Animal Shelter Outcomes

By: Eric Liao

Glossary

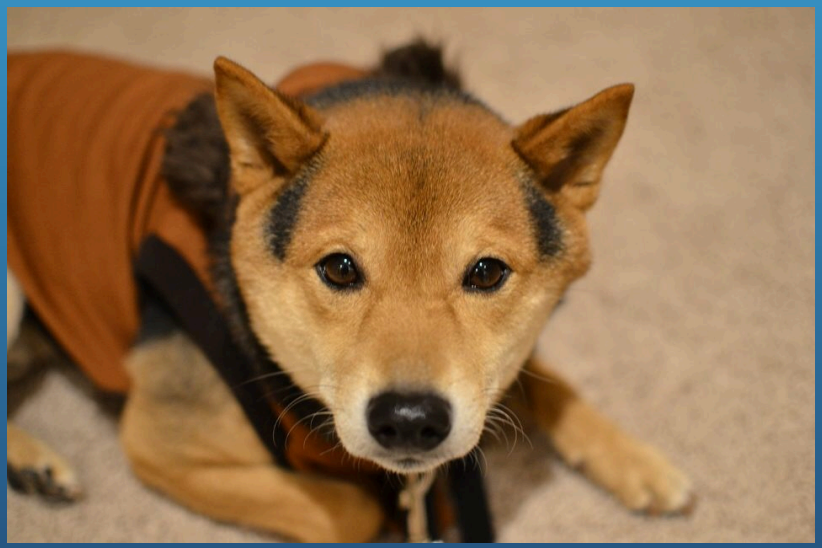
- Acquiring the Data
- Hypothesis
- Exploratory Analysis
- Dummy Variables
- Creating an Algorithm for Age
- More Analysis
- Cleaning Up my Data
- Machine Learning
- Machine Learning II
- Machine Learning III
- Machine Learning IV
- Next Steps
- Special Thanks
- Questions

Acquiring the Data

I will be looking at data from a Kaggle competition on Animal Shelter Outcomes. The "Outcomes" column has been removed from the test set, and I will create a model (from my train set) to see how accurately I can predict the missing outcomes. In order to complete this task, I will have to be picky on what categories I build my model out of, and see what yields the best results.

Hypothesis

I believe that "age" will play the most important role in correctly predicting the outcome of an animal shelter. Since most adopters prefer a younger puppy instead of an elder dog, age should be important in predicting adoptions and deaths.



Exploratory Analysis

```
train_df
```

	Name	DateTime	OutcomeType	OutcomeSubtype	AnimalType	SexuponOutcome	AgeuponOutcome	Breed	Color
AnimalID									
A671945	Hambone	2014-02-12 18:22:00	Return_to_owner	NaN	Dog	Neutered Male	1 year	Shetland Sheepdog Mix	Brown/White
A656520	Emily	2013-10-13 12:44:00	Euthanasia	Suffering	Cat	Spayed Female	1 year	Domestic Shorthair Mix	Cream Tabby
A686464	Pearce	2015-01-31 12:28:00	Adoption	Foster	Dog	Neutered Male	2 years	Pit Bull Mix	Blue/White
A683430	NaN	2014-07-11 19:09:00	Transfer	Partner	Cat	Intact Male	3 weeks	Domestic Shorthair Mix	Blue Cream
A667013	NaN	2013-11-15 12:52:00	Transfer	Partner	Dog	Neutered Male	2 years	Lhasa Apso/Miniature Poodle	Tan

- Mostly categorical, so I'll have to create dummy variables.
- Age is not in the same unit.

Creating Dummy Variables

```
BreedDummies = pd.get_dummies(train_df.Breed)
```

```
BreedDummies
```

	Abyssinian Mix	Affenpinscher Mix	Afghan Hound Mix	Airedale Terrier	Airedale Terrier Mix	...	Yorkshire Terrier/Norfolk Terrier	Yorkshire Terrier/Parson Russell Terrier	Yorkshire Terrier/Pomeranian	Yorkshire Terrier/Rat Terrier	Yorkshire Terrier/Toy Poodle
AnimalID											
A671945	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0
A656520	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0
A686464	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0
A683430	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0
A667013	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0
...	...	...	...	...	...	...	...	...	...	...	...
A702446	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0
A718934	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0
A698128	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0
A677478	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0
A706629	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0

26729 rows × 1380 columns

Creating an Algorithm for Age

```
def transform_to_numeric_age(age):
    num = int(age.split()[0])
    if 'week' in age:
        return  num / 4
    elif 'year' in age:
        return num * 12
    elif 'month' in age:
        return num
```

```
age = '12 months'
transform_to_numeric_age(age)
```

```
12
```

```
train_df.AgeuponOutcome = train_df.AgeuponOutcome.apply(transform_to_numeric_age)
```

Out of all my categories, I see that Age is the only one that has continuous variables. The problem is that not all of the dog's ages are in the same unit (year, month, day), so I need to create an algorithm to convert all ages to the same unit (month).

More Analysis

```
train_df.describe()
```

	AgeuponOutcome
count	26313.000000
mean	26.492114
std	35.732944
min	0.000000
25%	3.000000
50%	12.000000
75%	36.000000
max	240.000000

Cleaning Up the Data

```
train_df = train_df.join([AnimalDummies, SexDummies])
```

```
train_df.drop(['AnimalType', 'Breed', 'Color', 'SexuponOutcome', 'OutcomeSubtype'], axis=1, inplace=True)
```

```
len(BreedDummies.columns)
```

```
1380
```

```
for i in train_df.columns:
    print i
```

```
Name
DateTime
OutcomeType
AgeuponOutcome
Cat
Dog
Intact Female
Intact Male
Neutered Male
Spayed Female
Unknown
```

```
train_df.drop(['DateTime', 'Cat', 'Name', 'Unknown'], axis=1, inplace=True)
train_df.dropna(inplace=True)
```

Machine Learning

```
cross_validation.cross_val_score(model, train_X, train_y, cv = 10).mean()
```

```
0.50503864850506053
```

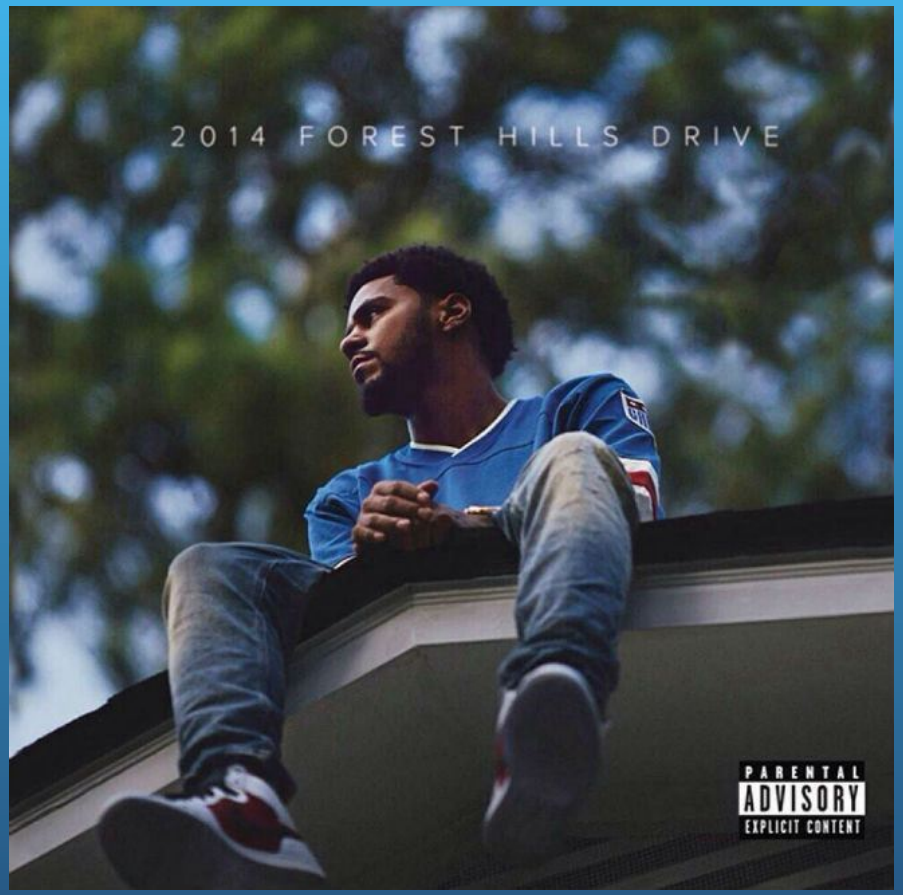
```
model.oob_score_
```

```
0.49800468740102616
```

- Since most of my data is categorical, there's no real way for me to do a linear regression model. Instead, I will be using random forest for my modeling.

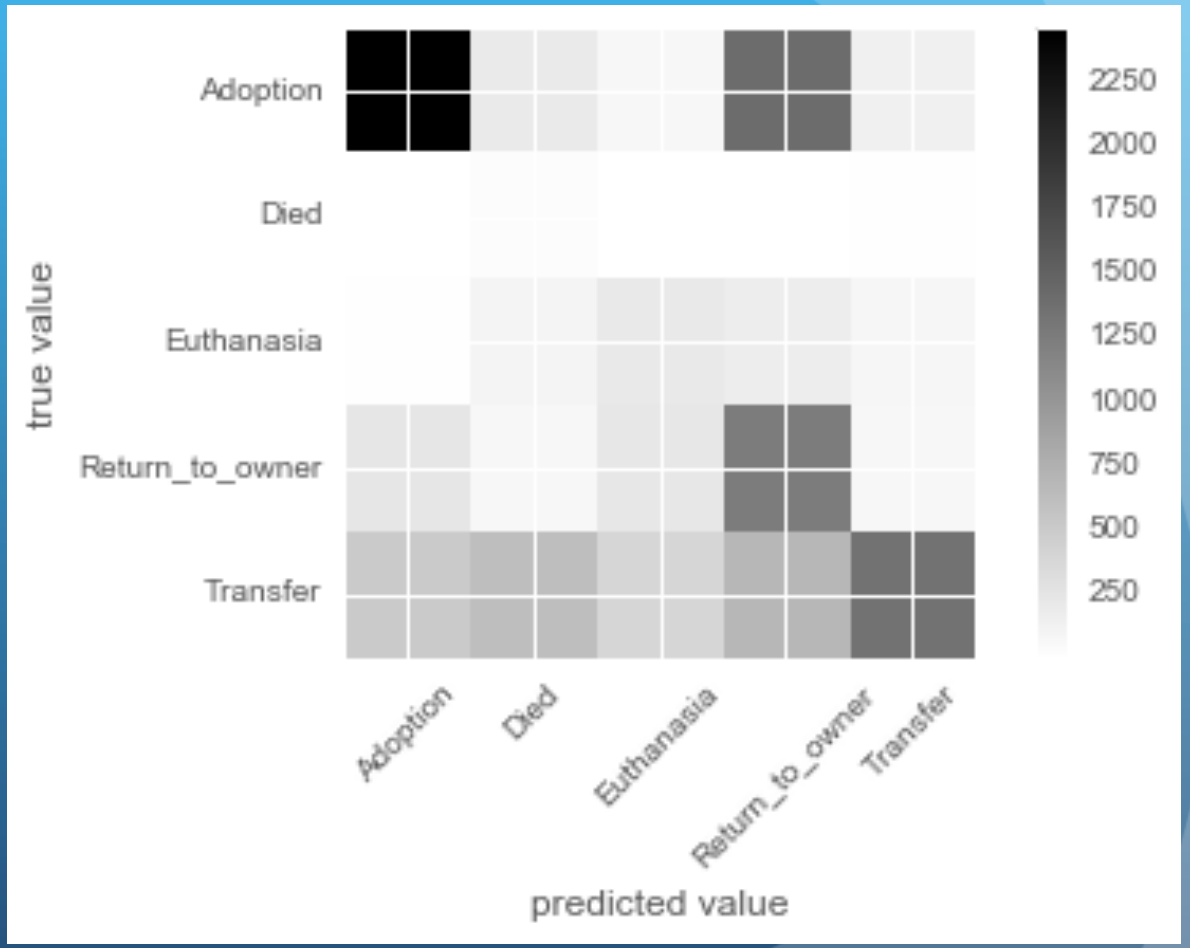
- My original random forest model actually had a higher cross validation score than my final. Since I am now using 20 estimators, instead of the original 10, I'm okay with this slight discrepancy, as my results should be more in tune with the true population.

Random Forest



Machine Learning II

Happy with my random forest model, I wanted to see if there were certain outcomes I could predict better than the others.



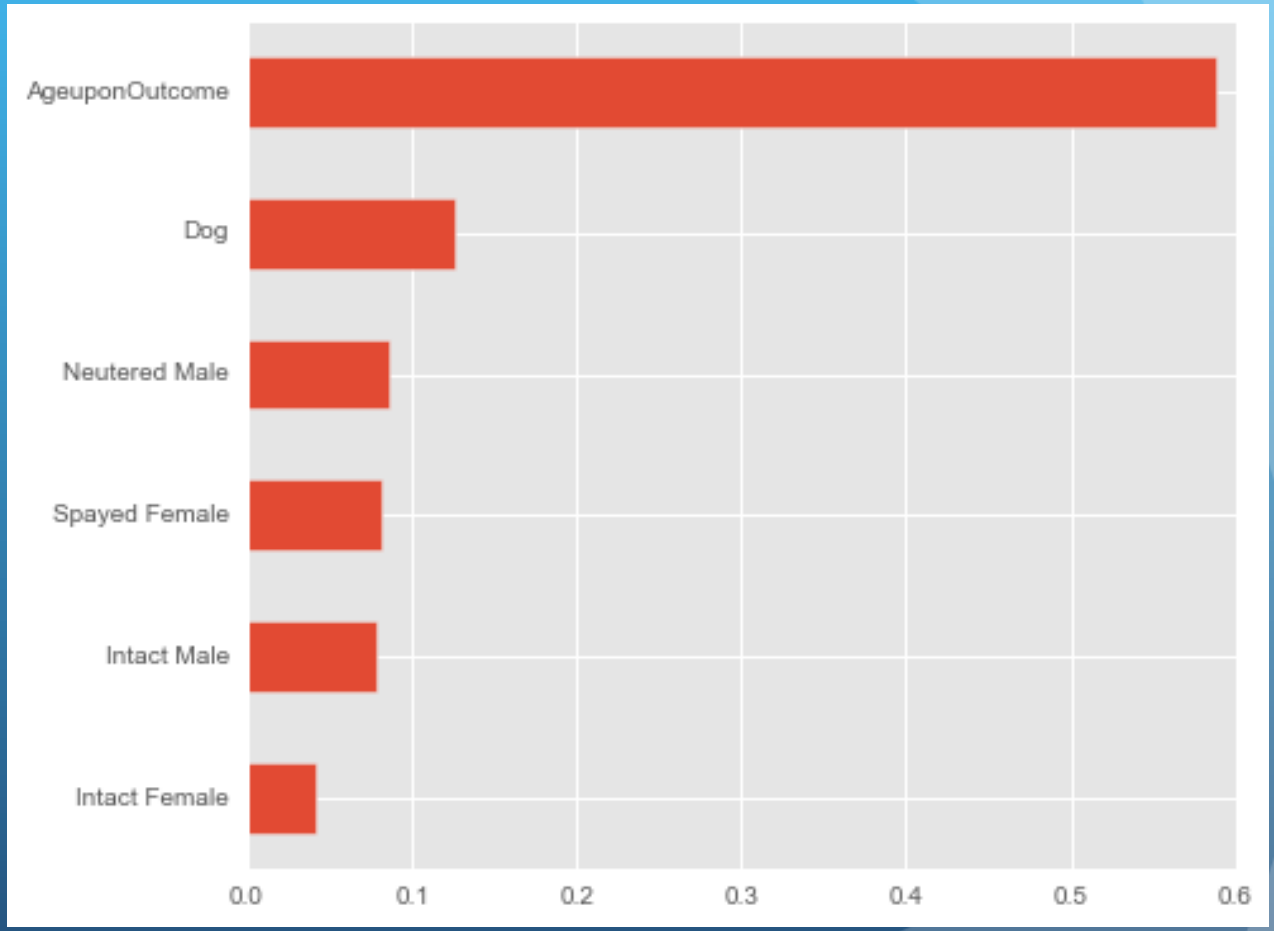
Machine Learning II (Cont'd)

```
print classification_report(test_y, y_pred)
```

	precision	recall	f1-score	support
Adoption	0.76	0.57	0.65	4320
Died	0.03	0.50	0.06	64
Euthanasia	0.23	0.35	0.28	628
Return_to_owner	0.35	0.66	0.46	1906
Transfer	0.80	0.37	0.51	3608
avg / total	0.66	0.51	0.54	10526

Machine Learning III

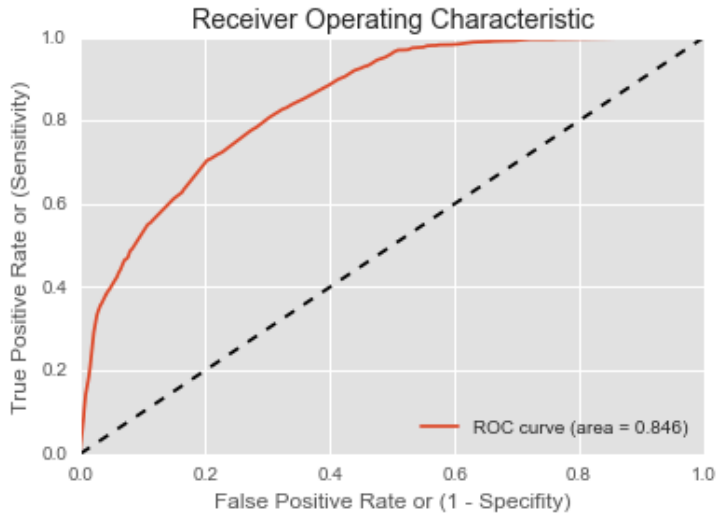
Now that I know what outcomes I can better predict, I wanted to see which features play the most importance on predicting a correct outcome.



Machine Learning IV

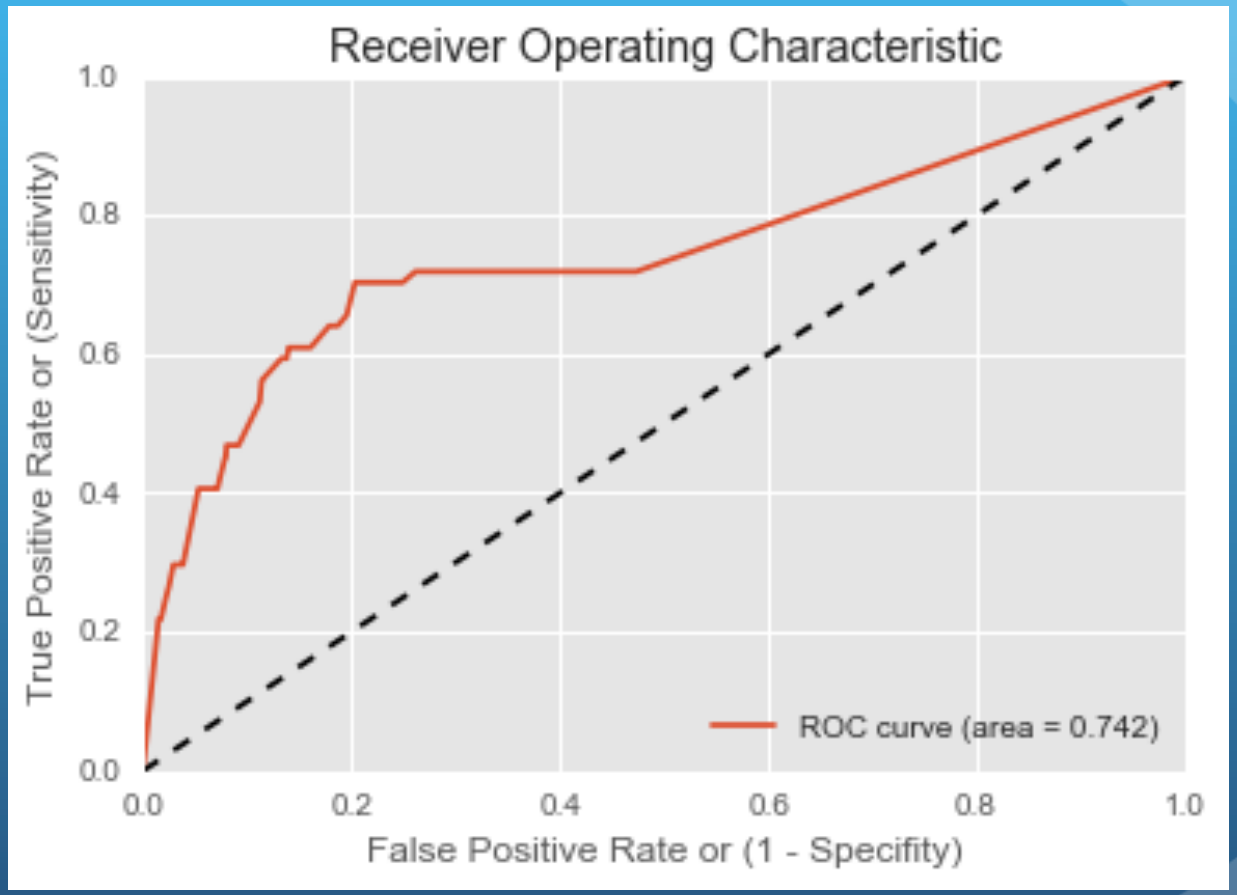
```
test_y_num = test_y.map({'Adoption':0, 'Died':1, 'Euthanasia':2, 'Return_to_owner':3, 'Transfer':4})
```

```
plot_roc_curve(TestDummies.reset_index(drop=True).ix[:,0], y_pred_proba[:,0])
```

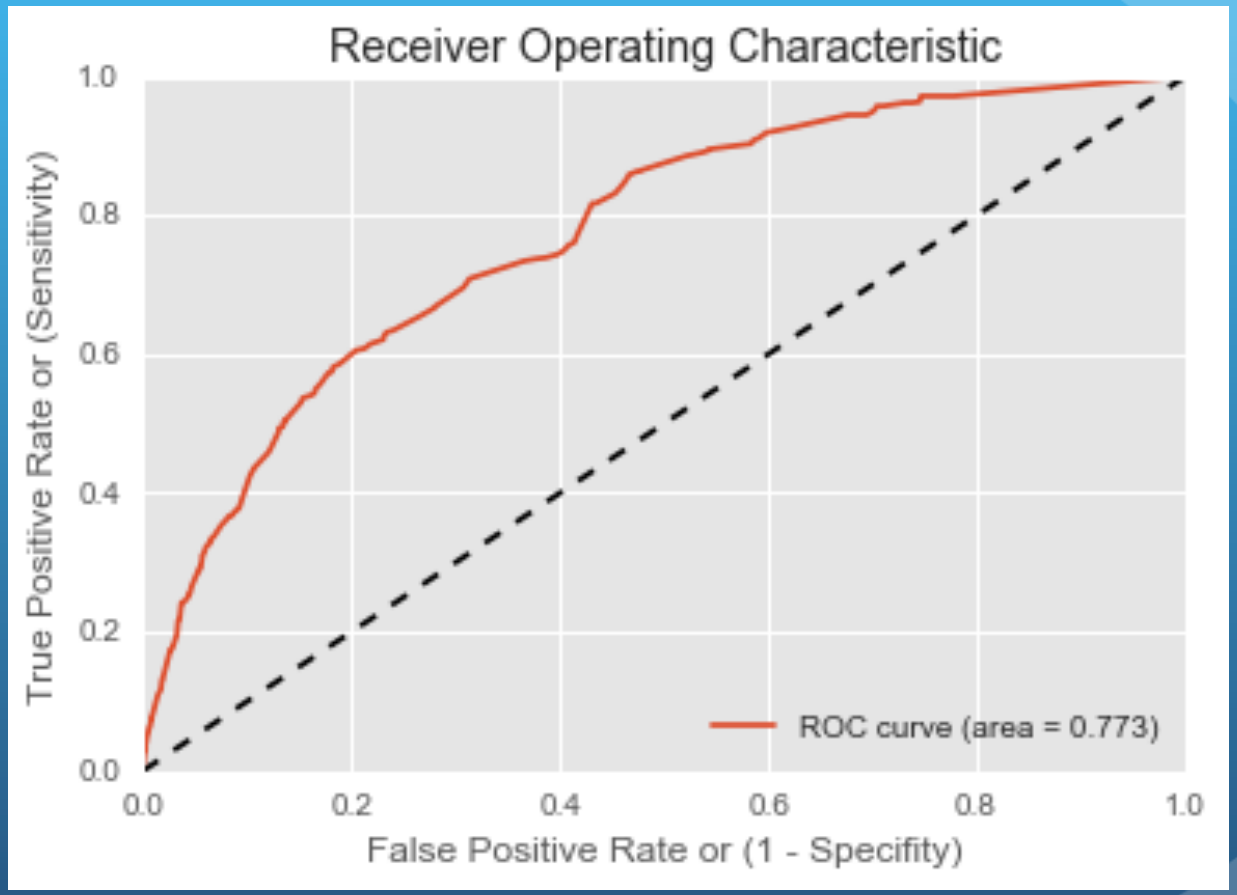


I will be using a ROC curve to determine my true positive rate in predicting the different outcomes.

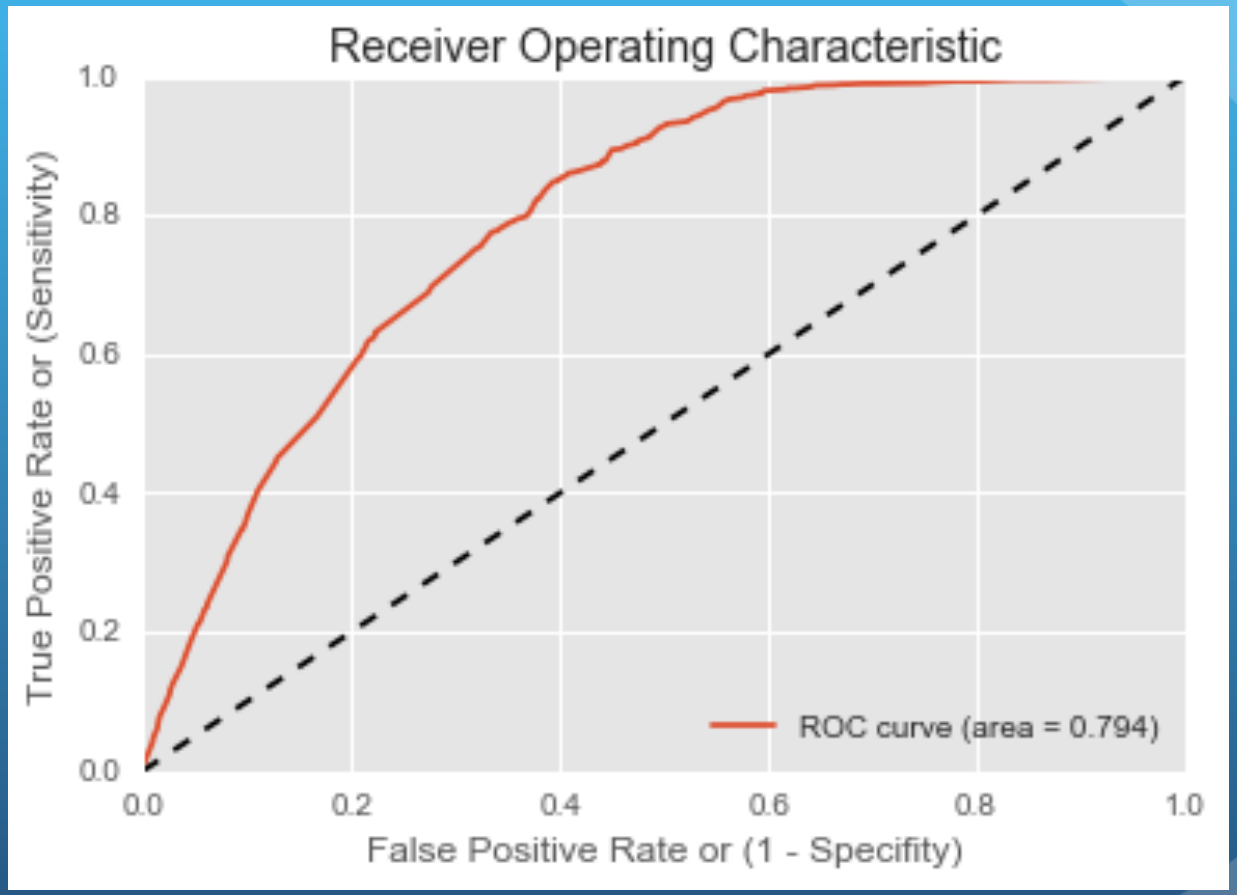
Machine Learning IV (Died)



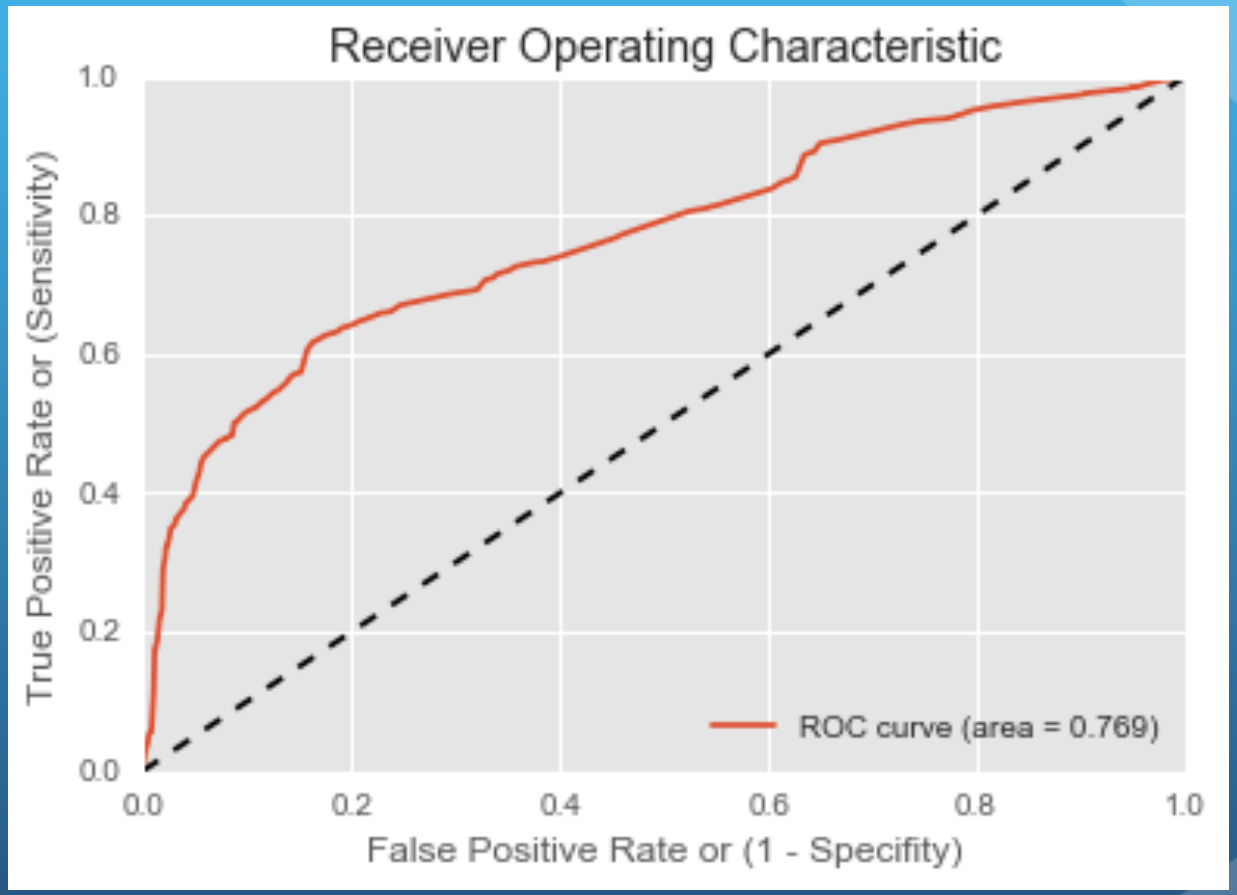
Machine Learning (Euthanasia)



Machine Learning IV (Return to Owner)



Machine Learning IV (Transfer)



Next Steps

- Acquire more data.
- Re-run my models to include cats.
- Look into the importance of other features for correctly predicting outcome.

Special Thanks!

- Ivan Corneillet, Jeremiah Gaw, & Azi Hussain
- SF-DAT-21

Questions?