Kaggle:
Petfinder.my Adoption Prediction

Prepared by :   Ruochi Zhang, Qi Lu, Xiaoqian Xu

CONTENTS

01 Introduction

02 Data Overview

03 Implementation

04 Result & Conclusion

01 Introduction

Introduction

Problem Description

- This Competition was initiated by PetFinder.my, which is a Malaysia's leading animal welfare platform aims at helping more pets find their home.
- In this competition, we should build a model to **predict the adoption speed** (0-4) category of 3984 pets in the test set. This is a **classification problem**.
- The accuracy is measured by **quadratic weighted kappa**.

Dataset Description

5 different types of dataset：

- **01. Main dataset** (csv files: 24 features, 14993 observations in the training set)
- **02. Images** (jpg files: 58311 images) → Google Vision API → **03. Image Metadata** (json)
- **04. Descriptions** → Google Natural language processing API → **05. Sentiment Data** (json)

Features

→

Target：adoption speed

0 - on the same day

1 - between 1 and 7 days (1st week)

2 - between 8 and 30 days (1st month)

3 - between 31 and 90 days (2nd & 3rd month)

4 - no adoption after 100 days

02 Data Overview

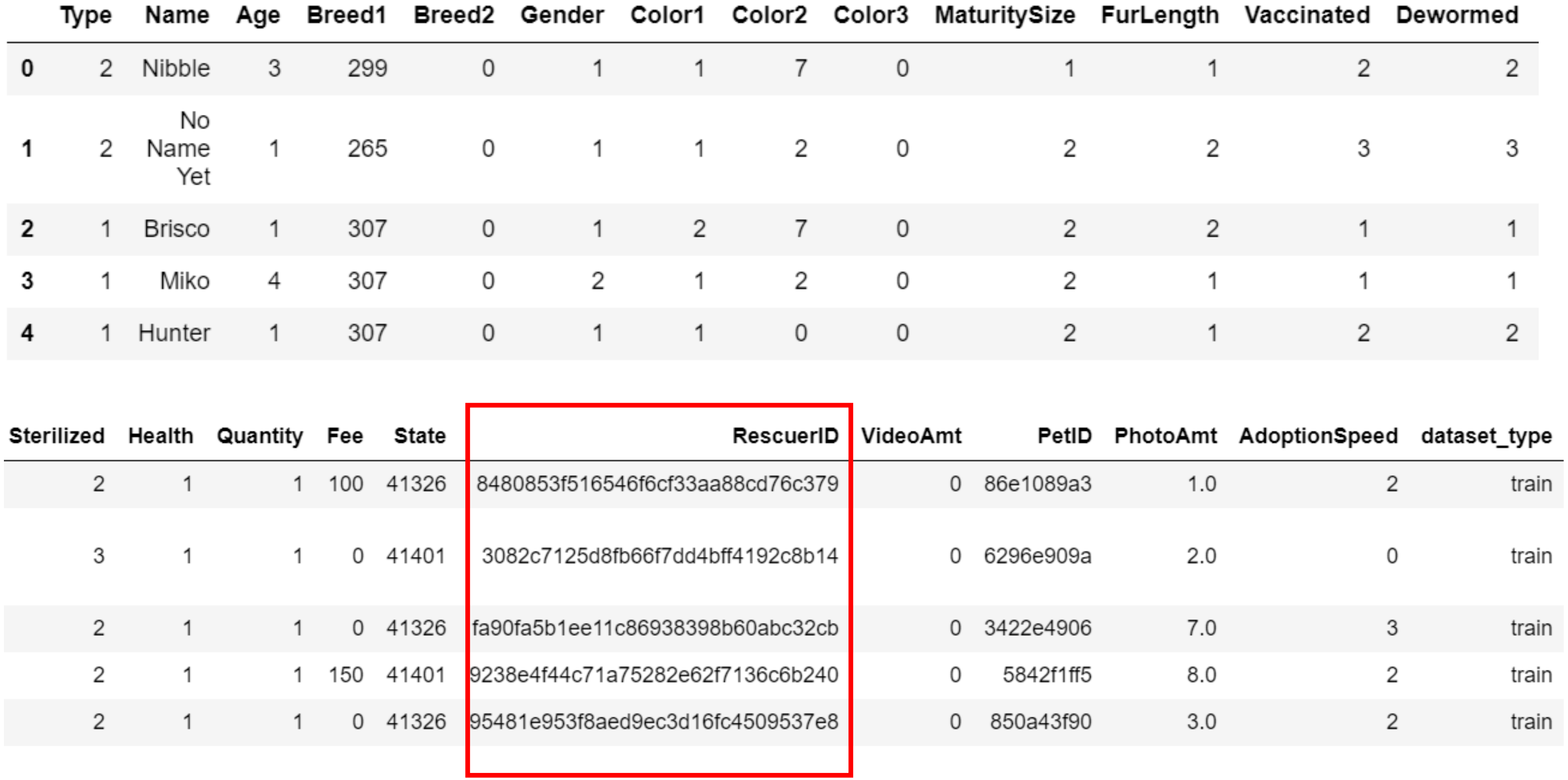
Data overview

Main dataset

Images & Image metadata

Description & Sentiment data

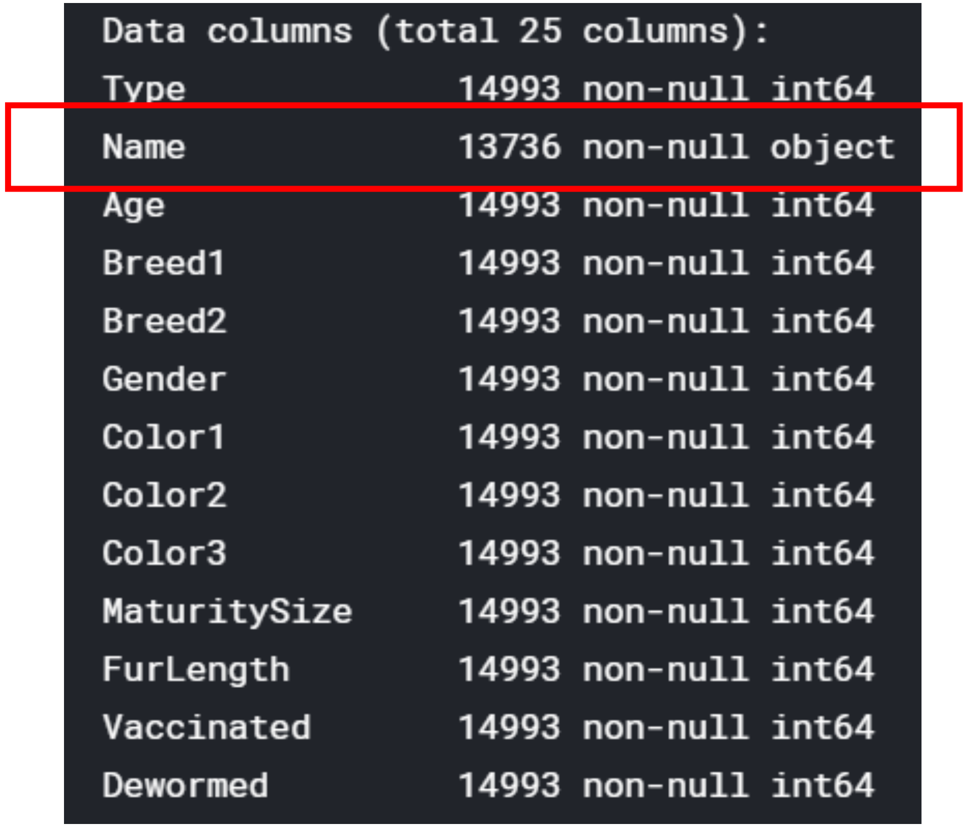
Data overview- main dataset



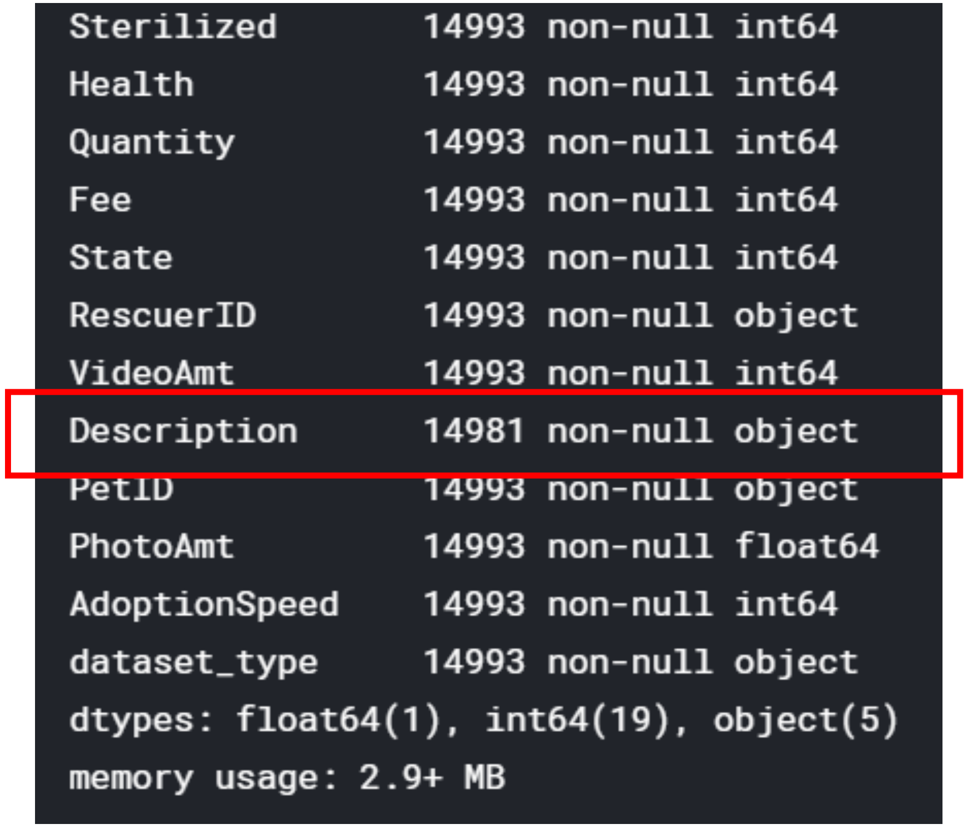
	Type	Name	Age	Breed1	Breed2	Gender	Color1	Color2	Color3	MaturitySize	FurLength	Vaccinated	Dewormed
0	2	Nibble	3	299	0	1	1	7	0	1	1	2	2
1	2	No Name Yet	1	265	0	1	1	2	0	2	2	3	3
2	1	Brisco	1	307	0	1	2	7	0	2	2	1	1
3	1	Miko	4	307	0	2	1	2	0	2	1	1	1
4	1	Hunter	1	307	0	1	1	0	0	2	1	2	2

Sterilized	Health	Quantity	Fee	State	RescuerID	VideoAmt	PetID	PhotoAmt	AdoptionSpeed	dataset_type
2	1	1	100	41326	8480853f516546f6cf33aa88cd76c379	0	86e1089a3	1.0	2	train
3	1	1	0	41401	3082c7125d8fb66f7dd4bff4192c8b14	0	6296e909a	2.0	0	train
2	1	1	0	41326	fa90fa5b1ee11c86938398b60abc32cb	0	3422e4906	7.0	3	train
2	1	1	150	41401	9238e4f44c71a75282e62f7136c6b240	0	5842f1ff5	8.0	2	train
2	1	1	0	41326	95481e953f8aed9ec3d16fc4509537e8	0	850a43f90	3.0	2	train

Data overview- main dataset

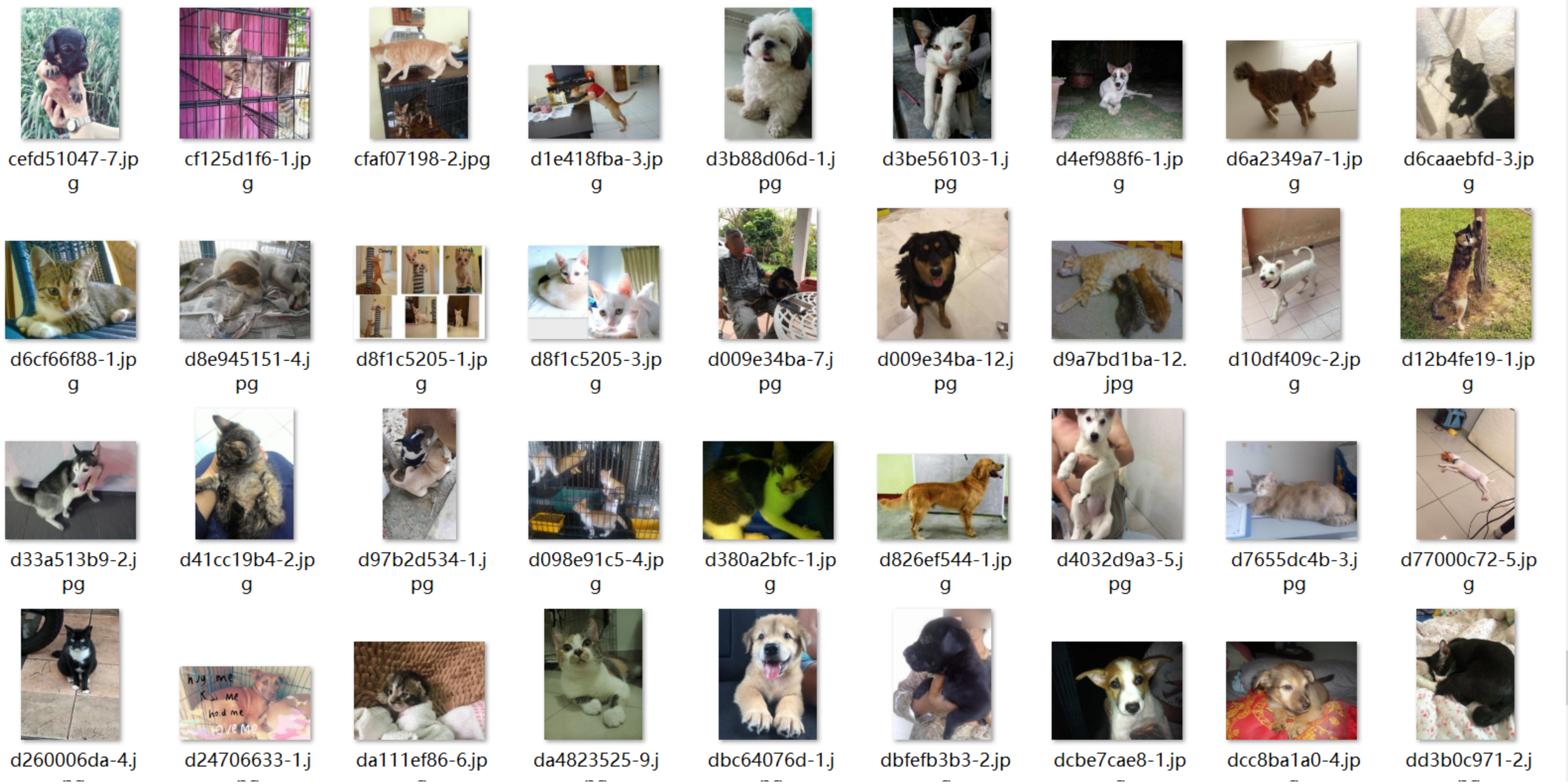


```
Data columns (total 25 columns):
Type             14993 non-null int64
Name             13736 non-null object
Age              14993 non-null int64
Breed1           14993 non-null int64
Breed2           14993 non-null int64
Gender           14993 non-null int64
Color1           14993 non-null int64
Color2           14993 non-null int64
Color3           14993 non-null int64
MaturitySize     14993 non-null int64
FurLength        14993 non-null int64
Vaccinated       14993 non-null int64
Dewormed         14993 non-null int64
```

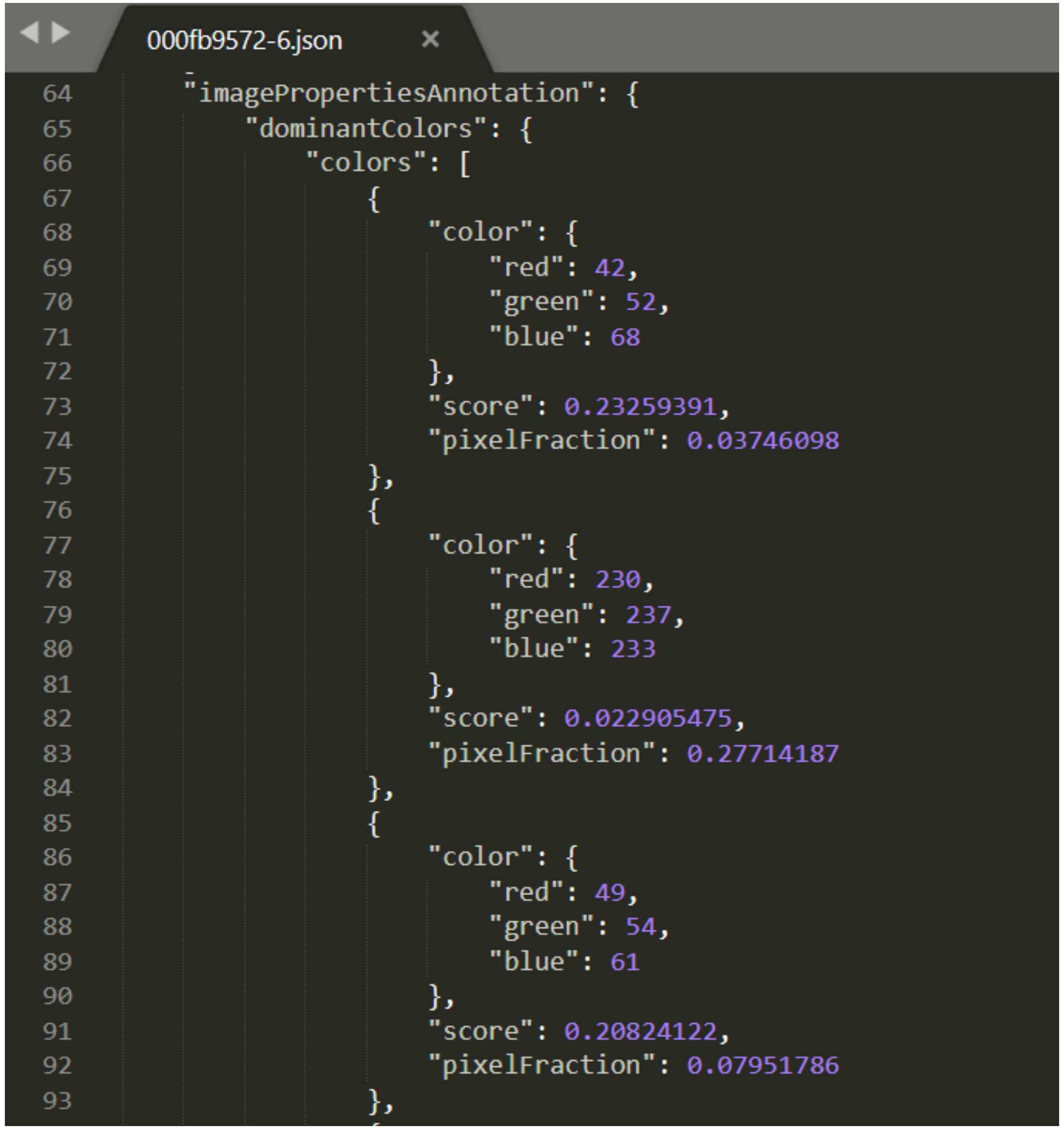


```
Sterilized       14993 non-null int64
Health           14993 non-null int64
Quantity         14993 non-null int64
Fee              14993 non-null int64
State            14993 non-null int64
RescuerID        14993 non-null object
VideoAmt         14993 non-null int64
Description      14981 non-null object
PetID            14993 non-null object
PhotoAmt         14993 non-null float64
AdoptionSpeed    14993 non-null int64
dataset_type     14993 non-null object
dtypes: float64(1), int64(19), object(5)
memory usage: 2.9+ MB
```

Data overview- images

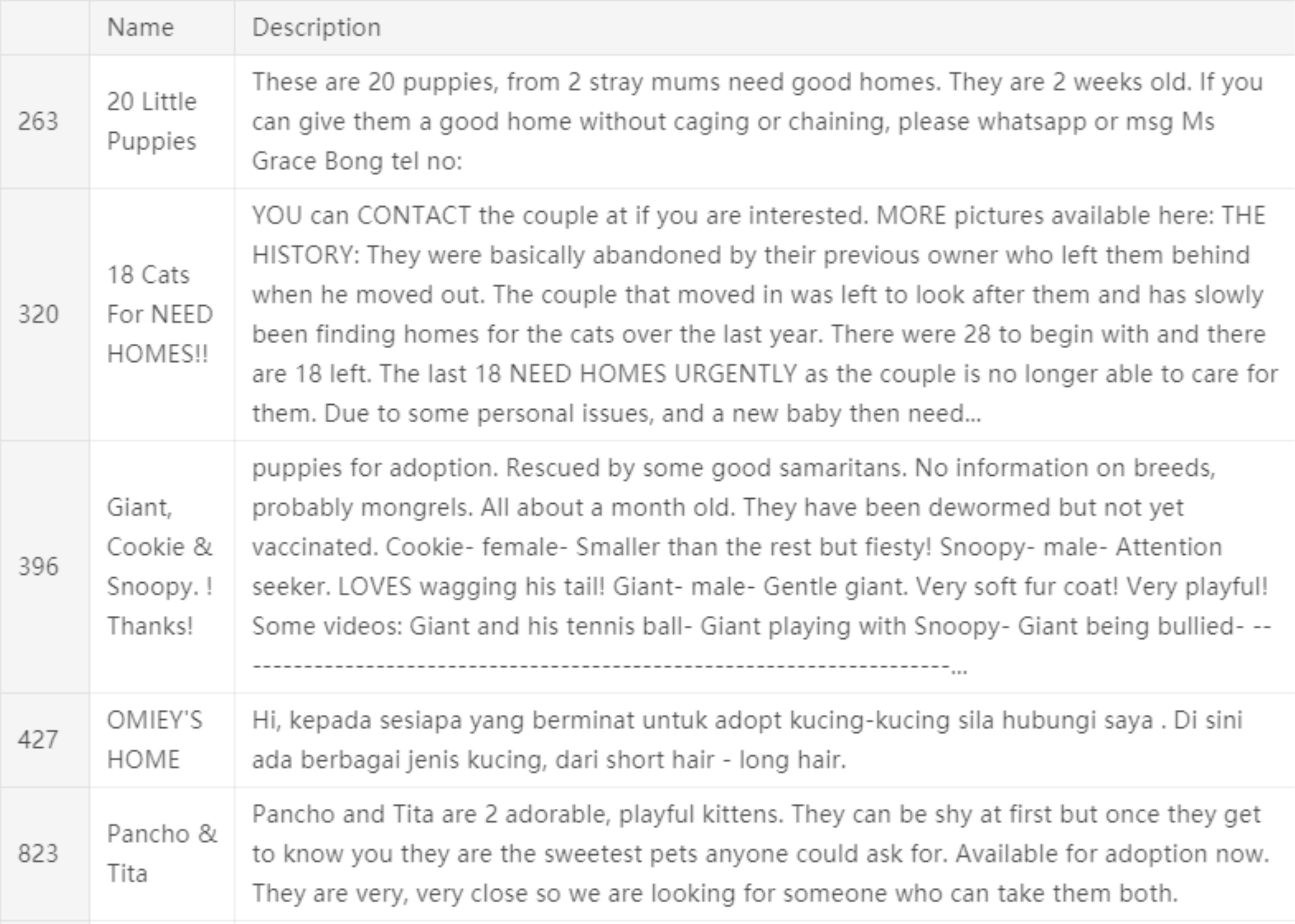


Data overview- image metadata



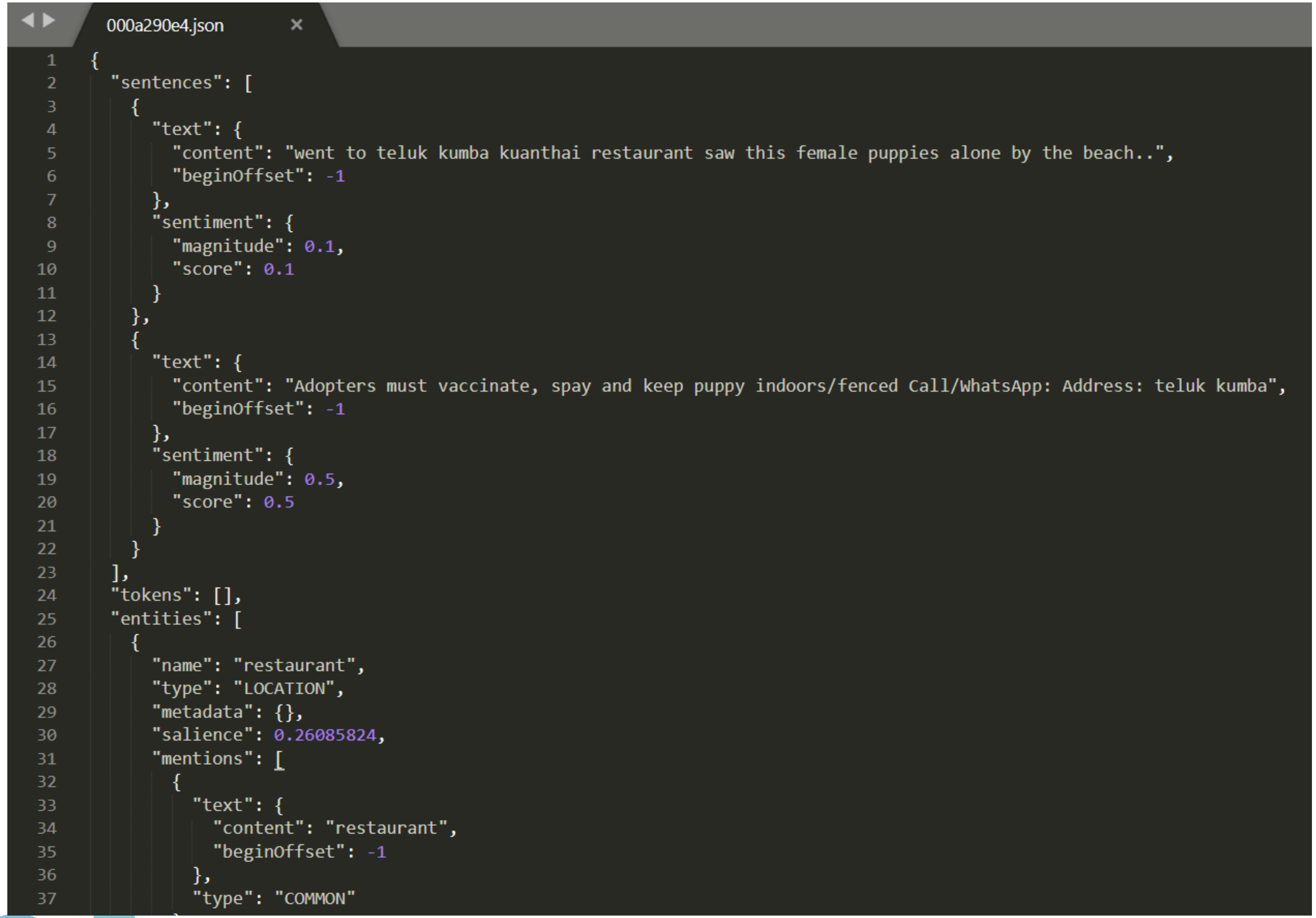
```
    "imagePropertiesAnnotation": {
        "dominantColors": {
            "colors": [
                {
                    "color": {
                        "red": 42,
                        "green": 52,
                        "blue": 68
                    },
                    "score": 0.23259391,
                    "pixelFraction": 0.03746098
                },
                {
                    "color": {
                        "red": 230,
                        "green": 237,
                        "blue": 233
                    },
                    "score": 0.022905475,
                    "pixelFraction": 0.27714187
                },
                {
                    "color": {
                        "red": 49,
                        "green": 54,
                        "blue": 61
                    },
                    "score": 0.20824122,
                    "pixelFraction": 0.07951786
                },
```

Data overview- description



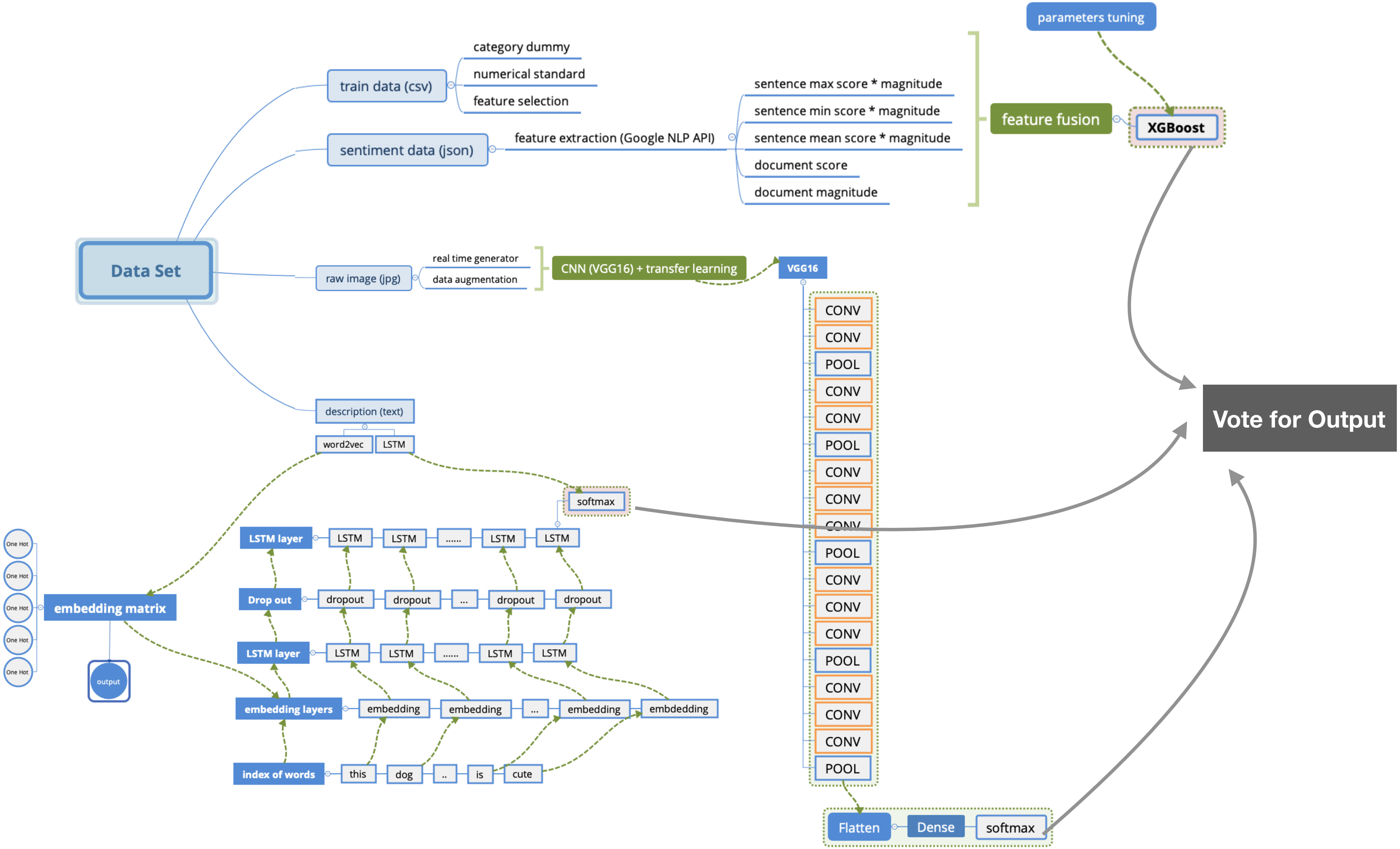
	Name	Description
263	20 Little Puppies	These are 20 puppies, from 2 stray mums need good homes. They are 2 weeks old. If you can give them a good home without caging or chaining, please whatsapp or msg Ms Grace Bong tel no:
320	18 Cats For NEED HOMES!!	YOU can CONTACT the couple at if you are interested. MORE pictures available here: THE HISTORY: They were basically abandoned by their previous owner who left them behind when he moved out. The couple that moved in was left to look after them and has slowly been finding homes for the cats over the last year. There were 28 to begin with and there are 18 left. The last 18 NEED HOMES URGENTLY as the couple is no longer able to care for them. Due to some personal issues, and a new baby then need...
396	Giant, Cookie & Snoopy. ! Thanks!	puppies for adoption. Rescued by some good samaritans. No information on breeds, probably mongrels. All about a month old. They have been dewormed but not yet vaccinated. Cookie- female- Smaller than the rest but fiesty! Snoopy- male- Attention seeker. LOVES wagging his tail! Giant- male- Gentle giant. Very soft fur coat! Very playful! Some videos: Giant and his tennis ball- Giant playing with Snoopy- Giant being bullied- -- ---...
427	OMIEY'S HOME	Hi, kepada sesiapa yang berminat untuk adopt kucing-kucing sila hubungi saya . Di sini ada berbagai jenis kucing, dari short hair - long hair.
823	Pancho & Tita	Pancho and Tita are 2 adorable, playful kittens. They can be shy at first but once they get to know you they are the sweetest pets anyone could ask for. Available for adoption now. They are very, very close so we are looking for someone who can take them both.

Data overview- sentiment data



```json
{
  "sentences": [
    {
      "text": {
        "content": "went to teluk kumba kuanthai restaurant saw this female puppies alone by the beach..",
        "beginOffset": -1
      },
      "sentiment": {
        "magnitude": 0.1,
        "score": 0.1
      }
    },
    {
      "text": {
        "content": "Adopters must vaccinate, spay and keep puppy indoors/fenced Call/WhatsApp: Address: teluk kumba",
        "beginOffset": -1
      },
      "sentiment": {
        "magnitude": 0.5,
        "score": 0.5
      }
    }
  ],
  "tokens": [],
  "entities": [
    {
      "name": "restaurant",
      "type": "LOCATION",
      "metadata": {},
      "salience": 0.26085824,
      "mentions": [
        {
          "text": {
            "content": "restaurant",
            "beginOffset": -1
          },
          "type": "COMMON"
```

03 Implementation



parameters tuning

category dummy

numerical standard

feature selection

train data (csv)

sentence max score * magnitude

sentence min score * magnitude

sentence mean score * magnitude

document score

document magnitude

feature fusion

XGBoost

sentiment data (json)

feature extraction (Google NLP API)

Data Set

raw image (jpg)

real time generator

data augmentation

CNN (VGG16) + transfer learning

VGG16

CONV

CONV

POOL

CONV

CONV

POOL

CONV

CONV

CONV

POOL

CONV

CONV

CONV

POOL

CONV

CONV

CONV

POOL

Flatten

Dense

softmax

Vote for Output

description (text)

word2vec

LSTM

softmax

One Hot

One Hot

One Hot

One Hot

One Hot

embedding matrix

output

LSTM layer

LSTM

LSTM

......

LSTM

LSTM

Drop out

dropout

dropout

...

dropout

dropout

LSTM layer

LSTM

LSTM

......

LSTM

LSTM

embedding layers

embedding

embedding

...

embedding

embdedding

index of words

this

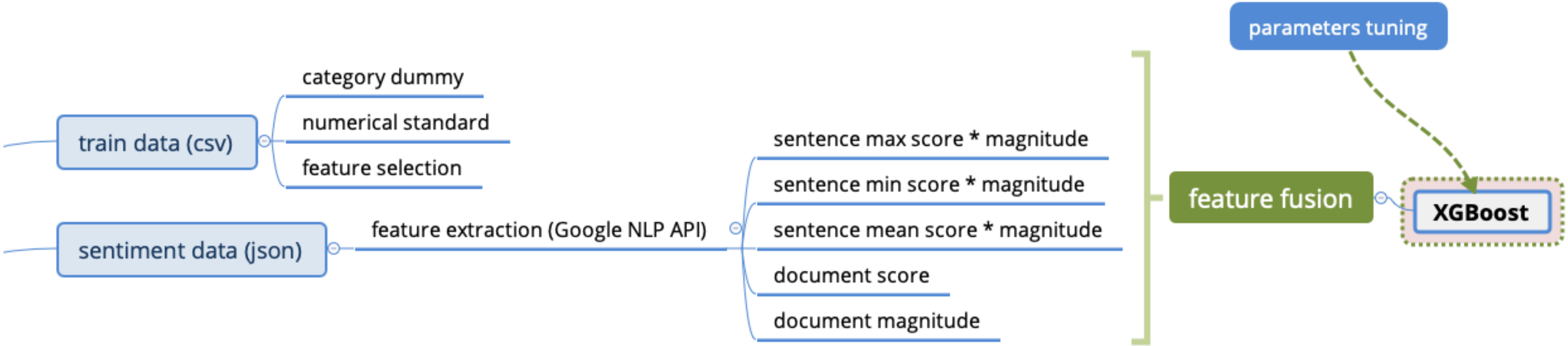
dog

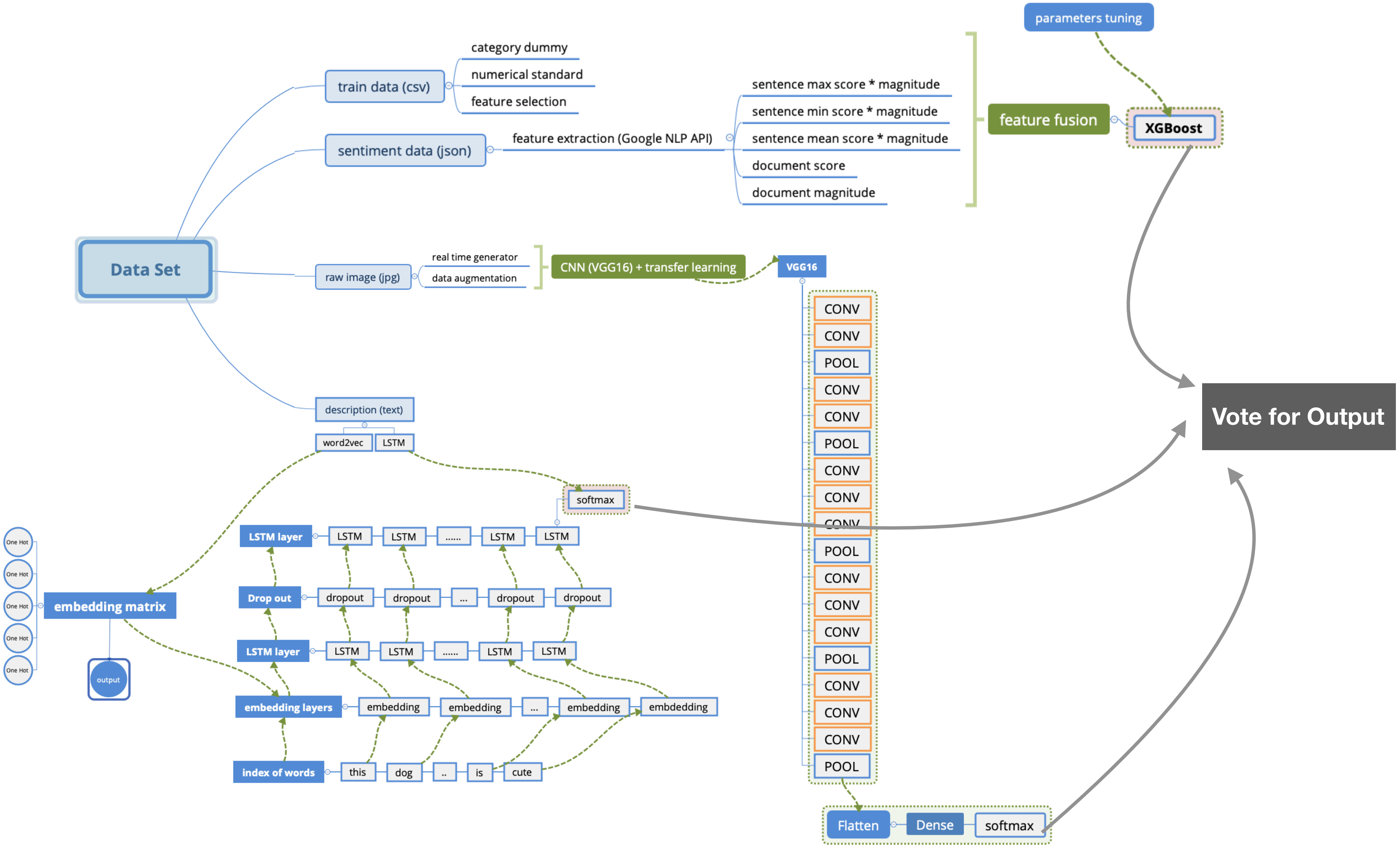
..

is

cute

Model 1


parameters tuning
train data (csv)
category dummy
numerical standard
feature selection
sentiment data (json)
feature extraction (Google NLP API)
sentence max score * magnitude
sentence min score * magnitude
sentence mean score * magnitude
document score
document magnitude
feature fusion
XGBoost



Data Set

train data (csv)

category dummy

numerical standard

feature selection

sentiment data (json)

feature extraction (Google NLP API)

sentence max score * magnitude

sentence min score * magnitude

sentence mean score * magnitude

document score

document magnitude

feature fusion

parameters tuning

XGBoost

raw image (jpg)

real time generator

data augmentation

CNN (VGG16) + transfer learning

VGG16

CONV

CONV

POOL

CONV

CONV

POOL

CONV

CONV

CONV

POOL

CONV

CONV

CONV

POOL

CONV

CONV

CONV

POOL

Flatten

Dense

softmax

description (text)

word2vec

LSTM

softmax

One Hot

One Hot

One Hot

One Hot

One Hot

embedding matrix

output

LSTM layer

LSTM

LSTM

......

LSTM

LSTM

Drop out

dropout

dropout

...

dropout

dropout

LSTM layer

LSTM

LSTM

......

LSTM

LSTM

embedding layers

embedding

embedding

...

embedding

embdedding

index of words

this

dog

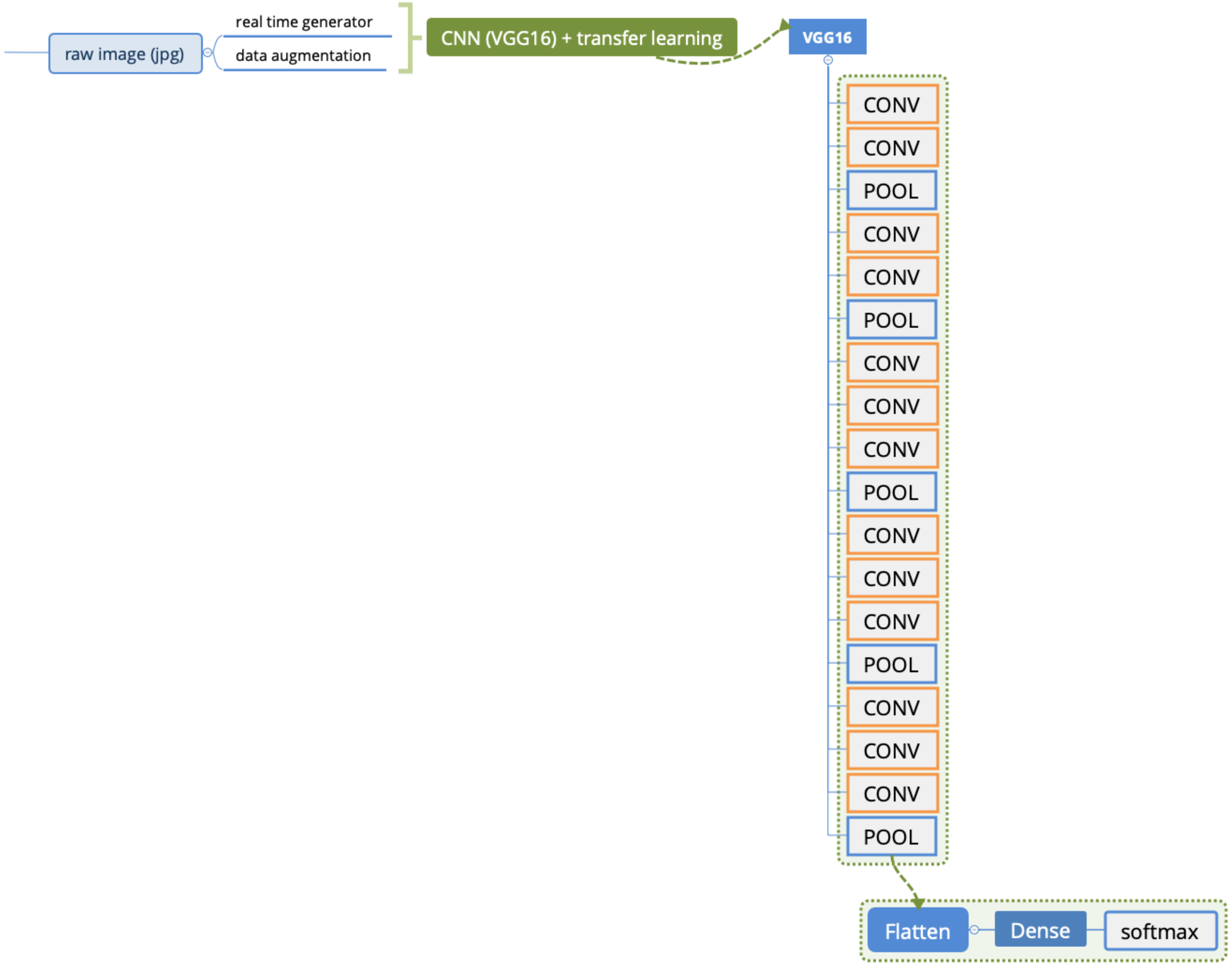
..

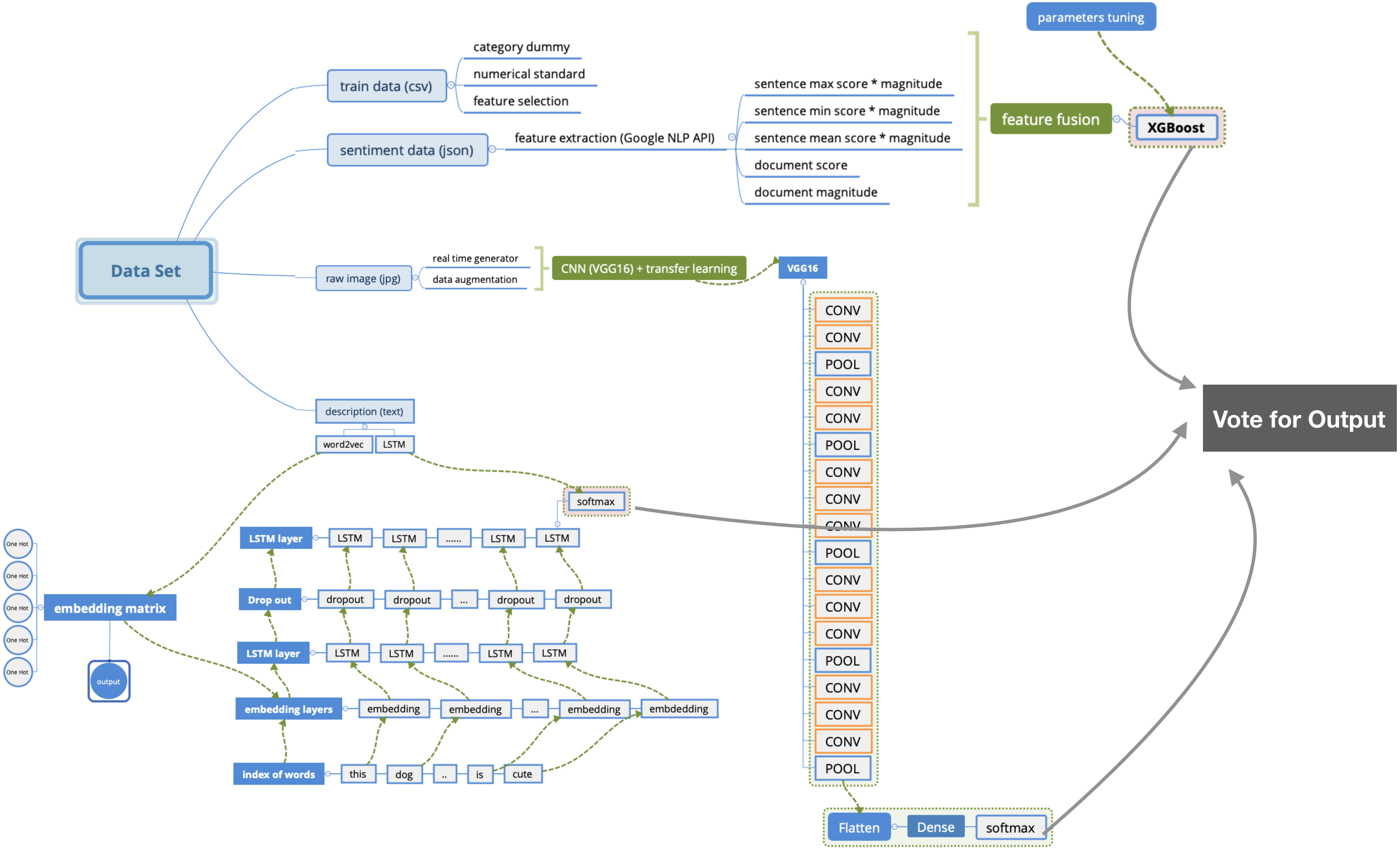
is

cute

Vote for Output

Model 2


raw image (jpg)
real time generator
data augmentation
CNN (VGG16) + transfer learning
VGG16
CONV
CONV
POOL
CONV
CONV
POOL
CONV
CONV
CONV
POOL
CONV
CONV
CONV
POOL
CONV
CONV
CONV
POOL
Flatten
Dense
softmax



parameters tuning

Data Set

train data (csv)

category dummy

numerical standard

feature selection

sentiment data (json)

feature extraction (Google NLP API)

sentence max score * magnitude

sentence min score * magnitude

sentence mean score * magnitude

document score

document magnitude

feature fusion

XGBoost

raw image (jpg)

real time generator

data augmentation

CNN (VGG16) + transfer learning

VGG16

CONV

CONV

POOL

CONV

CONV

POOL

CONV

CONV

CONV

POOL

CONV

CONV

CONV

POOL

CONV

CONV

CONV

POOL

Flatten

Dense

softmax

Vote for Output

description (text)

word2vec

LSTM

softmax

One Hot

One Hot

One Hot

One Hot

One Hot

embedding matrix

output

LSTM layer

LSTM

LSTM

......

LSTM

LSTM

Drop out

dropout

dropout

...

dropout

dropout

LSTM layer

LSTM

LSTM

......

LSTM

LSTM

embedding layers

embedding

embedding

...

embedding

embdedding

index of words

this

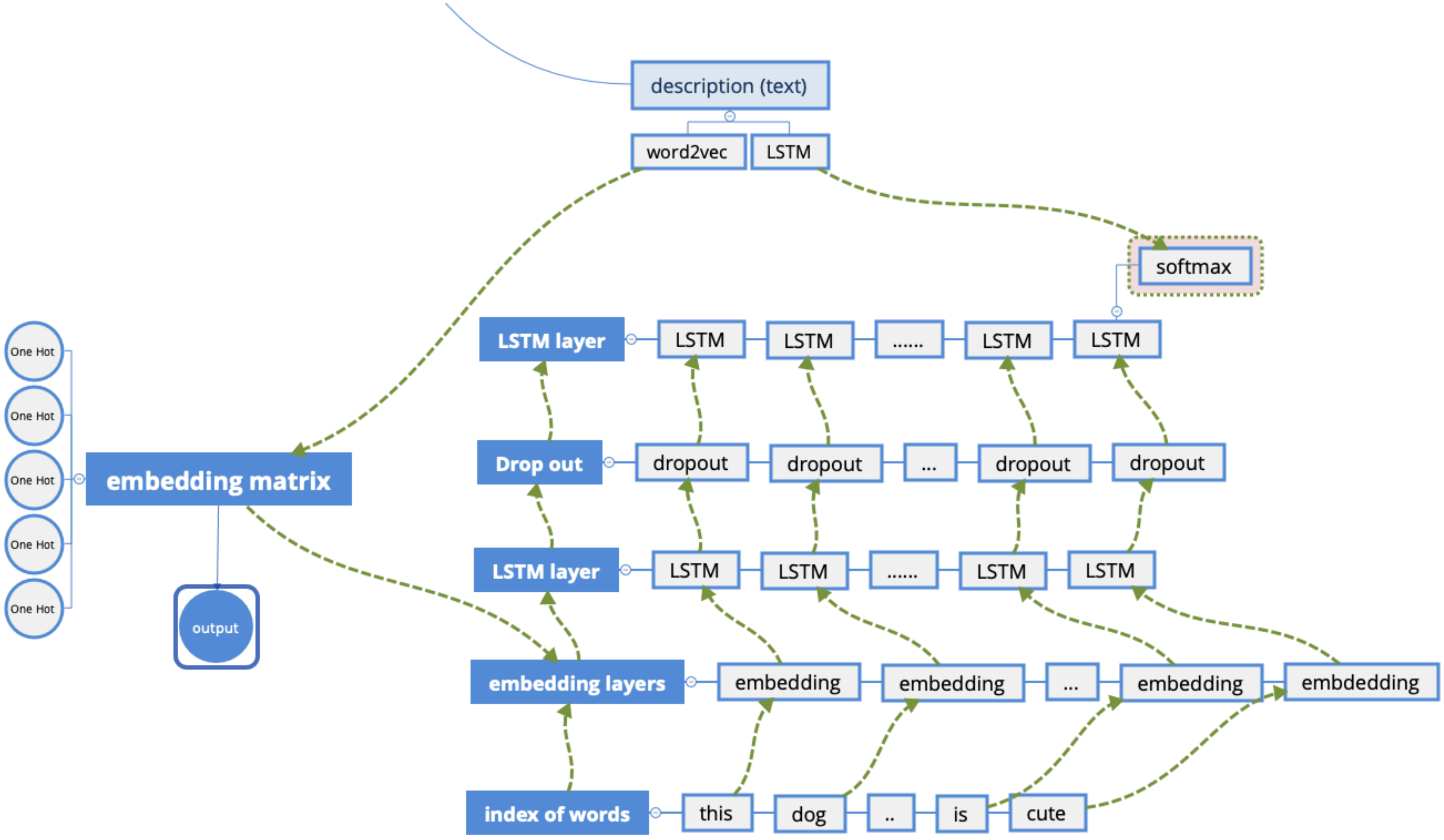
dog

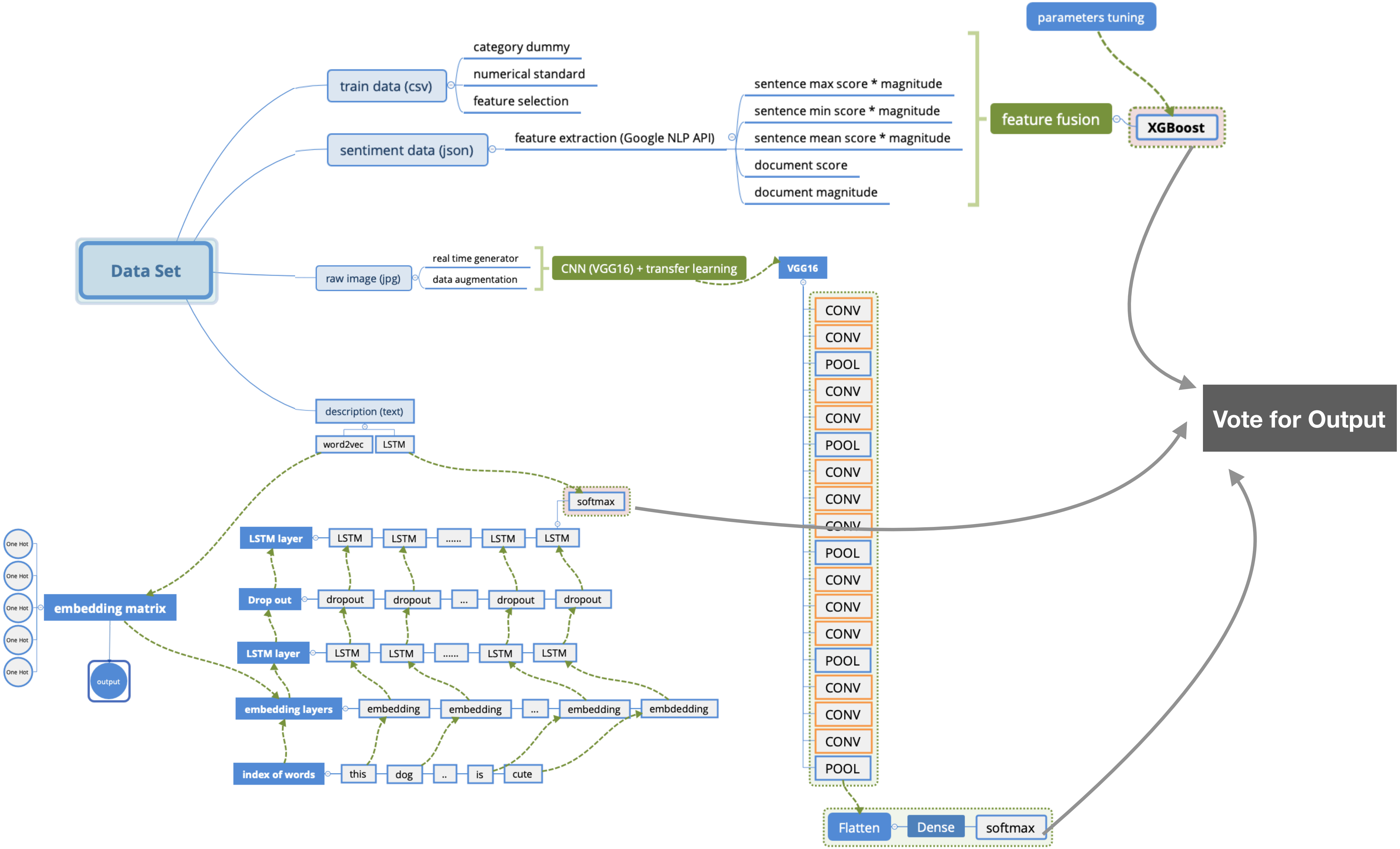
..

is

cute

Model 3


description (text)
word2vec
LSTM
softmax
One Hot
One Hot
One Hot
One Hot
One Hot
embedding matrix
output
LSTM layer
LSTM
LSTM
......
LSTM
LSTM
Drop out
dropout
dropout
...
dropout
dropout
LSTM layer
LSTM
LSTM
......
LSTM
LSTM
embedding layers
embedding
embedding
...
embedding
embdedding
index of words
this
dog
..
is
cute



Data Set

train data (csv)

category dummy

numerical standard

feature selection

sentiment data (json)

feature extraction (Google NLP API)

sentence max score * magnitude

sentence min score * magnitude

sentence mean score * magnitude

document score

document magnitude

feature fusion

parameters tuning

XGBoost

raw image (jpg)

real time generator

data augmentation

CNN (VGG16) + transfer learning

VGG16

CONV

CONV

POOL

CONV

CONV

POOL

CONV

CONV

CONV

POOL

CONV

CONV

CONV

POOL

CONV

CONV

CONV

POOL

Flatten

Dense

softmax

Vote for Output

description (text)

word2vec

LSTM

softmax

LSTM layer

LSTM

LSTM

......

LSTM

LSTM

Drop out

dropout

dropout

...

dropout

dropout

LSTM layer

LSTM

LSTM

......

LSTM

LSTM

embedding layers

embedding

embedding

...

embedding

embdedding

index of words

this

dog

..

is

cute

One Hot

One Hot

One Hot

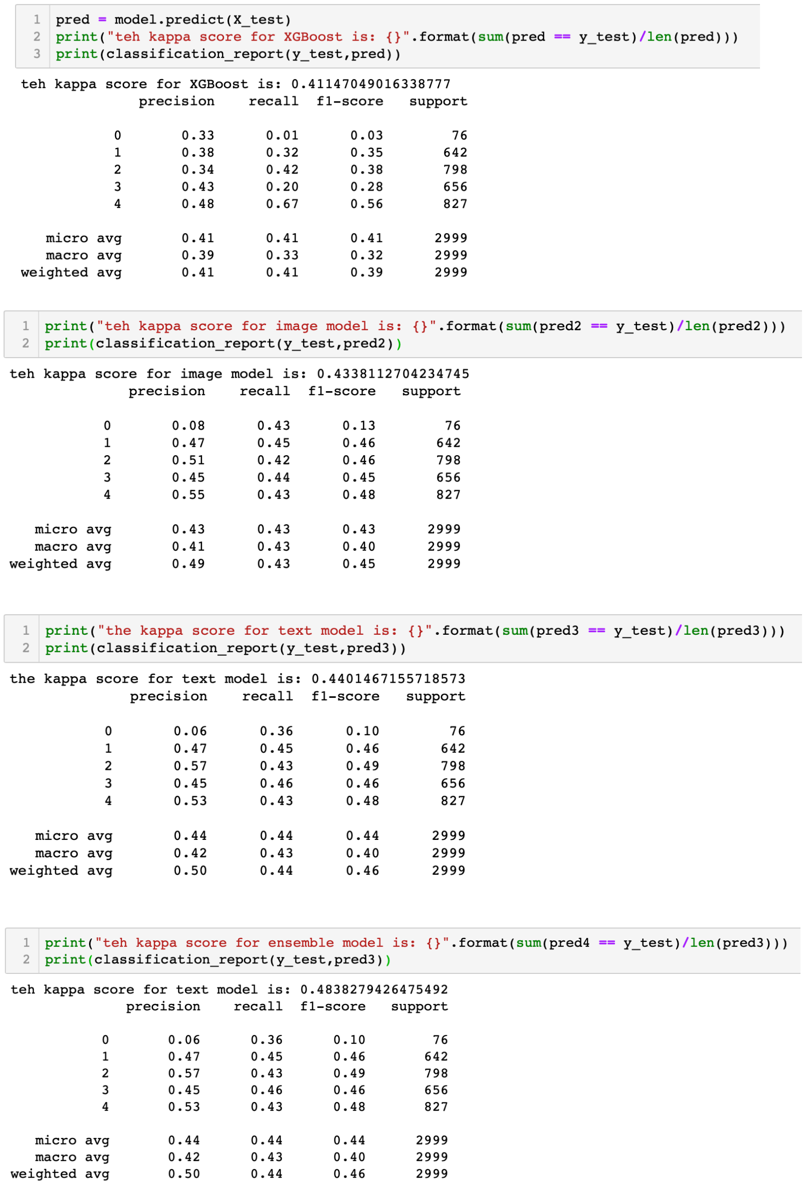
One Hot

One Hot

embedding matrix

output

04 Result & Conclusion



```
pred = model.predict(X_test)
print("teh kappa score for XGBoost is: {}".format(sum(pred == y_test)/len(pred)))
print(classification_report(y_test,pred))
```

```
teh kappa score for XGBoost is: 0.41147049016338777
              precision    recall  f1-score   support

           0       0.33      0.01      0.03        76
           1       0.38      0.32      0.35       642
           2       0.34      0.42      0.38       798
           3       0.43      0.20      0.28       656
           4       0.48      0.67      0.56       827

   micro avg       0.41      0.41      0.41      2999
   macro avg       0.39      0.33      0.32      2999
weighted avg       0.41      0.41      0.39      2999
```

```
print("teh kappa score for image model is: {}".format(sum(pred2 == y_test)/len(pred2)))
print(classification_report(y_test,pred2))
```

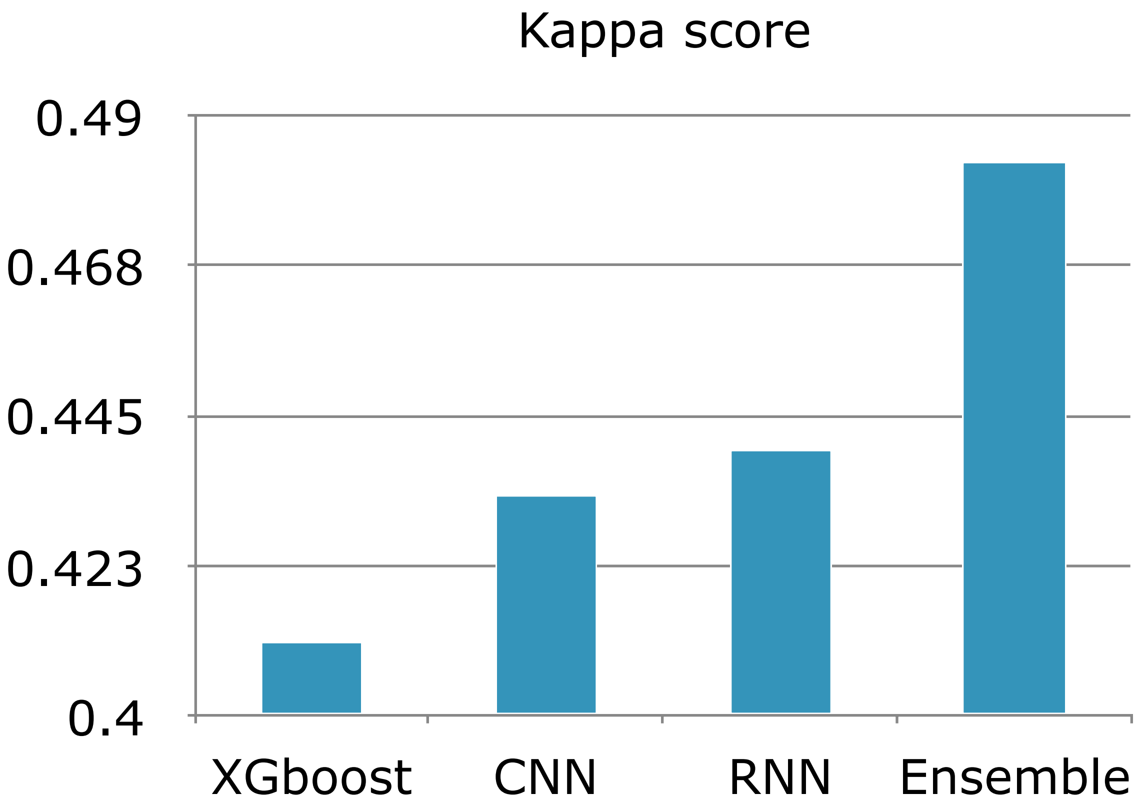
```
teh kappa score for image model is: 0.4338112704234745
              precision    recall  f1-score   support

           0       0.08      0.43      0.13        76
           1       0.47      0.45      0.46       642
           2       0.51      0.42      0.46       798
           3       0.45      0.44      0.45       656
           4       0.55      0.43      0.48       827

   micro avg       0.43      0.43      0.43      2999
   macro avg       0.41      0.43      0.40      2999
weighted avg       0.49      0.43      0.45      2999
```

```
print("the kappa score for text model is: {}".format(sum(pred3 == y_test)/len(pred3)))
print(classification_report(y_test,pred3))
```

```
the kappa score for text model is: 0.4401467155718573
              precision    recall  f1-score   support

           0       0.06      0.36      0.10        76
           1       0.47      0.45      0.46       642
           2       0.57      0.43      0.49       798
           3       0.45      0.46      0.46       656
           4       0.53      0.43      0.48       827

   micro avg       0.44      0.44      0.44      2999
   macro avg       0.42      0.43      0.40      2999
weighted avg       0.50      0.44      0.46      2999
```

```
print("teh kappa score for ensemble model is: {}".format(sum(pred4 == y_test)/len(pred3)))
print(classification_report(y_test,pred3))
```

```
teh kappa score for text model is: 0.4838279426475492
              precision    recall  f1-score   support

           0       0.06      0.36      0.10        76
           1       0.47      0.45      0.46       642
           2       0.57      0.43      0.49       798
           3       0.45      0.46      0.46       656
           4       0.53      0.43      0.48       827

   micro avg       0.44      0.44      0.44      2999
   macro avg       0.42      0.43      0.40      2999
weighted avg       0.50      0.44      0.46      2999
```



Reference

1. Wind, D. K., & Winther, O. (2014). Model selection in data analysis competitions. In 21st European Conference on Artificial Intelligence (ECAI 2014).
2. Liaw, A., & Wiener, M. (2002). Classification and regression by randomForest. R news, 2(3), 18-22.
3. Tong, S., & Chang, E. (2001, October). Support vector machine active learning for image retrieval. In Proceedings of the ninth ACM international conference on Multimedia (pp. 107-118). ACM.
4. Chen, T., & Guestrin, C. (2016, August). Xgboost: A scalable tree boosting system. In Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining (pp. 785-794). ACM.
5. Su, C. T., Chiu, C. C., & Chang, H. H. (2000). Parameter design optimization via neural network and genetic algorithm. INTERNATIONAL JOURNAL OF INDUSTRIAL ENGINEERING-THEORY APPLICATIONS AND PRACTICE, 7(3), 224-231.
6. Ben-David, A. (2008). Comparison of classification accuracy using Cohen’s Weighted Kappa. Expert Systems with Applications, 34(2), 825-832.
7. Li, N., & Wu, D. D. (2010). Using text mining and sentiment analysis for online forums hotspot detection and forecast. Decision support systems, 48(2), 354-368.
8. Liu, B., & Zhang, L. (2012). A survey of opinion mining and sentiment analysis. In Mining text data (pp. 415-463). Springer, Boston, MA.

Reference

9. Schmidhuber, J. (2015). Deep learning in neural networks: An overview. Neural networks, 61, 85-117.
10. Girija, S. S. (2016). Tensorflow: Large-scale machine learning on heterogeneous distributed systems.
11. Kosinski, M., Stillwell, D., & Graepel, T. (2013). Private traits and attributes are predictable from digital records of human behavior. Proceedings of the National Academy of Sciences, 201218772.
12. Goel, S., Hofman, J. M., Lahaie, S., Pennock, D. M., & Watts, D. J. (2010). Predicting consumer behavior with Web search. Proceedings of the National academy of sciences.
13. onahue, J., Anne Hendricks, L., Guadarrama, S., Rohrbach, M., Venugopalan, S., Saenko, K., & Darrell, T. (2015). Long-term recurrent convolutional networks for visual recognition and description. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 2625-2634).
14. Einav, L., Finkelstein, A., Mullainathan, S., & Obermeyer, Z. (2018). Predictive modeling of US health care spending in late life. Science, 360(6396), 1462-1465.
15. Guyon, I., & Elisseeff, A. (2003). An introduction to variable and feature selection. Journal of machine learning research, 3(Mar), 1157-1182.

THANK YOU FOR WATCHING