Final Project Report

Pet Adoption Likelihood (PAL)

Lily Nguyen lnguye78@ucsc.edu
Wan Fong wsfong@ucsc.edu
Andy Wong ankwwong@ucsc.edu

March 24, 2019

Abstract

In this project we built a machine learning model that could analyze the metadata of a pet's profile, provided by the adoption shelter, in order to predict how quickly a pet is likely to be adopted. We used an ensemble voting method, consisting of three individual classifiers: a Naive Bayes model, a Support Vector Machine (SVM) model, and a Logistic Regression model—all of which was implemented using the Scikit libraries. The performance of our model was evaluated using Scikit's cross validation tool. Within the timeframe of this project we were only able to reach an F1-score of about 38%, but we have also outlined ideas for how this model can be improved in the future. The code for this project is hosted on Github: https://github.com/nguyen-lily-p/cmps140.

Keywords: pet, adoption, machine learning, Naive Bayes, Support Vector Machine, Logistic Regression

1. Introduction

There is a strong correlation between the metadata associated with a pet and the speed at which it is adopted from a shelter. If we can find a way to predict the "adoptability" of a pet and find the underlying characteristics that lead to the animal being adopted more quickly, we can help shelters and animal rescues improve their pet's appeal in their online profile; we can potentially increase adoption rates, decrease animal suffering, and increase the welfare for many more pets than before.

This project was inspired by the kaggle competition PetFinder.my and our goals is to build a machine learning model that can find the underlying features that inform a pet's adoption speed and use this to predict the adoption speed for future pets. Broader impacts of this project could be to predict all kinds of things. For example given the right data set for cars we could predict how quickly the car would sell. It doesn't have to predict how fast it sells, instead given the right set of data about a automobile we can predict how often it breaks or if it is reliable based on certain features. This project idea uses features from specific data sets and priorities certain features and in turn would calculate and predict things based on those highlighted features.

2. Data

The data used in this project is provided by PetFinder.my as part of a competition hosted on Kaggle.com. The given data consists of multiple CSV files, with the main file containing various fields of information about each pet such as type of pet, color, age, gender, maturity size, health, etc. The sentiment analysis report on each pet's text description is provided in separate JSON files, with each JSON file corresponding to one specific pet.

We've separated our data based on whether the pet is a dog or a cat, and will implement two separate models for each. We decided to do this because of the possibility that some features may be stronger indicators for dogs than for cats, or vice versa. (For example: dogs vary more in size than cats do, and therefore Maturity Size may be a stronger factor in adoption speed for dogs, but may be a rather meaningless one for cats.)

	Adoption Speed					
Pet Type	**Class 0**	**Class 1**	**Class 2**	**Class 3**	**Class 4**	**Total**
Dog	168	1427	2153	1944	2394	8086
Cat	216	1522	1753	1232	1633	6356
Total	384	2949	3906	3176	4027	14442

As we can see from the table above, the data distribution for classes 1 - 4 are fairly close, with Class 1 having smaller representation than the other classes for both pets. However, Class 0 is significantly underrepresented in the data for both dogs and cats. This data imbalance could affect our models' performance; we will keep this in mind when revising our models, and will consider methods for dealing with an imbalanced data set.

3. Methodology

The input to our machine learning model is a csv file containing pets with features. What our model does it that it evaluates all these features and predicts an outcome. In our case the model will predict a class 0 through 4 for the adoption speed of each pet and output a text file of the evaluation of each classifier. The adoption speed of 0 will indicate that the pet will be adopted the same day it is posted. While the adoption speed of 4 will indicate that it will take more than 100 days to be adopted.

3.1 Research

We completed our initial research, which was all about learning the basics of solving a machine learning problem and figuring out how to approach our problem.

3.2 Set Up Work Environment

We created our Github repository and created our starter file for our main Python program. We also downloaded the data set from Kaggle.

3.3 Data Integration

We created some functions to read the data set into our Python program. Using the Pandas library, we read in the CSV data from the *train.csv* file and stored it in a Pandas dataframe object. We also read in the provided label code files (*breed_labels.csv*, *color_labels.csv*) to replace data codes with their actual values. For features that didn't have a provided label code file, we hard-coded dictionaries based on the description of the data fields from the Kaggle website. Additionally, some of the provided data was given as JSON files; that required us to parse through the JSON files to find the necessary information, and match that information to a data instance in the main data set based on the JSON file name and the data instance's *PetID* value.

3.4 Data Preprocessing

Our first preprocessing step was looking at the data to see if it needed to be cleaned. The only two problematic fields were the Name and Description columns, both of which contained unclean text data. These fields caused problems even when attempting to read in the data due to encoding issues; some of the text contained foreign language characters, and some also contained characters that were completely unrecognized. However, for reasons that will be discussed in the next section, these two features were dropped and therefore did not need to be cleaned.

In place of the Description column, we decided instead to include the sentiment analysis report on the Description column, provided in separate JSON files in the data set. However, some of the descriptions were unable to be analyzed, and therefore did not have an associated sentiment analysis report. We researched the ways to deal with missing data; in the end, since only 3.67% (551 out of 14,994) of the data was missing a value for that feature, we felt it was safe to simply drop those data instances.

We also considered whether we needed to perform feature scaling on our data, since the numerical fields (Age, Quantity, and Fee) all had very different ranges for values. However, the necessity of feature scaling is very much dependent upon the type of machine learning algorithm used: algorithms that learn weights for the features tend to be robust against unscaled features. Ultimately, we decided to not perform feature scaling and instead only use algorithms that assign weights to features (like Naive Bayes), and avoid algorithms that weigh all features equally (like KNN).

Our last consideration for preprocessing was binarization of features. Most of the features in our data set, such as Breed, Gender, and Fur Length, are categorical features. Obviously these feature values need to be turned into numbers in order to work with many algorithms, and these features were actually provided to us already encoded. (For example: the value for Breed is an integer, and the actual pet breed that number represents is listed in a separate file). However, this was not enough--these features are not ordinal, and we do not want our model to erroneously learn some sort of ordering between feature values. (For example: it would be disastrous for our model to assume 109 > 103, where 109 = Golden Retriever and 103 = German Shepherd.) We therefore decided to perform One Hot Encoding on all of our categorical features. Although this greatly increased the number of features (to about 500), in the end we felt that it was not an unreasonable amount.

3.5 Feature Engineering and Feature Selection

Our first step was to look at the provided data and eliminate any features that we felt were irrelevant to a pet's adoption speed. We immediately decided to drop the Rescuer ID and the State ID (which corresponds to the Malaysian state the pet was found in), as we believe that those features are not likely to significantly affect a pet's adoption likelihood. We also ended up dropping the Name and Description field because we felt that the amount of work it would take to clean and extract meaningful features from them could be an entire project in of itself. Also, we felt that incorporating the provided sentiment analysis report on the Description field as a feature would help mitigate the effect of dropping the Description field.

The sentiment analysis report on each data instance's description field contains both a score for emotion and a score for magnitude for every sentence in the description, as well as for the overall document. The emotion scores are between -1.0 and 1.0, where a negative value corresponds to negative emotions and a positive value corresponds to positive emotion. The magnitude score can be as little as 0.0, but has no upper bound; the higher the value, the stronger the emotion (whether negative of positive). We decided that a good way to use this information to assign a sentiment feature value to our data is to multiply the emotion and magnitude scores together; this essentially weighs the document's emotion using the magnitude (where documents with higher emotions will have higher values), while still preserving the overall sentiment of the data (negative documents will still have a negative value, positive documents will still have a positive value, and neutral documents will have values close to 0).

3.6 Implement the Machine Learning Model

Our program takes as input the data files provided by PetFinder.my on the Kaggle competition site. Our program performs the preprocessing and feature selection steps as outlined in the previous sections, outputting two new files: a preprocessed data set containing only dog data instances (*train_dog.csv*), and a preprocessed data set containing only cat data instances (*train_cat.csv*).

Our program then takes the two new data files and runs them through the actual machine learning models; we have two separate models, one for the dog training data and one for the cat training data. But the models for both are mostly the same. Each model consists of an ensemble voting method, and this ensemble method is composed of three individual classifiers. The individual classifiers are a Naive Bayes model (using the Gaussian Naive Bayes model from Scikit), a Support Vector Machine model (using the Linear Support Vector model from Scikit), and a Logistic Regression model (using the Logistic Regression model from Scikit). Each of the three individual classifiers takes in the provided data and generates their own model and their own predictions, which they send to the ensemble voting classifier. The voting classifier takes all of their predictions and, using a hard-voting system, makes a prediction based on the majority.

For evaluation, our program uses Scikit's cross_val_score to perform cross-validation on the model. It performs cross-validation for all of the individual classifiers as well as the ensemble classifier. It calculates several performance metrics (F1-score, accuracy, precision, recall) and writes the results to a text file.

See Appendix 8.4 for a diagram of the model and Appendix 8.5 for an example of the program's output.

4. Evaluation

The two models is evaluated on the training set, using cross-fold validation, with a *k*-value of 10. The performance is based on standard performance metrics, including quantities such as accuracy and F1-score. The more accurate and the closer our F1-score to 1 is what we determine as successful. Our model performs better than the expected value of randomly guessing (20% chance to guess adoption speed correctly since there are five classes), however our overall scores (especially for the Gaussian Naive Bayes model) are rather low.

As we can see from the results listed below, the overall performance of the dog model is better than the performance of the cat model (38% vs 35%). This could be due to the imbalanced data: there are about 2,000 more instances in the dog data set than in the cat data set.

Cross-Fold Validation (*k* = 10) for Dog Training Data				
Model	**Average**	**Average**	**Average**	**Average**

	F1-Score	Recall	Precision	Accuracy
Gaussian Naive Bayes	0.04612	0.04612	0.04612	0.04612
Linear Support Vector	0.38820	0.38770	0.38857	0.38819
Logistic Regression	0.38375	0.38375	0.38375	0.38375
Voting Ensemble Classifier	0.38499	0.38375	0.38474	0.38535

Cross-Fold Validation (*k* = 10) for Cat Training Data				
Model	**Average F1-Score**	**Average Recall**	**Average Precision**	**Average Accuracy**
Gaussian Naive Bayes	0.05176	0.05176	0.05176	0.05176
Linear Support Vector	0.35337	0.35384	0.35274	0.35321
Logistic Regression	0.35352	0.35352	0.35352	0.35352
Voting Ensemble Classifier	0.35132	0.35163	0.35132	0.35132

5. Conclusion

Over the course of this project, we found that implementing the machine learning model was surprisingly the easiest part of the project; the preprocessing, feature extraction, and model improvement steps proved to have been the most difficult. We've learned what kinds of steps are involved in preprocessing data before it can even be used in a machine learning model (text cleaning, feature scaling, label encoding, one-hot encoding, etc.).

We are rather disappointed with our final results; the final performance of our model is rather lackluster. We attempted to improve our model's performance by modifying the parameters for the classifiers. Some of the modifications—such as changing the penalty for the SVC model from L2 to L1, or changing the inverse of the regularization strength for the Logistic Regression model from 1.5 to 3.0—improved our model, but unfortunately these improvements were very minimal (there was only about +0.5% improvement for each change). However, if we had more time, we do have several ideas on how we would like to improve our model.

Given more time, we would like to further modify the parameters for each individual machine learning algorithm, as well as modify the class and model weights for the Voting Ensemble classifier. We would also consider exploring different machine learning algorithms and integrate them into our model. Additionally, we would add some more preprocessing steps to the data, such as methods that can help mitigate an unbalanced data set. We also wish we could have made use of some of the other provided data, such as the online profile descriptions for each pet or their pictures. We could have incorporated the profile descriptions as another feature dimension, using Bag of Words and a TF-IDF analysis on the text, and incorporated the pictures for each pet as another feature dimension, using image analysis techniques to identify relevant features.

6. Updated Milestones

Below is a table displaying the progress of our project based upon the completion of subgoals. More detailed information about what was accomplished in each subgoal was described in Section 3; pictures demonstrating this work can be found in the Appendix.

Due Date	Task	Progress
02/02/19	Research	Done (100%)
02/02/19	Set up work environment	Done (100%)
02/09/19	Data integration	Done (100%)
02/09/19	Data preprocessing	Done (100%)
02/15/19	Progress report	Done (100%)
02/16/19	Feature engineering and feature selection	Done (100%)
03/02/19	Implement model	Done (100%)
03/02/19	Evaluate performance of model	Done (100%)
03/09/19	Revise model	Done (90%)
03/11/19	Finalize project	Done (100%)
03/14/19	Poster presentation	Done (100%)

03/24/19	Submit final project report	Done (100%)

7. References

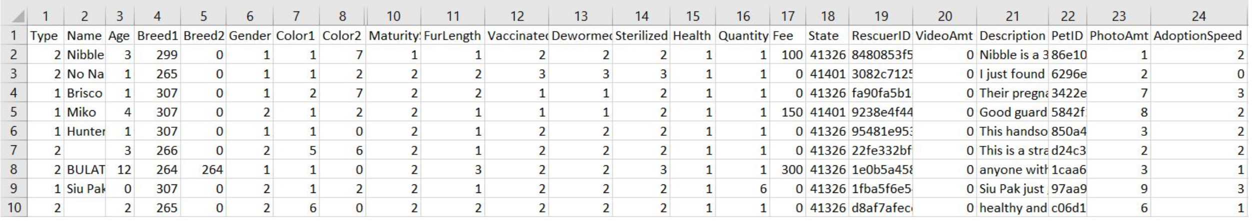
Below is a list of resources that have been used as references over the course of this project.

- “PetFinder.my Adoption Prediction.” *RSNA Pneumonia Detection Challenge | Kaggle*, PetFinder, www.kaggle.com/c/petfinder-adoption-prediction.
- “Documentation of Scikit-Learn 0.20.2.” *1.4. Support Vector Machines - Scikit-Learn 0.19.2 Documentation*, Scikit-Learn Developers, scikit-learn.org/stable/documentation.html.
- “Predicting Shelter Animal Outcomes: Team Kaggle for the Paws | Andras Zsom.” No Free Hunch, 5 Aug. 2016, blog.kaggle.com/2016/08/05/predicting-shelter-animal-outcomes-team-kaggle-for-the-paws-andras-zsom/.
- “Approaching (Almost) Any Machine Learning Problem | Abhishek Thakur.” No Free Hunch, 21 July 2016, blog.kaggle.com/2016/07/21/approaching-almost-any-machine-learning-problem-abhishek-thakur/.
- “Formulate Your Problem as an ML Problem | Introduction to Machine Learning Problem Framing | Google Developers.” Google, Google, developers.google.com/machine-learning/problem-framing/formulate.
- P.W.D. Kaggle, *Official Kaggle API*, GitHub repository, (2018) https://github.com/kaggle/kaggle-api.

8. Appendix: Tables and Images

Additional information about the project can be found below.

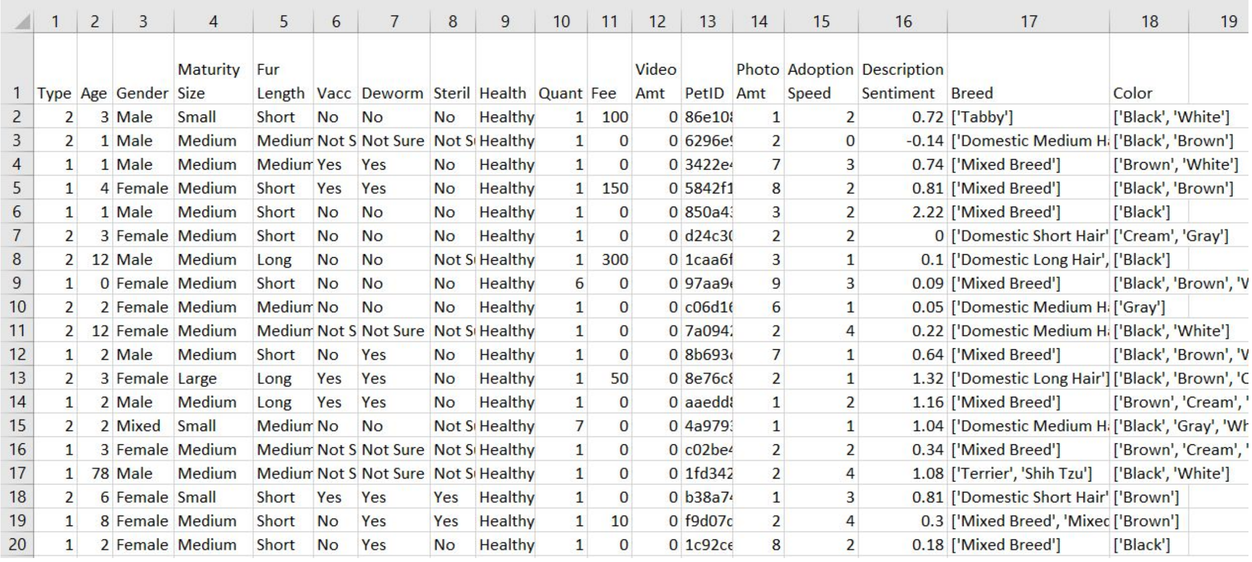
8.1 APPENDIX 1: Sample of the Original Data



	1	2	3	4	5	6	7	8	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
1	Type	Name	Age	Breed1	Breed2	Gender	Color1	Color2	Maturity!	FurLength	Vaccinatec	Dewormec	Sterilized	Health	Quantity	Fee	State	RescuerID	VideoAmt	Description	PetID	PhotoAmt	AdoptionSpeed
2	2	Nibble	3	299	0	1	1	7	1	1	2	2	2	1	1	100	41326	8480853f5	0	Nibble is a 3	86e10	1	2
3	2	No Na	1	265	0	1	1	2	2	2	3	3	3	1	1	0	41401	3082c7125	0	I just found	6296e	2	0
4	1	Brisco	1	307	0	1	2	7	2	2	1	1	2	1	1	0	41326	fa90fa5b1	0	Their pregna	3422e	7	3
5	1	Miko	4	307	0	2	1	2	2	1	1	1	2	1	1	150	41401	9238e4f44	0	Good guard	5842f	8	2
6	1	Hunter	1	307	0	1	1	0	2	1	2	2	2	1	1	0	41326	95481e953	0	This handso	850a4	3	2
7	2		3	266	0	2	5	6	2	1	2	2	2	1	1	0	41326	22fe332bf	0	This is a stra	d24c3	2	2
8	2	BULAT	12	264	264	1	1	0	2	3	2	2	3	1	1	300	41326	1e0b5a458	0	anyone with	1caa6	3	1
9	1	Siu Pak	0	307	0	2	1	2	2	1	2	2	2	1	6	0	41326	1fba5f6e5	0	Siu Pak just	97aa9	9	3
10	2		2	265	0	2	6	0	2	2	2	2	2	1	1	0	41326	d8af7afec	0	healthy and	c06d1	6	1

This is a picture of the original data, displayed in Excel, with all of the original fields and uncleaned data.

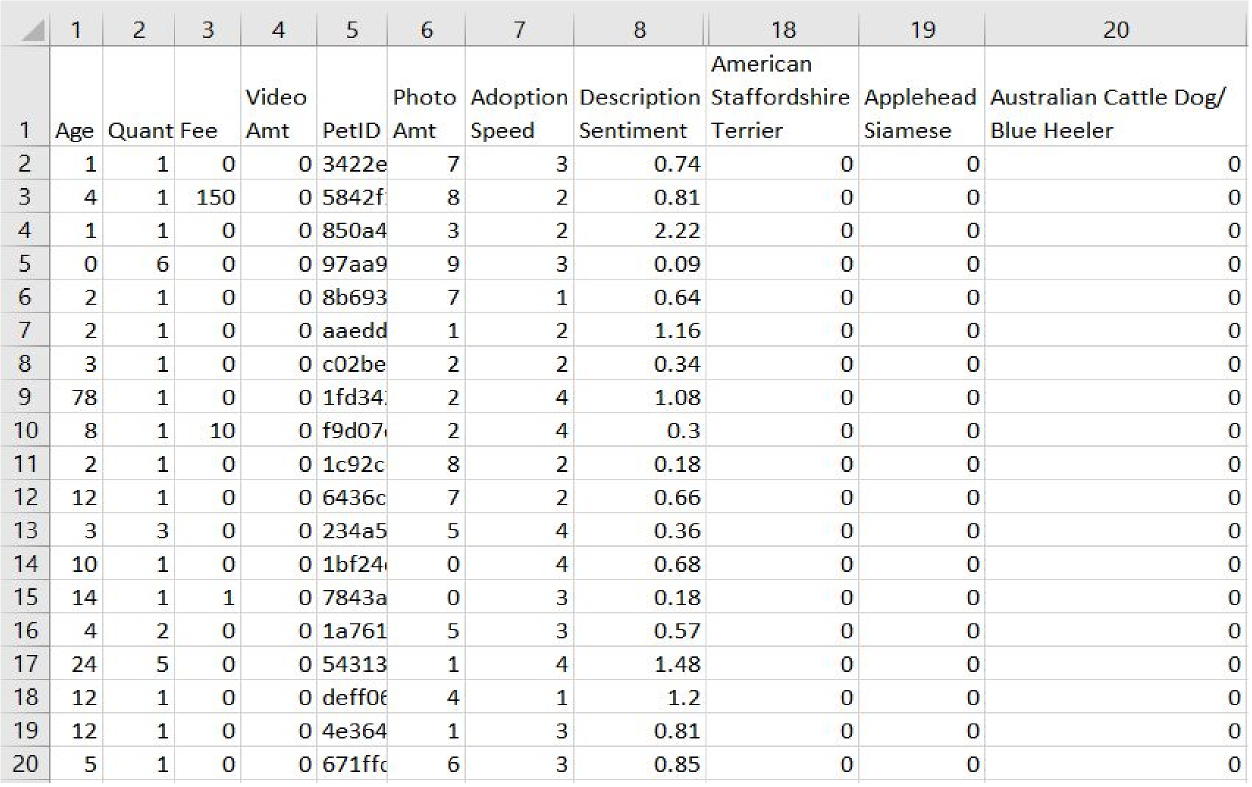
8.2 APPENDIX 2: Sample of the Modified Data



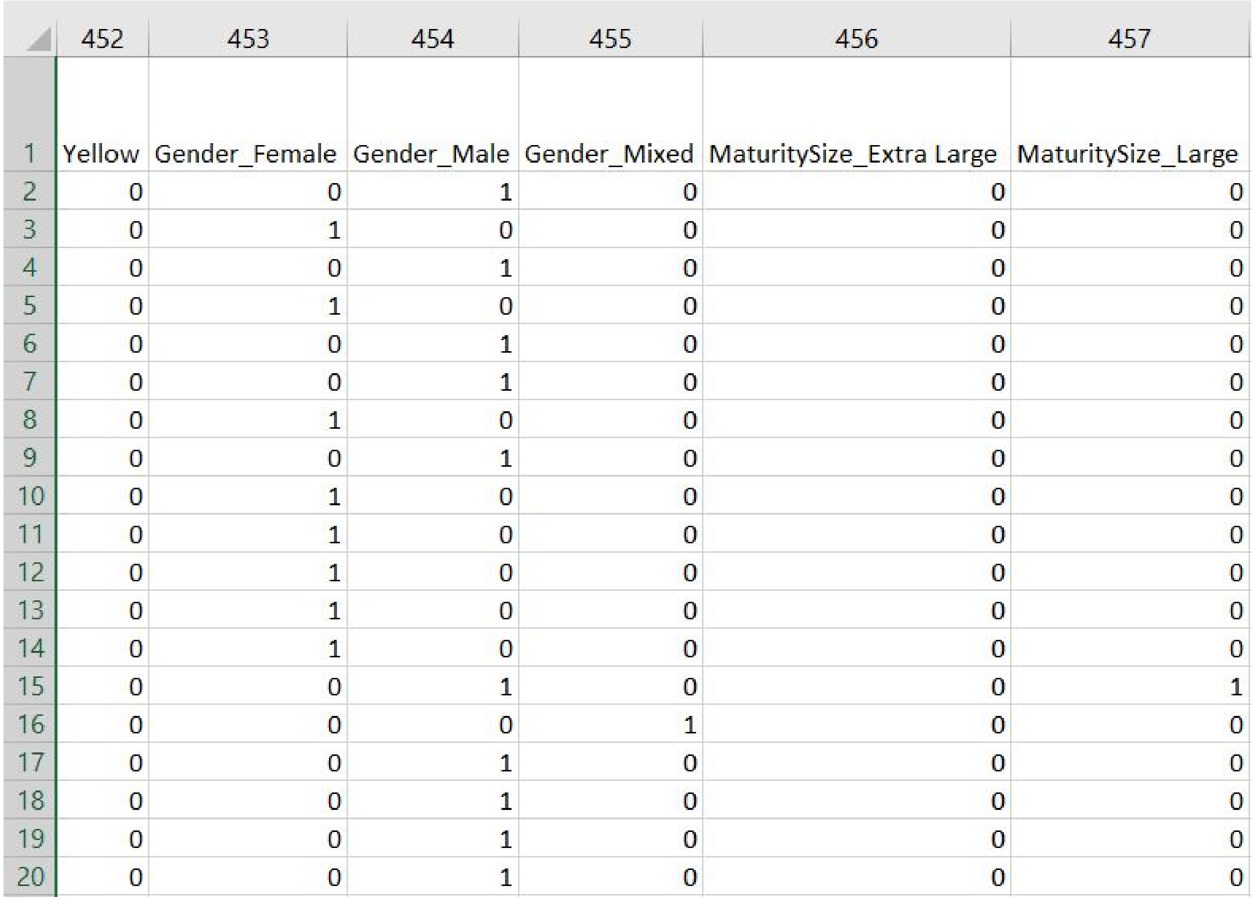
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
1	Type	Age	Gender	Maturity Size	Fur Length	Vacc	Deworm	Steril	Health	Quant	Fee	Video Amt	PetID	Photo Amt	Adoption Speed	Description Sentiment	Breed	Color	
2	2	3	Male	Small	Short	No	No	No	Healthy	1	100	0	86e10	1	2	0.72	['Tabby']	['Black', 'White']	
3	2	1	Male	Medium	Medium	Not S	Not Sure	Not S	Healthy	1	0	0	6296e	2	0	-0.14	['Domestic Medium H	['Black', 'Brown']	
4	1	1	Male	Medium	Medium	Yes	Yes	No	Healthy	1	0	0	3422e	7	3	0.74	['Mixed Breed']	['Brown', 'White']	
5	1	4	Female	Medium	Short	Yes	Yes	No	Healthy	1	150	0	5842f1	8	2	0.81	['Mixed Breed']	['Black', 'Brown']	
6	1	1	Male	Medium	Short	No	No	No	Healthy	1	0	0	850a4	3	2	2.22	['Mixed Breed']	['Black']	
7	2	3	Female	Medium	Short	No	No	No	Healthy	1	0	0	d24c3	2	2	0	['Domestic Short Hair'	['Cream', 'Gray']	
8	2	12	Male	Medium	Long	No	No	Not S	Healthy	1	300	0	1caa6	3	1	0.1	['Domestic Long Hair',	['Black']	
9	1	0	Female	Medium	Short	No	No	No	Healthy	6	0	0	97aa9	9	3	0.09	['Mixed Breed']	['Black', 'Brown', 'V	
10	2	2	Female	Medium	Medium	No	No	No	Healthy	1	0	0	c06d1	6	1	0.05	['Domestic Medium H	['Gray']	
11	2	12	Female	Medium	Medium	Not S	Not Sure	Not S	Healthy	1	0	0	7a094	2	4	0.22	['Domestic Medium H	['Black', 'White']	
12	1	2	Male	Medium	Short	No	Yes	No	Healthy	1	0	0	8b693	7	1	0.64	['Mixed Breed']	['Black', 'Brown', 'V	
13	2	3	Female	Large	Long	Yes	Yes	No	Healthy	1	50	0	8e76c	2	1	1.32	['Domestic Long Hair']	['Black', 'Brown', 'C	
14	1	2	Male	Medium	Long	Yes	Yes	No	Healthy	1	0	0	aaedd	1	2	1.16	['Mixed Breed']	['Brown', 'Cream', '	
15	2	2	Mixed	Small	Medium	No	No	Not S	Healthy	7	0	0	4a979	1	1	1.04	['Domestic Medium H	['Black', 'Gray', 'Wh	
16	1	3	Female	Medium	Medium	Not S	Not Sure	Not S	Healthy	1	0	0	c02be	2	2	0.34	['Mixed Breed']	['Brown', 'Cream', '	
17	1	78	Male	Medium	Medium	Not S	Not Sure	Not S	Healthy	1	0	0	1fd342	2	4	1.08	['Terrier', 'Shih Tzu']	['Black', 'White']	
18	2	6	Female	Small	Short	Yes	Yes	Yes	Healthy	1	0	0	b38a7	1	3	0.81	['Domestic Short Hair'	['Brown']	
19	1	8	Female	Medium	Short	No	Yes	Yes	Healthy	1	10	0	f9d07	2	4	0.3	['Mixed Breed', 'Mixec	['Brown']	
20	1	2	Female	Medium	Short	No	Yes	No	Healthy	1	0	0	1c92c	8	2	0.18	['Mixed Breed']	['Black']	

This is a picture of the data after some preprocessing and feature selection. The unwanted fields were dropped, the description sentiment score was added, some of the fields were combined, and the encoded data was replaced with their descriptive values.

8.3 APPENDIX 3: Sample of the Modified Data



	1	2	3	4	5	6	7	8	18	19	20
1	Age	Quant	Fee	Video Amt	PetID	Photo Amt	Adoption Speed	Description Sentiment	American Staffordshire Terrier	Applehead Siamese	Australian Cattle Dog/ Blue Heeler
2	1	1	0	0	3422e	7	3	0.74	0	0	0
3	4	1	150	0	5842f	8	2	0.81	0	0	0
4	1	1	0	0	850a4	3	2	2.22	0	0	0
5	0	6	0	0	97aa9	9	3	0.09	0	0	0
6	2	1	0	0	8b693	7	1	0.64	0	0	0
7	2	1	0	0	aaedd	1	2	1.16	0	0	0
8	3	1	0	0	c02be	2	2	0.34	0	0	0
9	78	1	0	0	1fd34	2	4	1.08	0	0	0
10	8	1	10	0	f9d07	2	4	0.3	0	0	0
11	2	1	0	0	1c92c	8	2	0.18	0	0	0
12	12	1	0	0	6436c	7	2	0.66	0	0	0
13	3	3	0	0	234a5	5	4	0.36	0	0	0
14	10	1	0	0	1bf24	0	4	0.68	0	0	0
15	14	1	1	0	7843a	0	3	0.18	0	0	0
16	4	2	0	0	1a761	5	3	0.57	0	0	0
17	24	5	0	0	54313	1	4	1.48	0	0	0
18	12	1	0	0	deff0	4	1	1.2	0	0	0
19	12	1	0	0	4e364	1	3	0.81	0	0	0
20	5	1	0	0	671ff	6	3	0.85	0	0	0

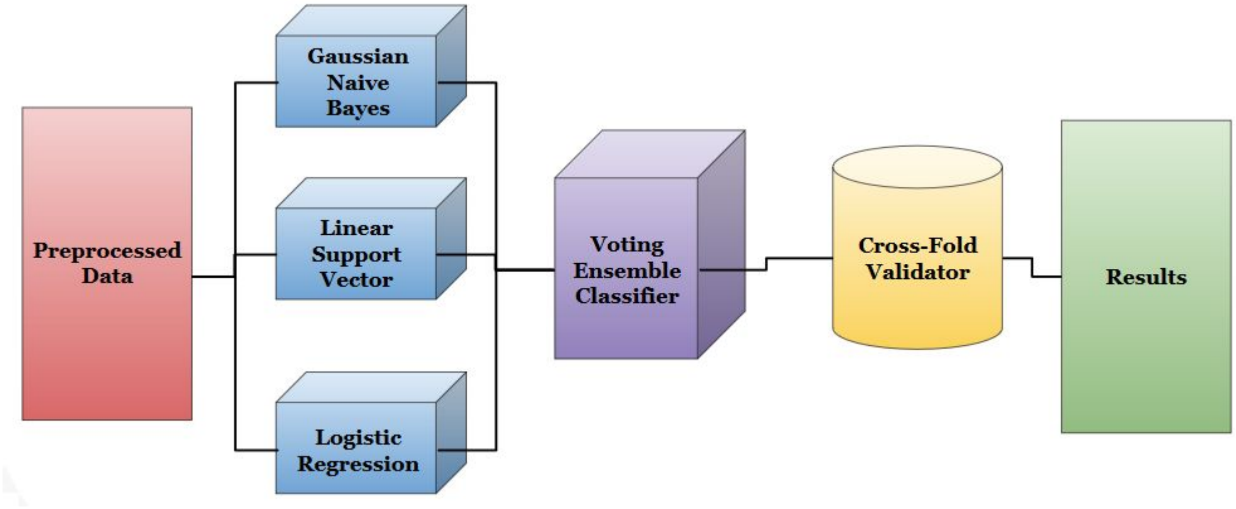


	452	453	454	455	456	457
1	Yellow	Gender_Female	Gender_Male	Gender_Mixed	MaturitySize_Extra Large	MaturitySize_Large
2	0	0	1	0	0	0
3	0	1	0	0	0	0
4	0	0	1	0	0	0
5	0	1	0	0	0	0
6	0	0	1	0	0	0
7	0	0	1	0	0	0
8	0	1	0	0	0	0
9	0	0	1	0	0	0
10	0	1	0	0	0	0
11	0	1	0	0	0	0
12	0	1	0	0	0	0
13	0	1	0	0	0	0
14	0	1	0	0	0	0
15	0	0	1	0	0	1
16	0	0	0	1	0	0
17	0	0	1	0	0	0
18	0	0	1	0	0	0
19	0	0	1	0	0	0
20	0	0	1	0	0	0

These are pictures of the data after some more preprocessing and feature selection. The data was split by pet type (dog and cat), which is why the Type column no longer appears--these particular photos are from the dog-specific data set. The categorical features were also one hot encoded. We can see that the Breeds column has been removed and instead has been replaced by columns for every type of dog breed. We can also see that the Gender column has been replaced by the binary features Gender_Female, Gender_Male, and Gender_Mixed.

8.4 APPENDIX 4: Model Diagram

This is a diagram of our implemented model.



8.5 APPENDIX 5: Example Output

Below is an example of the output that is produced by our program. Note that this file can also be found on the Github, in the "data" folder.

perf_dog_20190324191918.txt

```
+++ CROSS VALIDATION (CV = 10) +++

CLASSIFIER: naive_bayes.GaussianNB
    Average f1_micro: 0.0461218416208
    Average recall_micro: 0.0461218416208
    Average precision_micro: 0.0461218416208
    Average accuracy: 0.0461218416208

CLASSIFIER: svm.classes.LinearSVC
    Average f1_micro: 0.388198386224
    Average recall_micro: 0.387703178811
    Average precision_micro: 0.388570128516
    Average accuracy: 0.388198845935

CLASSIFIER: linear_model.logistic.LogisticRegression
    Average f1_micro: 0.383749808275
    Average recall_micro: 0.383749808275
    Average precision_micro: 0.383749808275
    Average accuracy: 0.383749808275

CLASSIFIER: ensemble.voting_classifier.VotingClassifier
    Average f1_micro: 0.384986210443
    Average recall_micro: 0.383746745961
    Average precision_micro: 0.384738070351
    Average accuracy: 0.385354898752
```