UC Pupeley: Improving Pet Adoptions from Online Posts Using Machine Learning

MIDS W207 Machine Learning Final Project
December 7, 2021

Team Members

1. Autumn Rains
2. Chandler Haukap
3. Carlos Moreno

Problem Statement

Owning a pet can be a benefit and highlight to one's life. The Center for Disease Control (CDC) in the United States explains that pet ownership “...can increase opportunities to exercise, get outside, and socialize. Regular walking or playing with pets can decrease blood pressure, cholesterol levels, and triglyceride levels. Pets can help manage loneliness and depression by giving us companionship. Most households in the United States have at least one pet." However, not all pets are fortunate enough to have a home. According to the American Society for the Prevention of Cruelty to Animals (ASPCA), "approximately 6.3 million companion animals enter U.S. animal shelters nationwide every year." (*Pet statistics* 2021) Pet adoption efforts are not unique to the United States. In Malaysia, PetFinder.my is the country's top web platform for pet adoption "featuring over 180,000 animals with 54,000 happily adopted. PetFinder collaborates closely with animal lovers, media, corporations, and global organizations to improve animal welfare." (*Petfinder.my - Pawpularity Contest* 2021)

Our project team found PetFinder's Kaggle competition with the main goal of studying the performance of existing pet photos using machine learning methods to understand and make a recommendation of how to improve adoption rates through better images. From the competition details: "Currently, PetFinder.my uses a basic Cuteness Meter to rank pet photos. It analyzes picture composition and other factors compared to the performance of thousands of pet profiles. While this basic tool is helpful, it's still in an experimental stage and the algorithm could be improved." (*Petfinder.my - Pawpularity Contest* 2021)

Objective

The objective of this project is to analyze metadata from raw images to identify the attributes of pet photos that most directly influence the popularity (‘pawpularity) of the pet’s profile on the adoption site. We will use various machine learning techniques with the ultimate goal of creating a machine learning pipeline that PetFinder.my can incorporate into their existing site to improve adoption rates. By improving pet adoption rates, cats and dogs in pet shelters will be able to find a 'furever' home.The main metric used in the competition to evaluate model’s performance is Root Mean Square Error (RMSE) of the predicted “Pawpularity” score. “Pawpularity” score is a 1 to 100 points score given to every image, where 100 represents the best possible image.

Approach/ Methodology

Our approach for this project included the following steps:

(1) Analyze features and images provided by competition to identify potential opportunities and challenges for Machine Learning (ML) models.
(2) Test preliminary Machine Learning (ML) models on the original metadata provided.
(3) Perform feature extraction to create additional image features to aid in ML models performance.
(4) Select the best combination of features and identify the best ML model configuration for best performance.
(5) Provide final recommendation and define next steps.

Block Diagram

The process that our project team followed is depicted below in Figure 1:

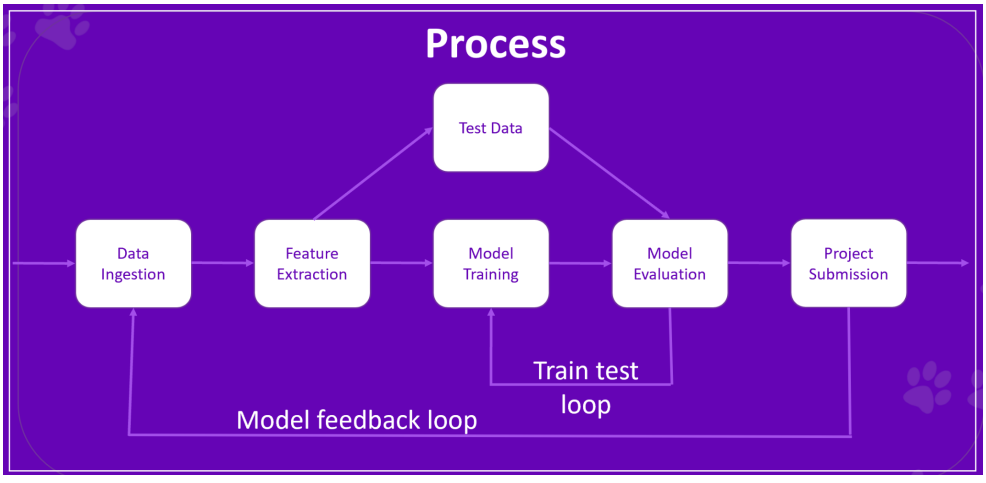


Figure 1: Project Process

Datasets (potential)

The dataset used for our project was found on Kaggle.com at the following address:

https://www.kaggle.com/c/petfinder-pawpularity-score/data

This was the primary dataset used for our project work.This dataset consists of 12 features that Petfinder.my extracted from the images, along with the images themselves and a “Pawpularity” score which serves as the target. There are 9,920 images in the dataset. An example of the dataset as well as images are seen below Table:

	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

Table 1: Dataset Features & Target

From the above training data, one can see there are 12 features excluding “Pawpularity” score and Image Id. Each feature is an indicator variable where 1 indicates that the feature is present in the image and 0 indicates that the feature is not present. The definitions of the features are as follows (obtained from the Kaggle project data dictionary):

1. **Focus** - Pet stands out against an uncluttered background, not too close / far.
2. **Eye**s - Both eyes are facing front or near-front, with at least 1 eye / pupil decently clear.
3. **Face** - Decently clear face, facing front or near-front.
4. **Near** - Single pet taking up a significant portion of the photo (roughly over 50% of photo width or height).
5. **Action** - Pet in the middle of an action (e.g., jumping).
6. **Accessory** - Accompanying physical or digital accessory / prop (i.e. toy, digital sticker), excluding collar and leash.
7. **Group** - More than 1 pet in the photo.
8. **Collage** - Digitally-retouched photo (i.e. with digital photo frame, combination of multiple photos).
9. **Human** - Human in the photo.
10. **Occlusion** - Specific undesirable objects blocking part of the pet (i.e. human, cage or fence). Note that not all blocking objects are considered occlusion.
11. **Info** - Custom-added text or labels (i.e. pet name, description).
12. **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.

The 'Pawpularity' score is on a scale from 0 - 100, with 100 being the desired outcome. A higher score indicates that an image performed better on the PetFinder.my website. (*Petfinder.my - Pawpularity Contest* 2021)

Pawpularity

As mentioned, Pawpularity is on a scale from 1 to 100 where 100 is the “best” score and 1 is the “worst”. Pawpularity is a derived value. According to Petfinder (*Petfinder.my - Pawpularity Contest* 2021):

> “The **Pawpularity Score** is derived from each pet profile's page view statistics at the listing pages, using an algorithm that normalizes the traffic data across different pages, platforms (web & mobile) and various metrics.”

Unfortunately, figure two demonstrates that, while petfinder claims that the score is normalized, it’s distribution is in fact skewed right. The mean score is approximately 38.26, and 78.7% of all scores were <= 50. In addition, there is a sizable spike at a score of 100. This will make it difficult to train a model because most models for continuous variables assume normally distributed targets.

It also suggests that the way Pawpularity is currently calculated might be flawed. When generating a customer engagement metric, it's always best practice to normalize. Alternatively, this could be a problem of inappropriate sampling techniques resulting in data that is not I.I.D.

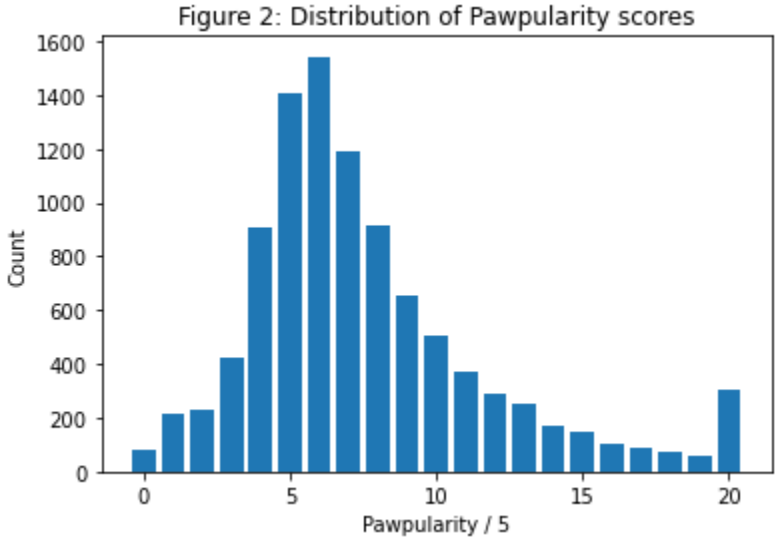


Figure 2: Distribution of Pawpularity Scores

The following figure presents the correlation matrix of the 12 original attributes to "Pawpularity". This matrix shows that none of the features are correlated with “Pawpularity” which indicates that potential ML models will need to extract additional features from the images themselves.

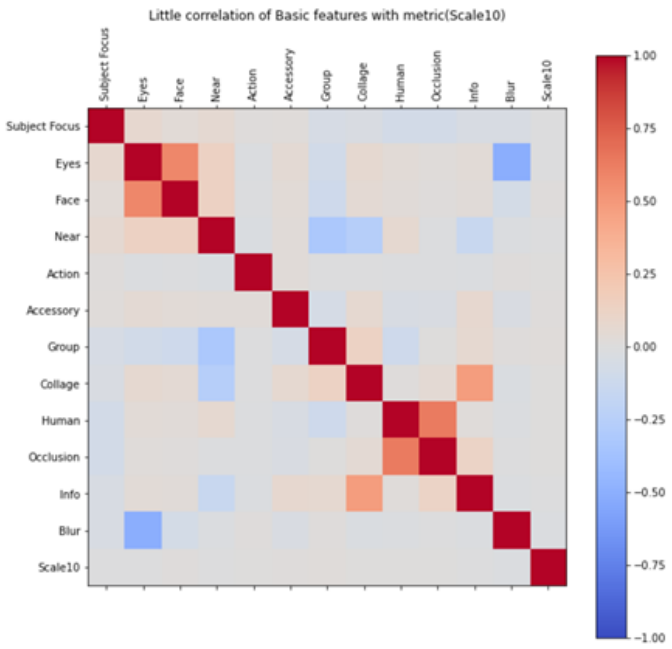


Figure 3: Correlation Matrix of Original Dataset

Images

Below are some examples of images from the dataset:

After reviewing the images, the following points are notable:

- The images were all different sizes and varying pixel quality.
- Some images were collages while others contained more than one animal.
- Images that appear similar, i.e. same breed and relatively same age, did not appear to have any correlation in scores.

- Some images with scores of 100 had the same 12 given attributes as images with scores of 1.
- The dataset contained duplicate images. These duplicates had different pawpularity scores.

What is considered success/ failure?

Our team had one major criteria for success based on the competition evaluation metric:

1. Root Mean Squared Error (RMSE) <= 10

This requires that our model produce estimates that are, on average, within 10 "Pawpularity" points of the true score. This parameter was chosen as the primary metric for success given the structure of the competition. Winners are chosen based on the lowest possible RMSE score. RMSE is generally used a measure of the difference between predicted values of a parameter in a sample or population by models and the actual observed value. (*sklearn metrics -mean_squared_error* 2021)

Evaluation Parameters

While a low RMSE was our main objective, we decided to consider a few other metrics during training. We did this to ensure that the models were robust enough to be performant on the larger withheld set that Petfinder uses for the competition. In addition to the RMSE we also considered F1 score, accuracy, and hamming loss as metrics to evaluate the performance of our models.

According to the authors of sklearn materials, "the F1 score can be interpreted as a harmonic mean of the precision and recall, where an F1 score reaches its best value at 1 and worst score at 0." Precision can be understood as the percentage of True positive predictions from all positive classes (both True and False positives). Recall on the other hand is the number of True Positives divided by the number of True Positives and False Negatives. Our project team opted to select F1 score to capture models that high a good balance of both precision and recall.

Accuracy is a measure of correctly predicting data labels. Hamming loss is similar in that it represents the fraction of labels that have been incorrectly predicted. A score for both of these parameters will be between 0 and 1. A higher score for accuracy is ideal while the opposite is true for hamming loss.(*Sklearn.metrics.f1_score* 2021)

Experiments

Our project team had multiple approaches to improve model performance.

PCA

Based on initial performance evaluation of the models using the dataset as provided, it was clear to the our project team that transformations would be a good next step to improve performance. Given the features of the dataset were derived from images, we determined that Principal Component Analysis (PCA) was an appropriate method for dimensionality reduction.

From sklearn documentation, "PCA is used to decompose a multivariate dataset in a set of successive orthogonal components that explain a maximum amount of the variance." In other words, it reduces the dimensional complexity of data and is ideal for transforming image data given that it reduces the number of features (components). For our project, PCA analysis on the dataset from Kaggle yielded that of the 12 components, 7 of them explain over 90% of the variance. This can be seen in Figure 2 below:

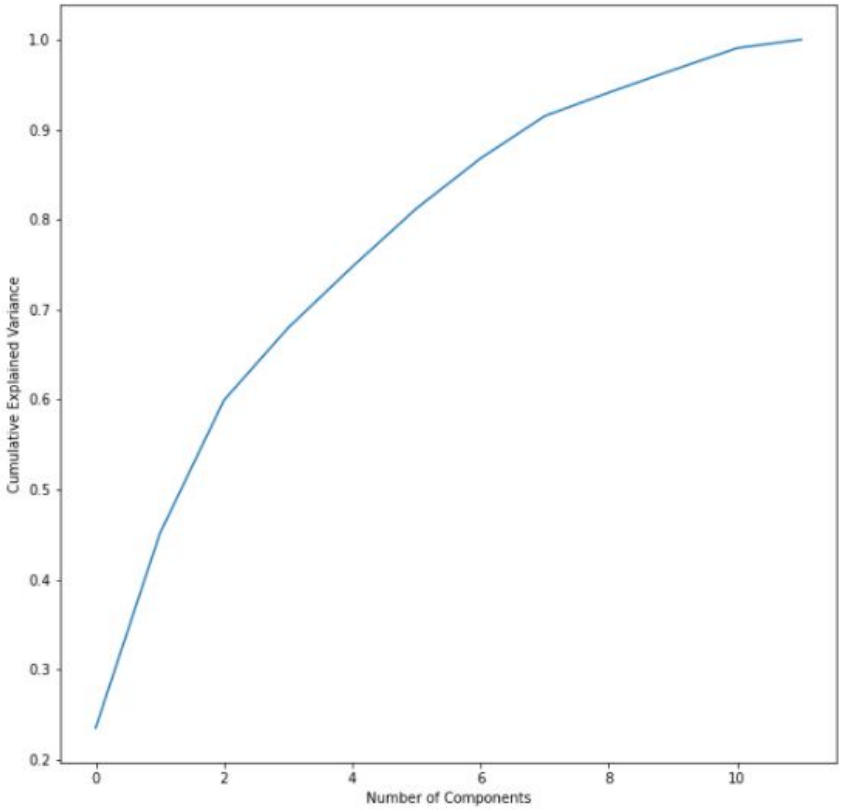


Figure 4: PCA of Original Dataset

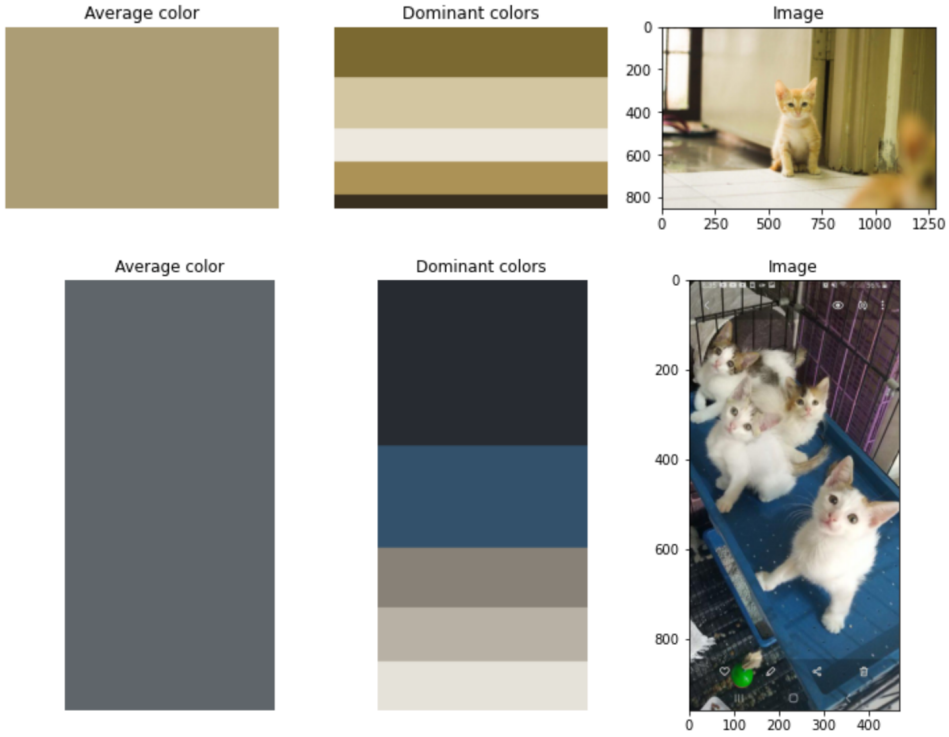
Therefore, one experiment performed on the dataset was to perform PCA with 7 components on the Kaggle dataset and rerun the baseline models. The top seven features were: ‘Eyes’, ‘Occlusion’, ‘Near’, ‘Group’, ‘Blur’, ‘Face’, and ‘Accessory.’ The RMSE was only slightly improved in some of the models by a few points of one or two, thus our team realized that additional experiments would be needed to achieve our project goals.

Color Palettes

When viewing images in rapid succession, humans can make decisions without even activating their conscious brain. A 2008 study of facial attractiveness found that participants took on average 150 milliseconds to judge a face as attractive or unattractive. While we do not have formal research of this kind for pet photos, it’s safe to assume that our initial judgement of pet photos is also rapid and subconscious. With this in mind, we spent some time identifying the color palette of our images to

see if they provided any explanatory power. (*(PDF) the appraisal of facial beauty is rapid but not mandatory* 2021)

The image's color palette is defined as the composition of 7 colors, with each color represented by three numbers (for the RGB values associated with a color). The seven attributes are the 5 most common colors for an image, the top color, and the average color for the image. Each color is represented by three values (RGB), thus a palette for an image includes 21 values (or features). We explored the "palette" of an image as a potential feature, as we hypothesized that the color palette of an image could influence the score. Below, there is an example of the color palette for two images.



Once the palette for all images was identified, we performed a correlation analysis to identify if this feature was correlated with the score (“Pawpularity”). The correlation matrix is shown below. Based on this analysis, we concluded that the image's palette is not correlated with the score, and it is unlikely to help differentiate images by itself. However, we would integrate these features with others, as the combination of features may help to correctly score the images.

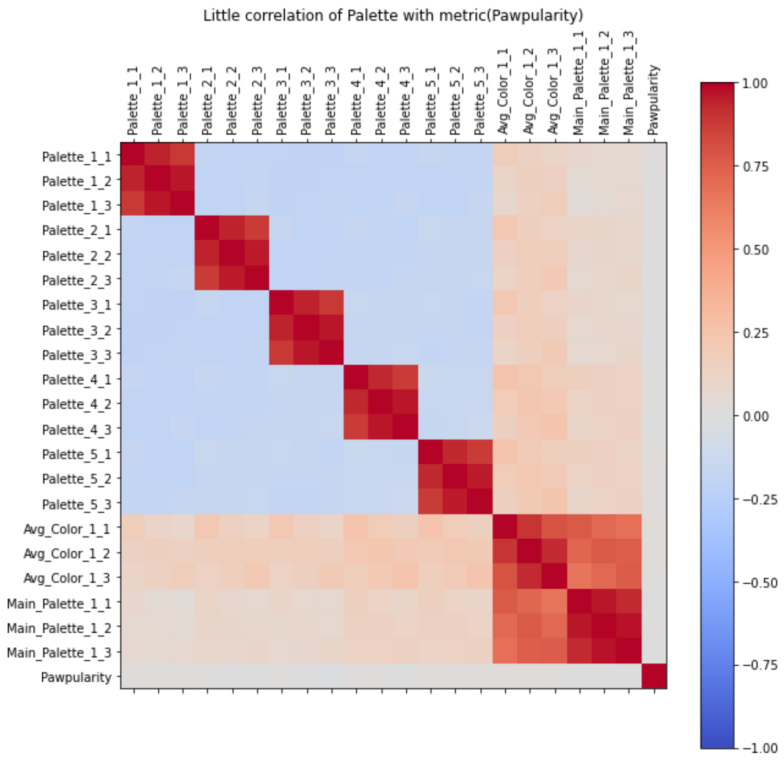


Figure 5: Correlation Matrix with Color Pallette

Bucketing

Using the 12 basic features ('Subject Focus', 'Eyes', 'Face', and so forth), and the “Pawpularity” scale (1-100), we tested several models using different configurations (i.e. KNN, NB, Multinomial NB, Gaussian NB, Logistic Regression, Decision Tree, Random Forest, AdaBooster, SVM and Neural Networks), we found that the best accuracy score was around 2.8%, f1_score around 1.2%, and RMSE of ~25.

The high RMSE was a concern as if we used the average “Pawpularity” score for all images as a prediction, we could achieve an RMSE of 20.6. A key challenge was that the models were not able to correctly classify any image with a “Pawpularity” score greater than 50. The following confusion matrix (Random Forest Model with 20 estimators, using Buckets of 10) illustrates the challenge:

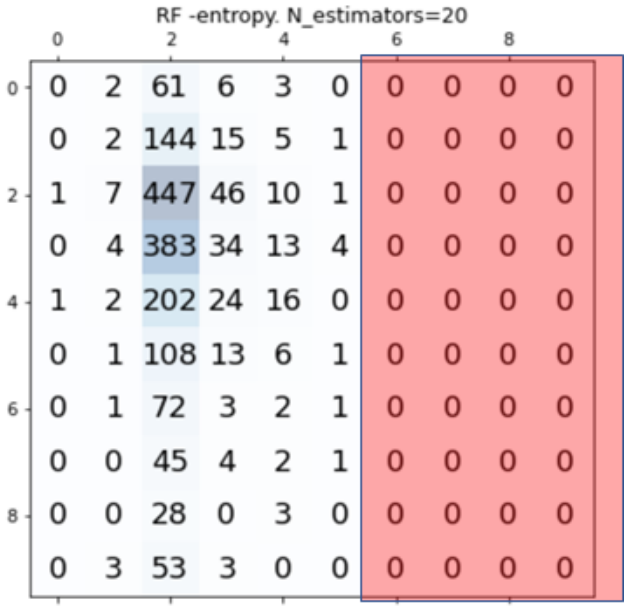


Figure 6: Confusion Matrix with Bucketing

As it can be seen in the confusion matrix, no image with a score of 50 or more is correctly classified. Thus, we explored the performance of multiple models with different configurations using images classified in different numbers of "Buckets" - 10 buckets, 5 buckets, 4 buckets, 3 buckets, and 2 buckets. The following graph presents the performance using these buckets.

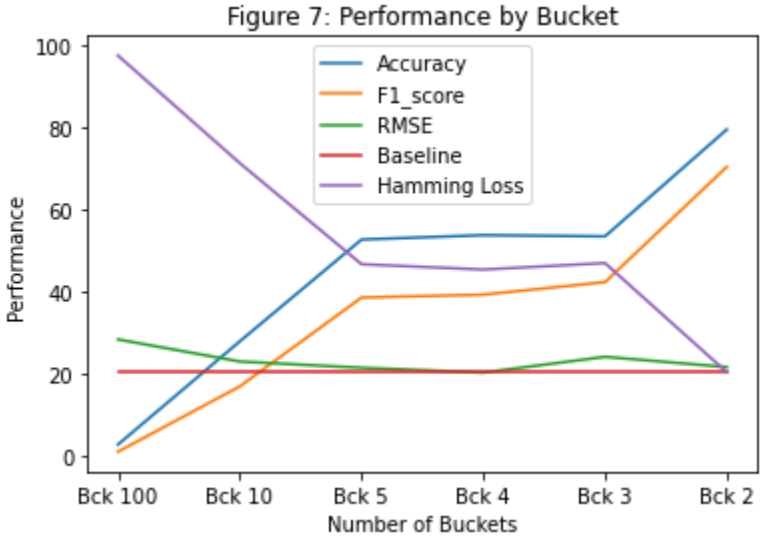


Figure 7: Performance by Bucket

After exploring performance for different buckets, we defined the following:

- Using 10 Buckets has the potential to significantly improve a model's performance given the noise in the data.

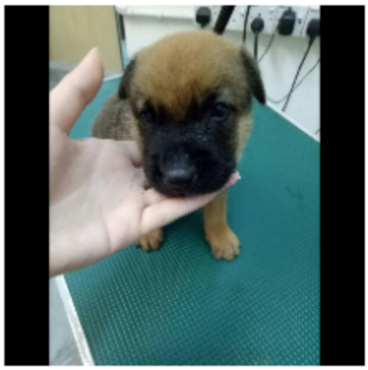
- Confirmed the need to identify additional image features that could help classify images with score above 5.
- Test different combinations of features together with different model configurations.

Image Classification

It seems intuitive that dog/cat breed and other objects in the images would influence the “Pawpularity”. To investigate this, we decided to try labeling the images.

There are a growing number of image classification models available online, so we decided not to reinvent the wheel. Instead, we chose the resnet_v2_152 classification model from TensorflowHub (*Tensorflow Hub* 2021). We found this model to have a good tradeoff between speed and accuracy. Plus, it contains 1,000 labels with multiple cat and dog breeds. The graph below presents an example of an Image Classification.

```
CPU times: user 4.22 s, sys: 674 ms, total: 4.89 s
Wall time: 3.46 s
CPU times: user 1.12 s, sys: 452 ms, total: 1.58 s
Wall time: 396 ms
True
(1)  225 - malinois: 0.8495094776153564
(2)  182 - Border terrier: 0.12605944275856018
(3)  255 - Leonberg: 0.02231677807867527
(4)  262 - Brabancon griffon: 0.0007704859017394483
(5)  235 - German shepherd: 0.0005434566410258412
(6)  174 - Norwegian elkhound: 0.0004126651620026678
(7)  202 - soft-coated wheaten terrier: 0.0001259345153812319
(8)  242 - boxer: 3.430302604101598e-05
(9)  192 - cairn: 3.33454372594133e-05
(10)  260 - chow: 3.1847794161876664e-05
```



Using this algorithm, we identified 360 additional features. These new features indicated the type of pet being displayed in the image. The correlation matrix indicated that some of these new features are correlated to “Pawpularity” which indicated that they could help to correctly differentiate images. The graph below presents the correlation matrix including the top 24 features and “Pawpularity”.

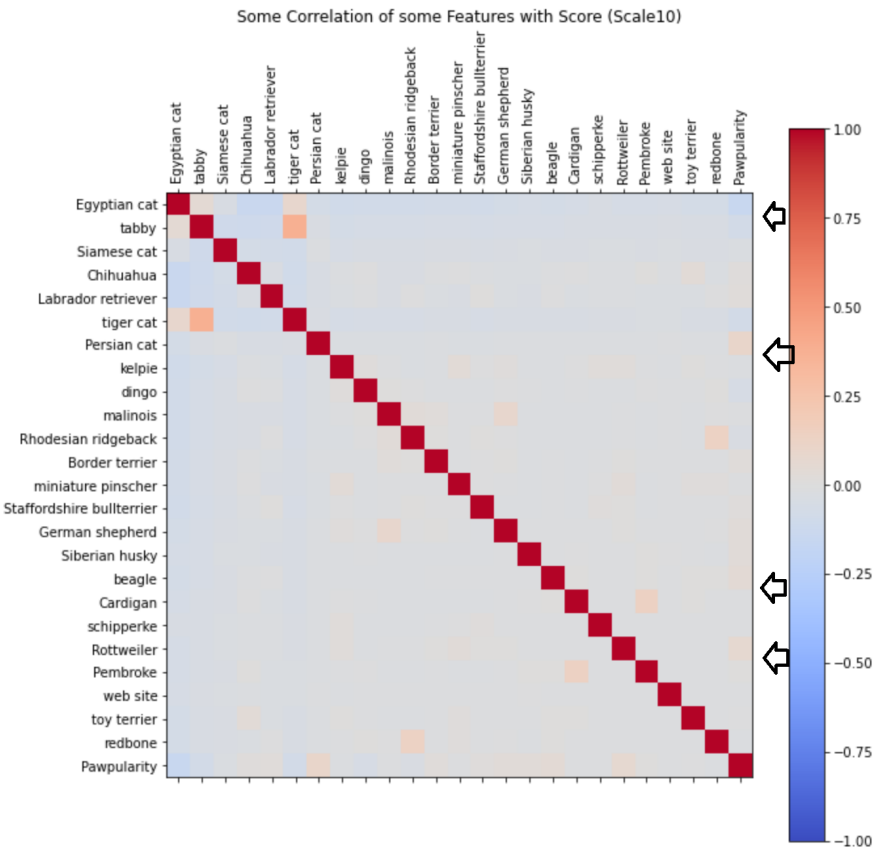


Figure 8: Correlation Matrix with Image Classification

Species

Given the number of labels generated by the classifier, we were concerned about overfitting. In an attempt to solve this problem, we tried mapping the labels based on species. Fol example, the labels “toy terrier”, “german shepherd”, and “chihuahua” all mapped to “dog” and the labels “egyptian cat”, “tiger cat”, and “siamese cat” all mapped to “Cat”. From there, we selected the maximum probability from both sets

$$P(dog) = P_{max}(\{dog\ labels\})$$
$$P(cat) = P_{max}(\{cat\ labels\})$$

Then used indicator variables determined by $P_{max}(dog) > 0.5$ and $P_{max}(cat) > 0.5$

Using a threshold of 0.5 divided the data into three distinct sets of similar size: Images labeled “dog”, images labeled “cat”, and images that the model did not recognize as either cats or dogs.

Using a t-test we determined that the cat and dog labels most likely did constitute separate subpopulations in the data set. However, the distributions of the subpopulations were not normally distributed.

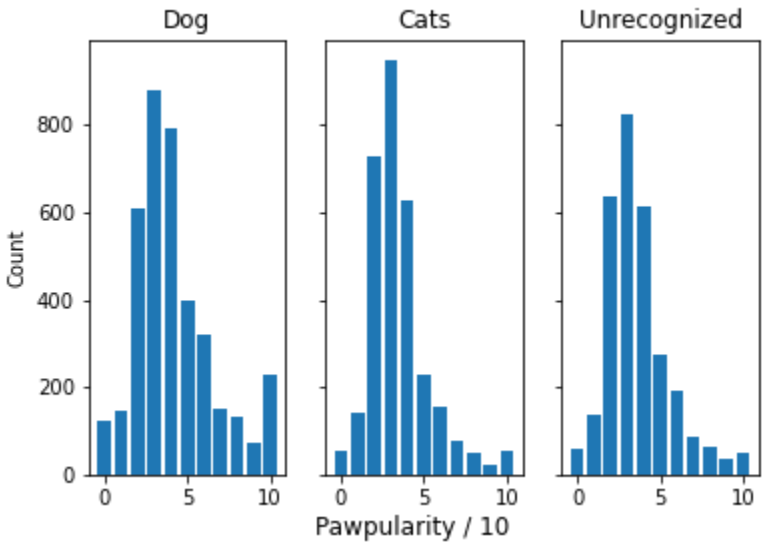


Figure 9: Distribution of Image Classifications Cat, Dog, None

Without a normal distribution the two labels were simply insufficient in describing the variance of Pawpularity scores compared to using all of the labels. We therefore abandoned this feature flattening technique in favor of the full feature set.

EigenFaces

Because we were working with images, it felt appropriate to see if eigenfaces would give us some explanatory power.

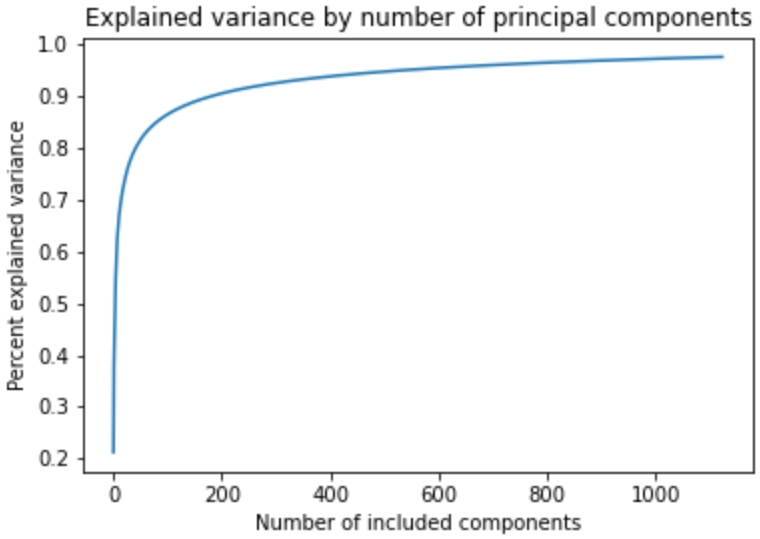


Figure 10: PCA of Original Features And EigenFaces

We started by scaling and padding the images then converting them to black and white. From there, we made the eigenfaces. The following images are the top five eigenfaces derived from the PCA.

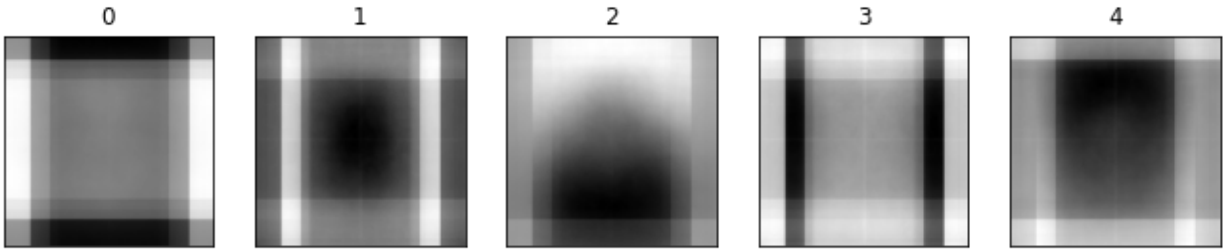


Figure 11: Example Output of EigenFaces

As you can see, there doesn't appear to be any recognizable pattern in the variance of the data and, after running some models using only these components, there doesn't appear to be any significance in the shared variance of the images. The images are simply too different to draw any conclusions from their eigenfaces.

Results

Several types of machine learning models were explored using different configurations. The table below shows the different configurations considered for the models:

MODEL	EVALUATED PARAMETERS			
KNN	**METRICS** 'euclidean', 'manhattan', 'chebyshev', 'minkowski'	**K-Values:** 1, 2, 3, 4, 5, 6, 7, 8 , 9 , 10, 11, 12, 13, 14, 15, 20	**Algorith** 'auto'	**Weight** "uniform"
Naive Bayes (NB)	**alpha:** 1.0e-10, 0.0001, 0.001, 0.01, 0.1, 0.5, 1.0, 2.0, 10.0, 20, 30 ,20 ,30 ,100			
Multinomial Naive Bayes (MNB)	**alpha:** 1.0e-10, 0.0001, 0.001, 0.01, 0.1, 0.5, 1.0, 2.0, 10.0, 20, 30 ,20 ,30 ,100			
Gaussian NB	**smoothing:** 1.0e-10, 0.0001, 0.001, 0.01, 0.1, 0.5, 1.0, 2.0, 10.0, 20, 30 ,20 ,30 ,100			
Logistic Regression	**Penalty** "l1", "l2"	**Solver** "liblinear", "newton-cg", "sag", "lbfgs"	**C** 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1, 1.1, 1.2, 1.3, 1.4, 1.5, 2, 3, 4, 5	**Multi Class** auto **Max Iter** 400
Decision Tree	**Criterion** "entropy", "gini"	**Max_Depth** 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12	**min_samples_split** 10	**random_state** 12345
Random Forest	**Criterion** "entropy", "gini"	**Number of Estimators** 5, 10, 15, 20, 25, 30, 35, 40, 50, 55, 60, 80, 100	**min_samples_split** 10	**random_state** 12345
AdaBoost	**Algorithm** "SAMME", "SAMME.R"	**Number of Estimators** 5, 10, 15, 20, 25, 30, 35, 40, 50, 55, 60, 80, 100	**learning_rate** 1.2	**random_state** 12345
SVM	**Kernel** "linear", "rbf", "poly", "LinearSVC"	**C** 0.5, 1, 1.5, 2, 2.5, 3, 4, 5, 10, 20	**LinearSVC**: max_iter=10000	**poly**: degree=2, gamma=1 **rbf**: gamma=0.7
Neural Networks	**Activation** "identity", "logistic", "tanh", "relu"	**layers** (10,10,10), (5,5,5), (3,3,3), (20, 20, 20)	**solver** "lbfgs", "sgd", "adam"	**alpha** 0.0001, 0.001, 0.01, 0.1, 0.2, 0.3, 0.5
Note: As some combination of parameters are not possible, testing was adjusted accordingly				

Table 2: Parameters Used for each Model

We created routines to test key combinations of parameters for the models outlined above, and the results were saved in an excel file. In total, we generated 335 different models using a 10 Buckets classification as the main outcome. The following table presents a summary of the performance of the best configuration for the models explored (by category). In the following sessions, we will further discuss the performance of the selected models.

MODEL	# Models	Winning Configuration	Accuracy	f1_score	RMSE_adj	Hamming Loss
KNN	64	k=13, metric=manhattan	26.46	21.96	19.65	73.54
Naive Bayes (NB)	14	alpha = 0.01	28.98	18.75	20.08	71.02
Multinomial Naive Bayes (MNB)	9	alpha = 0.01	28.98	20.56	19.27	71.02
Gaussian NB	9	smoothing=0.1	25.22	18.87	20.05	74.78
Logistic Regression	95	**c=4, solver=newton, penalty=l2**	**29.32**	**22.30**	**19.14**	**70.68**
Decision Tree	24	criterion=gini, n_estimators=20	27.91	21.72	19.31	72.09
Random Forest	26	**criterion=gini, n_estimators=20**	**28.31**	**23.17**	**19.03**	**71.69**
AdaBoost	26	algorithm=SAMME.R, n_estimatore=20	30.27	20.12	19.29	69.73
SVM	40	kernel=linear, c=1.5	29.65	20.84	19.24	70.35
Neural Networks	28	activation=identity, solver=sgd	31.28	23.07	19.20	68.72
Total Models for 10 Buckets:	335					

Table 3: Model Performance Results (Best Configurations)

Exploring Model Performance using 10 Bucket Classification

After exploring different predictive models, the two best configurations were as follows:

(1) Logistic Regression:

- C = 4
- solver="newton-cg“
- multi_class="auto“
- Penalty = l2
- max_iter=400

(2) Random Forest:

- Criterion = Entropy
- Max Depth = 11
- Num Estimators = 20
- min_samples_split=10

Although the Random Forest model was able to achieve a lower RMSE for some test/train splits, we selected logistic regression as the recommended model because it performed better on average. Therefore, we believe that logistic regression is more likely to prove effective when presented with novel data.

The models were evaluated using different number of features - the feature configurations were as follows:

- 12 Features = Original Features only.
- 21 Features = Palette Only
- 33 Features = Original Features + Palette
- 36 Features = Original Features + 24 key Image Features

- 57 Features = Original Features + Palette + 24 Key Image Features
- 100 Features = Top 100 Principal Component Features (out of 393 features) - representing 97.7% variance
- 372 Features = Original Features + 360 Image Features
- 393 Features = Original Features + Palette + 360 Image Features

The following graph presents the performance for the selected LogisticRegression model across all the different numbers of features.

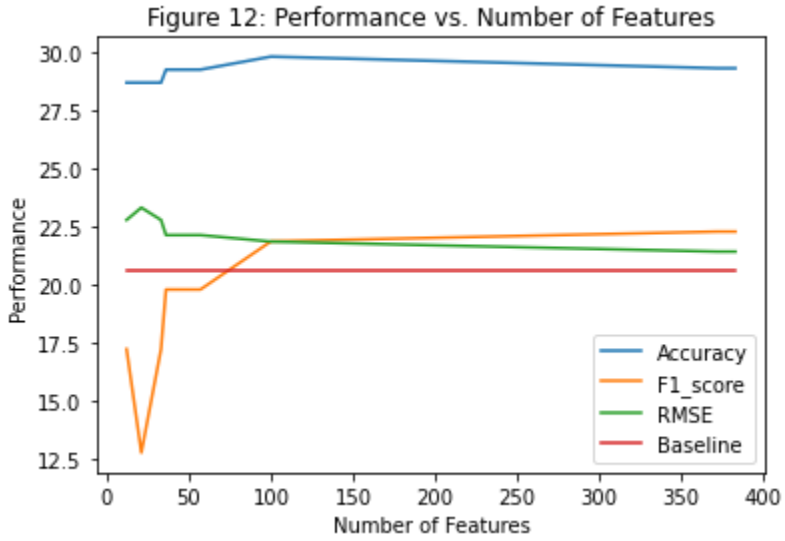


Figure 12: Performance of Logistic Regression Model Vs. Number of Features

While the addition of features helped improve the RMSE (green line) by lowering it from ~23 to 21.4, increasing f1_score (yellow line) from 17.25 to 22.3, as well as increasing accuracy(blue line) from 28.7% to 29.32%, the RMSE was still above the baseline (20.6 - red line: using overall image image score as prediction). The table below presents the performance values for our best performing Logistic Regression model.

Description	Original (12)	Palette (21)	Original (12) + Palette (21)	Original (12) + Key Image Features (24)	Original (12) + Palette (21) + Key Image Features (24)	PCA (100)	Original (12) + All Image Features (360)	Original (12) + All Image Features (360) + Palette (21)
N_Features	**12**	**21**	**33**	**36**	**57**	**100**	**372**	**393**
RMSE	22.80	23.33	22.80	22.15	22.15	21.87	21.44	21.44
Accuracy	28.70	28.70	28.70	29.26	29.26	29.82	29.32	29.32
F1_score	17.25	12.80	17.25	19.81	19.81	21.88	22.30	22.30
Hamming Loss	71.30	71.30	71.30	70.74	70.74	70.12	70.68	70.68

Table 4: Model Results - Logistic Regression (Best Model)

Reviewing strategies taken by other teams in the competition, it suggested that besides being able to correctly classify the images in the ten buckets, it was important to adjust the mean value used for

the bucket. Given that images were clustered between bucket 3 and 5, the mean for every bucket needed to be adjusted to reflect this distribution when calculating the RMSE..

The following table presents the original mean used by bucket, and the adjusted means. The calculation is as follows:

- Original Means = Average Score for Images in Train Data placed in a given bucket.
- Adjusted Means 1 = For each bucket add the average for all images (37.8) and divide by two.
- Adjusted Means 2 = For each Adjusted Mean 1, add 4 points (4 points was approximated by running linear regression of Buckets and “Pawpularity” - using the beta coefficient found).

Bucket	Average Score	Adj Average Score Using Total Avg	Adj Avg + Factor of 4.0
1	5	22	**26**
2	17	27	**31**
3	26	32	**36**
4	35	36	**40**
5	45	41	**45**
6	55	46	**51**
7	65	51	**55**
8	75	56	**60**
9	85	61	**65**
10	99	68	**72**
Total Avg	**37.82**		

Table 5: Original Means and Adjusted Means by Bucket

Using the adjusted means 2, the RMSE decreased, going below the baseline of 20.6 (green line). The following graph presents the results where the RMSE_adj (orange line) goes down to 19.14.

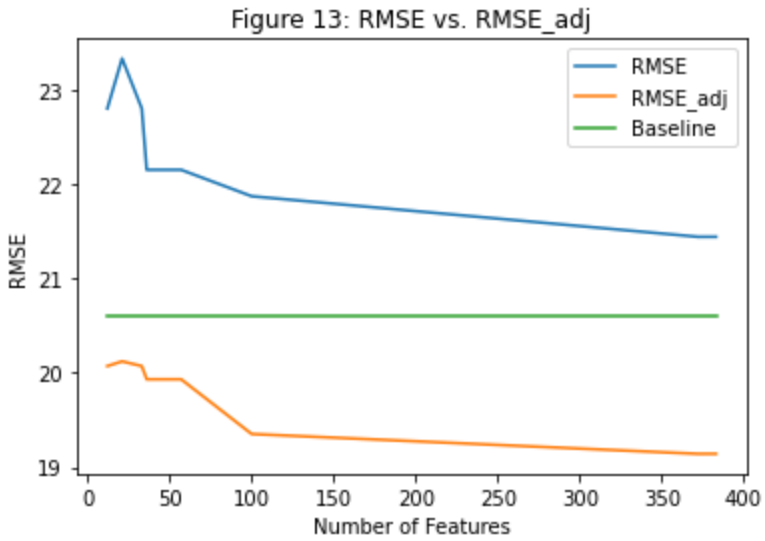


Figure 13: RMSE vs. Adjust RMSE per Number of Features

The following table presents the specific values for the RMSE_adj achieved by our best performing Logistic Regression model using the adjusted means for the buckets. The RMSE_adj went down from 20.07 (using 12 Original Features) to 19.14 (using Original Features and the 360 Additional Image Features identified).

Description	Original (12)	Palette (21)	Original (12) + Palette (21)	Original (12) + Key Image Features (24)	Original (12) + Palette (21) + Key Image Features (24)	PCA (100)	Original (12) + All Image Features (360)	Original (12) + All Image Features (360) + Palette (21)
N_Features	**12**	**21**	**33**	**36**	**57**	**100**	**372**	**393**
RMSE	22.80	23.33	22.80	22.15	22.15	21.87	21.44	21.44
RMSE ADJ	**20.07**	**20.12**	**20.07**	**19.93**	**19.93**	**19.35**	**19.14**	**19.14**
Accuracy	28.70	28.70	28.70	29.26	29.26	29.82	29.32	29.32
F1_score	17.25	12.80	17.25	19.81	19.81	21.88	22.30	22.30
Hamming Loss	71.30	71.30	71.30	70.74	70.74	70.12	70.68	70.68

Table 6: Model Results - Adjusted RMSE - Logistic Regression Model (Best Model)

Final Recommendation

In conclusion, the most stable and best performing model is Logistic Regression using 372 features (12 Original Features plus 360 Image Features) with a RMSE_adj of 19.14. Its main configuration is:

- C = 4
- solver="newton-cg“
- multi_class="auto“
- Penalty = l2
- max_iter=400

Note: using np.random.seed(0), 80/20 split for training/development datasets.

Our recommendation is as follows (presented in table below):

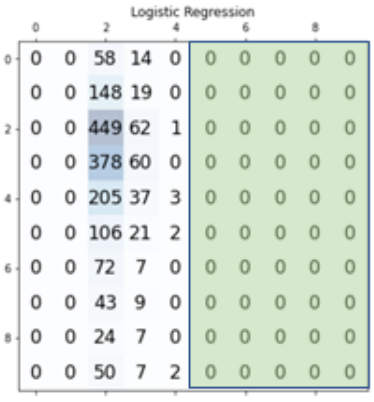
1. **For best performance as measured by RMSE**: Logistic Regression with 372 features (Original + All Image Features)
2. **For adequate performance as measured by RMSE with faster time**: Logistic Regression with top 100 Principal Components

Description	Original (12)	Palette (21)	Original (12) + Palette (21)	Original (12) + Key Image Features (24)	Original (12) + Palette (21) + Key Image Features (24)	PCA (100)	Original (12) + All Image Features (360)	Original (12) + All Image Features (360) + Palette (21)
N_Features	**12**	**21**	**33**	**36**	**57**	**100**	**372**	**393**
RMSE	22.80	23.33	22.80	22.15	22.15	21.87	21.44	21.44
RMSE ADJ	**20.07**	**20.12**	**20.07**	**19.93**	**19.93**	**19.35**	**19.14**	**19.14**
Accuracy	28.70	28.70	28.70	29.26	29.26	29.82	29.32	29.32
F1_score	17.25	12.80	17.25	19.81	19.81	21.88	22.30	22.30
Hamming Loss	71.30	71.30	71.30	70.74	70.74	70.12	70.68	70.68

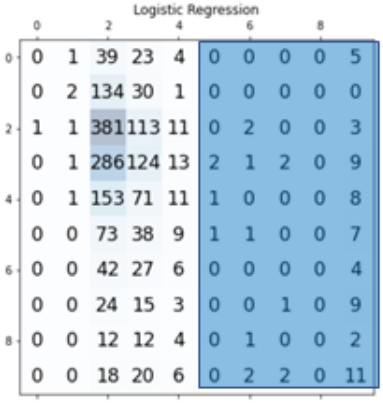
.

The Confusion Matrix showshow the suggested models improved their ability in correctly classifying images with a score greater or equal to 5 versus models using the 12 original features.

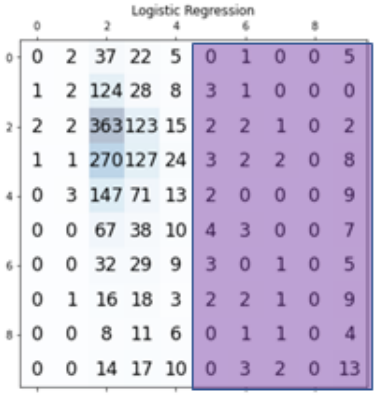
Using Original 12 Features



Using top 100 PC



Using 372 Features



Constraints

There were some constraints within this project. It was observed that the original dataset had duplicate images with varying popularity scores. Additionally, the popularity scoring did not seem to be consistent among similar images. For example, for two similar images with the same attributes (features), one would have a popularity scores of 100 while the second had a score of one. Last, the image quality generally tended to skew towards lower scores yielding an average

popularity score of 38.26. Of all the images, 78.5% of the images had a popularity score less than or equal to 50.

Standards

In order to better process images and to test a wide range of models, many Python Libraries were used. The following list includes all the libraries used in this project:

Package	Version
numpy	1.19.5
pandas	1.3.3
scipy	1.7.1
scikit-learn	1.0
matplotlib	3.4.3
opencv-python	4.5.4.60
scikit-image	0.18.3

Limitations of the Study

From the onset of this project, it became clear that the dataset was limited as detailed in the constraints section previously. In essence, the dataset represents individual performance of web pages with a single picture on them. There are many factors that can affect web page performance beyond the image found on the page, even for pet adoption photograph metadata. Additional metadata such as pet location or time of day of the posting could have been included by the project sponsors to add more features for a more in depth analysis of how popularity score was impacted.

Future Work

This competition is the most recent in a series related to improving pet images to increase pet adoptions by the project sponsor. Future work will most likely include deeper analysis of datasets with richer metadata or features. For our project team, we plan to continue improving models through experimentation to win the competition. At the very least, our team members will consider pet adoption from shelters given the realization of how many pets need homes!

Project Notebook/Git Repository

Our final project notebooks can be found at this link: https://github.com/chaukap/UCPupeley.git

References

(PDF) the appraisal of facial beauty is rapid but not mandatory. ResearchGate. (n.d.). Retrieved December 4, 2021, from https://www.researchgate.net/publication/5263130_The_appraisal_of_facial_beauty_is_rapid_but_not_mandatory.

Pet statistics. ASPCA. (n.d.). Retrieved December 1, 2021, from https://www.aspca.org/helping-people-pets/shelter-intake-and-surrender/pet-statistics.

Petfinder.my - Pawpularity Contest. Kaggle. (n.d.). Retrieved December 7, 2021, from https://www.kaggle.com/c/petfinder-pawpularity-score.

Sklearn.metrics.f1_score. scikit. (n.d.). Retrieved December 3, 2021, from https://scikit-learn.org/stable/modules/generated/sklearn.metrics.f1_score.html.

Sklearn.Metrocs.mean_squared_error. scikit. (n.d.). Retrieved December 2, 2021, from https://scikit-learn.org/stable/modules/generated/sklearn.metrics.mean_squared_error.html.

Tensorflow Hub. Tensorflow hub. (n.d.). Retrieved December 5, 2021, from https://tfhub.dev/google/imagenet/resnet_v2_152/classification/5.