PetFinder

張家華 李柏霆 鄭子萱 謝昕庭 許聖謙

Outline

- Problems
- EDA
- Data Pre-processing
- Methods & Evaluation
- Model Tunning
- Recommendations

Problems

1

Problem Adoption efficiency of stray pets

Strategy Predict adoption speed

Goal

Improve adoption rate.
Assist in the allocation of shelter resources.
Solve the problem of excessive stray pets in shelters.

Statistical Learning Goal

Predicting the probability of AdoptionSpeed

Classification

Supervised Learning

Data Description

Rows: 14993

Categorical : 18

Numerical : 5

Text: 1

Outcome variable

AdoptionSpeed

Methods

Classification

- KNN
- Logistic regression
- Naïve Bayes
- Guassian Process
- Bagging
- Random Forest
- XGboost

Explore Data Analysis

2

Data type

Continuous :

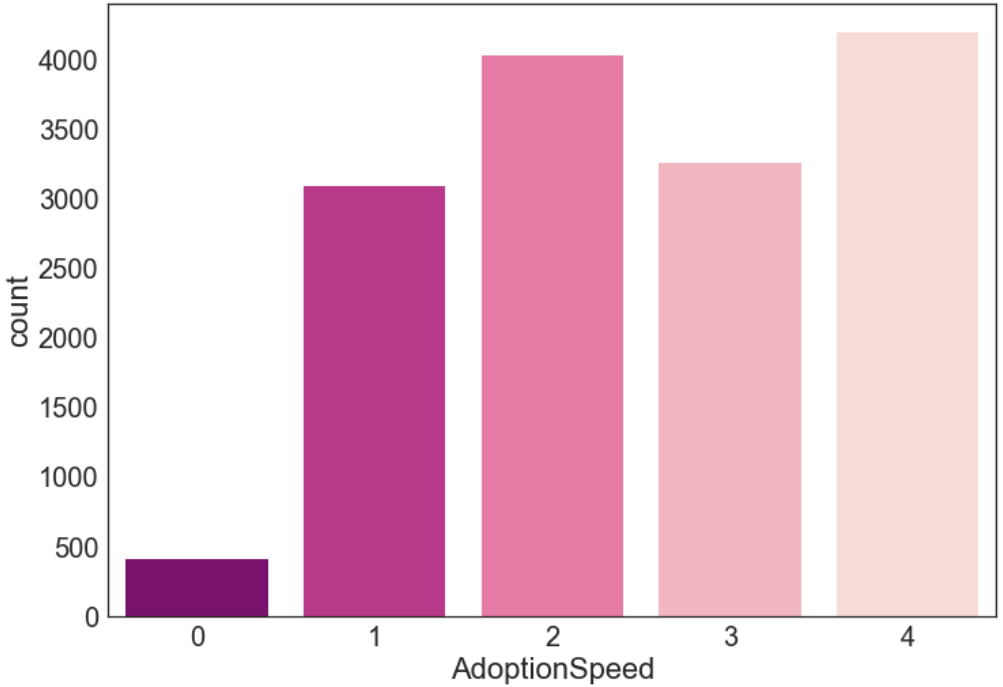
Age, Quantity, PhotoAmt, VideoAmt, Fee,

state_gdp, state_population(Derived variable)

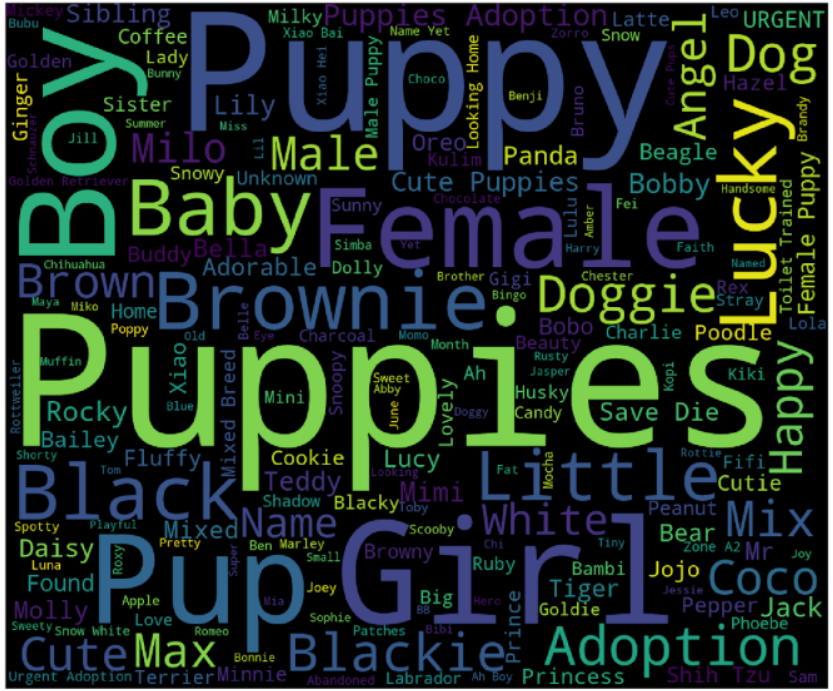
Discrete:

Most variables are discrete type.

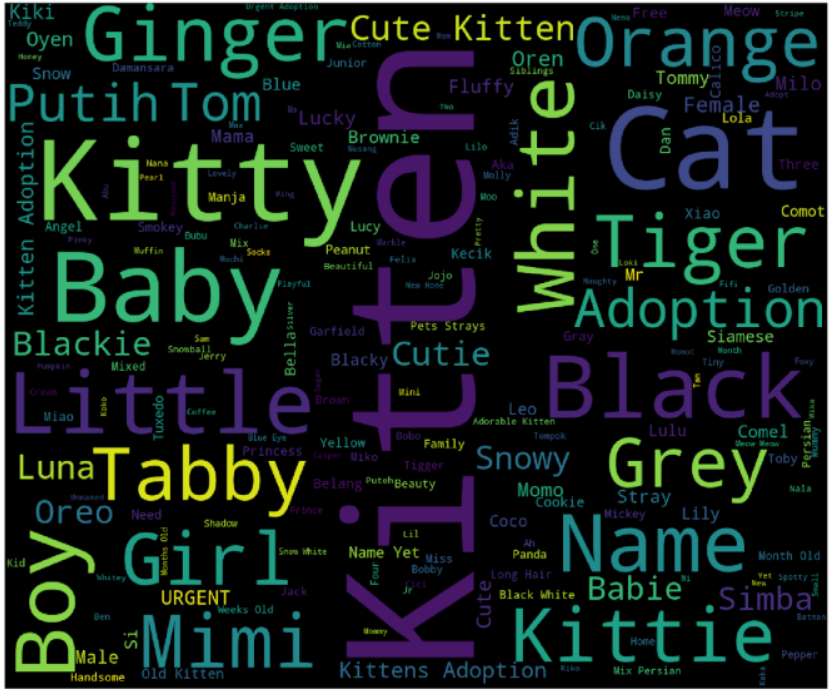
EDA- Adoption Speed



EDA- Name

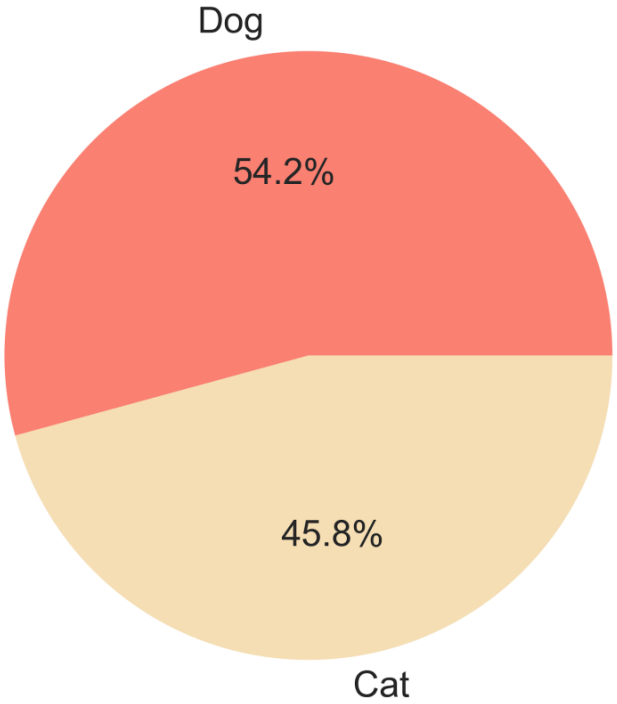


Dog

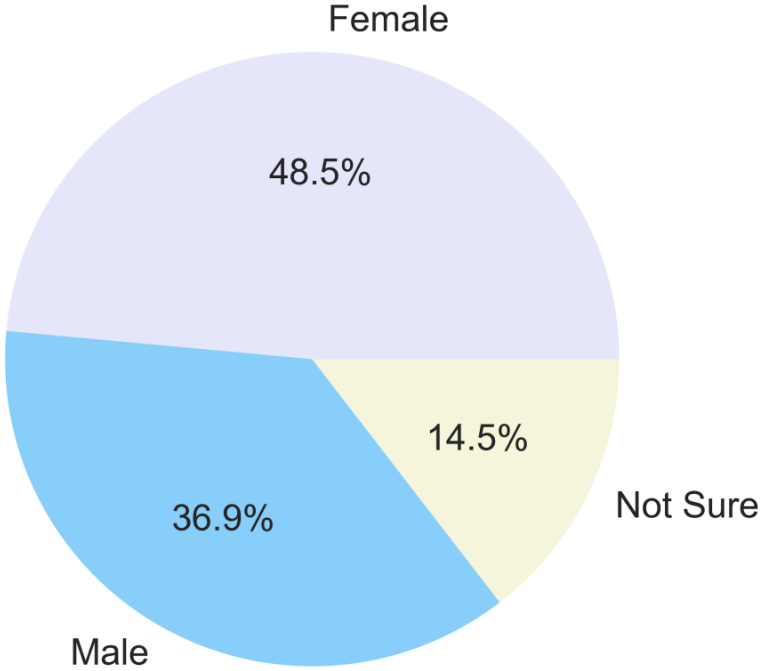


Cat

EDA- Type, Gender

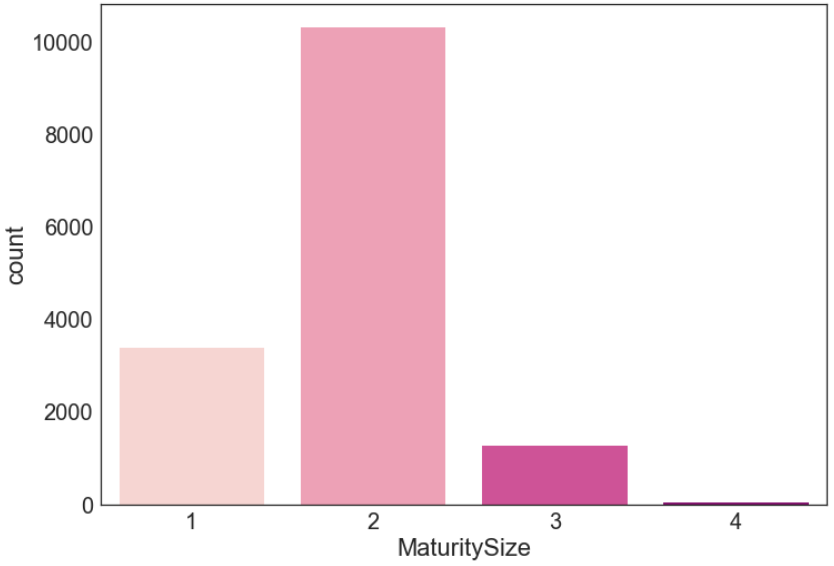


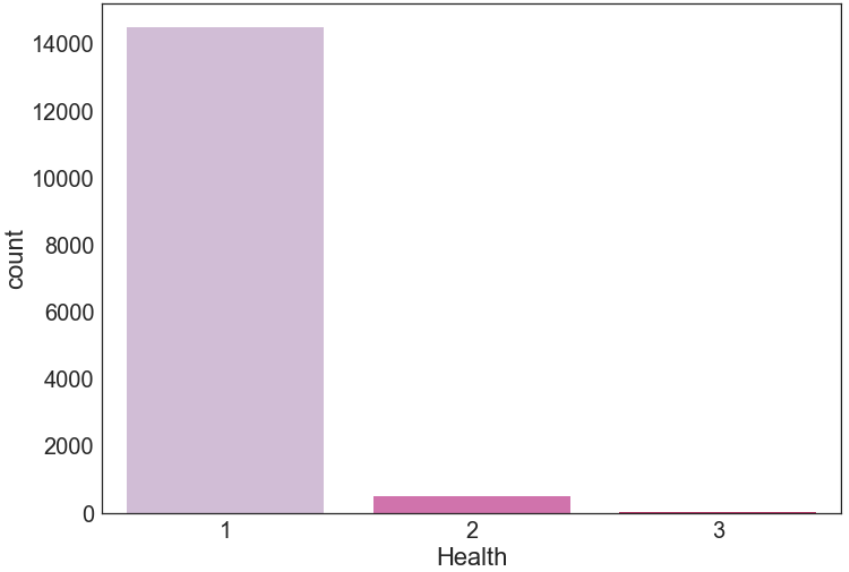
Type



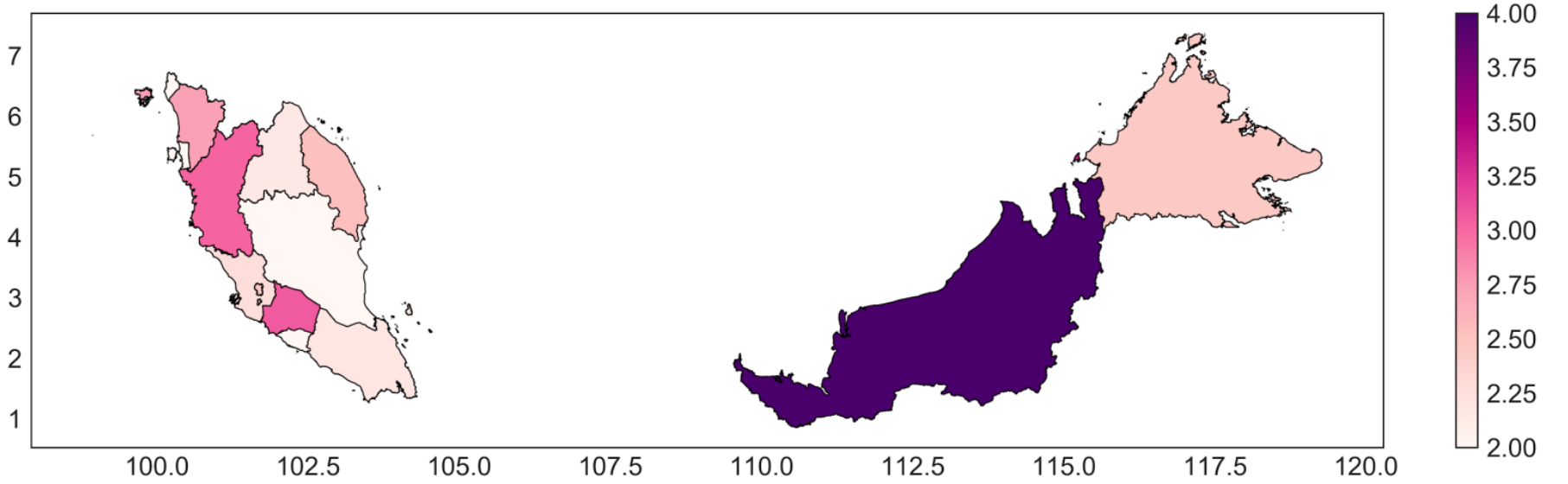
Gender

EDA- MaturitySize, Health

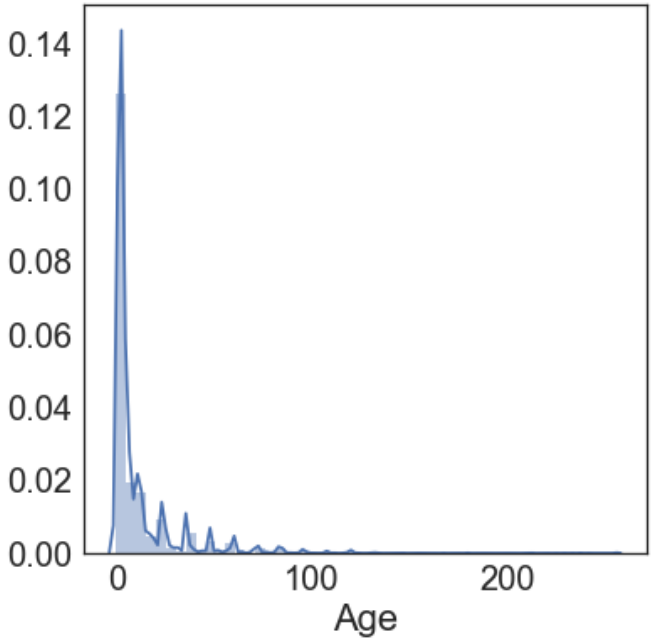


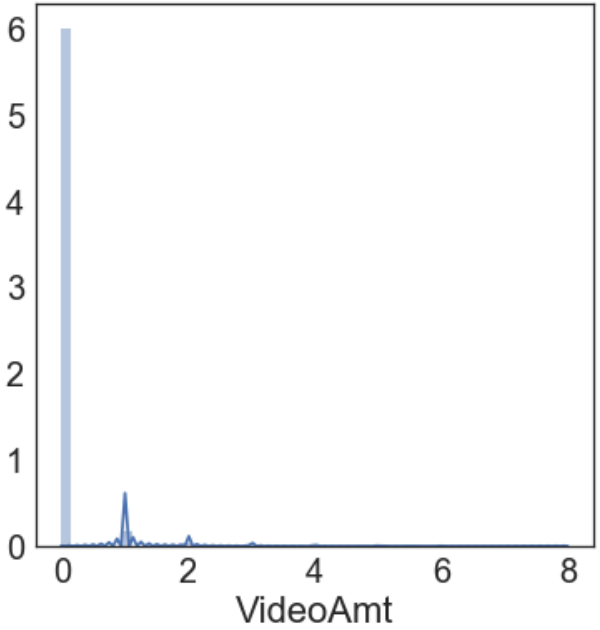


EDA- State

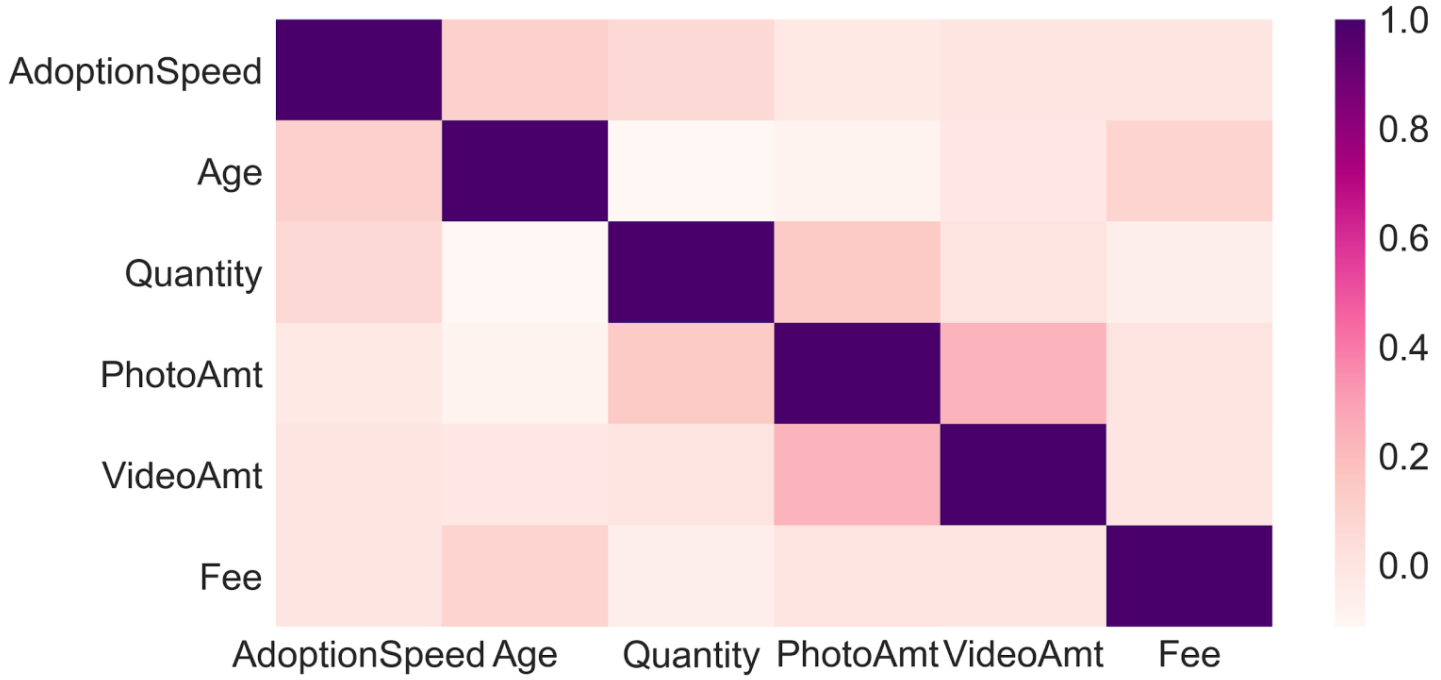

7
6
5
4
3
2
1
100.0
102.5
105.0
107.5
110.0
112.5
115.0
117.5
120.0
4.00
3.75
3.50
3.25
3.00
2.75
2.50
2.25
2.00

EDA- Age, VideoAmt

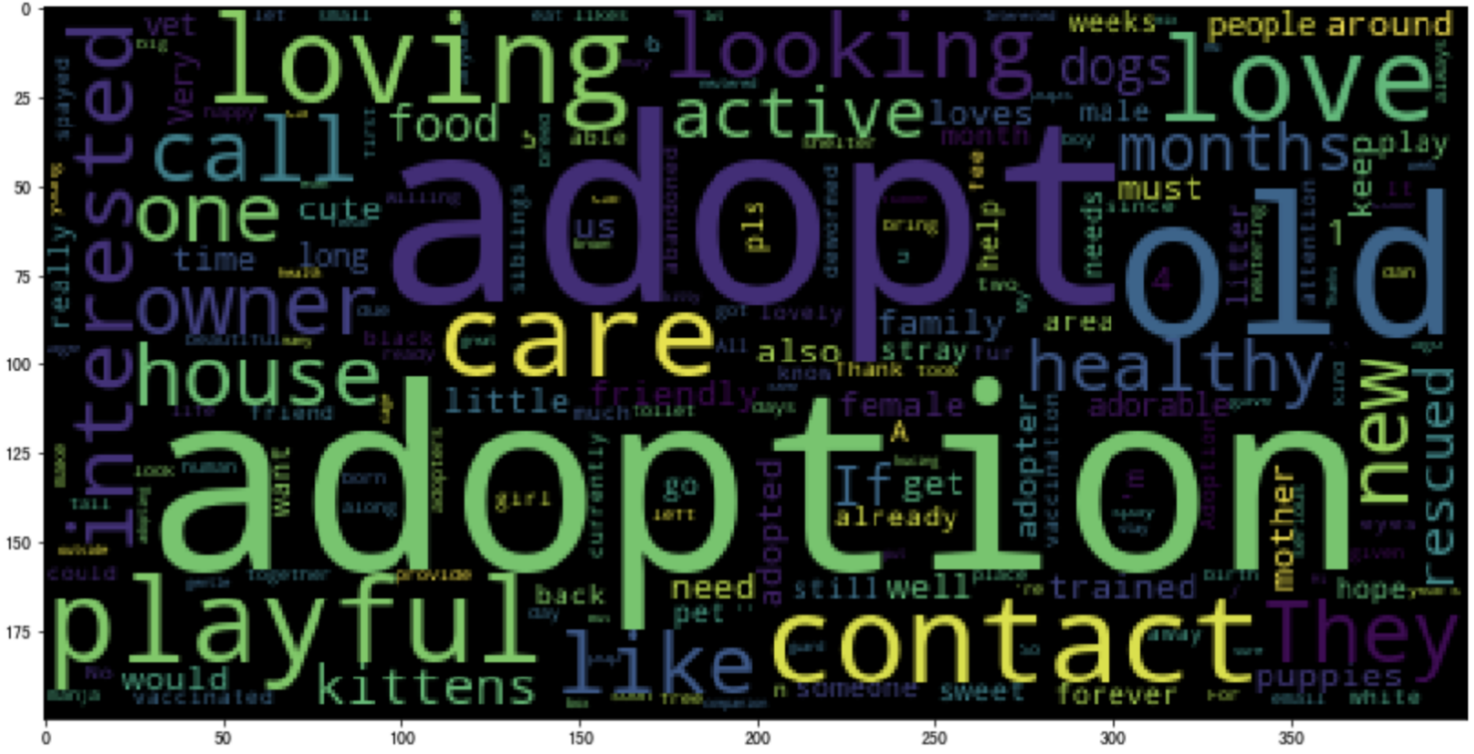




EDA- Correlation



EDA- Description



Data Pre-processing

Categorical & Numerical

Filling Value

Gender: Fill "Not sure" through male to female ratio
Name: Fill missing value with "No Name"

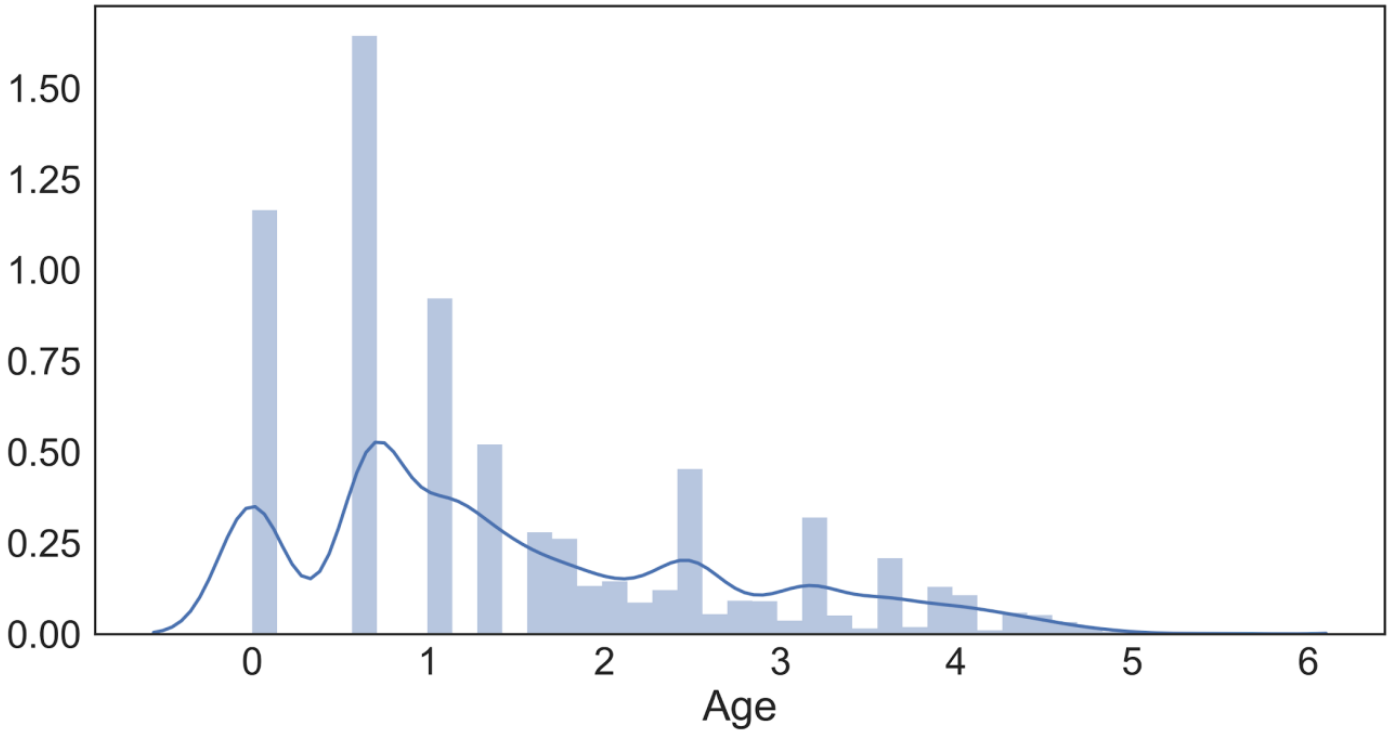
Derived Variables

StateGDP: Map state name to GDP
StatePopulation: Map state name to population

Data Transformation

MaturitySize, Health: Merge rare type to major type
VideoAmt: Binning
Age: Logarithm

Age



Description

Original data contains English, Chinese, and Malay

translate chinese description into English

```
from googletrans import Translator
```

```
translator = Translator()
```

```
def get_translation(ori_text):
    texts = ori_text.copy()
    for i, tx in enumerate(texts):

        isChinese = is_chinese(tx)
        if isChinese:
            try:
                trans_obj = translator.translate(tx, dest='en', src='zh-cn')
                texts.iloc[i] = trans_obj.text
            except BaseException as e:
                print("trans_err: ", i, tx)
                print(e)
        else:
            pass
    return texts
```

TF-IDF, TruncatedSVD

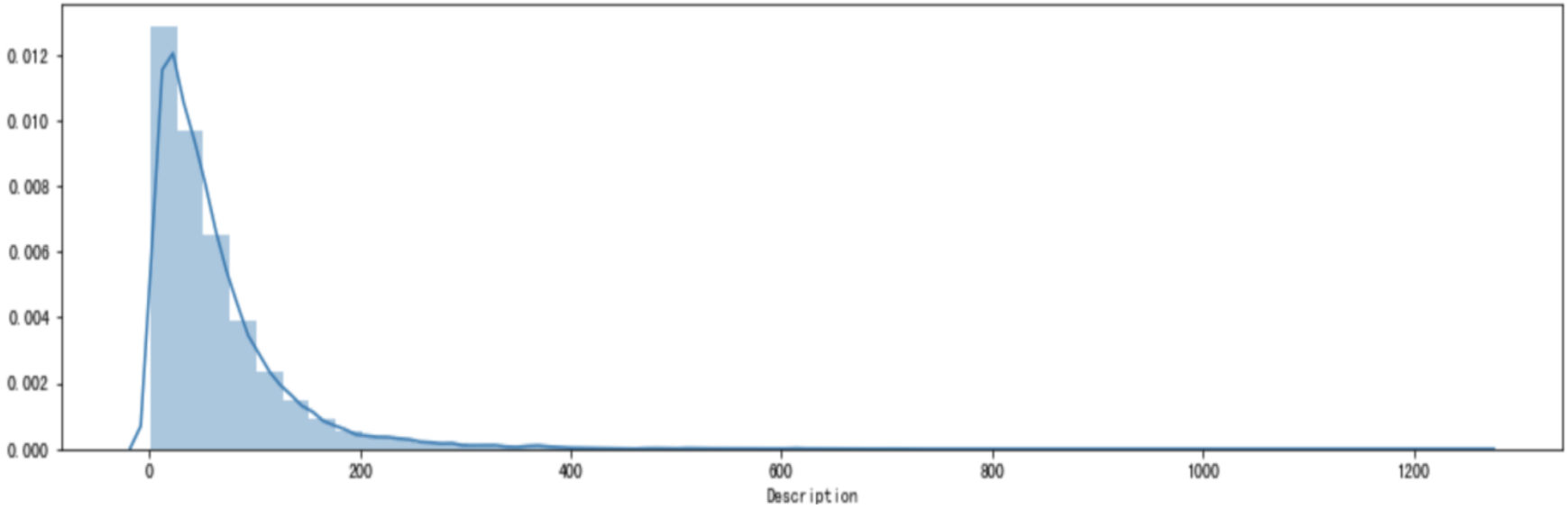
```
# tfidf
from sklearn.feature_extraction.text import TfidfVectorizer
vectorizer = TfidfVectorizer(stop_words = 'english', min_df = 0.03)
tfidf = vectorizer.fit_transform(after_trans)
```

```
from sklearn.decomposition import TruncatedSVD
n_components = 10
tsvd = TruncatedSVD(n_components=n_components)
tsvd_result = tsvd.fit_transform(tfidf)
```

```
print(tsvd_result.shape)
print(tsvd.explained_variance_ratio_)
print(tsvd_result)
```

```
(18965, 10)
[0.01071211 0.02892902 0.02756671 0.0245178  0.01880464 0.01856539
 0.0168846  0.0151018  0.01461099 0.01401708]
[[ 0.32888534 -0.10508095  0.13703928 ...  0.00750186 -0.16781752
   0.05596479]
 [ 0.20154541 -0.07696822 -0.02927354 ...  0.06006009  0.01305084
   0.08410268]
 [ 0.34384803 -0.05057657 -0.0963673  ... -0.05937973  0.04274726
  -0.02662512]
 ...
 [ 0.28087708 -0.09337685  0.17895598 ... -0.02665975  0.14000793
   0.09109794]
 [ 0.25020349 -0.08788644 -0.063428   ... -0.05483534  0.21303416
   0.23656155]
 [ 0.08744651 -0.01905805  0.02995471 ... -0.00889097  0.03372667
   0.00574143]]
```

Description Length



Description Features

```
   Type         Name  Age  Breed1  Breed2  Gender  Color1  Color2  Color3  \
0     2       Nibble    3     299       0       1       1       7       0
1     2  No Name Yet    1     265       0       1       1       2       0

   MaturitySize  ...    TSVD_1    TSVD_2    TSVD_3    TSVD_4    TSVD_5  \
0             1  ... -0.105081  0.137039 -0.055859  0.006113 -0.071988
1             2  ... -0.076968 -0.029274 -0.005811 -0.124465 -0.034767

     TSVD_6    TSVD_7    TSVD_8    TSVD_9  desc_len
0  0.114698  0.007502 -0.167818  0.055965        69
1  0.043573  0.060060  0.013051  0.084103        23

[2 rows x 36 columns]
   Type          Name  Age  Breed1  Breed2  Gender  Color1  Color2  Color3  \
0     2  Dopey & Grey    8     266     266       1       2       6       7
1     2       Chi Chi   36     285     264       2       1       4       7

   MaturitySize  ...    TSVD_1    TSVD_2    TSVD_3    TSVD_4    TSVD_5  \
0             1  ... -0.040790  0.102065 -0.073373  0.189612 -0.071576
1             2  ... -0.036608  0.098736 -0.023478 -0.046173  0.084160

     TSVD_6    TSVD_7    TSVD_8    TSVD_9  desc_len
0 -0.023036 -0.023981 -0.196318 -0.033491        97
1 -0.106820 -0.046059 -0.130618 -0.034740        77

[2 rows x 35 columns]
```

Methods & Evaluation

4

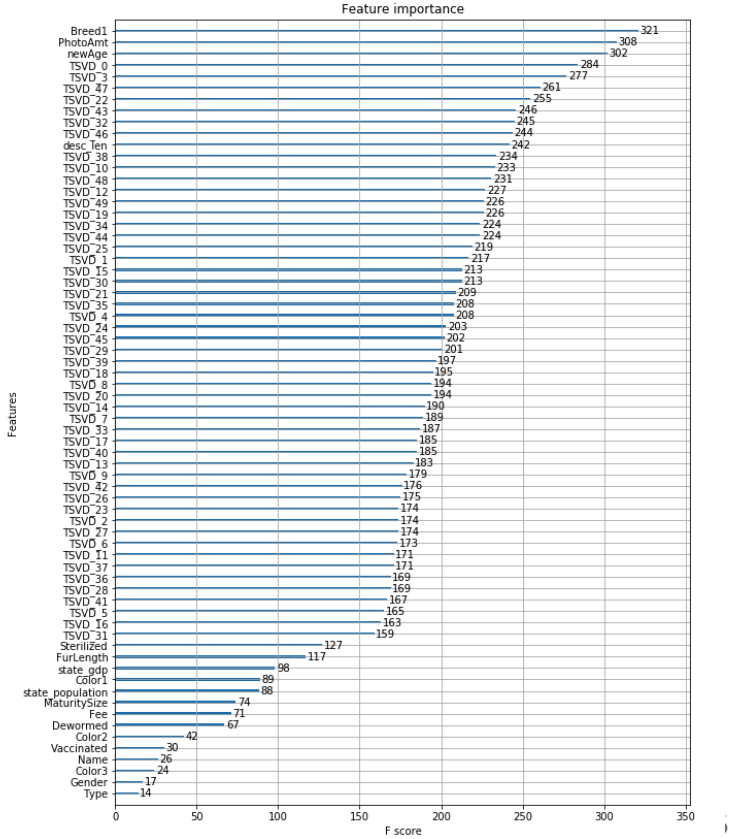
Methods	W/O Description		W/ Description	
	Training (75%)	Test (25%)	Training(75%)	Test(25%)
KNN (Benchmark)	53.82%	30.83%	54.03%	31.34%
Logistic Regression using PyStan	38.65%	37.51%	35.64%	30.78%
Logistic Regression using Sklearn	41.3%	38.8%	42.9%	39.0%
Naïve Bayes	35.4%	36.2%	36.9%	36.1%
Gaussian Process	35.67% (6hr)	35.50%	31.92%	31.44%
Bagging	82.81 %	39.85%	99.99%	43.45%
Random Forest	78.96%	40.76%	99.99%	46.01%
Xgboost	60.43%	40.78%	60.07%	42.41%

Other Findings & Comparisons

	Logistic Regression using Pystan	**Logistic Regression using Sklearn**
Optimization	full Bayesian statistical inference with MCMC sampling	solver : str, {‘newton-cg’, ‘lbfgs’, ‘liblinear’, ‘sag’, ‘saga’}, optional (default=’liblinear’).

Other Findings & Comparisons - Tree based model

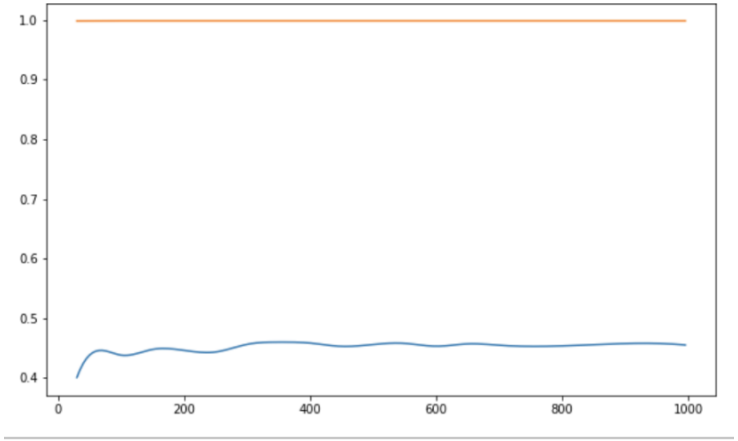
- Xgboost
  - Popular on Kaggle
  - Based on gradient boosting
  - Regularization term
  - Support parallel computation
- Feature importance
  - description feature has high importance



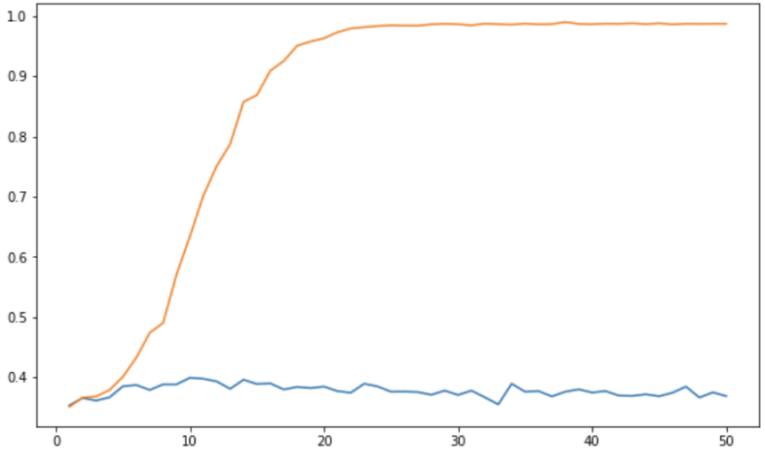
Model Tunning

Random Forest

n_estimators



max_depth



Recommendations

Recommendations

Use K-fold Cross-Validation and get more data to avoid overfitting problem

Implement sentiment analysis on description data

Models might require updates

Build recommendation system based on the model