PetFinder Adoption Speed Prediction Model

By: Brennan, Spring & Rosie as well as Anusha
GitHub:

Tim from petfinder.com

Why pet adoption?

The Addams Family Litter from petfinder.com

The topic we chose was pet adoption speed. We chose this topic because the members of our group have pets and love animals! Our datasource was taken from a featured code competition on Kaggle for Petfinder.my, Malaysia’s leading animal welfare platform. The competition calls for developing algorithms to predict how quickly a pet is adopted.

Link to Kaggle competition:

How is the data going to answer our question?

We will need to be able to predict the outcome of adoption speeds of the pets. Once we have that model, we can make other inferences based on the largest features that play into adoption speed prediction. As we have the adoption speed already, we can compare out model to actual results.

Wren from petfinder.com

Data Description

Baxter(adopted) from petfinder.com

Our data contains thousands of entries of pets, each of which contains information including

- Type - dog or cat
- Name - if given one
- Quantity - some come as a litter, some are bonded pairs, most are individuals
- Physical descriptors: color, breed, fur length, health, shots, sterilized, maturity size and age
- number of pictures or videos posted
- a written description of the pet(s)
- Fee from free and up
- RescuerID & State - who found, and where the pet is
- Adoption Speed - how long it took them to be adopted, and what we are trying to create a model to predict

Data Exploration

Calvin from petfinder.com

14993 rows, 24 columns before cleaning

We found 12 nulls in the description, and we dropped those entries as the number of nulls was low. We changed description to word length/count,to make it something we can include into the model, as we felt it was likely to be an important feature.

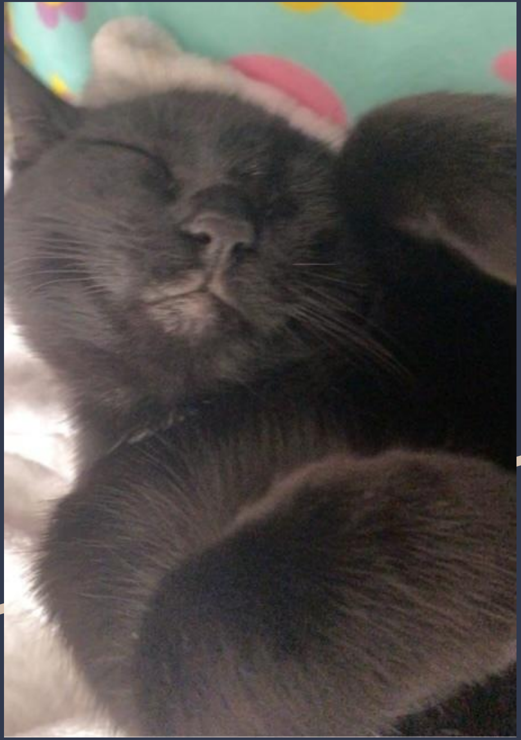
We binned quantity of pets, fee and photo amount into categories, as they each had too many unique values, and we wanted to scale the data before plugging it into the model.

Several columns were ID columns - we did not use in the model as they are just ‘noisy’ data.

Assumptions

- We expect the **type** of pet to have an impact on our model
- Black pets have been known to be less adoptable, so **color1** likely to have an impact
- The **health** of the pet might have a significant impact
- We assume **age** of pet to have a large impact. Everyone loves kittens and puppies!
- We expect **quantity of photos** to have an impact on adoption.
- **Description** will likely be a top half feature, as most people probably read them

Description of the analysis phase



Luca from petfinder.com

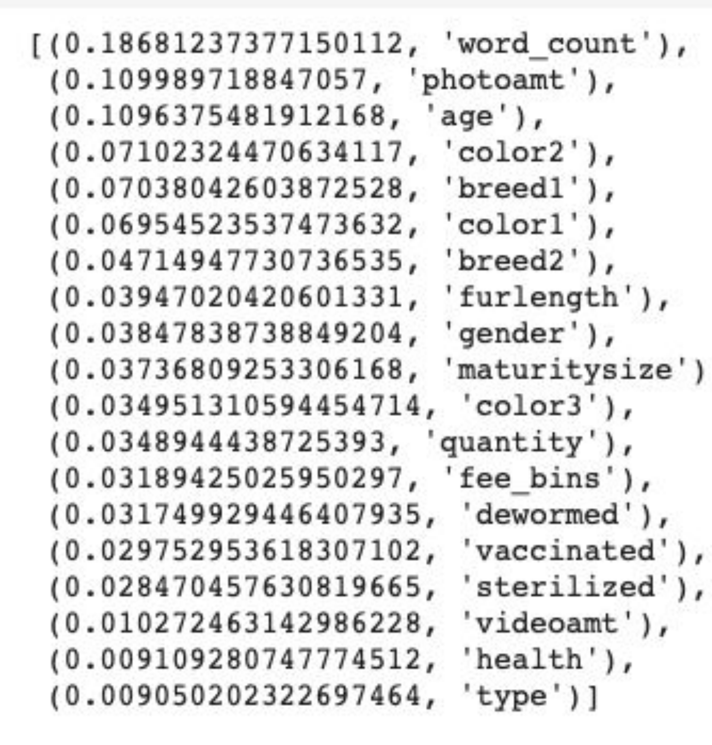
We used Supervised Machine Learning Model to predict adoption speed. We used the data we cleaned/transformed from exploration in the model.

We now have 14981 data points with **5 possible outcomes** for adoption speed

- 0 adopted on the same day as it was listed
- 1 adopted between 1 and 7 days after being listed
- 2 adopted between 8 and 30 days after being listed
- 3 adopted between 31 and 90 days 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

Target Accuracy 42.3+% as that would at least get us in the ‘bronze’. The top 10 (non-cheating) finalists were all at 44.1+% in the initial contest 3 years ago.

RandomForest Models



```
[(0.18681237377150112, 'word_count'),
 (0.109989718847057, 'photoamt'),
 (0.1096375481912168, 'age'),
 (0.07102324470634117, 'color2'),
 (0.07038042603872528, 'breed1'),
 (0.06954523537473632, 'color1'),
 (0.04714947730736535, 'breed2'),
 (0.03947020420601331, 'furlength'),
 (0.03847838738849204, 'gender'),
 (0.03736809253306168, 'maturitysize'),
 (0.034951310594454714, 'color3'),
 (0.0348944438725393, 'quantity'),
 (0.03189425025950297, 'fee_bins'),
 (0.031749929446407935, 'dewormed'),
 (0.029752953618307102, 'vaccinated'),
 (0.028470457630819665, 'sterilized'),
 (0.010272463142986228, 'videoamt'),
 (0.009109280747774512, 'health'),
 (0.009050202322697464, 'type')]
```

We used a standard scaler prior to the train/test/split to standardize the data for the model

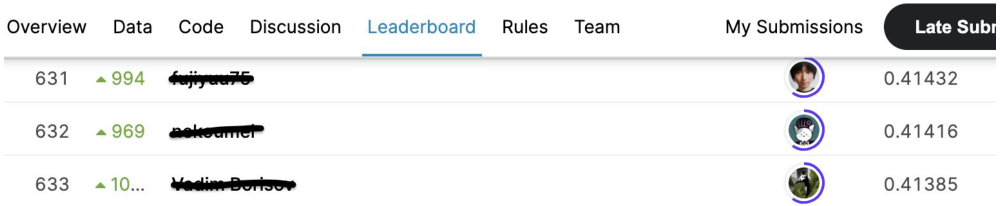
We will use feature importance pulled from first model to improve to try new model runs based on dropping low features.

75/25 train/test split

First model run all features: accuracy 41.4%

Feature importance: see image left

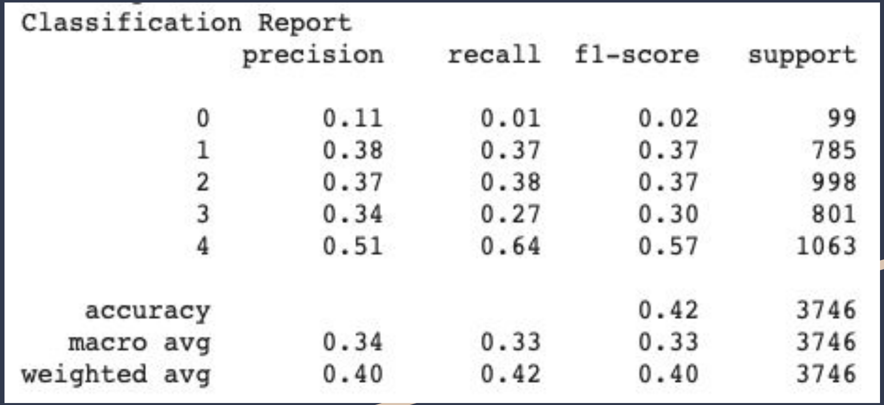
Comp ranking: around 632/633 out of 2023 entries, not in bronze though



631	▲994	[redacted]	0.41432
632	▲969	[redacted]	0.41416
633	▲10...	[redacted]	0.41385

RF Models cont.

Second model run less type, health and video amount features: accuracy 41.6% best model created in RF at 75/25 split, see image left of classification report.

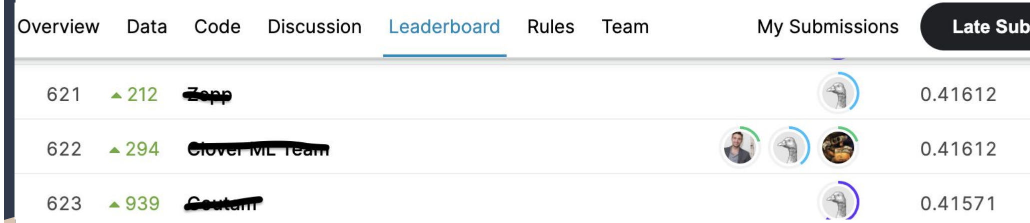


```
Classification Report
              precision    recall  f1-score   support

           0       0.11      0.01      0.02        99
           1       0.38      0.37      0.37       785
           2       0.37      0.38      0.37       998
           3       0.34      0.27      0.30       801
           4       0.51      0.64      0.57      1063

    accuracy                           0.42      3746
   macro avg       0.34      0.33      0.33      3746
weighted avg       0.40      0.42      0.40      3746
```

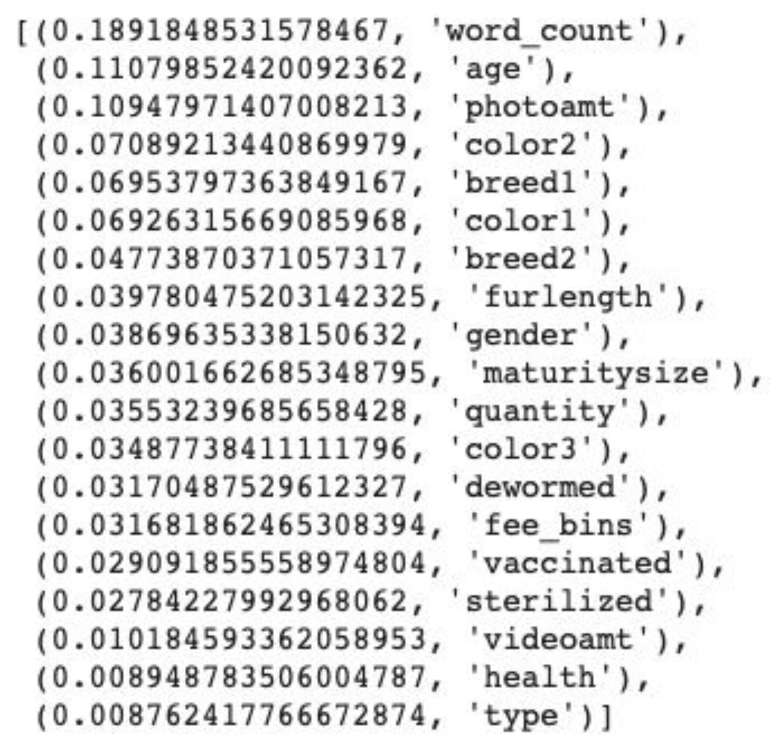
Comp ranking: around 621/622 out of 2023 entries, not in bronze though



621	▲212	[redacted]	0.41612
622	▲294	[redacted]	0.41612
623	▲939	[redacted]	0.41571

Ran several other RF iterations 75/25 - dropping more features, changing word count feature, etc. - *but none produced as strong of model as the first two passes*

RF Models cont.



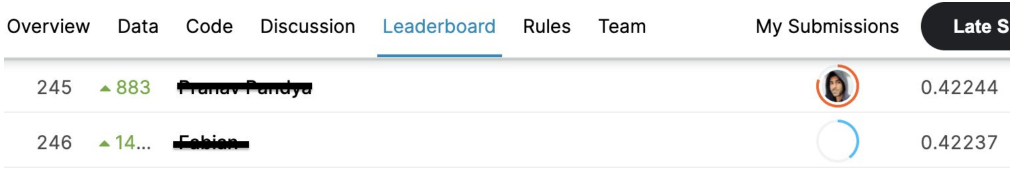
```
[(0.1891848531578467, 'word_count'),
 (0.11079852420092362, 'age'),
 (0.10947971407008213, 'photoamt'),
 (0.07089213440869979, 'color2'),
 (0.06953797363849167, 'breed1'),
 (0.06926315669085968, 'color1'),
 (0.04773870371057317, 'breed2'),
 (0.039780475203142325, 'furlength'),
 (0.03869635338150632, 'gender'),
 (0.036001662685348795, 'maturitysize'),
 (0.03553239685658428, 'quantity'),
 (0.03487738411111796, 'color3'),
 (0.03170487529612327, 'dewormed'),
 (0.031681862465308394, 'fee_bins'),
 (0.029091855558974804, 'vaccinated'),
 (0.02784227992968062, 'sterilized'),
 (0.010184593362058953, 'videoamt'),
 (0.008948783506004787, 'health'),
 (0.008762417766672874, 'type')]
```

Adjusted **split to 80/20** and use best two model parameters from first iterations

First model run (technically model 9) all features: accuracy 42.2%

Feature importance - changed slightly from first model age and photo amount switched places, image left

Competition ranking: 245/246 out of 2023 entries, not in bronze (but only barely).

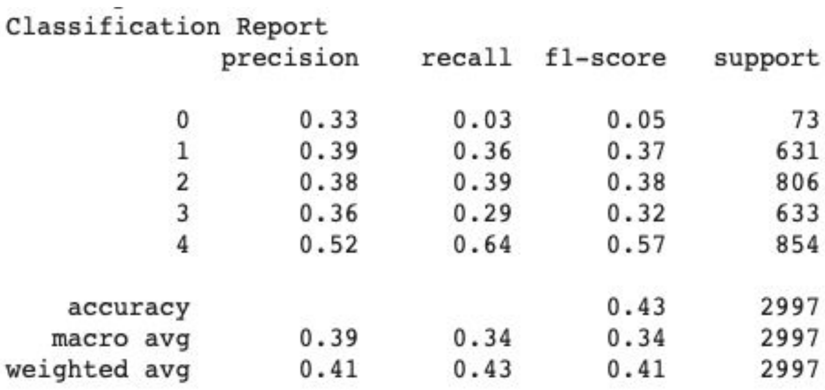


245	▲883	[redacted]	0.42244
246	▲14...	[redacted]	0.42237

The Final Model – In the bronze!

Second model run (technically model 10) less type, health and video amount features: accuracy 42.5% our best model thus final model

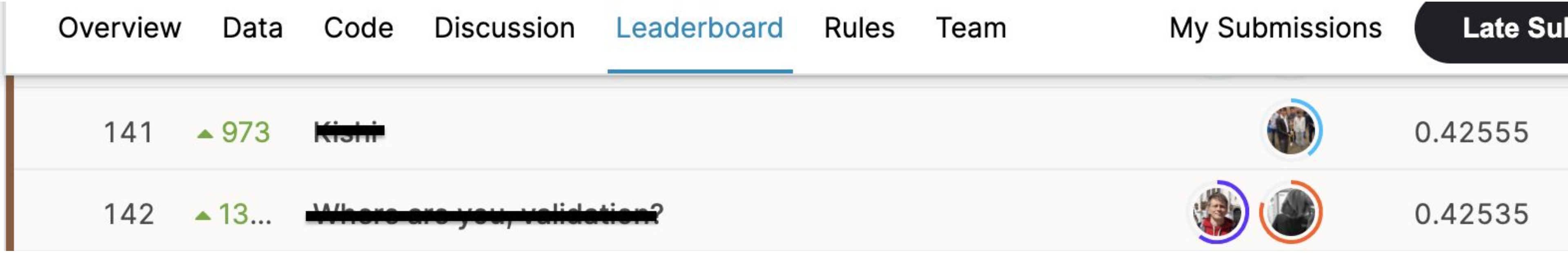
Competition ranking: 141/142 out of 2023 entries, **in bronze! Minimum target accuracy goal hit.**



```
Classification Report
              precision    recall  f1-score   support

           0       0.33      0.03      0.05        73
           1       0.39      0.36      0.37       631
           2       0.38      0.39      0.38       806
           3       0.36      0.29      0.32       633
           4       0.52      0.64      0.57       854

    accuracy                           0.43      2997
   macro avg       0.39      0.34      0.34      2997
weighted avg       0.41      0.43      0.41      2997
```



141	▲ 973	[redacted]	0.42555
142	▲ 13...	[redacted]	0.42535

Findings

Bonnie from petfinder.com

Top 3 features

- Word count
- Age
- Photo amount

Features that didn't matter/ held back model

- Type of pet
- Health
- Video Amount

Some of these results were surprising to us, especially the type and word_count features. Once we dropped the bottom three features, our models improved.

Suggestions based on findings:

Fluff from petfinder.com

- **Have a description,** not too long less than 200 words but also more than 40 seems to be key at the faster adoption speeds.
- **Have photos of the pet(s)** between at minimum 1 images and up to 5 appears to be the ticket for faster adoption speeds. More than 5 waste of resource
- Age matters the bulk of all pets regardless of speed were under 65 months, and there appears to be a correlation with adoption speed and age as age goes up so does length of time to adopt. **Perhaps consistent photos/descriptions would help with older pets.**
- **No video needed.** One of the least important features so either not enough videos to fairly judge or videos aren't needed thus a waste of resources.
- Focus on Health of the pet also not necessary. If putting in description probably a 'waste of words.

If had more time, what we could have done differently

Things we could have tried to better our model:

Different binning/categorization of features for the in model

Other machine learning model types to compare, for example k-means or SMV

Full NLP on description - top feature when in ‘word_count’ form, looking at sentiment, etc. could have likely lead to better model results.

Other limitations

Mango from petfinder.com

The Data --

Limited amount of data points in Adoption Speed 0, which is also where even our best model struggled the most with predicting.

Adoption Speed already binned prior to our accessing the data. Given that adoption speed is based on a timeline, had this not been categorized we could have performed a Linear Regression Model which may have given us more clues as to where things weren't aligning, and perhaps could have produced a more accurate model.

Lastly, if there were images provided perhaps an analysis could have been done on those to see how the image itself might impact the adoption timeline.

Technology Used

- Python and Pandas used for Data Cleaning and Analysis. Natural Language Toolkit(NLTK) to remove stop words in the description column within our dataset (second cleaning pass)
- AWS RDS Postgres for the database, pgAdmin 4 to interface
- SQLAlchemy to connect the database and Google Colab notebooks
- sklearn library to run our model, using RandomForestClassifier
    - Analytical functions used: accuracy_score, classification_report, confusion_matrix, and feature_importances_
- Using Tableau to create visuals, and as the platform from which we will present, integrating slides from Google Slides

Project sources