.1 Voting

3.2.2 Weighted Voting

3.2.3 LGB, XGB and CAT

//TODO: Add diagrams

4 Conclusion

Present last (best) score What we have learned? Further ideas for improving?

References

[1] Kaggle.com. Petfinder.my adoption prediction — kaggle, 2019. Last accessed 09 March 2019.

[2] Google Cloud. Natural language api, 2019. Last accessed 09 March 2019.

[3] Google Cloud. The google vision api, 2019. Last accessed 09 March 2019.

[4] Kaggle.com. Ideas for image features and image quality — kaggle, 2019. Last accessed 10 March 2019.

[5] Scikit-learn.org. Documentation scikit-learn: machine learning in python — scikit-learn 0.20.3 documentation, 2019. Last accessed 10 March 2019.

[6] TensorFlow. Tensorflow guide, 2019. Last accessed 10 March 2019.

[7] Keras.io. About keras models - keras documentation, 2019. Last accessed 10 March 2019.