TERM PROJECT:
Research Project: Kaggle Competition
Data Mining and Knowledge Discovery (KSE525)

Yannis FLET-BERLIAC - s151399 (DTU)

Handed in the 22th of June

Abstract.

As part of the course KSE525 taught in KAIST by professor Jae-Gil Lee, Data Mining and Knowledge Discovery, the students have been given the opportunity to participate to the Kaggle competition named "Shelter Animal Outcomes". In that competition the participants were asked to make the best possible outcome predictions for a set of animals living in a shelter - the critical part here was to use the most relevant data analysis tools and select the most appropriate model to tackle the given problem.

1 Introduction

The aim for the competition is to help improve outcomes for shelter animals. According to several attributes corresponding to the animals (which we will discuss later in this report), the model has to be able to find the best prediction possible for the outcome of each of those animals. Given that, the shelter administrators will be able to understand trends in animal outcomes, gain access to useful insights and eventually help them – but also other shelters – to focus on specific animals who need particular attention or treatment.

First of all, we can rephrase the problem as this: we want to predict the outcome of the animals as they leave the Animal Center. By outcome, the problem means: Adoption, Died, Euthanasia, Return to owner, or Transfer. Then we will talk about the different attributes in the given data below, in the next section.

Adoption	Died	Euthanasia	Return to owner	Transfer

Table 1: Outcomes

2 The data

When we sign into the competition, different data is given: the training set, the test set, and a sample of submission to help us upload the right format of output.

In the training set are shown the outcomes (the targets) but not in the test set. At the submission procedure, we have to upload the predictions that our model retrieves from the test set – our model will be trained on the training set.

The training set is composed of 26730 rows and 10 columns. Each row represents an animal, each columns represents an attribute. Those attributes are as follows:

AnimalID	Name	DateTime	OutcomeType	OutcomeSubtype
AnimalType	SexuponOutcome	AgeuponOutcome	Breed	Color

Table 2: My caption

- AnimalID is the identification number of the animal.
- Name is the name of the animal.
- DateTime is the date and hour of the reported outcome.
- OutcomeType is the type of outcome (see above).
- OutcomeSubtype is some detail about the outcome.
- AnimalType is the type of the animal - either cat or dog.
- SexuponOutcome is the sex of the animal at the time of the reported outcome - either Neutered Male, Spayed Female, Intact Male, Intact Female or Unknown.
- AgeuponOutcome is the age of the animal at the time of the reported outcome.
- Breed is the breed of the animal.
- Color is the color of the animal.

3 Short note about the script

We will use Python for the whole project, please find the script "script.py" attached to the report. As a main framework we will use the package Pandas which helps to generate and work with tables. It is also very useful when it comes to convert table from or to .csv files.

The script has been commented each time a new function has been written for a better understanding. The script is divided into four parts: Importing needed Python packages, Loading the data, Processing the train and test sets and Generate the model.

4 Feature engineering

First of all, we decided to create features for weekdays, months and years. Indeed, depending on the weekday different outcomes may occur - the same can be observed depending on the month. For instance around Christmas the number of adoption may increase for expected reasons.

Then, one of the most massive impact over the outcome is probably the age of the animal. It has been translated into number of days. Moreover to emphasize the scale difference from cats and dogs compared to humans, the values have been engineered with a tanh function. Tanh(x) allows us to give more importance to the fact that animals get more adopted during their early years of life.

Another feature we decided to engineer was the name. Simply by showing if the animal has already a name, or not.

When it comes to the "value" of an animal, the color and the breed have a big impact over the outcome.

- The color data has been massively engineered so that we obtain information about what we can call the shade of the animal AND the color. One can easily discover that there are 8 shades: "Merle", "Brindle", "Tiger", "Smoke", "Cream", "Point", "Tick", and "Tabby". Features for each of them have been created meaning that for instance given the shade "Tiger", if an animal is "Tiger" the corresponding cell receives a True, if not the cell receives a False. The same process has been done for the colors.

- The breed data has also been engineered to extend the features for each animal. Some of them – similarly to their color attribute – have several of them. it will result in an animal firing with a True for instance for two columns corresponding to its to breeds. The "Mix" information has been independently used to again add some knowledge on the animal (eg. the function isMix(breeds) returns True if the breed is mixed, False if not).

In addition to that, for the breed and the color, a feature has been added for the number of different colors/breeds, because sometimes it is more relevant to see an animal as "mixed" or not (speaking both about color or breed) than to know exactly the kind of color/breed of the animal. Exactely the same process has been done with the word "Mix" attribute that we could find in some breed values of the set. Finally, some attributes have been found to be irrelevant such as 'AnimalID' and 'OutcomeSubtype' for obvious reasons regarding the AnimalID, and because it did not appear in the test set for OutcomeSubtype.

As an end-note, the number of days, months or years for each animal has been kept intentionally because one can think that when the shelter gives the information of how old an animal is, the number of days, months or years really means something for the potential adopters. The corresponding features have been created and the final trainings of the model showed better results when those latter were introduced.

5 The models

After pre-processing the data, we first decided to use the Random Forest Classifier from sklearn to test our data and make the very first submissions on Kaggle.

We split our training set with test_size = 0.2, created the Random Forest with n_estimators = 500 and max_features = 'auto'. After running it we obtained a score of 0.686 accuracy and a corresponding log-loss of 0.81114 on Kaggle.

But it was not that good, and considering the rankings of other participants, better could be done. So I decided to use a boosting model (XGBClassifier from sklearn). We had to take into account that it would introduce over-fitting. Hopefully, we can use within the XGBClassifier function the lambda parameter (regularization parameter) which will reduce the size of our model on training and thus reduce the risks of over-fitting. The objective function has been chosen to be "softprob" as the needed output was likelihood probabilities between the five different outcomes. Moreover, a quite huge amount of time has been spend to tune all of those parameter concluding that a good compromise was obtained for lambda = 3. Other parameters have been tuned such as the learning_rate (0.2), n_estimators (400) which is the number of boosting stages to perform in the process, max_depth (6) which is the maximum depth of an individual estimator, subsample (0.8) and colsample_bytree (0.8) which are respectively the fraction of sample and columns to be used in the training step. Those two latter were in part very useful again to avoid over-fitting.

6 Results

Another great functionality of the XGBoost Classifier from sklearn is the possibility to display the log-loss values as the model gets trained. It gave some insight on how the model was fitting the data and help understand the relevance, or irrelevance, of the different pre-processing parts that have been tested on the data (eg. the number of letters in an animal's name has finally been removed as the log-loss score finally improved when the corresponding function was block-commented).

After several attempts using the same model and trying to tune the parameters as best as possible, we obtained a log-loss score of 0.73369 on the public leader board.

7 Conclusion

Using a boosting model for this project finally gave enthusiastic results, and even more when the parameters were tuned. This highlighted one more time that it is of great importance to spend time tuning the parameters of the model we use. Another critical part has also been the feature engineering process by understanding the data, trying to use empathy to understand how the potential adopters may choose an animal, and also learn what matters when it comes to the characteristics of a dog or a cat. This was my first Kaggle competition and

it was a very rich experience, this made me very eager to try again to upgrade my results as much as possible before the final submission deadline.

8 Identity in the leader board

Please find bellow both my Kaggle Profile Page and my ranking on the Leader Board

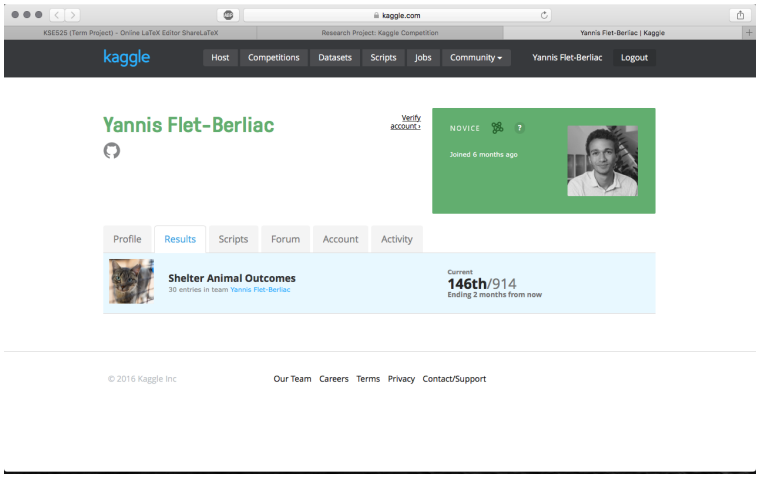

Figure 1: Kaggle's Profile Page

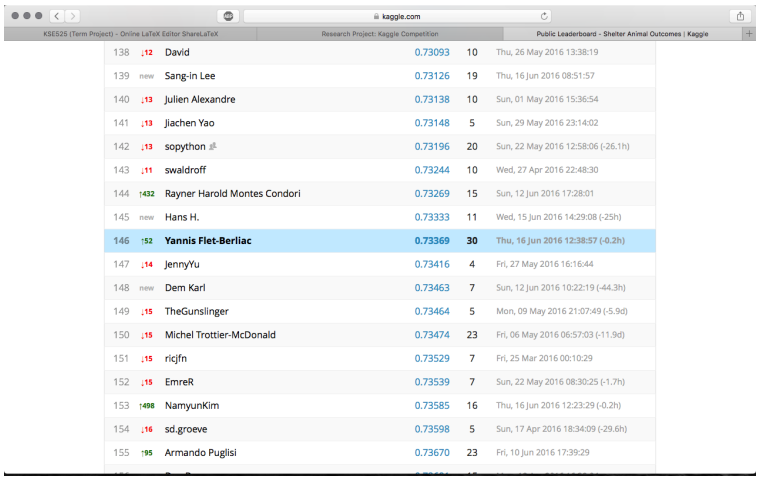

Figure 2: The Public Leader Board targeting my name