PetFinder: Predicting Adoption Speed of Pets

Jinyi Liu

Business Analytics & Statistics

December 6, 2023

THE UNIVERSITY OF TENNESSEE KNOXVILLE

Outline

1. Motivation

2. Experiment

3. (Very Preliminary) Results

4. Discussion

Outline

1. Motivation

2. Experiment

3. (Very Preliminary) Results

4. Discussion

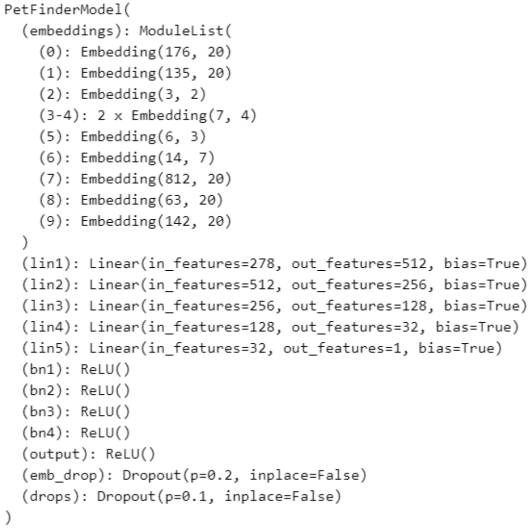
Motivation

1. In the midterm I used XGBoost to predict the adoption speed of pets. The quadratic weighted κ is 0.4694.
2. I did not consider the sparcity of the categorical variables such as the breed and color of the pets.
3. Therefore, in this project I employ an embedding layer to deal with the categorical variables and combine it with continuous features to build a neural network model.

Outline

1. Motivation
2. Experiment
3. (Very Preliminary) Results
4. Discussion

Experiment



```
PetFinderModel(
  (embeddings): ModuleList(
    (0): Embedding(176, 20)
    (1): Embedding(135, 20)
    (2): Embedding(3, 2)
    (3-4): 2 x Embedding(7, 4)
    (5): Embedding(6, 3)
    (6): Embedding(14, 7)
    (7): Embedding(812, 20)
    (8): Embedding(63, 20)
    (9): Embedding(142, 20)
  )
  (lin1): Linear(in_features=278, out_features=512, bias=True)
  (lin2): Linear(in_features=512, out_features=256, bias=True)
  (lin3): Linear(in_features=256, out_features=128, bias=True)
  (lin4): Linear(in_features=128, out_features=32, bias=True)
  (lin5): Linear(in_features=32, out_features=1, bias=True)
  (bn1): ReLU()
  (bn2): ReLU()
  (bn3): ReLU()
  (bn4): ReLU()
  (output): ReLU()
  (emb_drop): Dropout(p=0.2, inplace=False)
  (drops): Dropout(p=0.1, inplace=False)
)
```

Figure: Network

Outline

1. Motivation

2. Experiment

3. (Very Preliminary) Results

4. Discussion

(Very Preliminary) Results

Loss

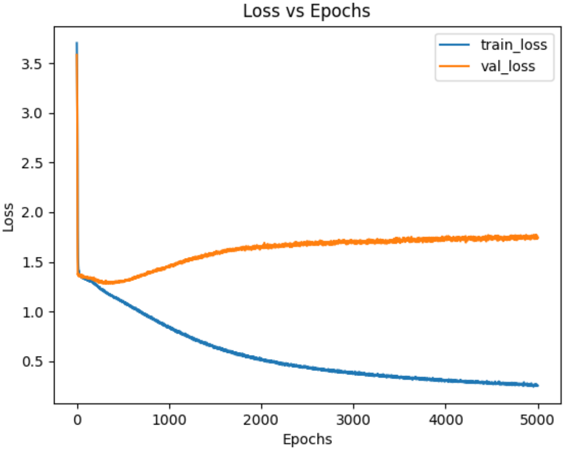


Figure: Loss (Stratified)

(Very Preliminary) Results

Weighted Kappa

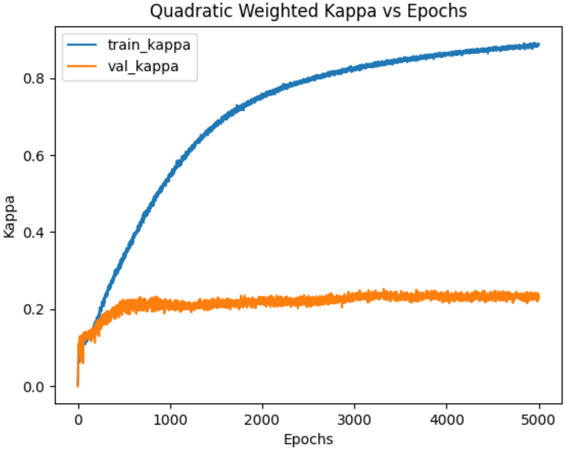


Figure: Kappa (Stratified)

Outline

1. Motivation
2. Experiment
3. (Very Preliminary) Results
4. Discussion

Discussion

1. The model is not good enough since the κ score is lower than the XGBoost model.
2. During the feature engineering process, many NaN values are generated. We need to find a way to deal with them in neural network since XGBoost can handle NaN values without any problem.
3. I try to fill the NaN by 0 and mean value, but the result is not good.
4. I might need to enlarge the network and add more layers.