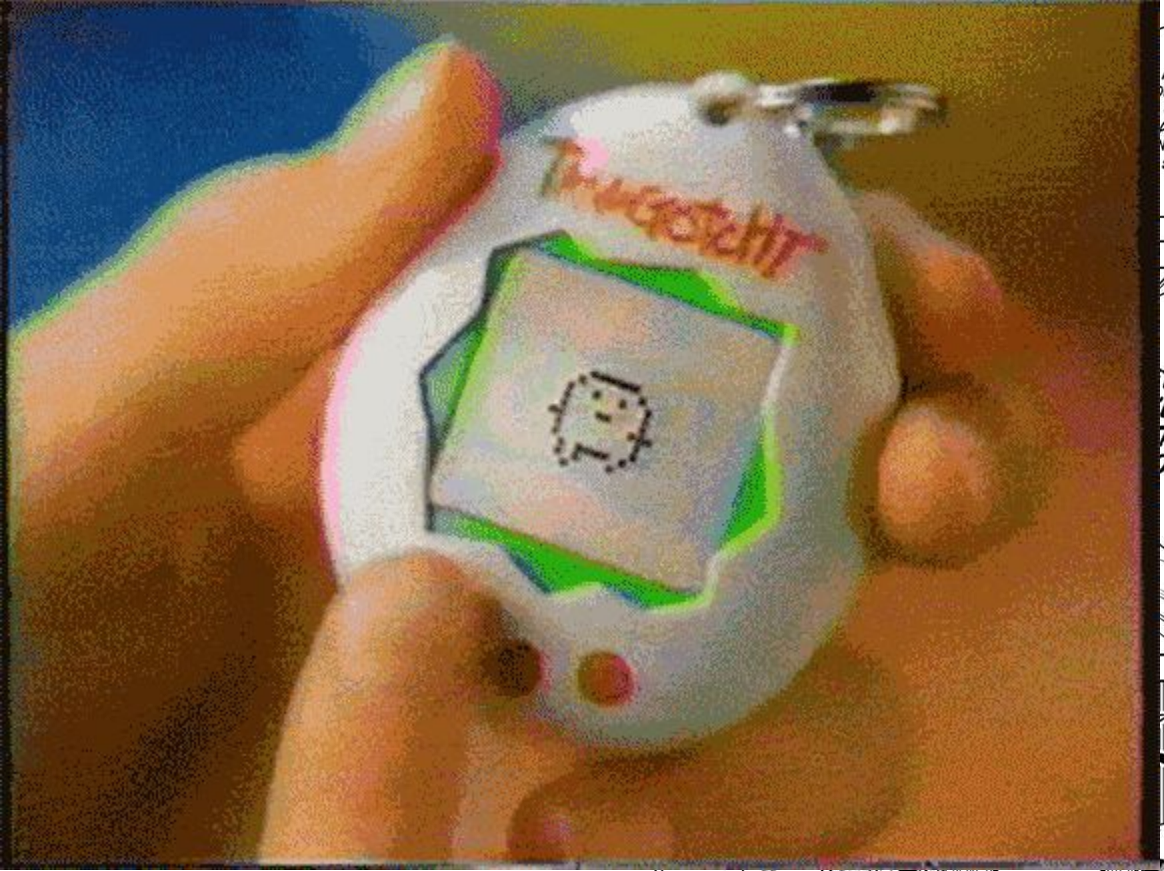
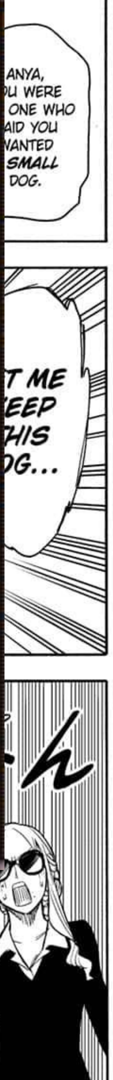
Pet sale price in the UK

Kevin Juandi
24.06.2022

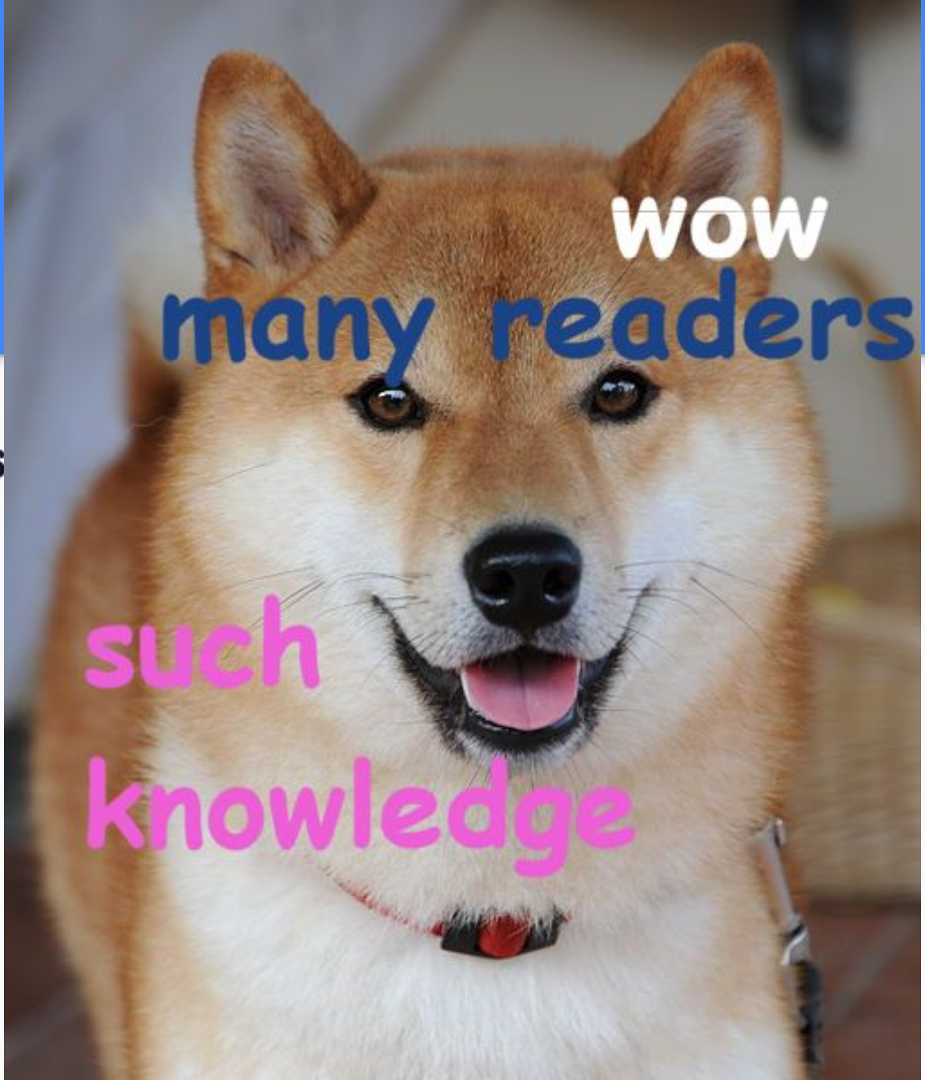
ANYA, YOU WERE THE ONE WHO SAID YOU WANTED A **SMALL** DOG.
WELL NOW I WANT A BIG DOG! **THIS** BIG DOG!
YOU NEED TO BE REASON-ABLE!
LET ME KEEP THIS DOG...
...OR I WILL NEVER GO TO SCHOOL AGAIN!
ガーん
GASP
WHAT?!
WHAT ARE YOU TALKING ABOUT?!
THIS DOESN'T HAVE ANYTHING TO DO WITH—
HE'S MINE MINE MINE!
...



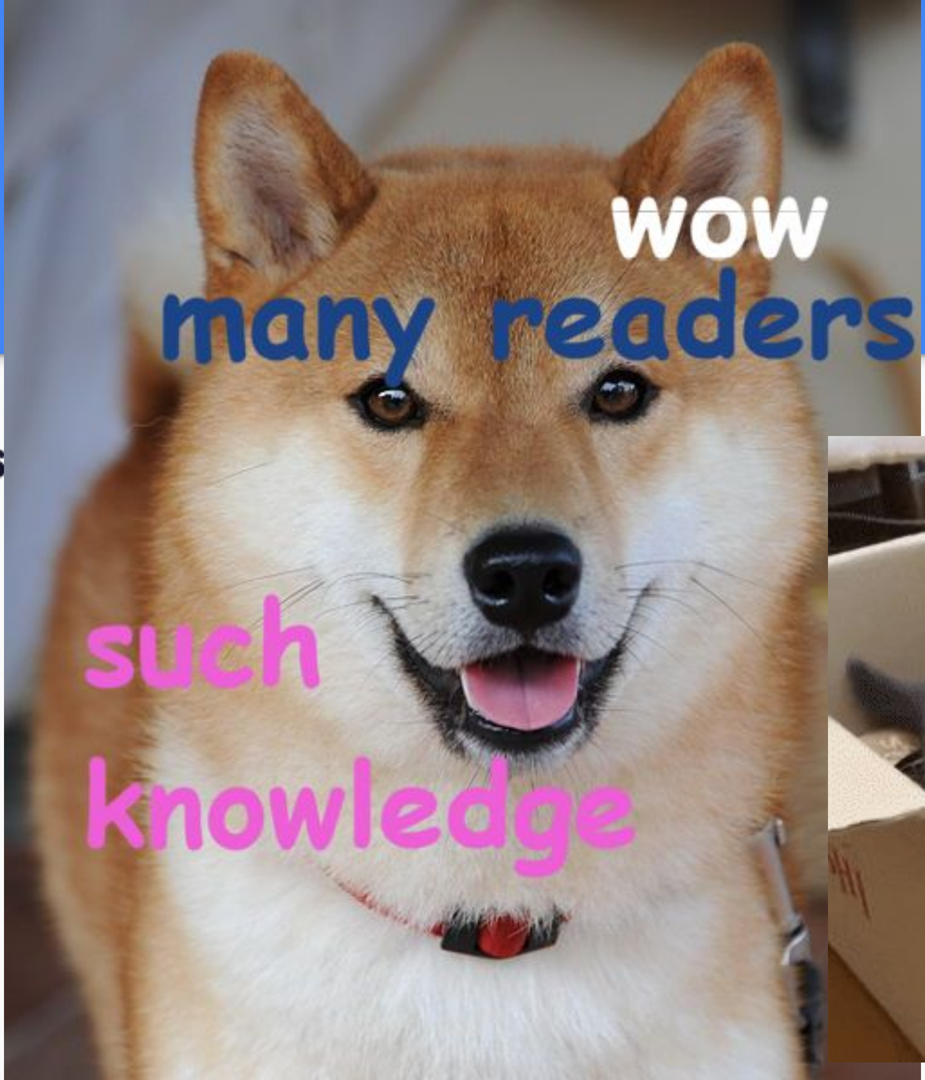

MINE
MINE
MINE!
WHAT?!
...

Pets4Homes

Pets4Homes



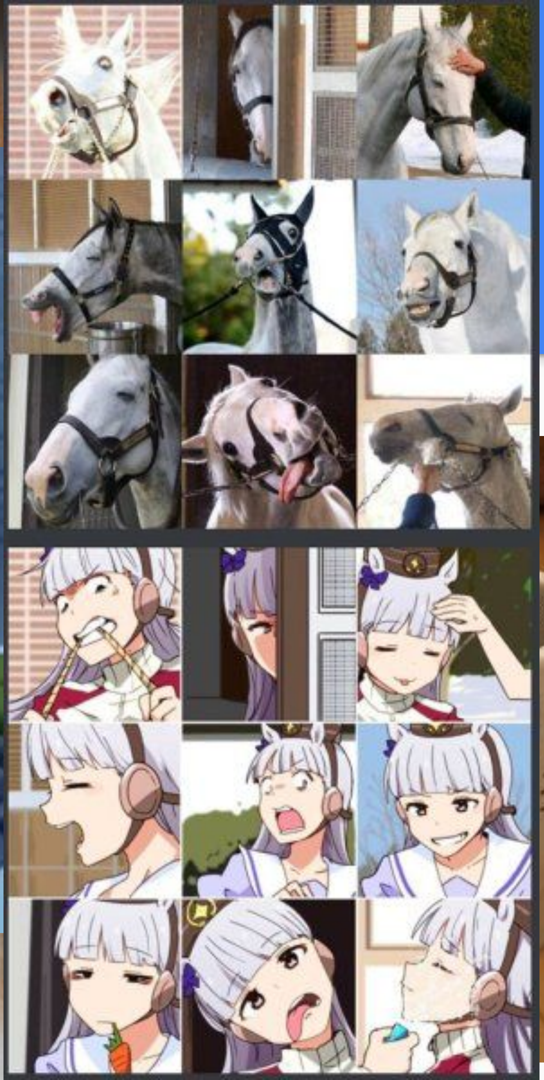
Pets4Homes




wow
eaders
e

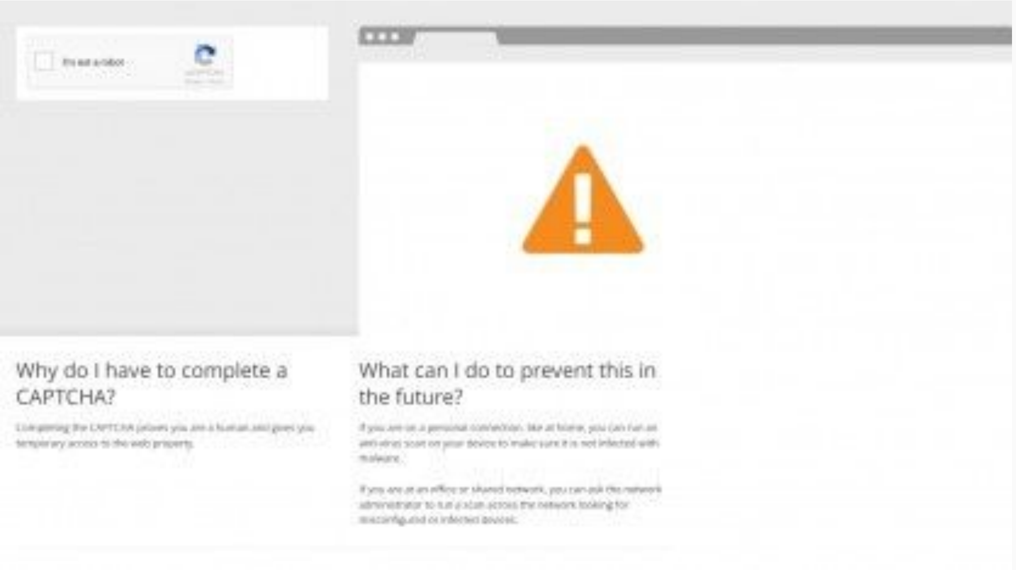

peteneems

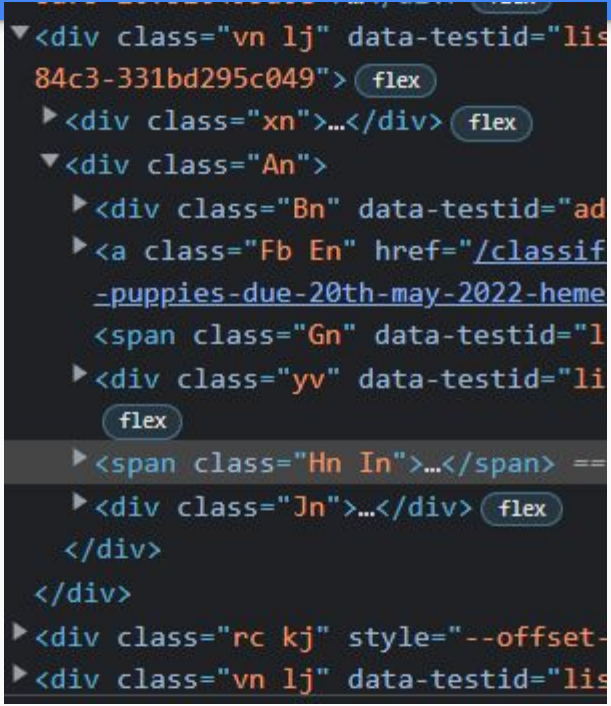




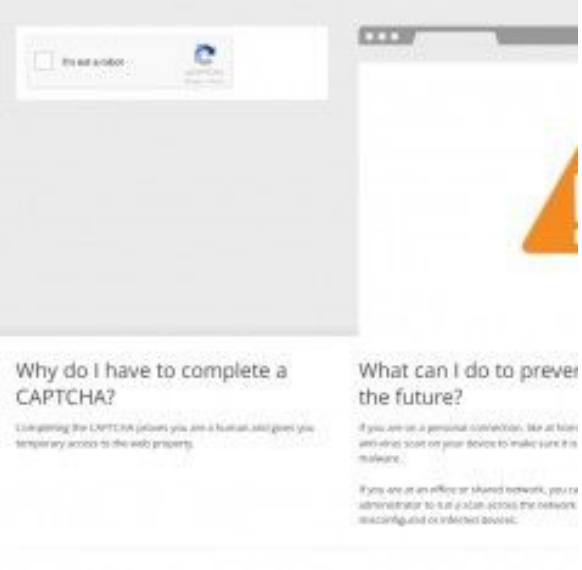
Scraping the Data

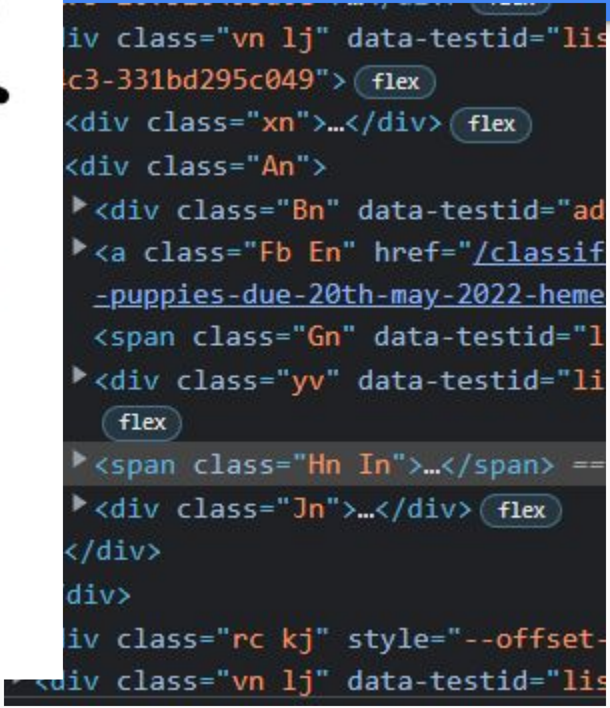
Scraping the Data





Scraping the Da





Scraping the Data

1. Use selenium to open each pages individually
2. Create new CSV file for each city/region
3. Write an alarm with windsound to alert html tag change

Processing the Data

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 48781 entries, 0 to 48780
Data columns (total 11 columns):
 #   Column           Non-Null Count  Dtype  
---  ------           --------------  -----  
 0   Title            48781 non-null  object 
 1   price            48781 non-null  float64
 2   species          48781 non-null  object 
 3   age              48781 non-null  object 
 4   gender           48781 non-null  object 
 5   description      48781 non-null  object 
 6   seller_name      48781 non-null  object 
 7   seller_location  48781 non-null  object 
 8   seller_type      48781 non-null  object 
 9   listing_type     48781 non-null  object 
 10  pet_type         48781 non-null  object 
dtypes: float64(1), object(10)
memory usage: 4.5+ MB
```

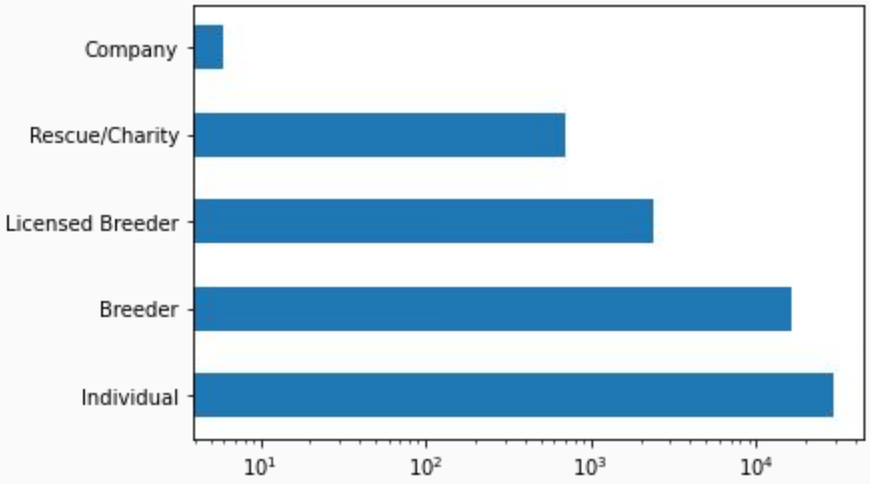
Processing the Data

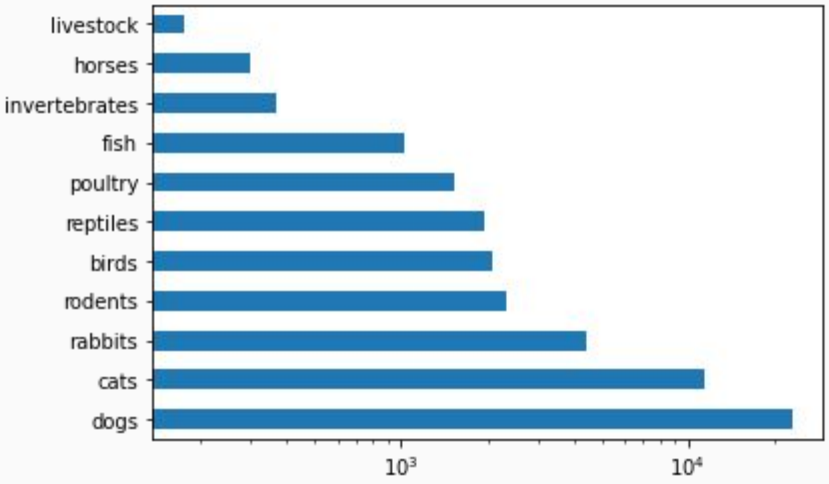
```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 48781 entries, 0 to 48780
Data columns (total 11 columns):
 #   Column           Non-Null Count  Dtype  
---  ------           --------------  -----  
 0   Title            48781 non-null  object 
 1   price            48781 non-null  float64
 2   species          48781 non-null  object 
 3   age              48781 non-null  object 
 4   gender           48781 non-null  object 
 5   description      48781 non-null  object 
 6   seller_name      48781 non-null  object 
 7   seller_location  48781 non-null  object 
 8   seller_type      48781 non-null  object 
 9   listing_type     48781 non-null  object 
 10  pet_type         48781 non-null  object 
dtypes: float64(1), object(10)
memory usage: 4.5+ MB
```

```
#a quick look of number of unique values
df.nunique()
```

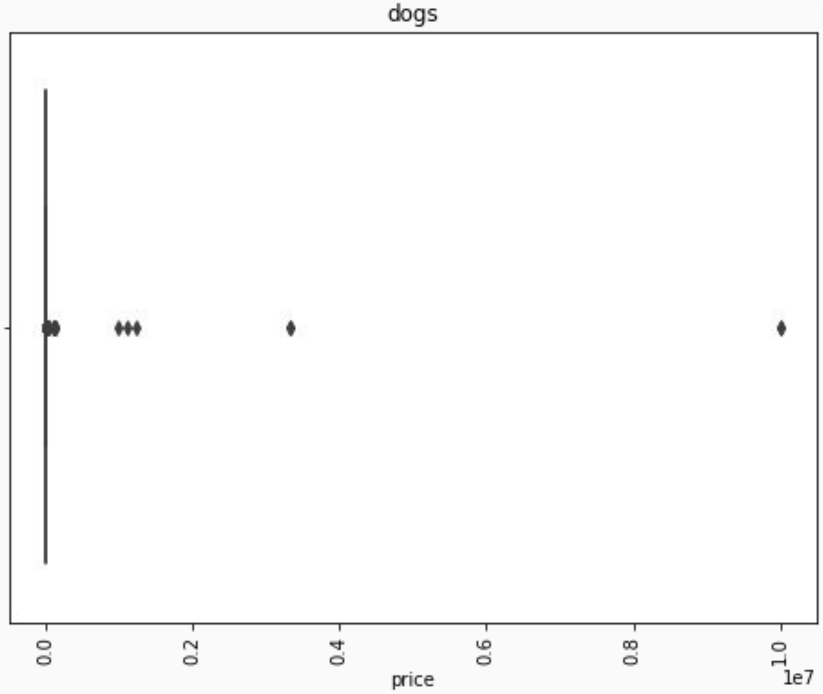
executed in 80ms, finished 09:24:53 2022-06-22

```
Title              37519
price                394
species              438
age                  101
gender               139
description        44796
seller_name        20520
seller_location     1083
seller_type            5
listing_type           3
pet_type              11
dtype: int64
```

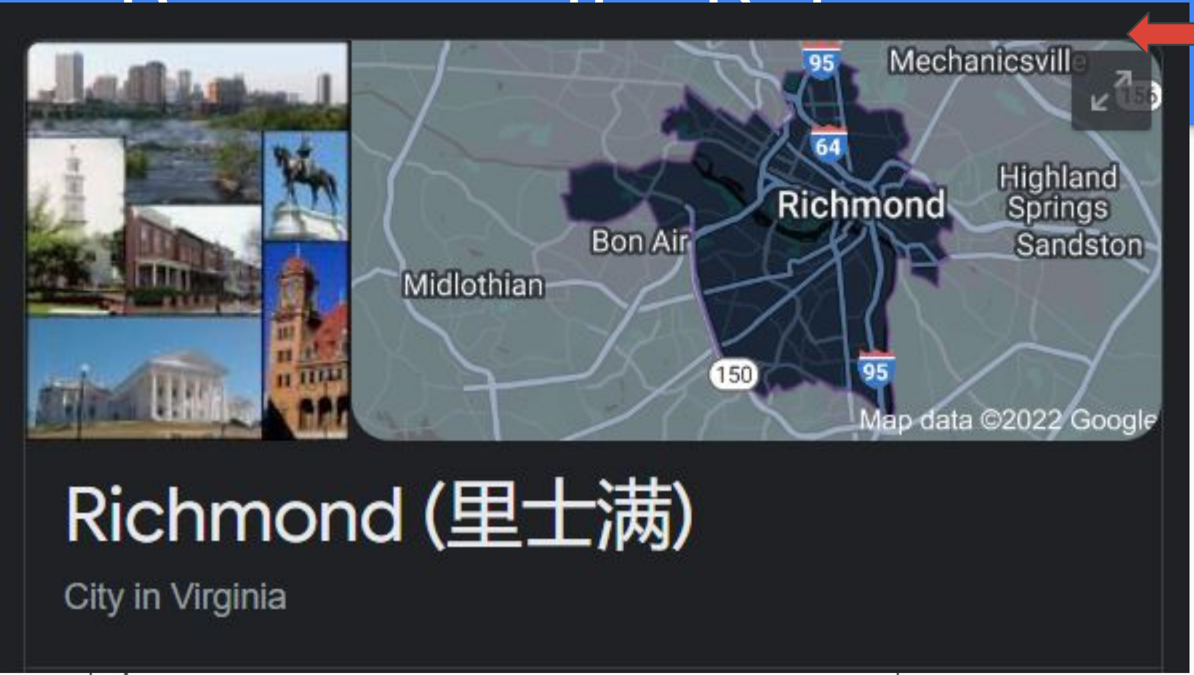

Company
Rescue/Charity
Licensed Breeder
Breeder
Individual
10^1
10^2
10^3
10^4


livestock
horses
invertebrates
fish
poultry
reptiles
birds
rodents
rabbits
cats
dogs
10^3
10^4

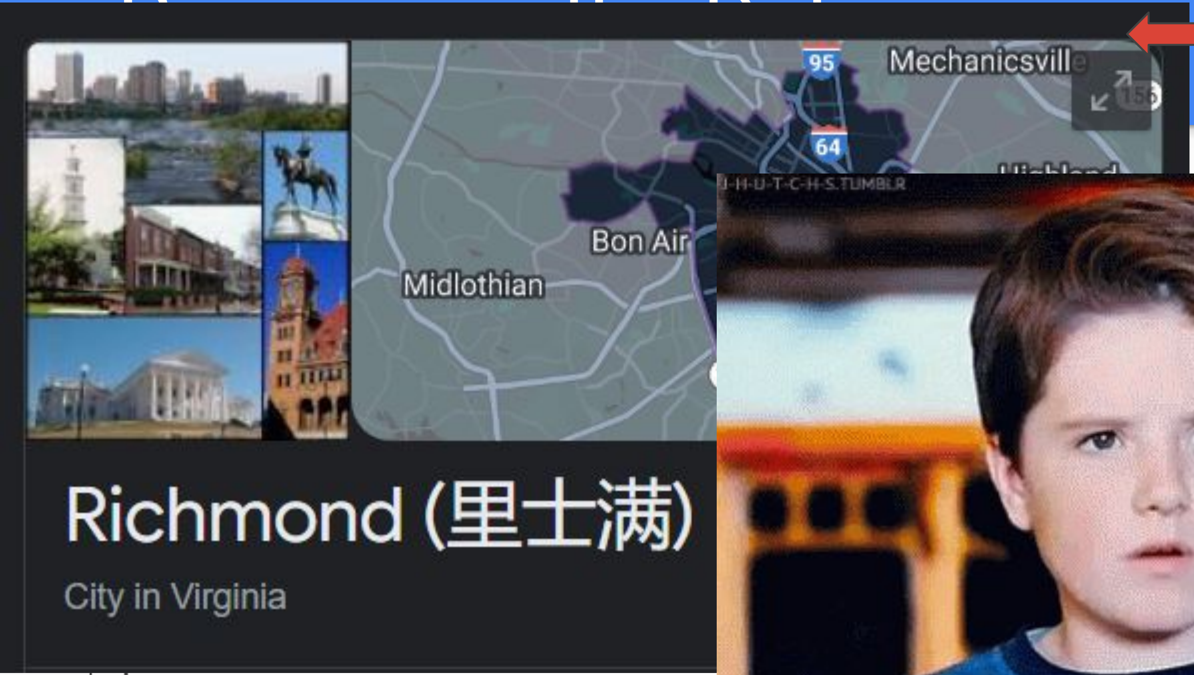
Processing the Data




```
df['seller_location'].value_counts().tail(50)

里士满                  1
St Helens              1
Darvel                 1
Renfrew                1
Irvine                 1
Crieff                 1
Larkhall               1
Erskine                1
Bellshill              1
Troon                  1
Johnstone              1
Ayr                    1
Лондон                 1
Helensburgh            1
Άλτον                  1
Chislehurst            1
West Linton            1
Corbridge              1
Barton-upon-Humber     1
Houghton Le Spring     1
Belper                 1
Ilkeston               1
Heanor                 1
Felixstowe             1
Haverhill              1
```




```
df['seller_location'].value_counts().tail(50)

里士满                  1
St Helens              1
Darvel                 1
Renfrew                1
Irvine                 1
Crieff                 1
Larkhall               1
Erskine                1
Bellshill              1
Troon                  1
Johnstone              1
Ayr                    1
Лондон                 1
Helensburgh            1
Άλτον                  1
Chislehurst            1
West Linton            1
Corbridge              1
Barton-upon-Humber     1
Houghton Le Spring     1
Belper                 1
Ilkeston               1
Heanor                 1
Felixstowe             1
Haverhill              1
```




```
df['seller_location'].value_counts().tail(50)
```

```
里士满          1
St Helens      1
Darvel         1
Renfrew        1
Irvine         1
Ilkeston       1
Heanor         1
Felixstowe     1
Haverhill      1
```

Processing the Data

```
#examining gender column for unique values
df['gender'].value_counts()
```

executed in 16ms, finished 09:24:53 2022-06-22

```
unknown                 11934
Mixed                    6199
Male                     3719
1 male                   2857
Female                   2796
                        ...
03 male / 03 female         1
03 male / 01 female         1
03 male / 02 female         1
7 male / 8 female           1
05 male / 1 female          1
Name: gender, Length: 139, dtype: int64
```

```python
#examining gender column for unique values
df['gender'].value_counts()
```

executed in 16ms, finished 09:24:53 2022-06-22

```
unknown                11934
Mixed                   6199
Male                    3719
1 male                  2857
Female                  2796
                       ...
03 male / 03 female        1
03 male / 01 female        1
03 male / 02 female        1
7 male / 8 female          1
05 male / 1 female         1
Name: gender, Length: 139, dtype: int64
```

```
Mixed       20614
unknown     11934
Male         8850
Female       7085
Mare          139
Gelding       117
Stallion       42
Name: gender, dtype: int64
```

```
#examining gender column for unique
df['gender'].value_counts()
```

executed in 16ms, finished 09:24:53 2022-06-22

```
unknown                  11934
Mixed                     6199
Male                      3719
1 male                    2857
Female                    2796
                        ...
03 male / 03 female          1
03 male / 01 female          1
03 male / 02 female          1
7 male / 8 female            1
05 male / 1 female           1
Name: gender, Length: 139, dtype: int64
```

```
Mixed       20614
unknown     11934
Male         8850
Female       7085
Mare          139
Gelding       117
Stallion       42
Name: gender, dtype: int64
```

```
df['age'].str.startswith('Due').value_counts()
```

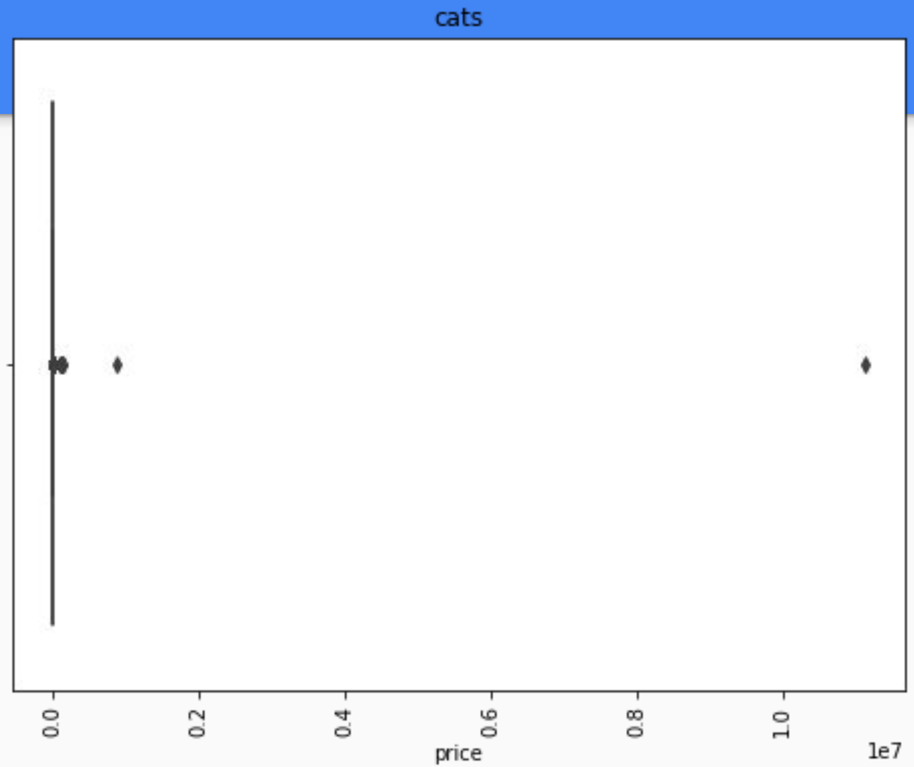
executed in 27ms, finished 09:25:54 2022-06-22

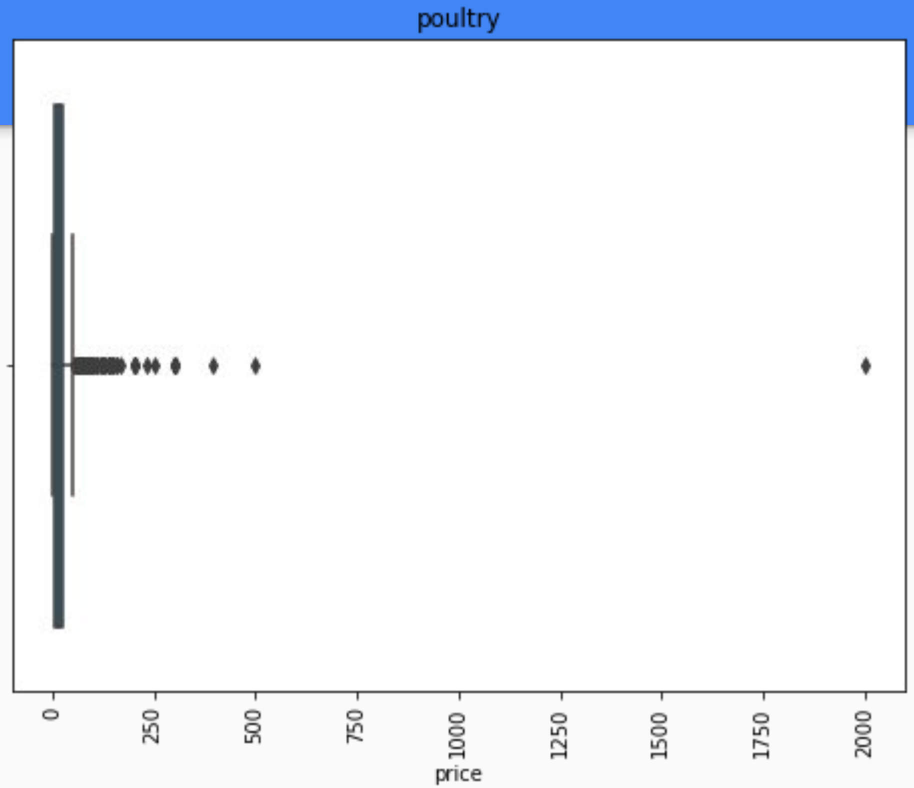
```
False    48636
True       145
Name: age, dtype: int64
```

```
#checking age column
df['age'].value_counts().tail(20)
```

executed in 14ms, finished 09:25:42 2022-06-22

```
2001              3
16 years          3
2000              3
Due in 6 weeks    3
2003 years        2
Due in 6 days     2
18 years          2
2004 years        2
20 years          2
2002 years        2
1998              1
2000 years        1
24 years          1
Due in 7 weeks    1
42 years          1
1999              1
66 years          1
2019 years        1
23 years          1
17 years          1
Name: age, dtype: int64
```


cats
0.0
0.2
0.4
0.6
0.8
1.0
1e7
price

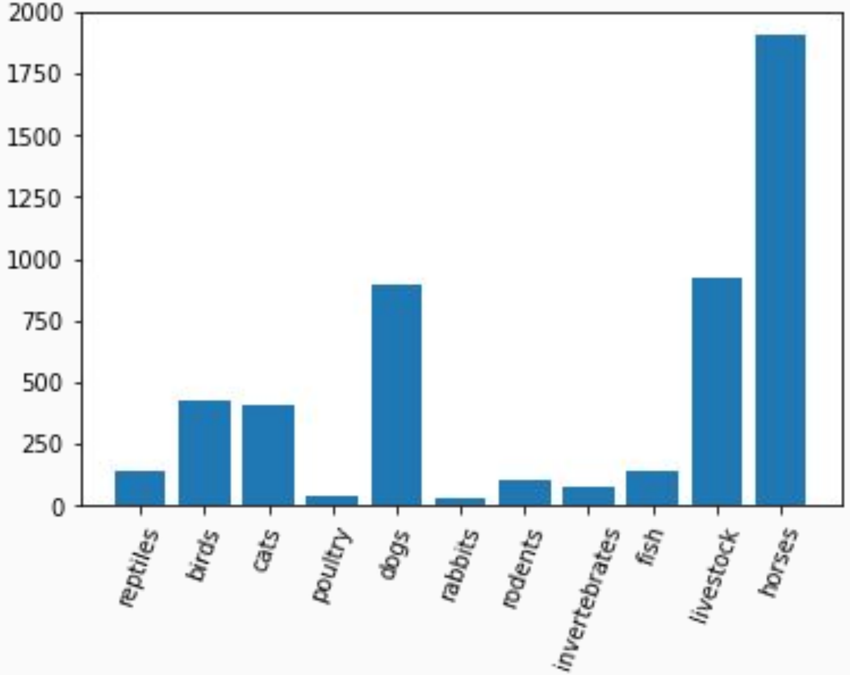

poultry
0
250
500
750
1000
1250
1500
1750
2000
price

1. Correlation rather low
2. Majority of listing is not even 1 years old

	price	year
price	1.000000	-0.106195
year	-0.106195	1.000000

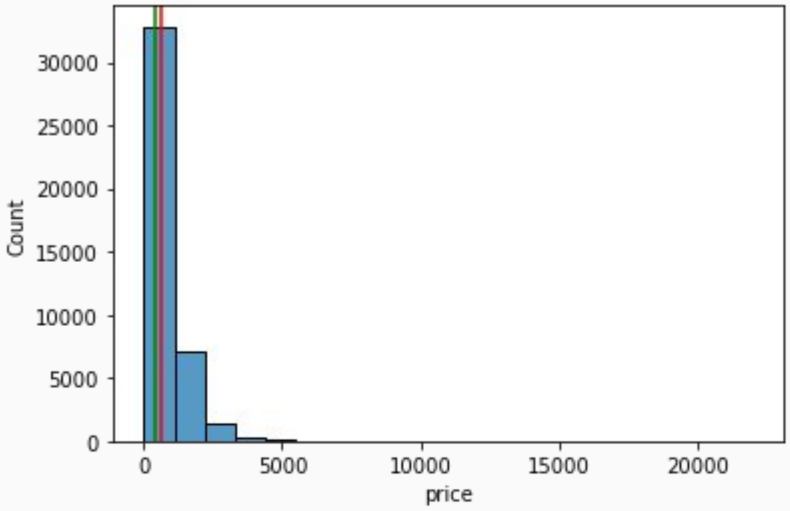
	price	year
count	41664.000000	41664.000000
mean	656.732335	0.946517
std	805.959946	1.722896
min	0.000000	0.000000
25%	100.000000	0.153846
50%	392.500000	0.250000
75%	1000.000000	1.000000
max	22000.000000	66.000000

Standard Deviation of Price



High or Low

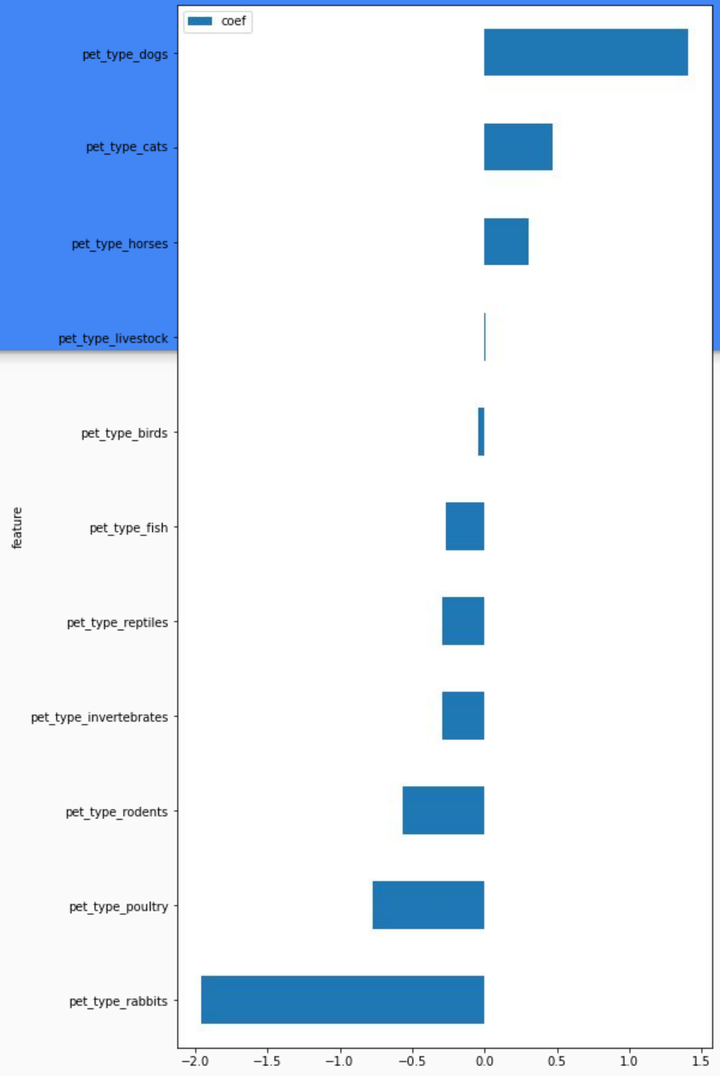
- Median is 392.5



Logistics regression

With only pet type as variables

- the cv score is 0.7921409736053113
- the training score is 0.7921409957481826
- the test score is 0.79016



Logistics regression

With more variables such as seller type, listing type, species or breed and gender:

- the cv score is 0.8982305867213576
- the training score is 0.9030311342751337
- the test score is 0.90024

NLP Model

- Use description text to predict price class
- Extremely resource intensive

The winner was Random Forest

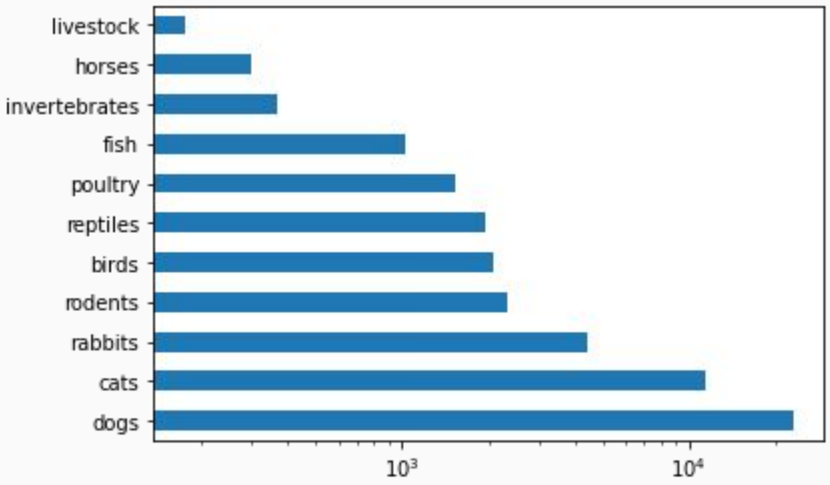
- the cv score is 0.8650048197933474
- the training score is 0.993553696337950
- the test score is 0.8676

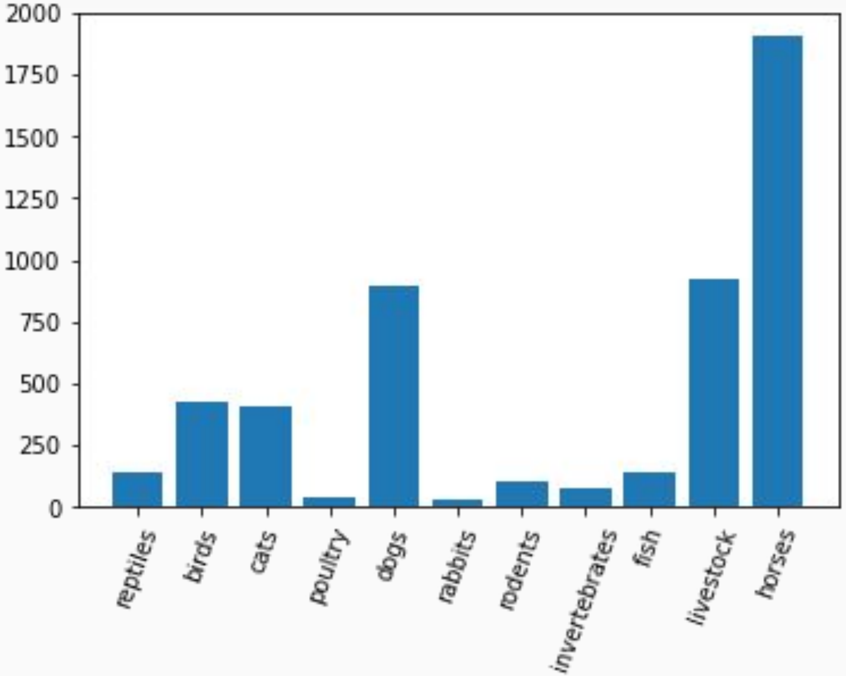
```
RandomForestClassifier(max_features=300, n_estimators=10)
accuracy score is 0.9557651689708141
auc score is 0.9557651689708141
------------------------------------------------------------

the confusion matrix is:
 [[20199   633]
 [ 1210 19622]]
------------------------------------------------------------

and the classification report:
              precision    recall  f1-score   support

 lower price       0.94      0.97      0.96     20832
higher price       0.97      0.94      0.96     20832

    accuracy                           0.96     41664
   macro avg       0.96      0.96      0.96     41664
weighted avg       0.96      0.96      0.96     41664
```

Price Prediction

Ridge	0.5222435083569612
Lasso	0.5228068232308152
Elastic Net	0.5229001110885271
Decision Tree	0.48605925061713284
Bagging	0.5166823132596303
Random Forest	0.49772672687446845
Ada Boost	0.3907683994473204
Gradient Boost	0.5316628271074635
Neural Net	0.5190056431272434

Number of entries and standard deviation





Comparison with and without dogs

Ridge	0.5222435083569612	0.3657269556683256 4	0.5310472091257473
Lasso	0.5228068232308152	0.3659987749711451 4	0.530634729719761 6
Elastic Net	0.5229001110885271	0.3661667051768921	0.5323566954322898
Decision Tree	0.4860592506171328 4	0.3201275719248156 7	0.501739066514634 2
Bagging	0.5166823132596303	0.348130411571521 7	0.5681024881252817
Random Forest	0.4977267268744684 5	0.2989360623029004	0.5654032982753289
Ada Boost	0.3907683994473204	0.1855568664582719	0.2092885634699999
Gradient Boost	0.5316628271074635	0.3697957086680038	0.5869383815391225
Neural Net	0.5190056431272434	0.3623905748734906	0.5219623002989864

Prediction from best model

- Rather terrible
- Model tends to overestimate the value
- When it doesn’t, it underestimate by a lot
- Especially prominent for dogs

```
count     41664.000000
mean         -0.165941
std         471.110310
min      -16601.398979
25%         -76.769833
50%          39.695575
75%         143.470811
max        5602.940704
Name: Gboost_difference, dtype: float64
```

Conclusion

- The age of the animal hardly affect price
- Most listing are animals under one year old
- Predicting sale price accurately is difficult
- More should be done in cleaning the data
- Convert location data to counties
- Multiclass regression model is perhaps better approach to the problem