Capstone Project 2
Petfinder Milestone Report 3

Table of Contents

Capstone Project 2
Petfinder Milestone Report 3

Objectives:

The objective for this report is to share a second round of machine learning results to build a model to predict the speed at which a pet is adopted. In this round the following changes to try and improve the model are applied...

1. Outliers (pets older than 12 years old are removed from the data)
2. Scalarization is applied to features where this might help the algorithms produce a better model
3. Added XGBoost algorithm because it should be a good algorithm for adoption rates and it also provides a means to assess feature importance
4. Hyper-parameter tuning is applied to the algorithms
5. Word-2-Vec is applied to the Pet Description for use in NLP
6. An ensemble (Random Forest or XGBoost) combines the best performing non-NLP with the best performing NLP algorithm

As a reminder, the classification areas are...

0 - Pet was adopted on the same day 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

About the Data:

Petfinder has provided the data at this link, which consists of 14,993 pets (54% dogs and 46% cats) that were up for adoption. Along with various features, this data also contains a column called “Adoption Speed”, which is the predictor variable for this project.

Removing outliers modified the data from 14,993 pets to 14,978; so 15 pets were over the age of 12 years old.

Machine Learning

Removal of Outliers and Application of Scaling

The removal of outliers and the application of scaling to the features did not improve the accuracy of the algorithms.

Models:

XG Boost provided a better model than any of the other models used to date.

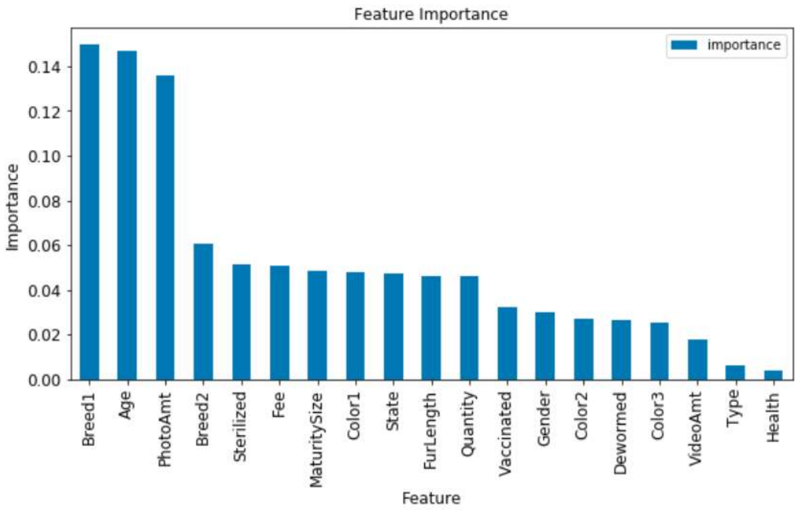
Algorithm	Accuracy Score	Precision Score	Recall
XGBoost	40%	41%	40%
Random Forest	36%	35%	36%

Algorithm	Accuracy Score	Precision Score	Recall
Logistic Regression	35%	33%	35%
KNN	31%	34%	31%
Naive Bayes	34%	36%	34%

Feature Importance:

The addition of XGBoost also enabled a relatively simple means to assess feature importance. Per the chart below, several interesting points can be made:

1. The most significant features driving adoption rates are Breed, Age, and presence of photos.
2. Based on the "Type" (dog or cat) there is not a significant preference/adoption speed difference
3. Interestingly, the "Health" of the animal does not play a significant role (more analysis on this might be beneficial, i.e. are most animals marked as healthy? Are unhealthy animals not put up for adoption?, etc...
4. "Quantity" is in middle of feature importance (more analysis on what happens to the adoption rate as the quantity increases from 1 to 2 pets, to 3, to 4, etc...



Hyper-parameter Tuning:

Decided to only apply hyper-parameter tuning to the two best algorithms, which are XGBoost and Random Forest. The parameters selected for tuning and the best value for each parameter are summarized below...

XGBoost:
Based on a talk by Owen Zhang at ODSC Boston 2015 titled "Open Source Tools and Data Science Competitions", he summarized common parameters to tune as...

Parameter	Default Value	Best Value for this Model
colsamplebytree	1	1
colsamplebylevel	1	1
Tree Size (max depth)	3	10
learning rate	0.1	1

Parameter	Default Value	Best Value for this Model
Min Leaf Weight (min child weight)	1	1
Row Sampling (subsample)	1	0.5

RandomForest:

Based on the article, “Tuning Random Forest Model” at this link, the parameters below are being run through GridSearchCV...

Parameter	Default Value	Values to run in GridSearchCV
n_estimators	10	200 to 2000, increments of 10
max_depth	None	10 to 110, increments of 10
min_sample_split	2	2,5,10
min_samples_leaf	1	25,50,100

The hyper-parameter tuning did not change the results of accuracy scores for XGBoost nor for Random Forest.

Natural Language Processing (NLP) with Word2Vec and XGBoost

Leveraging the Pet Description feature, Word2Vec was used to create a vectorized version of each pet’s description. The vector description was then run thru the XGBoost algorithm. Results were very similar to the non-NLP models previously covered and are summarized below...

Algorithm	Accuracy Score	Precision Score	Recall
XGBoost using Word2Vec	35%	37%	35%

Stacking Models

The XGBoost model using 20 nonNLP features prediction results and the XGBoost model using NLP’s prediction results were combined into an array; then used as input to a new XGBoost model. The results obtained were the best to date at 44% accuracy; a 4% improvement over the previous best model.

Algorithm	Accuracy Score	Precision Score	Recall
Stacked Model	44%	36%	44%

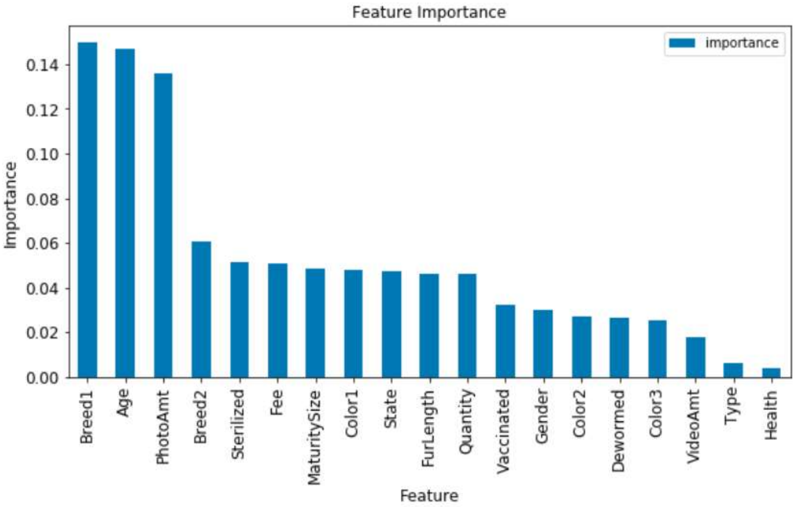
Summary of Key Findings:

1. The best model was the Scaled model that combined prediction results of the nonNLP XGBoost and NLP XGBoost models. Accuracy was 44%.
2. Removal of outliers did not significantly change the model results. This is probably due to the low number of only 15 outliers (pets over the age of 12) out of the 14,993 records.
3. Adding Scalarization to the data did not significantly change the model results. This is probably due to most of the features are already in a tight range. Only the age of the pet was a range of values (0 to 12 years)

4. Using XGBoost on the 20 nonNLP features paid off; it generated the best performing model using the 19 features (accuracy score of 40%) and is the only model to achieve 100% precision score in any adoption category (pets adopted on same day).
5. Using NLP with Word2Vec & XGBoost on the Pet Description feature yielded a model that produced results in the mid-30’s for accuracy, precision, and recall. Not a significantly different result from the nonNLP models.

Recommendations:

1. This analysis included 19 features to predict the adoption rate of dogs/cats. Based on the feature importance results a fourth round using XGBoost and RandomForest with only the top three features; and possibly a fifth round using the top 11 features might improve the model’s accuracy and precision.



2. Consider running the KNN algorithm using only the top three features. The reduction in dimensionality may improve this algorithm’s ability to group similar adoption trends, and produce a model with improved accuracy and precision.
3. Additional research into the contribution of Natural Language Processing (NLP) to model accuracy should be considered, specifically engagement of deep learning.
4. The best results were obtained by Scaling two models (one NLP and one nonNLP). Considering the breakout of Feature Importance (above bar chart) into two primary contribution levels (features 1 – 3 in one group and features 4 – 11 in a second group), consider stacking 3 separate models – model for features 1 – 3, model for features 4 – 11, and the NLP with Word2Vec model.