Kaggle Pawpularity Contest

Predicting shelter pet popularity from profile images

Alex Teboul, Shea Dettling, Hima Spandana

Overview

1. Exploratory Data Analysis
2. Model Building
    - Metadata Models
    - Convolutional Neural Networks
    - Pre-trained Models (Xception)
3. Conclusion

PetFinder.my

Exploratory Data Analysis

Tutorial Series

Tutorial Part 1: EDA for Beginners
Updated 20d ago
27 comments · PetFinder.my - Pawpularity Contest

123
Gold •••

Tutorial Part 2: Model Building using the Metadata
Updated 11d ago
Score: 20.52197 · 19 comments · PetFinder.my - Pawpularity Contest

47
Silver •••

Tutorial Part 3: CNN Image Modeling 1
Updated 19d ago
Score: 20.6781 · 5 comments · PetFinder.my - Pawpularity Contest

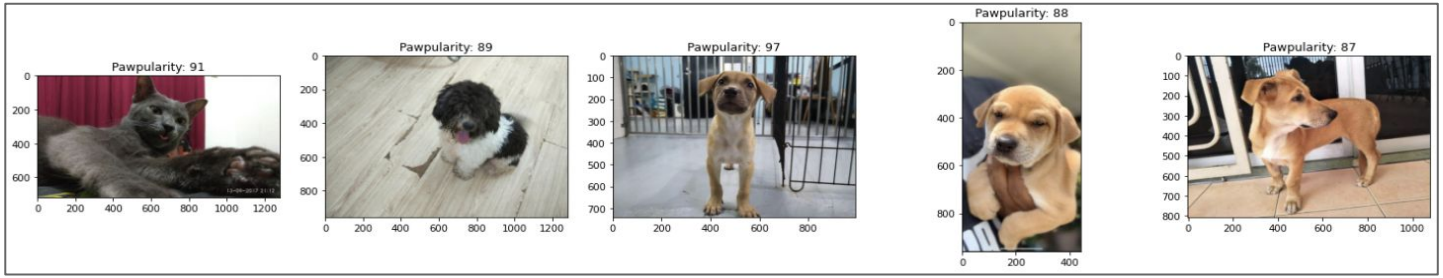
20
Bronze •••

Dataset

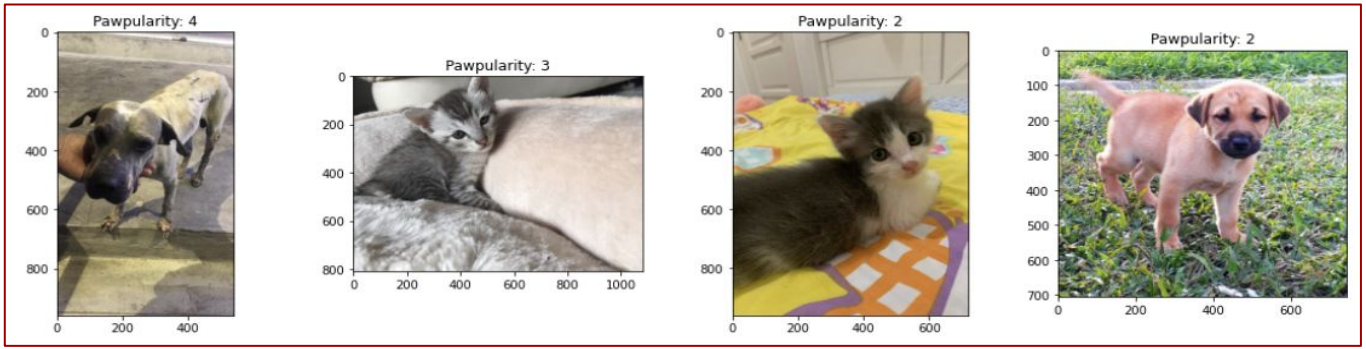
1. **Metadata**: train.csv | 9912 rows x 14 columns | 12 features

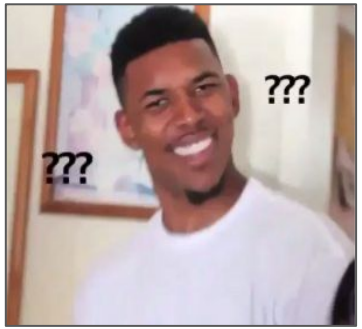
	Id	Subject Focus	Eyes	Face	Near	Action	Accessory	Group	Collage	Human	Occlusion	Info	Blur	Pawpularity
0	0007de18844b0dbbb5e1f607da0606e0	0	1	1	1	0	0	1	0	0	0	0	0	63
1	0009c66b9439883ba2750fb825e1d7db	0	1	1	0	0	0	0	0	0	0	0	0	42
2	0013fd999caf9a3efe1352ca1b0d937e	0	1	1	1	0	0	0	0	1	1	0	0	28
3	0018df346ac9c1d8413cfcc888ca8246	0	1	1	1	0	0	0	0	0	0	0	0	15
4	001dc955e10590d3ca4673f034feeef2	0	0	0	1	0	0	1	0	0	0	0	0	72

2. **Pet Images**: 9912 images

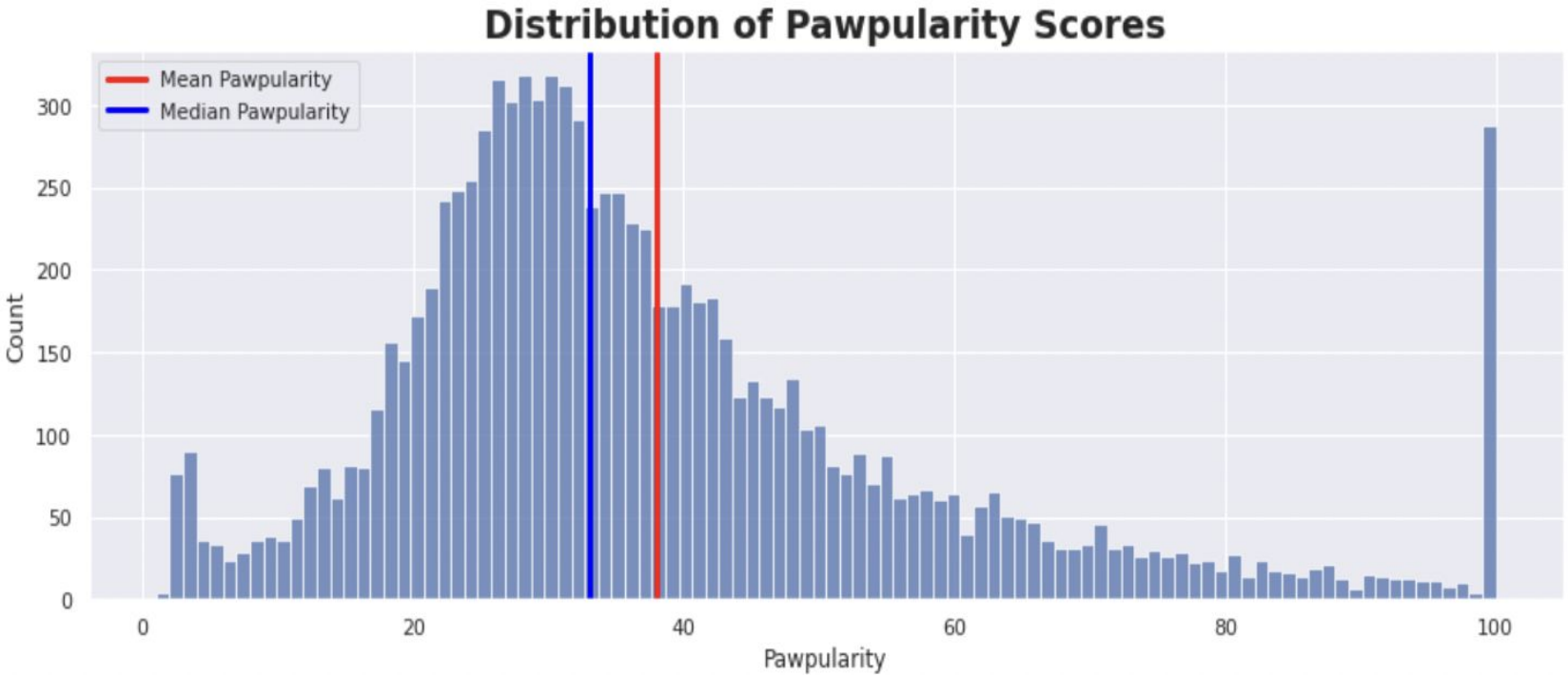


The Problem With Pawpularity



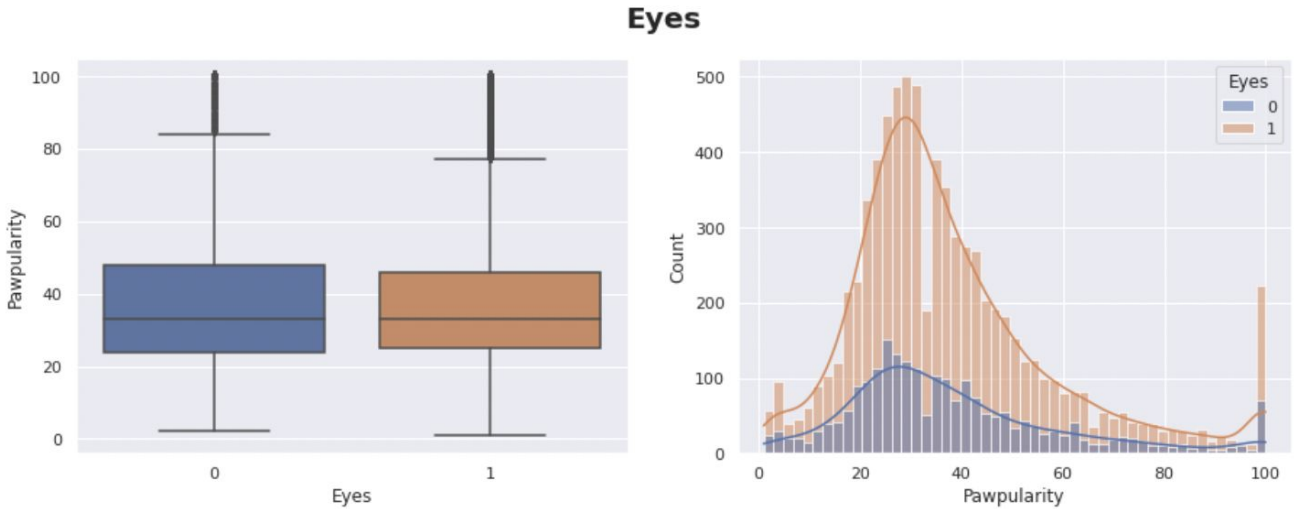


Pawpularity



Example of features

- The 12 feature variables are binary and are poorly correlated with Pawpularity

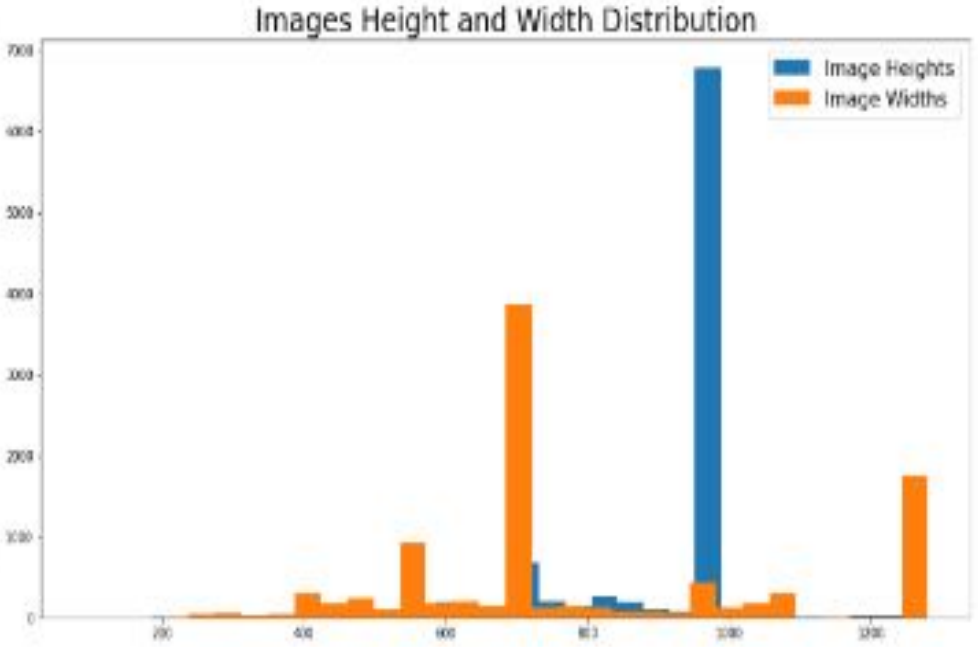


Duplicate Images

Image EDA

```
Width Statistics

count     9858.000000
mean       804.522520
std        270.248645
min         90.000000
25%        676.000000
50%        720.000000
75%        960.000000
max       1280.000000
dtype: float64
```

```
Height Statistics

count     9858.000000
mean       904.418340
std        156.872505
min        113.000000
25%        909.250000
50%        960.000000
75%        960.000000
max       1280.000000
dtype: float64
```



Model Building

- Metadata Models
- Convolutional Neural Networks
- Pre-trained Models (Xception)

- # Metadata Models

Metadata Models

Model	RMSE
4.1 Decision Tree Regression	20.857
4.2 Decision Tree Classification	22.900
5.1 Ordinary Least Square Regression	20.827
5.2 Ridge Regression	20.827
6.1 Bernoulli Naive Bayes Classification	23.468
7.1 Random Forest Regression	20.838
7.2 Histogram-based Gradient Boosting Regression	20.924

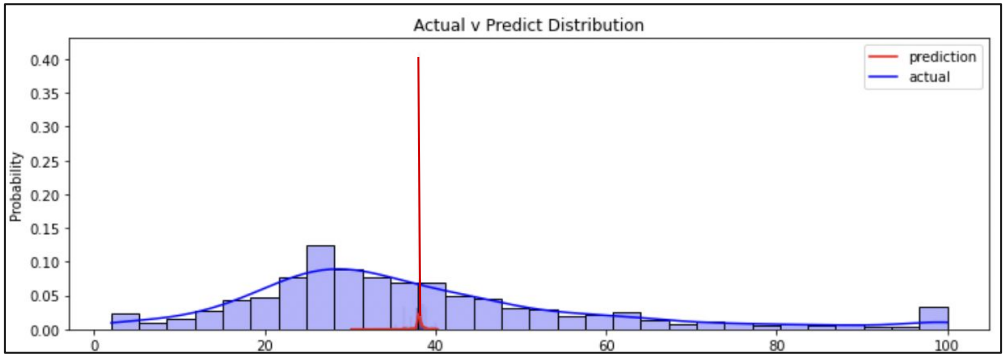
Ensemble Techniques

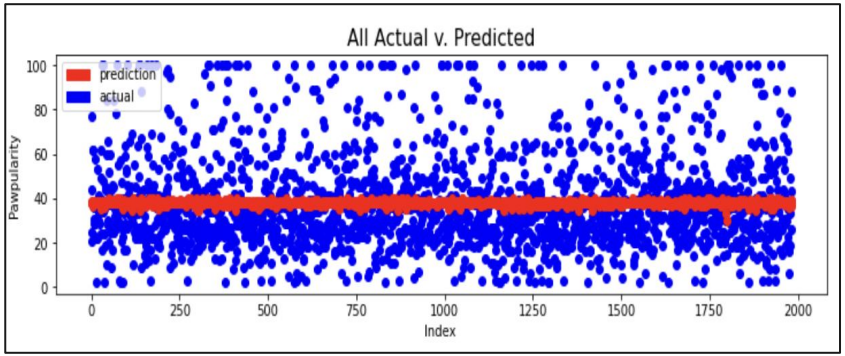
	rmse	reg_score	test_score	train_score
AdaBoost	22.09	0.07	-0.1	-0.04
Bagging	21.05	0.09	-0.01	0.05
Gradient Boost	21.07	0.03	-0.08	0.01
Random Forest	21.49	0.04	-0.01	0.06
XGB	21.57	0.05	-0.07	0.04

Metadata Models

1. Models built using only the tabular metadata learn to guess the mean pawpularity of 38 for nearly all pet images.

2. The manually created metadata features are not correlated with Pawpularity.





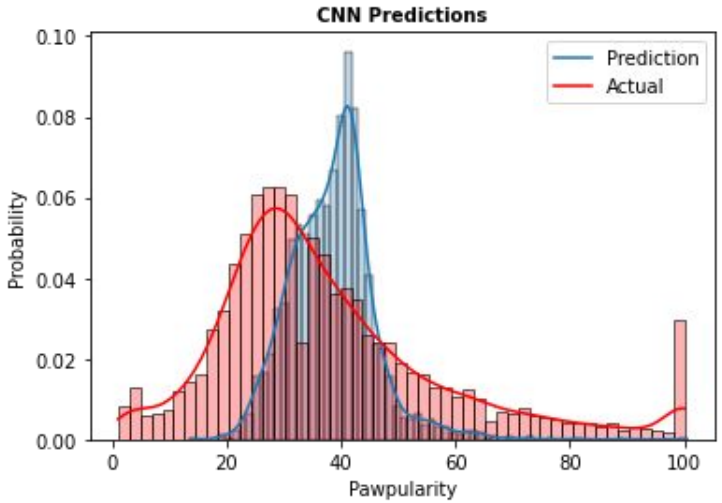
- Convolutional Neural Networks

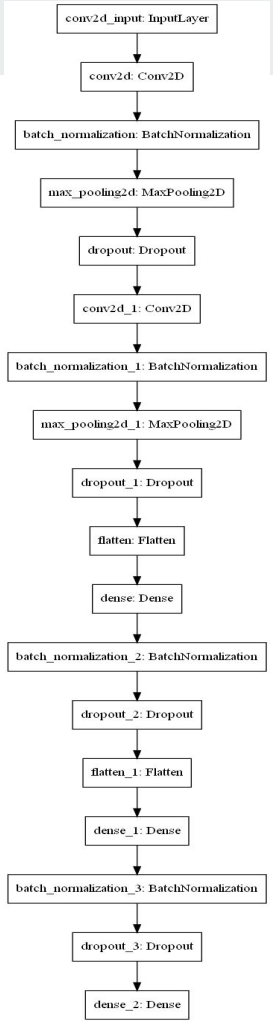
Convolutional Neural Networks (CNN)

1. Multiple image processing techniques, data augmentation techniques, and model architectures are tried:
    a. Ex. Compress into standard size 48x72, which was a downsized version of the most common aspect ratio to cause least amount of tear.
    b. Ex. Convert to grayscale and normalize value range to [0,1]
    c. Ex. Add a channel dimension (=1) to conform to conv2d requirements
    d. ex. Grid searched layer depth, activations, map depth, nodes quantity, optimizer, loss, dropout

Convolutional Neural Networks (CNN)

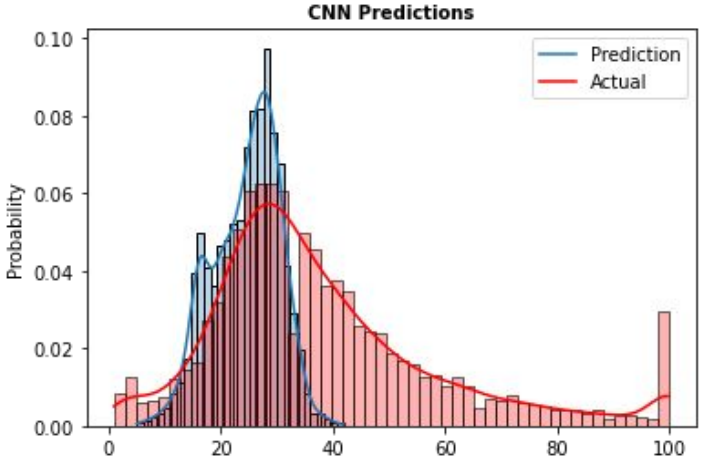
- Two conv layers with 32 then 64 kernels, two dense layers with 100 nodes, linear output layer (w/ learning decay)

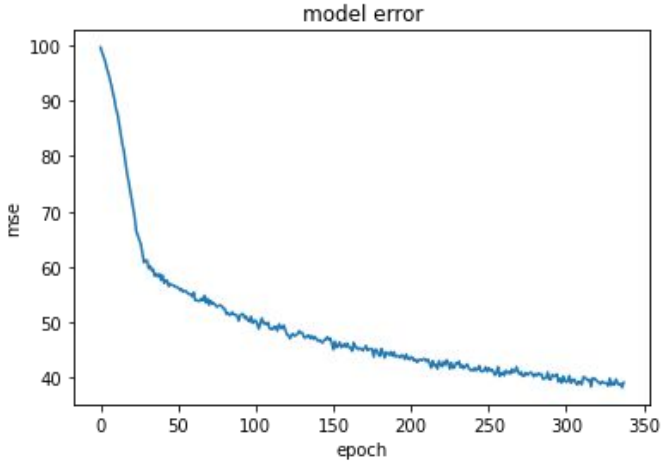




Hyperparameter Tuning - Grid Searching

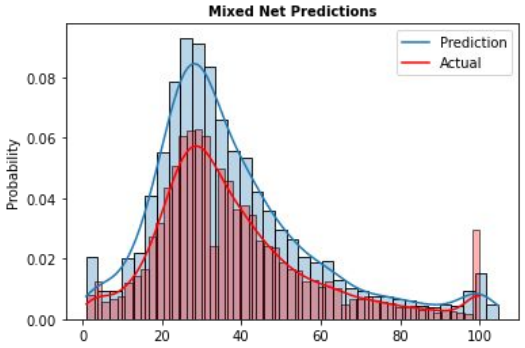
- While the prediction distribution still is clearly off, loss has improved to ~35 from 64

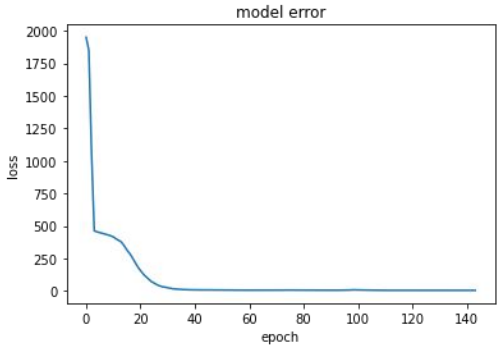


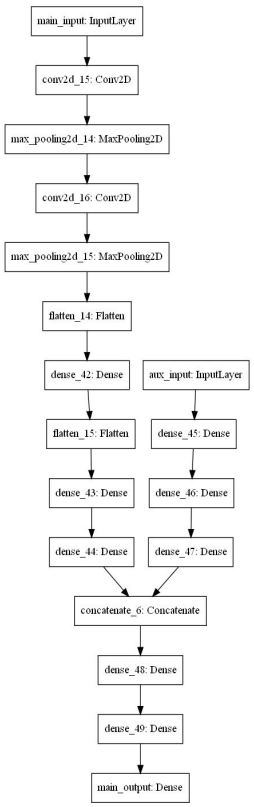


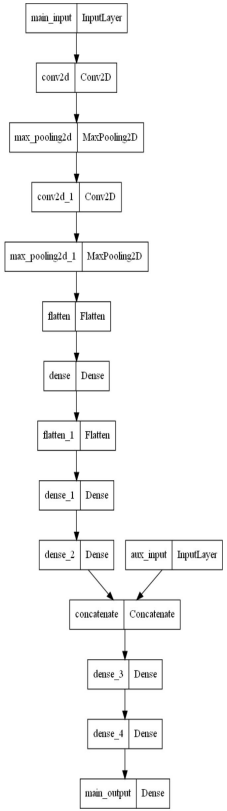
Mixed Net (Image CNN merged w/ Metadata MLP)

- Two architectures: one w/ an ANN for metadata where prediction is combined with prediction from CNN to form a 2 input for a final ANN
- Second was including the CNN prediction as a feature in the metadata to feed forward as a 13 dim input for an ANN

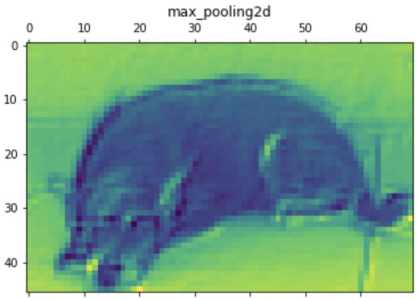


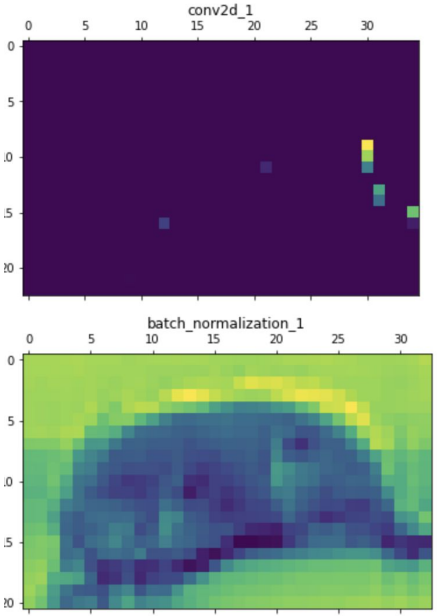


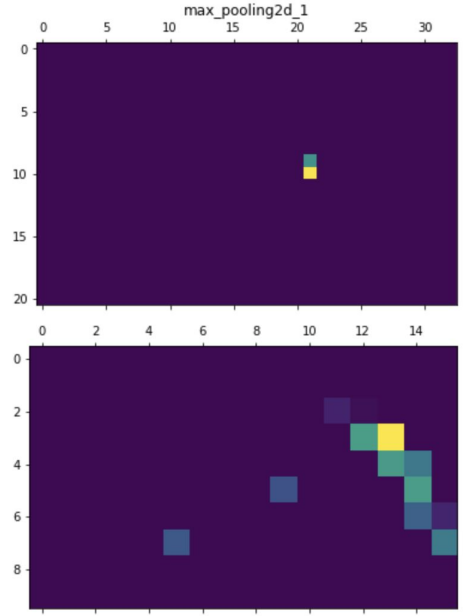




Visualizing Feature Maps



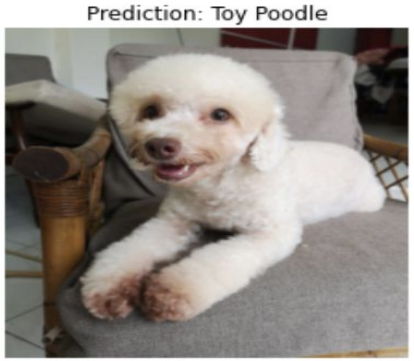




- Pre-trained Models (Xception)

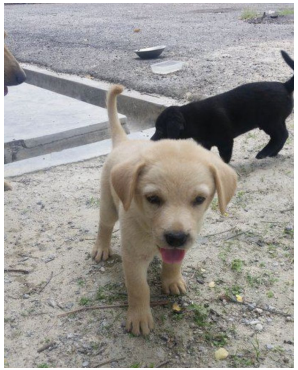
Pre-trained Models (Xception)

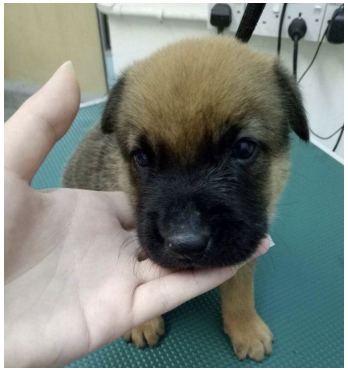
```
xception = tf.keras.applications.xception.Xception(
    include_top=True, weights='imagenet', input_tensor=None,
    input_shape=None, pooling=None, classes=1000,
    classifier_activation='softmax'
)
classifier_model = xception

IMAGE_SHAPE = (299, 299)
%%capture
xception_clf = tf.keras.Sequential([
    hub.KerasLayer(classifier_model, input_shape=IMAGE_SHAPE+(3,))
])
result = xception_clf.predict(example_dog[np.newaxis, ...])
predicted_class = tf.math.argmax(result[0], axis=-1)
labels_path = tf.keras.utils.get_file('ImageNetLabels.txt',
    'https://storage.googleapis.com/download.tensorflow.org/data/ImageNetLabels.txt')
imagenet_labels = np.array(open(labels_path).read().splitlines())

plt.imshow(example_dog)
plt.axis('off')
predicted_class_name = imagenet_labels[predicted_class]
_ = plt.title("Prediction: " + predicted_class_name.title())
```



Pre-trained Models (Xception)

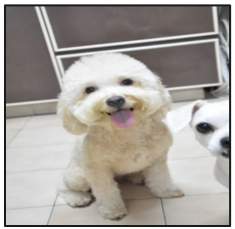
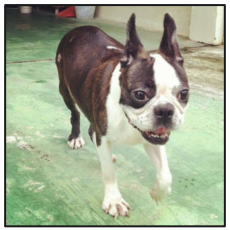
	Id	Subject Focus	Eyes	Face	Near	Action	Accessory	Group	Collage	Human	Occlusion	Info	Blur	Pawpularity	img_path	Breed
0	0007de18844b0dbbb5e1f607da0606e0	0	1	1	1	0	0	1	0	0	0	0	0	63	../input/petfinder-pawpularity-score/train/000...	Labrador retriever
1	0009c66b9439883ba2750fb825e1d7db	0	1	1	0	0	0	0	0	0	0	0	0	42	../input/petfinder-pawpularity-score/train/000...	tabby
2	0013fd999caf9a3efe1352ca1b0d937e	0	1	1	1	0	0	0	0	1	1	0	0	28	../input/petfinder-pawpularity-score/train/001...	Border terrier
3	0018df346ac9c1d8413cfcc888ca8246	0	1	1	1	0	0	0	0	0	0	0	0	15	../input/petfinder-pawpularity-score/train/001...	Staffordshire bullterrier
4	001dc955e10590d3ca4673f034feeef2	0	0	0	1	0	0	1	0	0	0	0	0	72	../input/petfinder-pawpularity-score/train/001...	French bulldog
5	001dd4f6fafb890610b1635f967ea081	0	0	1	0	0	0	0	0	0	0	0	1	74	../input/petfinder-pawpularity-score/train/001...	Japanese spaniel
6	0023b8a3abc93c712edd6120867deb53	0	1	1	1	0	0	0	0	1	1	0	0	22	../input/petfinder-pawpularity-score/train/002...	Labrador retriever
7	0031d6a9ef7340f898c3e05f92c7bb04	0	1	1	0	0	0	1	1	0	0	1	0	35	../input/petfinder-pawpularity-score/train/003...	Rottweiler
8	0042bc5bada6d1cf8951f8f9f0d399fa	0	1	1	1	0	0	0	0	0	0	0	0	53	../input/petfinder-pawpularity-score/train/004...	tabby
9	0049cb81313c94fa007286e9039af910	0	1	1	1	0	0	0	0	0	0	0	0	21	../input/petfinder-pawpularity-score/train/004...	Egyptian cat





Xception Results

1. Different breeds do appear to have different mean Pawpularity scores. Makes sense given people search the website by pet species/breed. But base model still makes errors.

Predicted Breed	Maltese dog	Golden Retriever	Persian cat	Studio couch	Egyptian cat	Boston Bull
Average Pawpularity	86.8	52.9	52.4	39.3	30.9	27.0
Example						

BreedBoost! - Slight Improvements in RMSE

```
#RMSE comparison!
example_y_actual    = train['Pawpularity'].tolist()
example_y_predicted = train['Predicted_Pawpularity'].tolist()
example_MSE = mean_squared_error(example_y_actual, example_y_predicted)
example_RMSE = math.sqrt(example_MSE)
print("Breed Averages Guess on Training Data RMSE:", round(example_RMSE,2))
print("BreedBoosted Metadata RandomForest:", round(RF_reg_RMSE,2))
print("Metadata Models RMSE: 20.52")
```

```
Breed Averages Guess on Training Data RMSE: 18.74
BreedBoosted Metadata RandomForest: 19.91
Metadata Models RMSE: 20.52
```

Conclusion

1. Pawpularity not correlated with metadata - poor models.
2. Basic CNNs lack ability to learn features and improve RMSE.
3. Pretrained Models see some slight improvements with RMSE.
4. Hard to predict shelter pet profile click rates based solely on pet profile pictures.