Pawpularity Contest

Austin Jin, Chandni Shah, Matt Lyons
W207 Final Project, Fall 2021

Berkeley
UNIVERSITY OF CALIFORNIA

Agenda

1. Project Description
2. Data Description
3. Exploratory Data Analysis
4. Tabular Data Models
5. Pixel Data Models
6. Combined Data Model
7. Overall Summary (Challenges, Surprises, etc)
8. Missing Data / Noise
9. Q&A

Berkeley
UNIVERSITY OF CALIFORNIA

Project Description

PetFinder.my

- Malaysia's leading animal welfare platform
- Uses basic Cuteness meter to rank pet photos
- Analyzes picture composition and other factors compared to performance of thousands of pet profiles

Competition

- Analyze raw images and metadata to predict the "Pawpularity" of pet photos
- Train and test model on PetFinder.my's thousands of pet profiles
- Winning versions will offer accurate recommendations that will improve animal welfare

Data Description

Pawpularity Score

- Derived from each pet profile's page view statistics at the listing pages that uses an algorithm to normalize traffic data
- Duplicate clicks, crawler bot accesses, and sponsored profiles are excluded from the analysis

Photo Metadata

- Manually labeled each photo for key visual quality and composition parameters
- Not used for deriving Pawpularity score but beneficial for better understand the content

Training Data

- train/ folder contains training set photos of the form {id}.jpg, where {id} is a unique Pet Profile ID
- train.csv contains metadata for each photo in the training set and target, the photo's Pawpularity score.

Tabular Metadata

"Tabular Metadata: Each pet photo is labeled with the value of 1 (Yes) or 0 (No) for each of the following features. These labels are not used for deriving the Pawpularity score.

- Focus - Pet stands out against uncluttered background, not too close / far.
- Eyes - Both eyes are facing front or near-front, with at least 1 eye / pupil decently clear.
- Face - Decently clear face, facing front or near-front.
- Near - Single pet taking up significant portion of photo (roughly over 50% of photo width or height).
- Action - Pet in the middle of an action (e.g., jumping).
- Accessory - Accompanying physical or digital accessory / prop (i.e. toy, digital sticker), excluding collar and leash.
- Group - More than 1 pet in the photo.
- Collage - Digitally-retouched photo (i.e. with digital photo frame, combination of multiple photos).
- Human - Human in the photo.
- Occlusion - Specific undesirable objects blocking part of the pet (i.e. human, cage or fence). Note that not all blocking objects are considered occlusion.
- Info - Custom-added text or labels (i.e. pet name, description).
- Blur - Noticeably out of focus or noisy, especially for the pet's eyes and face. For Blur entries, "Eyes" column is always set to 0."

Tabular EDA

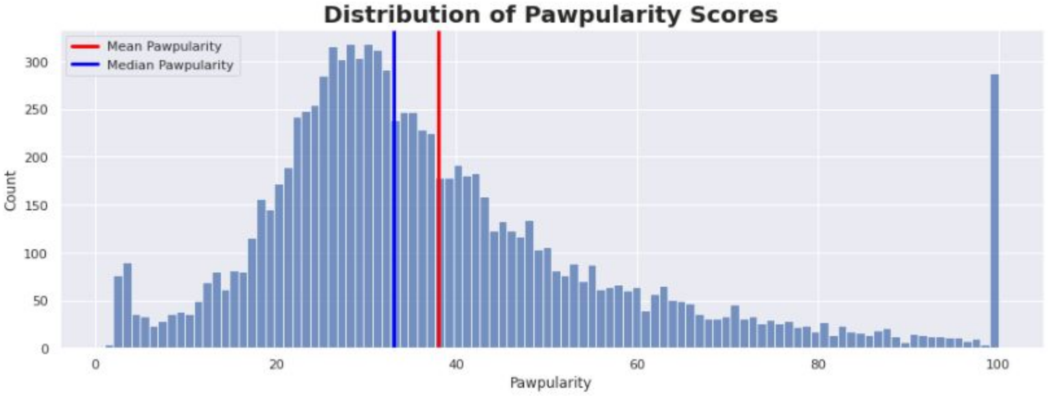
```python
train_df = pd.read_csv('./petfinder-pawpularity-score/train.csv')
train_df.head()
```

	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

```python
print(test_df.shape)
print(train_df.shape)
```

```
(8, 13)
(9912, 14)
```

	Pawpularity
count	9912.000000
mean	38.039044
std	20.591990
min	1.000000
25%	25.000000
50%	33.000000
75%	46.000000
max	100.000000



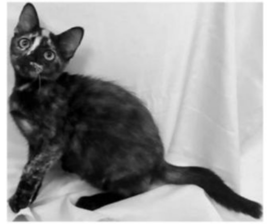
Berkeley
UNIVERSITY OF CALIFORNIA

EDA Results

- Distribution of Pawpularity scores are skewed with a small curve close to zero Pawpularity as well along with 300 Pawpularity scores at 100
- Distribution of Pawpularity scores is very similar for each variable and class
  - Features doesn't seem to influence the Pawpularity scores as much
- Found out that a winning solution requires the use of images and not the .csv metadata
- Found out that reshaping the images will be needed when building the models
  - In order to do so, we needed to retrieve the image filenames without the directory and .jpg at the end so that we can search the ID column in the train_df dataframe for Pawpularity scores

Score: 100

Score: 25th Percentile



Berkeley
UNIVERSITY OF CALIFORNIA

Tabular Data Models: Summary

We used three models with tabular data to predict scores. We quickly learned models performed better omitting outlier scores == 100, as suspected from EDA. Below are the results omitting 100 values from training:

Linear Regression

- Started simple with LR
- RMSE = 18.4451
- 46.83% of the predicted labels were within 10 points of the correct label

KNN Regression

- Used GridSearchCV to search best parameters
- k=175 was optimal and computationally reasonable
- RMSE = 18.4635
- 45.79% of the predicted labels were within 10 points of the correct label

Decision Tree & Random Forest

Decision Tree

- RMSE = 18.5166
- 46.21% of the predicted labels were within 10 points of the correct label

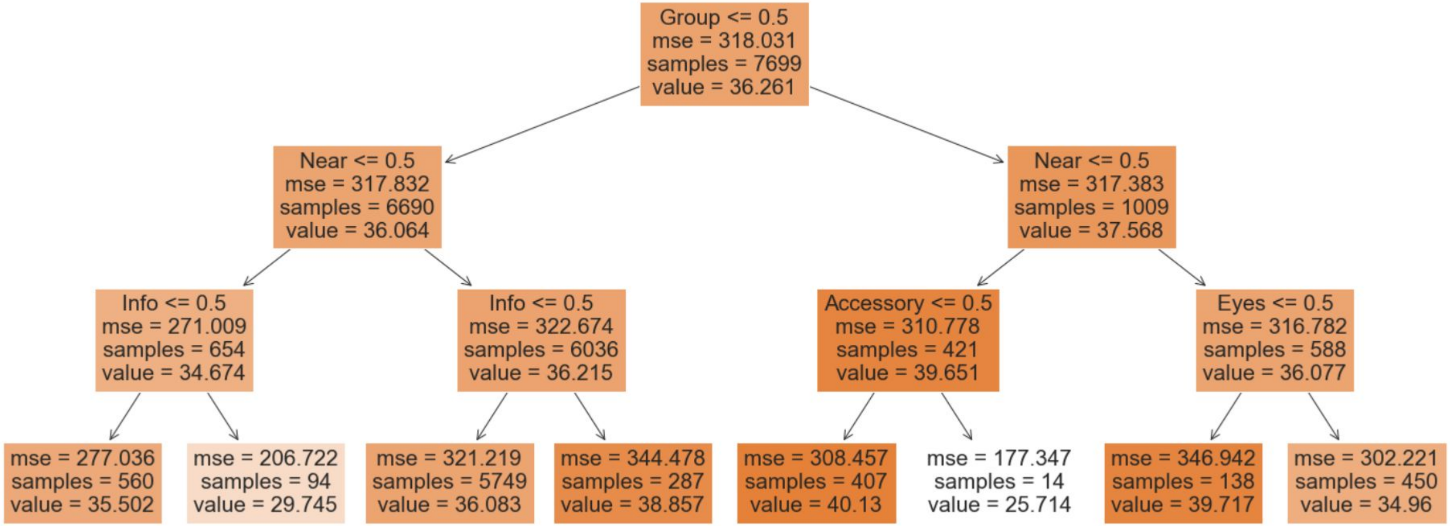
Random Forest

- Ensemble of 100 trees
- RMSE = **18.4401**
- 46.63% of the predicted labels were within 10 points of the correct label

Berkeley
UNIVERSITY OF CALIFORNIA

Tabular Data Models: Decision Tree

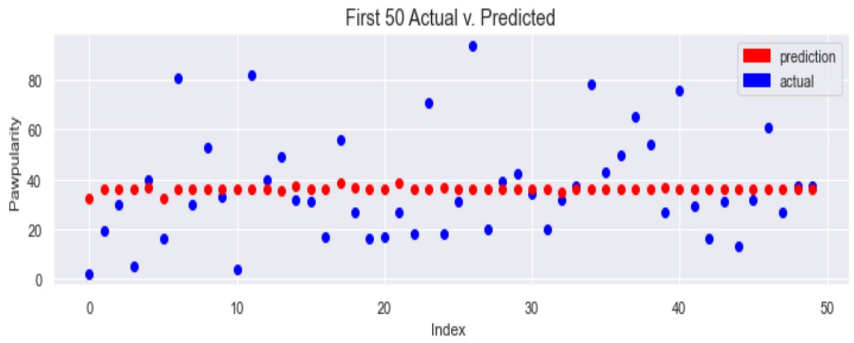
Below is our decision tree visualized with max_depth = 3, min_samples_split = 10:

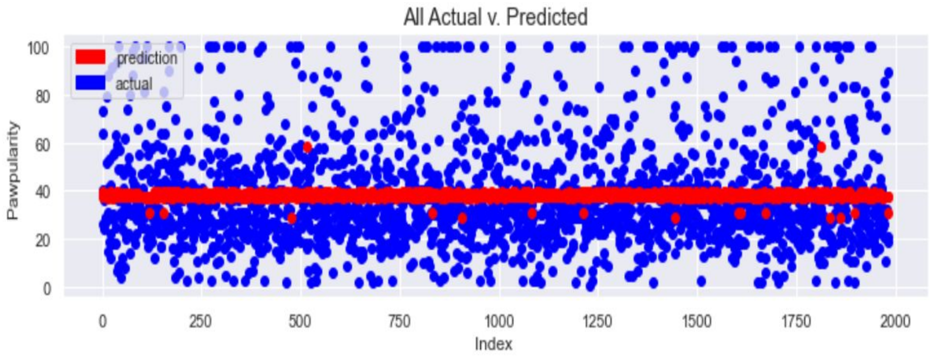


Berkeley
UNIVERSITY OF CALIFORNIA

Tabular Data Models: Random Forest

While the Random Forest Ensemble was our best scoring model for tabular data, we noticed a reliance on the mean to generate predictions:





Berkeley
UNIVERSITY OF CALIFORNIA

Pixel Data Preparation/Baseline Models

- Resized to 300x300 grayscale,
- Added padding to retain proportions

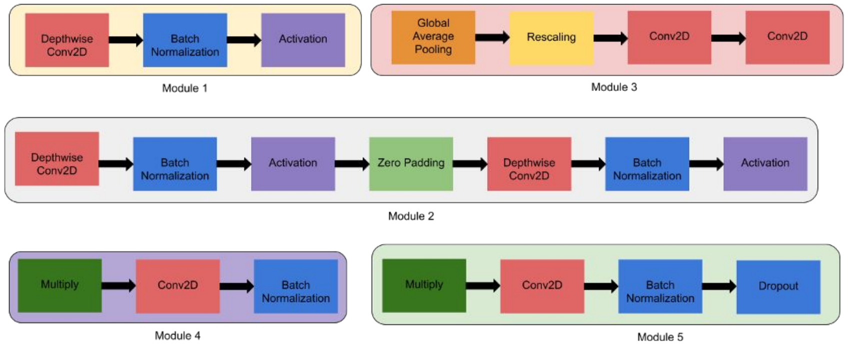
Baseline KNN: RMSE = 20.81
KNN w/ outliers excluded: RMSE = 18.46

Baseline Linear regression: RMSE = 27.06
Linear regression no outliers: RMSE = 18.45

Pixel Data CNNs

```
Layer (type)                 Output Shape              Param #
=================================================================
input_1 (InputLayer)         [(None, 128, 128, 3)]     0
conv2d (Conv2D)              (None, 61, 61, 16)        2368
conv2d_1 (Conv2D)            (None, 61, 61, 32)        4640
batch_normalization (BatchNo (None, 61, 61, 32)        128
conv2d_2 (Conv2D)            (None, 31, 31, 32)        9248
batch_normalization_1 (Batch (None, 31, 31, 32)        128
dropout (Dropout)            (None, 31, 31, 32)        0
conv2d_3 (Conv2D)            (None, 31, 31, 64)        18496
batch_normalization_2 (Batch (None, 31, 31, 64)        256
conv2d_4 (Conv2D)            (None, 16, 16, 64)        36928
batch_normalization_3 (Batch (None, 16, 16, 64)        256
dropout_1 (Dropout)          (None, 16, 16, 64)        0
conv2d_5 (Conv2D)            (None, 16, 16, 128)       73856
batch_normalization_4 (Batch (None, 16, 16, 128)       512
max_pooling2d (MaxPooling2D) (None, 8, 8, 128)         0
conv2d_6 (Conv2D)            (None, 8, 8, 128)         147584
batch_normalization_5 (Batch (None, 8, 8, 128)         512
dropout_2 (Dropout)          (None, 8, 8, 128)         0
flatten (Flatten)            (None, 8192)              0
dense (Dense)                (None, 512)               4194816
dropout_3 (Dropout)          (None, 512)               0
dense_1 (Dense)              (None, 1)                 513
=================================================================
Total params: 4,490,241
Trainable params: 4,489,345
Non-trainable params: 896
```

→ RMSE ~18.5 (not great, not terrible) --padding outperformed resizing



EfficientNet 200+ Layers, Many Hours of Training:

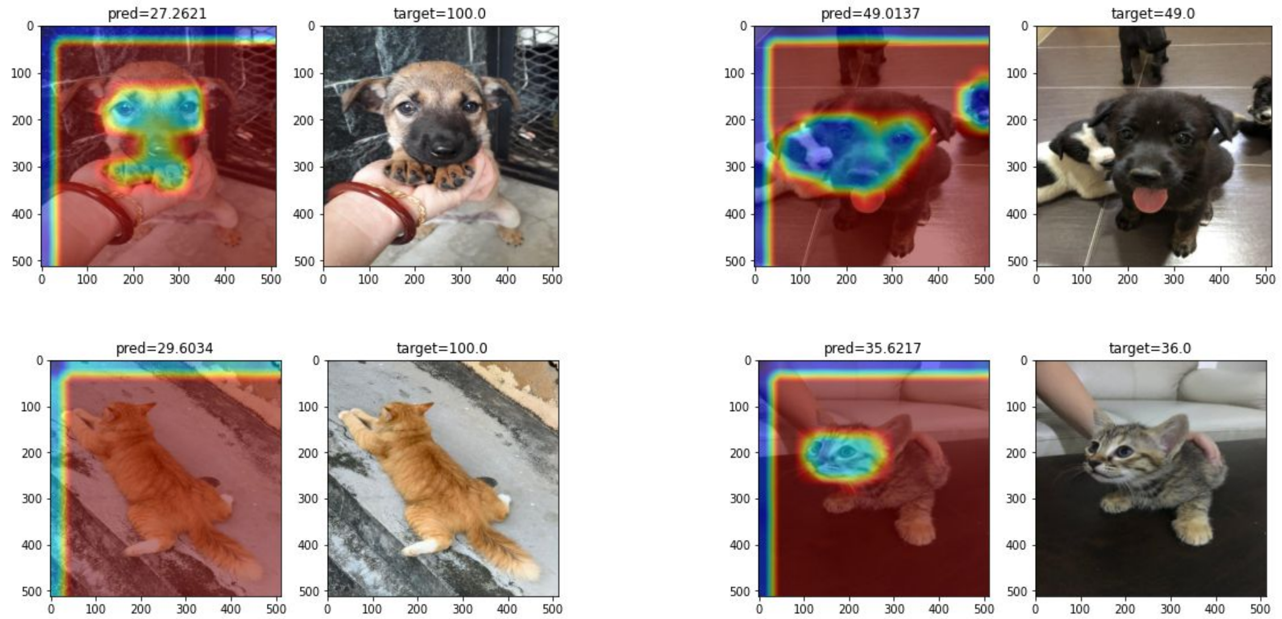
EfficientNet b3 Model RMSE + LightGBM = **16.5786**

Berkeley
UNIVERSITY OF CALIFORNIA

Grad-CAM (Gradient-weighted Class Activation Mapping

Good Region
Bad Score

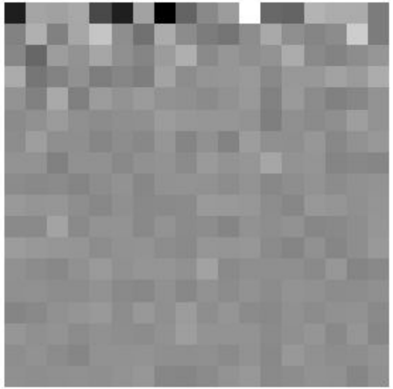
Bad Region
Bad Score

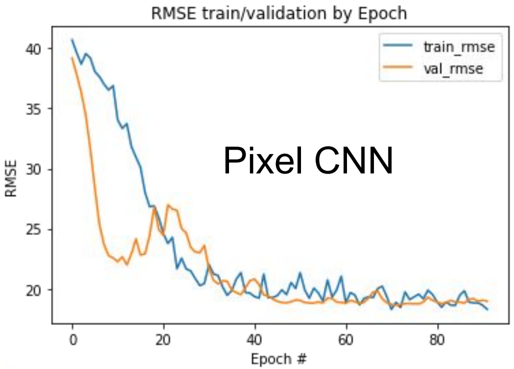


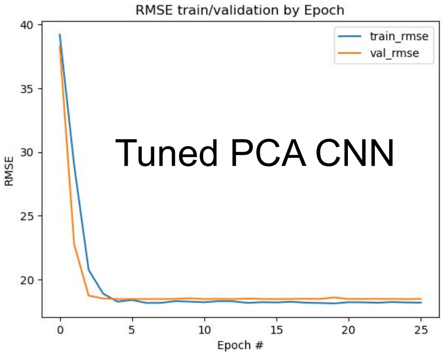
Good Regions
Good Scores

Pixel Data -> PCA

- Noisy images--many pixels might not be useful in discriminating
- Performed PCA and fed back through (differently structured) CNN
    - 90000 -> 324 (18 x 18) (300 PCs explained ~91% of variance)
    - MUCH faster training and tuning
    - Similar performance (RMSE = ~18.47)
- Images became very abstract
        - Training on PCA was more like “hitting a wall” compared to pixel CNN







Berkeley
UNIVERSITY OF CALIFORNIA

Combined Data Model: Tabular and Pixel

Given our separate tabular and pixel data results, we combined the two data sources

- Features: 12 tabular features + 324 PCA components from pixels
- Models:
  - Random Forest Ensemble because it was the best performing tabular model
  - Histogram-based GBM because LightGBM worked well with the EfficientNet CNN
- Results:
  - Improved scores compared to tabular data models
  - Random Forest outperformed Histogram-based GBM
    - RMSE = 17.8775
    - 47.48% of predicted labels are within 10 points of the correct label

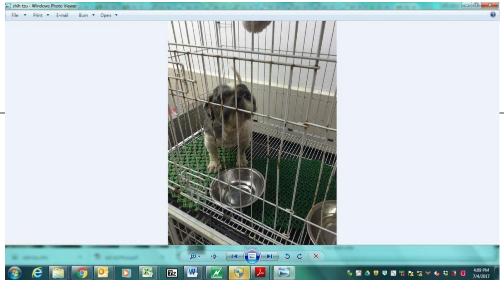
Berkeley
UNIVERSITY OF CALIFORNIA

Overall Summary

Challenges	Surprises	Methods
Understanding Image Scores	Unexpected Data Skews	Digit Classification
Transforming Images	Lowest RMSE - Random Forest	PCA, Flattening/Padding Images
Creating High-Performing Models	No Insights On People Scoring	KNN and Linear Regression
Defining CNN Model Architecture	Lack Of Results From CNN Model	CNN, Random Forest/Decision Tree
Accurately Predicting Pawpularity Scores Based on Images	Similar Distribution Of Pawpularity Scores For Each Variable/Class	Packages such as tensorflow, scikit, sklearn, matplotlib, etc.

Missing Data / Noise

Missing Data	Noise
Insights On How People Scored	Distribution Skew Of Pawpularity Scores - e.g. 300 Pawpularity Scores at 100
Website Mechanics	Unnecessary background items and colors
External Influential Factors (e.g. # of clicks per pet)	



Thank you!

Berkeley
UNIVERSITY OF CALIFORNIA