CS583A: Course Project

Hantao Ren, Xiaopeng Yuan, and Yunfan Ye

May 17, 2019

1 Summary

We pariticipe an active competition of prediction on adption speed on cats and dogs. The final model we choose is Ensemble model, a model integrated XGBoost, LightGBM and Neural Network. Performance is evaluated on RMSE. In the public leaderboard, our score is 0.41912; we rank 622 among the 2023 teams. In the private leaderboard, our score is 0.42292; we rank 220 among the 2023 teams.

2 Problem Description

Problem. 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 Malaysias 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. Animal adoption rates are strongly correlated to the metadata associated with their online profiles, such as descriptive text and photo characteristics. As one example, PetFinder is currently experimenting with a simple AI tool called the Cuteness Meter, which ranks how cute a pet is based on qualities present in their photos.

Features and Conclusions from EDA:

Adoption Spped. Target Variable: 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).

Pet Type. Cats are more likely to be adopted early than dogs and overall the percentage of not adopted cats is lower. It maybe because this dataset is small and could contain bias. On the other hand more dogs are adopted after several months.

Pet Names. Less than 10% of pets don't have names, but they have a higher possibility of not being adopted.

Age. Young pets are adopted quite fast and most of them are adopted.

Breeds. Non-pure breed pets tend to be adopted more and faster, especially cats.

Maturity Size. Medium sized pets are most common and they have slightly less chances to be adopted, and There are almost no Extra Large pets.

Fur Length. Most pets have short fur, few pets have long fur. Pets with long hair tend to have a higher chance of being adopted.

Quantity. The number of pets in one advertisement, quantity 1 is the most common number. This feature will become very useful after aggregation.

Fee. Most pets are free and it seems that asking for a fee slightly desreased the chance of adoption. Also free cats are adopted faster than free dogs

State. About 90% advertisements come from three states, and interestingly top-2 and top-3 states have lower rates of adoption.

PhotoAmt. The number of photoes appeared in the advertisement, also like quantity. This feature will become very useful after some aggregation.

Challenges. The most challege is to deal with three types of data: Image, Text and data frame.

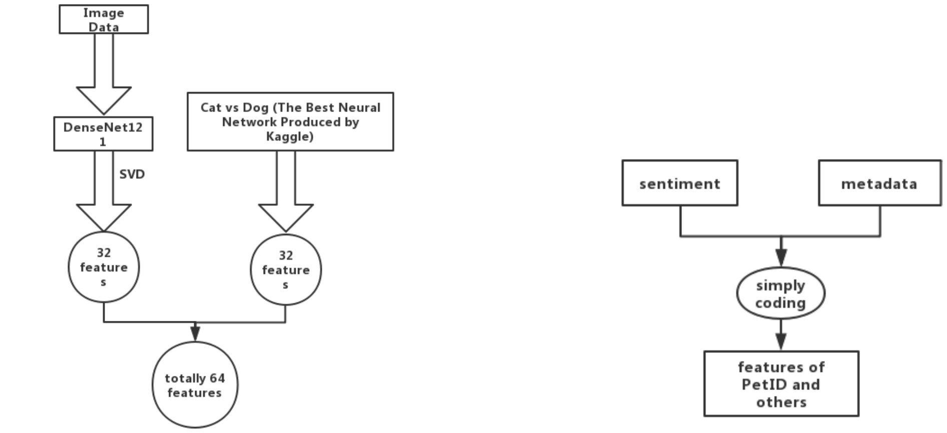
3 Feature Engineering

Image Data. We have use two networks to extract features, the first one is DenseNet121 and the second is the best Neural Network model from another Kaggle competition: Cat vs Dog. As you can see in Figure 1 (a).

JSON File Data. we have two kinds of json file, sentiment and metadata, they have PetID in common, so we just coding to connect them. As you can see in Figure 1 (b).

Text Data. We have three parts including text data, one comes from our regular description, the other two are from descriptions in metadata and sentiment JSON file. Here we have use a pretrained network again. As you can see in Figure 2 (a)

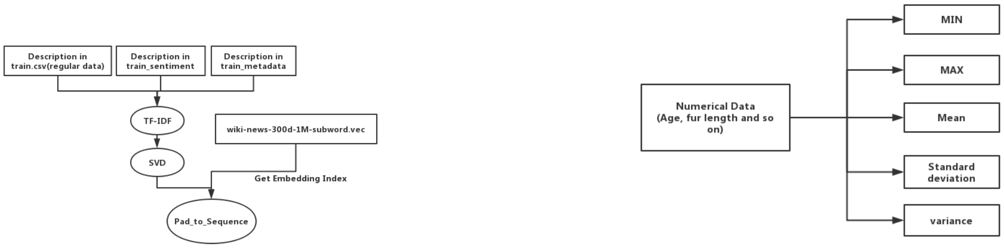
Other Attempts. Besides those special engineering, we also create many aggregation features, like the maximum, minimum, mean, standard deviation, and variance of most numerical data like age fur length, fee and so on. As you can see in Figure 2 (b).



Data.png File.png

(a) Image Data. (b) JSON File Data.

Figure 1: Feature Enginnering.

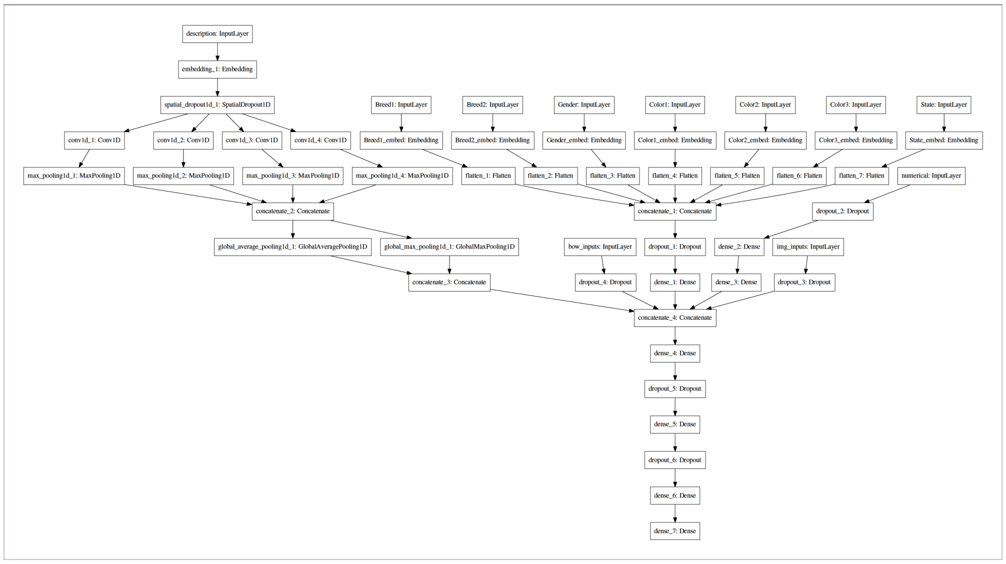

Description in train.csv(regular data)
Description in train_sentiment
Description in train_metadata
TF-IDF
SVD
wiki-news-300d-1M-subword.vec
Get Embedding Index
Pad_to_Sequence
Numerical Data (Age, fur length and so on)
MIN
MAX
Mean
Standard deviation
variance

Data.png Attempts.png

(a) Text Data. (b) Other Attempts.

Figure 2: Feature Enginnering.

Construction.png



(a) Neural Network.

Figure 3: Neural Network Graph.

4 Solution

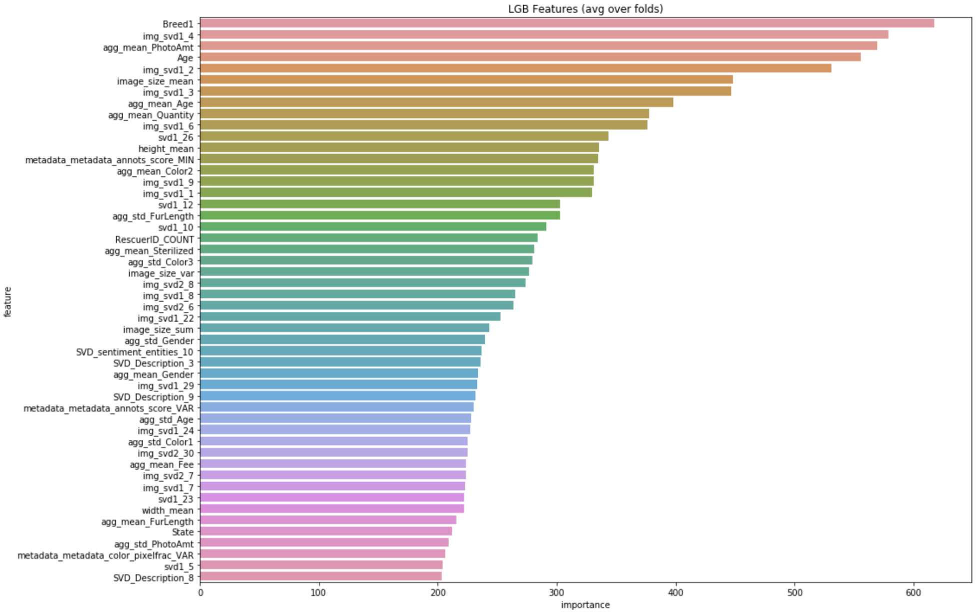
Model. We have applied XGBoost, LightGBM and Neural Network (NN) in our problems. For NN model, the approach used are embedding layer, CNN and fully connected layer.

Implementation. We implement the Neural Network using four parts: text, categorical data, numerical data and data after svd in images. As you can see the details in the Figure 3. Also, the method used to improve the performance is ensemble three models together on Ridge Regression model. The predictions from XGBoost, LightGBM and NN are the three new features on Ridge Regression to get new coefficients so that we can make predictions on test data. We tune the parameters using a 5-fold cross-validation on Bayes Search optimization approach. For XGBoost, we used the GPU to accelerate the speed on hyperparameters tuning and predictions.

Feature Importance. The most important features are from the original data such as image data and train data. As you can see in Figure 4.

Evaluation. First, we tune the parameters on RMSE and test the model performance on Quadratic Weighted Kappa which is our competition requirements and score calculations basis. To calculate Quadratic Weighted Kappa, we need to follow six steps.

Step 1. We shall be calculating a confusion matrix between the Predicted and Actual values. Here is a great resource to know more about confusion matrix.



Importance.png

(a) Feature Importance.

Figure 4: Feature Importance.

Step 2. Under step-2 each element is weighted. Predictions that are further away from actuals are marked harshly than predictions that are closer to actuals. We will have a less score if our prediction is 5 and actual is 3 as compared to a prediction of 4 in the same case.

Step 3. Step-3: We create two vectors, one for preds and one for actuals, which tells us how many values of each rating exist in both vectors.

Step 4. Step-4: E is the Expected Matrix which is the outer product of the two vectors calculated in step-3.

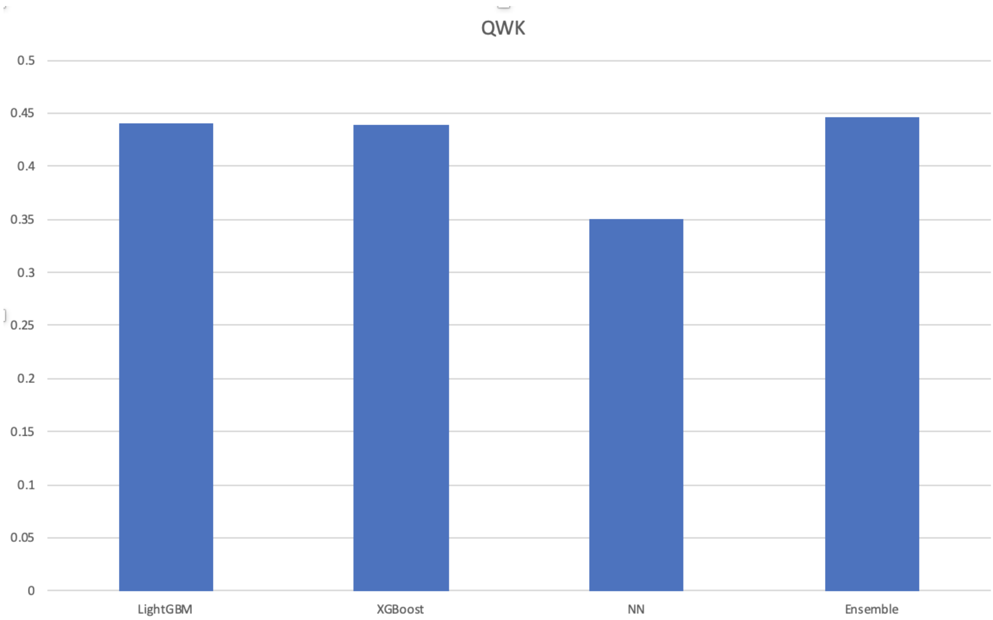
Step 5. Step-5: Normalise both matrices to have same sum. Since, it is easiest to get sum to be '1', we will simply divide each matrix by it's sum to normalise the data.

Step 6. Step-6: Calculated numerator and denominator of Weighted Kappa and return the Weighted Kappa metric as 1-(num/den

5 Compared Methods

The difference on performance among these models as shown on Figure 5.

Compared.png

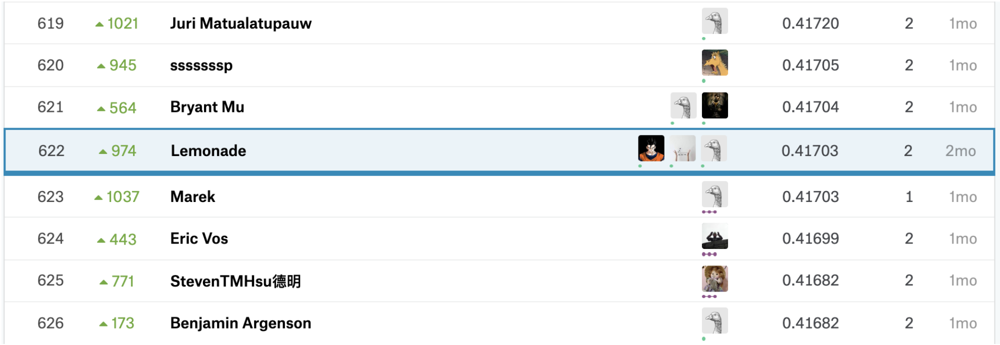


(a) Results.

Figure 5: Comparison Results Graph.

6 Outcome

We participated in an active competition. Our score is 0.42292 in the latesubmissions and 0.41912 in the final submission when the competition is active. We rank 622/2023 in the private leaderboard. We have raised the ranking to 10% on late submissions. The screenshots are in Figure 6.



619	▲1021	**Juri Matualatupauw**	0.41720	2	1mo
620	▲945	**sssssssp**	0.41705	2	1mo
621	▲564	**Bryant Mu**	0.41704	2	1mo
622	▲974	**Lemonade**	0.41703	2	2mo
623	▲1037	**Marek**	0.41703	1	1mo
624	▲443	**Eric Vos**	0.41699	2	1mo
625	▲771	**StevenTMHsu德明**	0.41682	2	1mo
626	▲173	**Benjamin Argenson**	0.41682	2	1mo

(a) Private leaderboard.

Figure 6: Our rankings in the leaderboard.

References

Kernel, "Kernel Paltform", Kaggle. Available from: https://www.kaggle.com/c/petfinder-adoption-prediction/kernels. [09 May 2019]