Team Pet Adoption
Progress Report I

By: Yuwen Dai, Allison Yuan, and Fred Watts

Section 1: Introduction of the project background and objectives

Our team has chosen to study animal welfare. Millions of abandoned or lost animals end up in shelters. These shelters are often overcrowded, and some euthanize animals that aren't adopted quickly. We propose that studying adoption data may help shelters and rescue organizations match with potential adopting homes more effectively to promote adoptions. Easing shelter overcrowding reduces animal suffering.

In this project, our main datasets are Malaysian animal shelter records from a Kaggle competition. Contestants compete to produce algorithms to predict adoption rates from traits in the shelter animals' online profile such as breed, color, gender, health, the quality of profile images, and the quality of descriptive text. The goal of our project is to leverage data science techniques to identify relevant features and uncover their relationship to the adoption successful rate and speed of pets, and therefore improve pet adoption experience.

Section 2: Dataset description

These animal shelter records are available at the Kaggle competition website (*https://www.kaggle.com/c/petfinder-adoption-prediction/*). The main dataset contains 14,993 observation and 27 variables, with adoption records on both dogs and cats (*Fig. 1*). After investigating this initial data, our team decided its analytical uses were limited, so we searched for other datasets on pet adoption motives.

First, we determined that the breed variable was too diverse to draw large sample sizes for individual breeds (for instance, out of 176 different breeds, 37 had only one observation), and it seemed that breed was not a strong factor in decision-making regardless. However, we knew that certain breeds have strong tendencies toward particular personality types and energy levels. We predicted that those traits would drive adoption decisions more so than breed itself while being easier to measure. We found data on those traits from the American Kennel Club and Cat Fanciers Association websites (*akc.com, cfa.com*). Also, there are significant portion of breed types (e.g. 37 breed types only have one observation) doesn't have enough observation. To group pets by personality and energy level can make sure each category has sufficient observation counts.

Second, we hypothesized that the adoption speed and success rate depended on how the supply of shelter pets related to the number of local people looking to adopt. The Kaggle data provided the pet numbers and the state where they were located, but it said nothing on the local human population. To approximate this, we found each state population on a Malaysian government website (*pmr.penerangan.gov.my*). With this new data, we can add state-level pet to population ratios to reflect this supply and demand relationship.

Variable Name	Variable Description	Value Definition
PetID	Unique hash ID of pet profile	
AdoptionSpeed	Categorical speed of adoption. The value to predict.	Lower is faster.
Type	Type of animal	*1 = Dog, 2 = Cat*
Name	Name of pet	*Empty if not named*
Age	Age of pet when listed, in months	
Breed1	Primary breed of pet	*Refer to BreedLabels dictionary*
Breed2	Secondary breed of pet	*Refer to BreedLabels dictionary*
Gender	Gender of pet	*1 = Male, 2 = Female, 3 = Mixed, if profile represents group of pets*
Color1	Color 1 of pet	*Refer to ColorLabels dictionary*
Color2	Color 2 of pet	*Refer to ColorLabels dictionary*
Color3	Color 3 of pet	*Refer to ColorLabels dictionary*
MaturitySize	Size at maturity	*1 = Small, 2 = Medium, 3 = Large, 4 = Extra Large, 0 = Not Specified*
FurLength	Fur length	*1 = Short, 2 = Medium, 3 = Long, 0 = Not Specified*
Vaccinated	Pet has been vaccinated	*1 = Yes, 2 = No, 3 = Not Sure*
Dewormed	Pet has been dewormed	*1 = Yes, 2 = No, 3 = Not Sure*
Sterilized	Pet has been spayed / neutered	*1 = Yes, 2 = No, 3 = Not Sure*
Health	Health Condition	*1 = Healthy, 2 = Minor Injury, 3 = Serious Injury, 0 = Not Specified*
Quantity	Number of pets represented in profile	
Fee	Adoption fee	*0 = Free*
State	State location in Malaysia	*Refer to StateLabels dictionary*
RescuerID	Unique hash ID of rescuer	
VideoAmt	Total uploaded videos for this pet	
PhotoAmt	Total uploaded photos for this pet	
Description	Profile write-up for this pet.	Primary language is English, with some Malay or Chinese.

Fig. 1: Training Data Dictionary

Section 3: Exploratory data analysis on the combined dataset

In this section, we applied a set of exploratory data analysis tools, including summary statistics, frequency tables, correlation matrix, histogram, boxplot, etc to examine the variable distribution pattern, correlation and interaction with other variables. We also conducted data quality check at both variable and observation level to check missing values and extreme values in variables, and to confirm there is no logic error entries in the data.

Overall, our main Kaggle dataset had high-quality data. It required only a few treatments to clean, which will be discussed in detail in the following sections.

Summary Statistics

First of all, we run summary statistics analysis on all variables (*Fig. 2*) to examine potential missing value, extreme values, and skewed distribution in the variables. At first glance, it seems there is no missing value issue in the dataset. However, we noticed that there are some variables (i.e. breed1, and age) have 0 value, which logically equals to missing. In later section, we will discuss in detail on how to handle those missing values. Based on Fig. 2, we also discovered that age variable has extremely high values. We will discuss the extreme value treatment in a later section (*Fig. 7*). We also found that variables fee, VideoAmt, and PhotoAmt have left skewed distribution. We will discuss our treatment on the skewed variables later in detail (*Fig. 6*).

Correlation Matrix

First, we run correlation between all explanatory variables with the target variable (adoption speed). Unfortunately, all the explanatory variables have weak correlation with the target variable. Their correlation values range between -0.1 and 0.1 (*Fig. 4*). There are so many levels in the target variable, and the target variable only has weak correlation with all explanatory variables. We hypothesized that the adoption speed is indifferent to the features provided in the dataset, but possibly the ultimate adoption decision has stronger correlation with some of the available features. Therefore, we created an adoption flag binary variable. In this simplified variable 1 means the pet was adopted within 100 days since listed, and 0 means the pet was not adopted within 100 days since listed. This new binary variable will be our new target variable for our modeling efforts.

Frequency Table

From the summary statistics, we also noticed that some of the variables are categorical variables. Therefore, we run frequency tables on those variables to check count, row/column percentages by themselves, and cross-tabulated with the target variable (i.e., adoption speed or adoption flag). We found a few variables particularly notable (*Fig. 3*). For example, most maturity sizes have adoption rates near the average (~71%), except for the largest size of 4. These very large pets average a 90% adoption success rate. However, the overall correlation between maturity size and adoption rate is low (-0.023). This size outlier is likely caused by the extremely low event rate for extra large pets. There are only 33 extra large pets, or 0.22% of the population. In the modeling stage, we will pay attention to this and explore some machine learning techniques (e.g. random forest, naive Bayes classifiers) to mitigate it.

Most categorical variables in the dataset had insignificant impact on adoption success rates, except for Vaccinated, Dewormed, Sterilized, and State variables. In the modeling stage, we will convert the multi-level categorical variable into several binary indicators so that certain significant levels in those variables can be easily captured in the model specification (*Fig. 3*).

	Type	Age	Breed1	Breed2	Gender	\
count	14993.000000	14993.000000	14993.000000	14993.000000	14993.000000	
mean	1.457614	10.452078	265.272594	74.009738	1.776162	
std	0.498217	18.155790	60.056818	123.011575	0.681592	
min	1.000000	0.000000	0.000000	0.000000	1.000000	
25%	1.000000	2.000000	265.000000	0.000000	1.000000	
50%	1.000000	3.000000	266.000000	0.000000	2.000000	
75%	2.000000	12.000000	307.000000	179.000000	2.000000	
max	2.000000	255.000000	307.000000	307.000000	3.000000	

	Color1	Color2	Color3	MaturitySize	FurLength	\
count	14993.000000	14993.000000	14993.000000	14993.000000	14993.000000	
mean	2.234176	3.222837	1.882012	1.862002	1.467485	
std	1.745225	2.742562	2.984086	0.547959	0.599070	
min	1.000000	0.000000	0.000000	1.000000	1.000000	
25%	1.000000	0.000000	0.000000	2.000000	1.000000	
50%	2.000000	2.000000	0.000000	2.000000	1.000000	
75%	3.000000	6.000000	5.000000	2.000000	2.000000	
max	7.000000	7.000000	7.000000	4.000000	3.000000	

	Vaccinated	Dewormed	Sterilized	Health	Quantity	\
count	14993.000000	14993.000000	14993.000000	14993.000000	14993.000000	
mean	1.731208	1.558727	1.914227	1.036617	1.576069	
std	0.667649	0.695817	0.566172	0.199535	1.472477	
min	1.000000	1.000000	1.000000	1.000000	1.000000	
25%	1.000000	1.000000	2.000000	1.000000	1.000000	
50%	2.000000	1.000000	2.000000	1.000000	1.000000	
75%	2.000000	2.000000	2.000000	1.000000	1.000000	
max	3.000000	3.000000	3.000000	3.000000	20.000000	

	Fee	State	VideoAmt	PhotoAmt	AdoptionSpeed
count	14993.000000	14993.000000	14993.000000	14993.000000	14993.000000
mean	21.259988	41346.028347	0.056760	3.889215	2.516441
std	78.414548	32.444153	0.346185	3.487810	1.177265
min	0.000000	41324.000000	0.000000	0.000000	0.000000
25%	0.000000	41326.000000	0.000000	2.000000	2.000000
50%	0.000000	41326.000000	0.000000	3.000000	2.000000
75%	0.000000	41401.000000	0.000000	5.000000	4.000000
max	3000.000000	41415.000000	8.000000	30.000000	4.000000

Fig. 2: Frequency Tables

AdoptionSpeed	0	1	2	3	4
MaturitySize					
1	0.040943	0.256259	0.260383	0.188807	0.253608
2	0.021931	0.185250	0.275303	0.229015	0.288501
3	0.034127	0.241270	0.242063	0.196825	0.285714
4	0.060606	0.212121	0.333333	0.303030	0.090909

Adopted Flag	0	1
MaturitySize		
1	0.253608	0.746392
2	0.288501	0.711499
3	0.285714	0.714286
4	0.090909	0.909091

AdoptionSpeed	0	1	2	3	4
Type					
1	0.020905	0.176463	0.266109	0.239670	0.296852
2	0.034980	0.241218	0.272992	0.190934	0.259875

Adopted Flag	0	1
Type		
1	0.296852	0.703148
2	0.259875	0.740125

AdoptionSpeed	0	1	2	3	4
Gender					
1	0.028902	0.231756	0.285043	0.200325	0.253974
2	0.028034	0.187715	0.262608	0.229628	0.292016
3	0.021101	0.202294	0.251376	0.219725	0.305505

Adopted Flag	0	1
Gender		
1	0.253974	0.746026
2	0.292016	0.707984
3	0.305505	0.694495

AdoptionSpeed	0	1	2	3	4
FurLength					
1	0.022025	0.187216	0.269301	0.226385	0.295073
2	0.029845	0.221414	0.269353	0.207237	0.272151
3	0.067961	0.308252	0.268204	0.186893	0.168689

Adopted Flag	0	1
FurLength		
1	0.295073	0.704927
2	0.272151	0.727849
3	0.168689	0.831311

Fig.3: Adoption Rates Over Some Major Variables

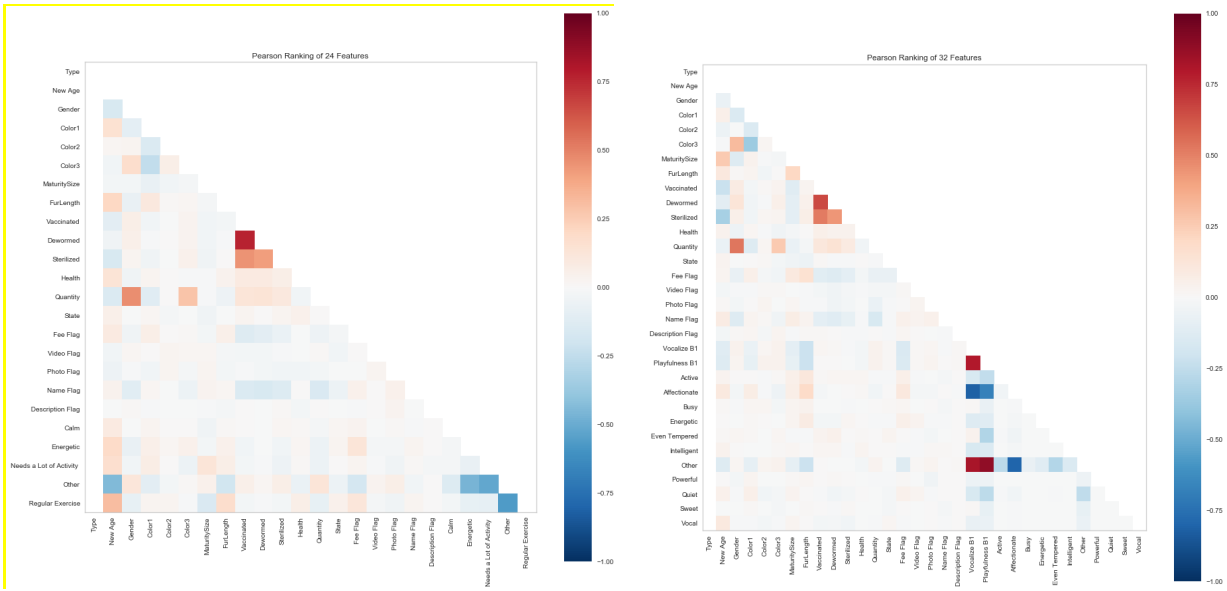

Fig.4: Rank2D plot for Dogs and Cats

Data Quality Check

Next, we performed a data quality check. We searched to see whether any variables had missing or contradictory information in any observation. For example, if Color3 !=0 in any observation when Color2 == 0, or any observation when breed2 != 0 while breed1 == 0 (multi-numbered variables exist to express mixed traits, but unmixed traits default to the lower number). We found that the color variables were coded correctly, but there were five cases where Breed1 equaled to 0 while Breed2 didn't. Given there are only five cases, we assumed this was data entry error and assigned the Breed1 equals to the Breed2 value for those five cases.

Data Treatment

Based on summary statistics on the data, we also created a few new variables:

- A "Mixed Breed" variable. The Kaggle dataset comes with two variables related to breed type. Breed1 has too many categories, and many lack sufficient observation count for statistical modeling. And Breed2 has many missing values. Therefore, it is more useful to create a binary "Mixed Breed" variable based on whether Breed1 == 'mixed' OR Breed1 != 'mixed' and breed2 != 0.
- A flag indicator for skewed variable. The Fee variable is continuous, but it skews highly toward the leftmost value of 0 (i.e. there are many free pets), and there are some extremely high outlier values on the other end as well. This highly skewed distribution could easily become insignificant in the model or potentially bias the estimation. A binary variable would ameliorate this. Therefore, we created a "No Fee" flag to show whether Fee == 0. We did a similar treatment on the VideoAmt and PhotoAmt for the same reason.

```
In [43]: pd_train_new['MixedBreed']=np.where((pd_train_new['Breed1 New']==307)|((pd_train_new['Breed1 New']!=pd_train_new['Breed2'])
                                                                                &(pd_train_new['Breed2']!=0)) , 'Yes', 'No')
         pd_train_new
```

Fig.5: 'Mixed Breed' Code Sample

We searched the number of values missing in all variables and found Breed1 and Age have logically missing values. Given Breed1 is a categorical variable and we intended to replace breed variables with energy level and personality type variables (see Section 2), we decide not to further process those missing values in Breed1 variable. While, we are replacing the missing values for Age with the median.

We discovered that the Age variable (measuring the pet's age in months) shows extreme values at the right end (i.e., unrealistically old ages suggesting input error) which would bias statistical measures. To solve this, we capped the age at its 99th percentile, which is 84 months (*Fig. 7 and 8*).

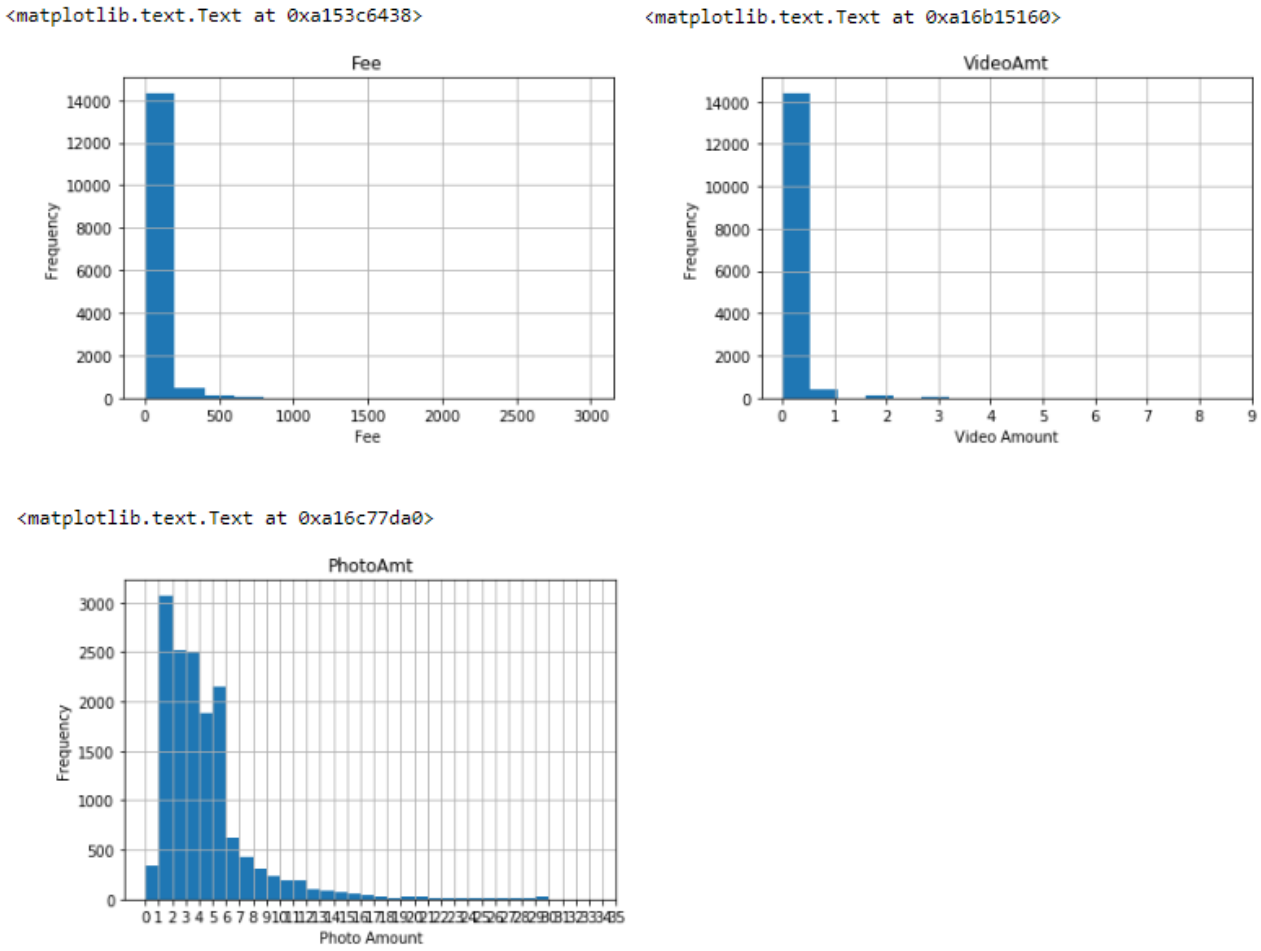

Fig. 6: Bar Graphs Showing High-Outlier Continuous Variables

```
count    14993.000000
mean        10.452078
std         18.155790
min          0.000000
25%          2.000000
50%          3.000000
75%         12.000000
max        255.000000
Name: Age, dtype: float64
```

```
count    14993.000000
mean        10.154333
std         16.363657
min          0.000000
25%          2.000000
50%          3.000000
75%         12.000000
max         84.000000
Name: New Age, dtype: float64
```

```
pd_train_new['New Age']=np.where(pd_train_new['Age']>84,84,pd_train_new['Age'])
```

Fig. 7: Age Variable Statistics before and after Percentile Cap Before

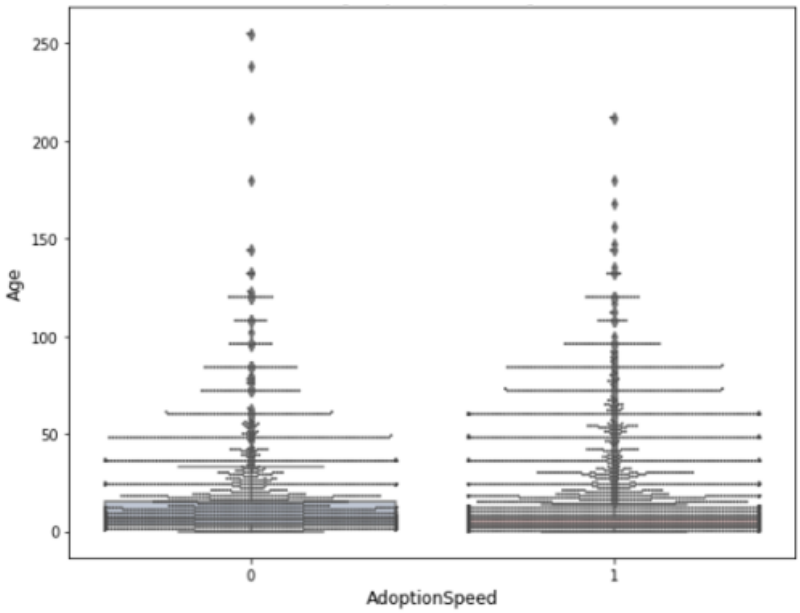

Fig. 8: Age by Adopted Flag

Section 4: Preliminary Model Development and Feature Selection

In our dataset, there are two types of animals, dogs and cats. These have different attributes and potentially attract different patterns of adopters. For example, dog adopters would look for the energetic level while cat adopters would look for vocalization. Also the data indicate so as shown in the figures below.

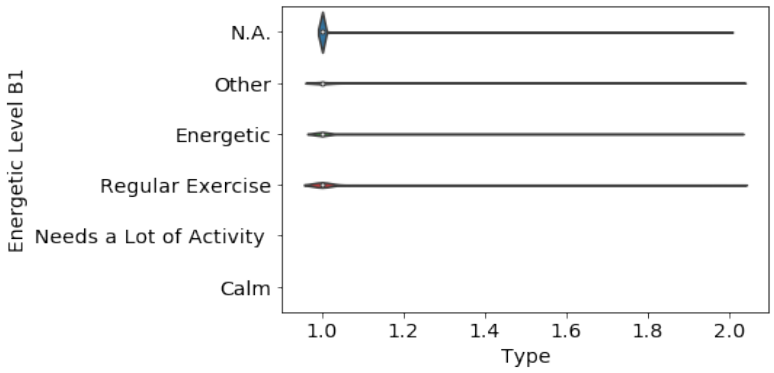

Fig 9. Violin plot of Energetic Level by Animal Type

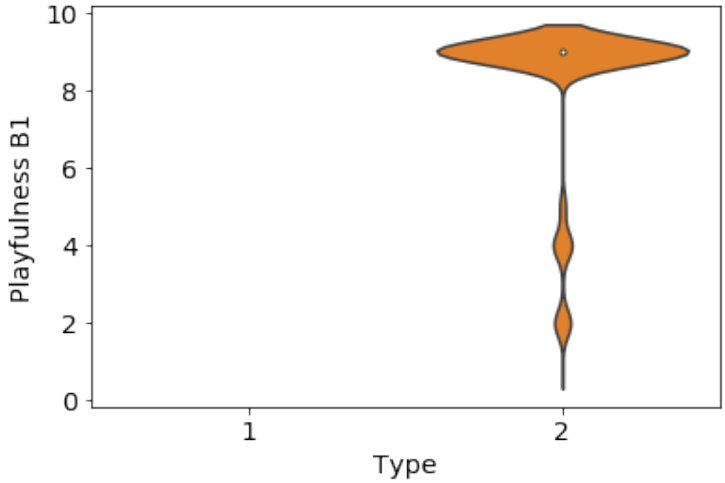

Fig 10. Violin plot of Playfulness by Animal Type

Therefore, it made sense to split the data by animal and independently develop models for each. model. First, we plot a learning curve using random forest model to explore the potential optimal training/testing data split. However, the learning curve indicate that there is no obvious model performance difference along different data split. Therefore, we use rule of thumb to split the data by 7:3 ratio. Given 8132 observations in dog data and 6861 observations in cat data, the 7:3 split should give enough observations for training and testing. Giving more weight on the training dataset, it can potentially feed as much information as possible for the model development and hyperparameters tuning. Additionally, we checked the data balance and confirmed that the data is fairly balanced (Fig 12) .

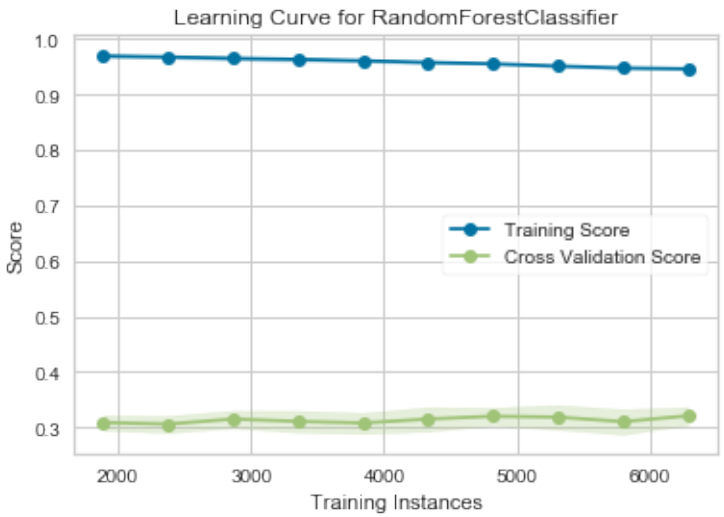

Fig 11. Learning Curve for Random Forest Classifier

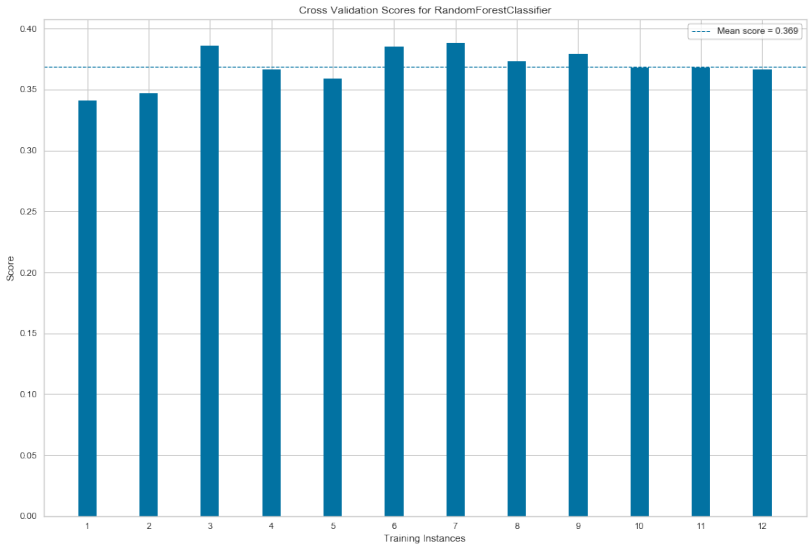

Fig 12. Cross Validation Scores for Random Forest Model

Given our target variable is a multi-level categorical variable, we assessed that it is a classification problem. We have explored a set of classification methods, including KNN, Random Forest, XGBooting, and SVC. And we followed the following steps to ultimately identified our final champion models for dogs and cats respectively:

1. Hyperparameter tuning on the primary hyperparameter in each candidate model to find the best hyperparameter value based on accuracy score, which is the ratio of correctly predicted observations to the total observations. An example plot of accuracy score vs hyperparameter is shown in Fig 13.
2. Calculate the classification report and feature importance plot based on the optimal hyperparameter selected from step 1 on each model. One caveat is that SVC classifier

doesn't have feature importance, because the kernel function we used is RBF. An example classification report and feature importance plot are shown in Fig 14.

3. Run ROC plot on all the optimal models specified in Step 2, Fig 15.
4. Select the best classification model based on F1 score from the candidate models in Step 2. Our project is a multi-class classification problem, in order to account for potential class imbalance, we used micro F1 score as the selection criterium. XGBoosting has the best performance for dogs, while SVC performs the best for cats.

SVC: 0.3889344262295082 KNeighborsClassifier: 0.3737704918032787 RandomForestClassifier: 0.3729508196721312 **XGBClassifier:** **0.39057377049180325**	**SVC: 0.3705682370082565** KNeighborsClassifier: 0.3516270033997086 RandomForestClassifier: 0.34191355026711995 XGBClassifier: 0.35988343856240895

5. Run grid search on expanded hyperparameters to further fine tune the models. For XGBoosting, we fine tuned on n_estimator, learning_rate, subsample, max_depth, colsample_bytree, min_child_weight. For SVC, we fine tuned on gamma, and C. Fig 16 shows the detailed model parameters for our final champion models for dogs and cats, and their performance (e.g. classification report).

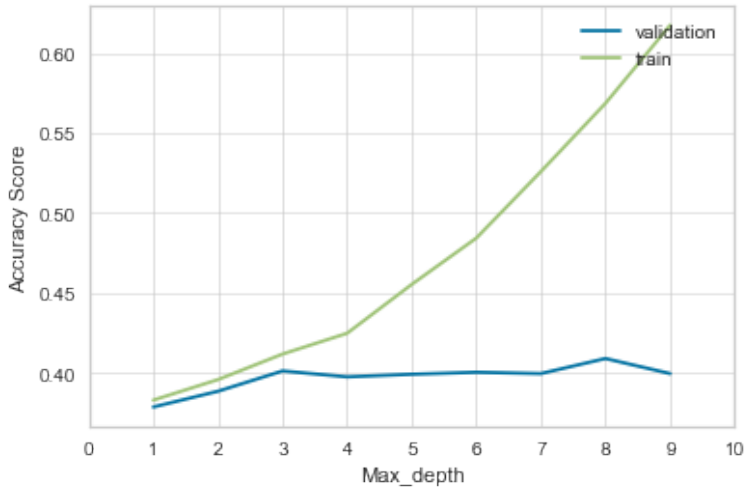

Fig 13. Accuracy Score vs Max_depth for XGBoosting

	precision	recall	f1-score	support
0	0.18	0.06	0.09	50
1	0.31	0.32	0.31	422
2	0.34	0.40	0.37	684
3	0.38	0.20	0.26	582

4	0.51	0.64	0.57	702
micro avg	0.40	0.40	0.40	2440
macro avg	0.34	0.32	0.32	2440
weighted avg	0.39	0.40	0.38	2440

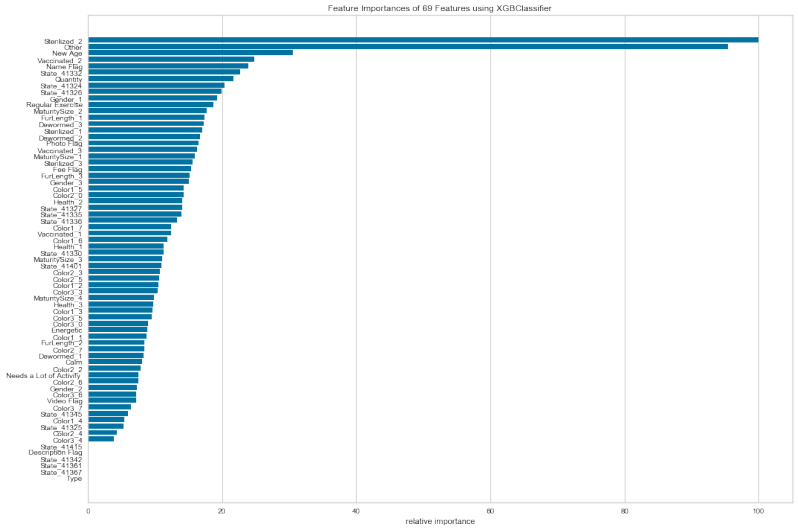

Fig 14. Classification Report and Feature Importance Plot for XGBoosting

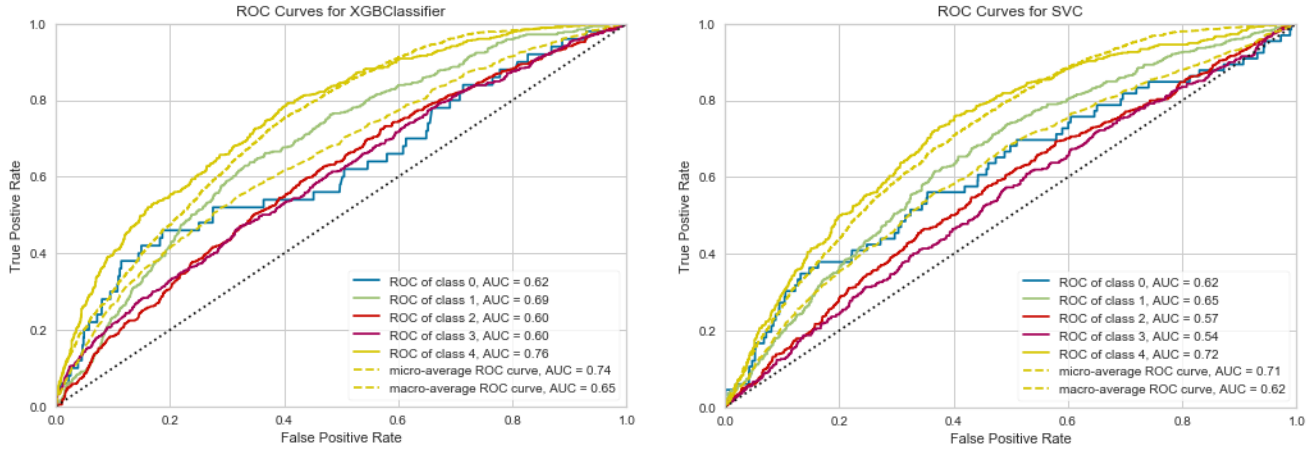

Fig 15. ROC Curves on XGBoosting and SVC

```
XGBClassifier(base_score=0.5, booster='gbtree', colsample_bylevel=1,
       colsample_bynode=1, colsample_bytree=0.9142538281542341, gamma=0,
       learning_rate=0.17308319853302173, max_delta_step=0, max_depth=6,
       min_child_weight=2, missing=None, n_estimators=377, n_jobs=1,
       nthread=None, objective='multi:softprob', random_state=0,
       reg_alpha=0, reg_lambda=1, scale_pos_weight=1, seed=None,
       silent=None, subsample=0.46955765814009887, verbosity=1)

Detailed classification report:

The model is trained on the full development set.
The scores are computed on the full evaluation set.

              precision    recall  f1-score   support

           0       0.11      0.04      0.06        50
           1       0.36      0.26      0.31       461
           2       0.33      0.37      0.35       646
           3       0.34      0.32      0.33       603
           4       0.49      0.58      0.54       680

   micro avg       0.39      0.39      0.39      2440
   macro avg       0.33      0.32      0.32      2440
weighted avg       0.38      0.39      0.38      2440
```

```
SVC(C=1, cache_size=200, class_weight=None, coef0=0.0,
  decision_function_shape='ovr', degree=3, gamma=0.1, kernel='rbf',
  max_iter=-1, probability=False, random_state=None, shrinking=True,
  tol=0.001, verbose=False)

Detailed classification report:

The model is trained on the full development set.
The scores are computed on the full evaluation set.

              precision    recall  f1-score   support

           0       1.00      0.03      0.06        66
           1       0.37      0.40      0.38       505
           2       0.35      0.39      0.37       585
           3       0.24      0.06      0.10       389
           4       0.41      0.59      0.48       514

   micro avg       0.37      0.37      0.37      2059
   macro avg       0.47      0.30      0.28      2059
weighted avg       0.37      0.37      0.34      2059
```

Fig 16. Final Champion Model Parameters for Dogs and Cats

Section 5: Member Contribution

Yuwen: Designed the hypothesis, the entire project data analytical and model methodological approach and steps to execute. Designed all the functionalities and Python code used in data wrangling, EDA tests and analysis, model and feature selection, hyperparameter tuning, and model diagnostics used in Python. Analyzed all the results, prepared the final Jupyter Notebook script based on Allison's draft code, and authored all the project reports and the final presentation PowerPoint deck.

Allison: Created and wrote Python code for data cleaning and wrangling, EDA, and all modelling parts such as feature selection, classification performance, and hyperparameter tuning. Updated and debugged code based on discussions with the team members and did research for various functions and packages in Python. Cleaned up final Jupyter Notebook scripts.

Fred: Researched and proposed secondary data sources (geography and population). Participated and tested wrangling solutions. Proofread and edited all reports. Co-drafted and edited PowerPoint deck. Reviewed EDA.

Section 6: Code Summary

In our team's Georgetown Analytics Github repository, we have included a jupyter notebook (*PetAdoption-EDA-v1.4.ipynb*) with results from the first three parts of the pipeline (*Fig. 9*).

Raw data:

The code uses the *train.csv* dataset along with outside datasets (see Section 2) of breed behavior and state records to create Activity Level, Personality, Vocalize and Playfulness for each breed, and populations for each state.

Wrangling Module:

We created several additional variables: Fee Flag, Photo Flag, Video Flag, Adopted Flag, MixedBreed (see Section 3). For data processing, we capped Age at 99% and created a 'New Age' variable. For Breed1, we found five records with a 0 in Breed1 and a value in Breed2. We replaced these five records of Breed1 with their Breed2 value.

Exploratory Data Analysis:

We created a summary of all variables, a correlation matrix and Rank2D plot for all variables, two-way frequency tables for categorical variables including row percentages,

histograms for continuous variables, boxplot and percentile distributions for variables with extreme values like Age, and Violin plot of animal personality by Animal Type

Machine Learning Model Development and Feature Selection

We started our model exploration on random forest, KNN, XGBoosting, and SVC by using:

- Cross validation Analytics: Learning curve and cross validation score
- Training and testing data split
- Hyperparameter tuning: grid search, accuracy score vs hyperparameter plot
- Feature importance ranking and visualization: feature importance plot, AUC ROC plot
- Classification report