The Pawpularity Context

Team name; Bucket of Data

Presented by
Sukhwinder Singh Kaur, 1818430
Zhong Xingyu, 1820539
Ahmed Mohmed elzamarany, 1823917
Pujit Golchha, 1823413
Sreehari Pradeep Kumar, 1837310
Ashish Udai Mandayam,1822720

Submitted to
Data Mining 1 team
Prof. Dr.Tobias Weller
University of Mannheim

December 2021

Contents

1 Problem Statement **1**

2 Data Understanding **1**

3 Image Pre-Processing **2**

4 Algorithms **3**

4.1 Experimentation with the ML Algorithms 4

4.1.1 CNNs 5

4.1.2 Transfer Learning 7

5 Evaluation Results **8**

6 Conclusion **8**

List of Figures

1 Re-sized images 3

2 Top 5 Best features 5

3 Convolutional Neural Networks (CNNs)[1] 6

4 CNNs model 7

5 duplicates 9

List of Tables

1 Evaluation Measures on different models 9

List of Abbreviations

CNNs Convolutional Neural Networks. i, 5, 6, 7, 8

RMSE Root Mean Square Error. 1, 5, 6, 8, 9

1 Problem Statement

A picture is worth a thousand words. But did you know a picture can save a thousand lives? Millions of stray animals suffer on the streets or are euthanized in shelters every day around the world. You might expect pets with attractive photos to generate more interest and be adopted faster. But what makes a good picture? With the help of data science, you may be able to accurately determine a pet photo's appeal and even suggest improvements to give these rescue animals a higher chance of loving homes.

In this project we will focus on the prediction of the chances of a pet being adopted beforehand, which is referred to as the Pawpularity score. Currently, this is part of a competition in Kaggle. [2]

That will be done applying different tools and algorithms from Machine Learning to the data, which is given in two different formats, one part already preprocessed and the other which needs to be processed.

As a way to measure and compare the results from the different applied techniques, we will use the Root Mean Square Error (RMSE). So we will try to reduce these errors by modifying the default parameters of the algorithms and applying further techniques such as cross validation or feature selection.

2 Data Understanding

In the project we are going to deal with two types of data. On the one side, we have the raw pictures provided in the competition website, a total number of 9920 images.

On the other hand, we will deal with the metadata, which doesn't need any type of special treatment and can be directly used for the techniques. The metadata is in the cleanest possible format available, and hence no Data preprocessing was required. In these files, each image is labeled with a unique id. Furthermore, there are 12 features represented in a binary format, 1 meaning "Yes" and 0 meaning "No". Their description is as follows:[3]

1. Focus - Pet stands out against uncluttered background, not too close / far.
2. Eyes - 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 significant portion of 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", which is the Target Variable, 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[3]. This data set has been split into train and test data set with a ratio of 90:10 i.e. the Train set with 8920x12 and the Test set with 992x12.

3 Image Pre-Processing

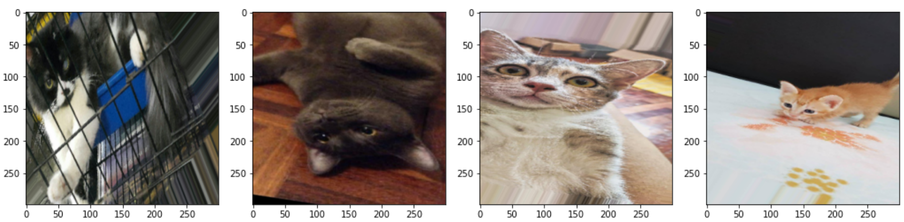
When we want to use the images, we have to treat them conveniently, in order to extract the best information. For the image processing, we use the so-called ImageDataGenerator from the keras module. The ImageDataGenerator helps us to stream the data as it might not be feasible or efficient to store all the images in memory. We have tried reading and storing images in arrays, without using generators, but because the images are very large, we run out of memory all the time.

The ImageDataGenerator has parameters which help us to resize the images, as well as augment the images in real-time, while the model is still training. Some of the modifications or changes that can be done are:

- Zoom the image.
- Rotation.

- Brightness
- Shifts.
- Random Flips.

Some images generated are shown in figures 1.



(a) (b) (c) (d)

Figure 1: Re-sized images

Once we are done with the transformation, we are able to start using the data to train our model, by generating images in batches, instead of reading everything at once which only leads to memory instabilities.

4 Algorithms

In this project we have used multiple algorithms, with respect to deep learning for the images data-set and machine learning for the metadata data-set. Deep Learning algorithms have been used for the image data-set such as convolution Neural Network (CNN) and other pre-trained models such as ”InceptionResNetV2”, ”NasnetLarge” and ”EfficientNetB1” have been used. Supervised machine learning algorithms have been used for the metadata data-set, especially regression models such as linear regression, decision tree regressor, support vector regressor, random forest regressor LassoLARS.

Linear Regression: It’s a form of regression model that draws a straight line through all of the data points based on their relationship. This line can be used to forecast values in the future. In the lecture, this was also explained.

Support Vector Machine/Regression: It is a form of regression model for predicting discrete values. The SVR, unlike other regression models, aims to fit the best line inside a threshold value, rather than minimizing the error between the real and projected value. This was also discussed in detail during the lecture.

Decision Tree Regression: It's a form of regression model that's used to forecast recurring values. It detects an object's characteristics and trains a model in the form of a tree to anticipate future data and create meaningful continuous output.

Random Forest Regression: A random forest is an ensemble technique that uses several decision trees and a technique called Bootstrap and aggregation, sometimes known as bagging, to solve both regression and classification problems. Instead of depending on individual decision trees, the main idea is to aggregate numerous decision trees to determine the final outcome. As a fundamental learning model, random Forest uses several decision trees.

LassoLARS: It is a combination of 2 different regression algorithms. Least Angle Regression (LARS) gives a response by a linear combination of factors for high-dimensional data. It has something to do with forward step-wise regression. The most correlated variable is chosen in each step in an equiangular direction between the two predictors using this procedure. LassoLars is a lasso model-based implementation of the LARS algorithm. Lasso regression is a sort of regularized linear regression with an L1 penalty that is widely used. This causes the coefficients for input variables that don't contribute much to the prediction task to diminish[4].

Here the said models are experimented with the complete metadata data-set and Metadata with Feature Selection

4.1 Experimentation with the ML Algorithms

For a few ML Algorithms, "RandomizedSearchCV" has been used to find the best "Hyperparameters", instead of manual "hyperparameter" tuning. "RandomSearchCV" and "GridSearchCV" both have the same goal: to identify the optimum parameters for improvement of the ML models. However, in "RandomSearchCV" not all parameters are checked. Instead, the search is randomized, with all other parameters remaining constant while the ones under test are changeable. Also, feature selection was performed to remove unwanted and unimportant features. This actually helped in improving the accuracy of the models and reducing its complexity. This helped in the reduction of Over-fitting of the models.

In the Feature Selection process, "SelectKBest" and Chi^2 are used. Chi^2 Score weeds out irrelevant features and helps select the best and important features for the prediction. The table 2 the Top 5 features with the best possible score

	feature	Score
4	Action	110.521692
10	Info	110.037115
6	Group	107.661366
9	Occlusion	106.320074
7	Collage	98.043067

Figure 2: Top 5 Best features

After selecting these features, the rest of the features are dropped and a separate Data-Frame was created to test out the above said models. A decrease in the RMSE score was observed after dropping the features with less score. The results can be observed in the Evaluation section 5.

4.1.1 CNNs

CNNs are a specialized kind of neural network for processing data that have a known grid-like topology, for example images[5].

The network employs a mathematical operation called convolution. The convolution it's nothing but an element-wise multiplication between a size patch of the input image and a Filter or Kernel, which is then summed up to get a single value. The filter used is a two dimensional array, which is smaller than the input image and it is intentional, as it allows the same filter to be multiplied to different parts of the input. The filter is applied systematically to each fixed size patch of the image, from left to right and then from top to button. This process is shown in the figure 3.

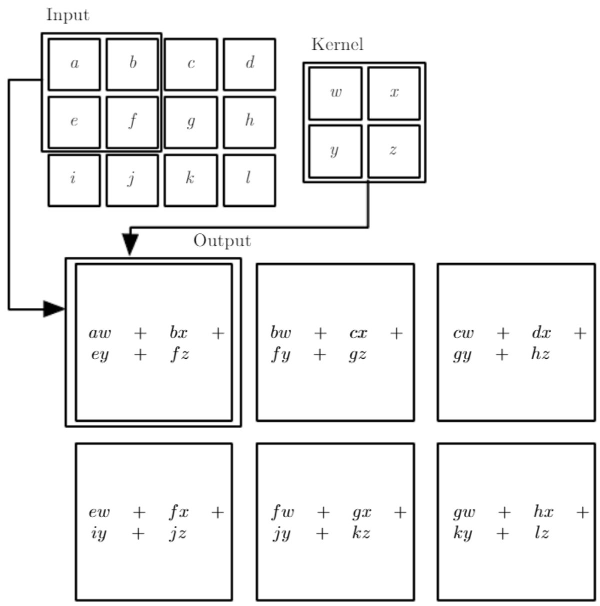


Figure 3: CNNs[1]

Convolutions layers are typically followed by pooling layers which downsize the feature resolutions. For a given size of pooling window, pooling takes either the average or the maximum value of the region.

Our CNN architecture consists of three convolution layers, followed by a max pooling layer and a final dense layer, which has a linear activation function, because the Pawpularity score is a continuous variable. The input images used in our model are generated using ”ImageDataGenerators”, images are randomly rotated by up to 45 degrees, horizontally flipped, height and width shifted, and zoomed by up to 15%.These processes are applied only to training data to improve generality of the model, and new images are generated for each batch. Our CNNs model has achieved a RMSE of 20.01 and has a structure as shown in figure 4.

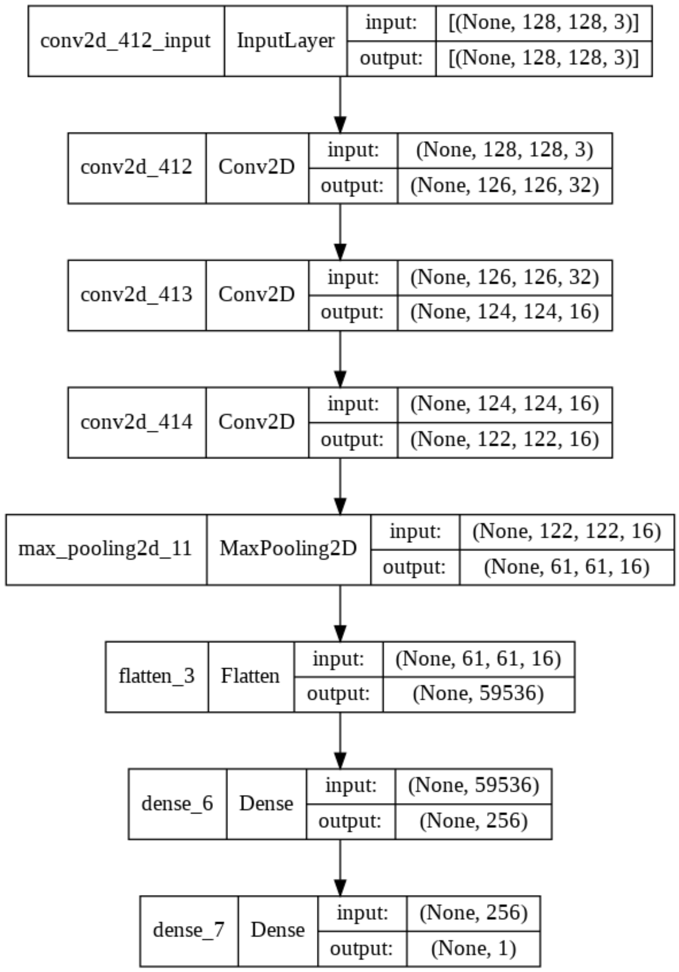


Figure 4: CNNs model

4.1.2 Transfer Learning

Transfer learning allows one to store knowledge gained in solving problems in one domain and apply to relevant tasks in other areas. In our application, we use pre-computed networks such as ”InceptionResNetV2”,” NasnetLarge” and ”EfficientNetB1” as feature extractors. These models are those which have obtained the best accuracy on the ”ImageNet data-set”. The ImageNet data-set contains 1.2 million images with 1000 classes. We have used the pre-computed weights of these models along with some extra layers to predict the Pawpularity score of the image. The last ten layers of these pre-trained models have also been trained according to our data-set, while the weights of the other layers have been frozen. Due to memory restrictions we were not able to train all the layers from the models [6] [7]. The best results are given by the "EfficientNetB1", which has an RMSE of 19.96, while the other models have an RMSE greater than 21.

"EfficientNetB1" achieves better results because they have a better accuracy than the other pre-trained models on the "ImageNet dataset". They are 8.4 times smaller and 6.1 faster than the best models. As compared to the CNNs we trained, the pre-trained models are giving almost the same results, with no improvements at all.

Another option we considered was using the metadata along with the images,and then train the neural network to see if our results improve. We also created new features such as the Clustering label and the Principal Components, and merged them to the metadata. The RMSE increased using this approach, and hence we concluded that the metadata was not adding information to our model.

5 Evaluation Results

With respect to the Supervised Machine Learning Regression Algorithms, the Kaggle Competition suggested using the RMSE as a metric to find out the performance of the model's accuracy in predicting the Pawpularity Score. The measurement shows how accurate our forecasts are and how far off they are from the actual data. The discrepancy between the values predicted by a model and their actual values is measured using this metric. Table 1 shows the RMSE for the different approaches discussed in 5. The results are very similar, but in general the RMSE is better when using feature selection.

6 Conclusion

Although using various techniques and approaches, the value of RMSE didn't drop than 19. After getting these results the data were examined manually and duplicates of photos were found, but does have different pawpularity score. Figure 5 shows an example for this inconsistency. We conclude that the main problem is in the data set itself. The reason for this inconsistency might be due to the score being user based and can vary from one person to another. For future work, changing the type of data for this use case is recommended. Traffic data for the website can be more useful.

MODELS/ALGORITHMS	**RMSE w. F. S.**	**RMSE not w. F.S.**
Linear Regression	20.29	19.62
Support Vector Regressor(with RSCV)	20.81	-
Support Vector Regressor	20.79	19.98
Decision Tree	20.27	19.62
Random Forest	20.25	19.63
Random Forest(with RSCV)	20.25	-
LassoLARS	20.27	19.63
Cnn	20.04	-
Transfer learning	19.96	-
Cnn with Metadata	20.61	-

Table 1: Evaluation Measures on different models

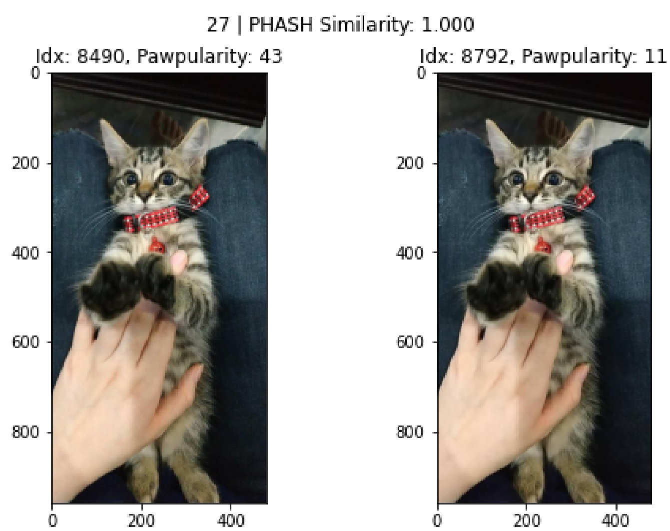


Figure 5: duplicates

References

[1] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.

[2] The pawpularity contest. https://www.kaggle.com/c/petfinder-pawpularity-score/overview.

[3] The data set of pawpularity contest. https://www.kaggle.com/c/petfinder-pawpularity-score/data.

[4] Jason Brownle. How to develop lasso regression models in python. https://machinelearningmastery.com/lasso-regression-with-python/, 2020.

[5] Amruta Kadlaskar. Image classification using convolutional neural network with python. https://www.analyticsvidhya.com/blog/2021/06/image-classification-using-convolutional-neural-network-with-python/, 2021.

[6] Tejan Irla. Transfer learning using inception-v3 for image classification. https://medium.com/analytics-vidhya/transfer-learning-using-inception-v3-for-image-classification-867004112 2019.

[7] Thuwarakesh Murallie. Transfer learning: The highest leverage deep learning skill you can learn. https://towardsdatascience.com/transfer-learning-in-deep-learning-641089950f5d, 2021.