Pawpularity Contest

Predicting Popularity Of Shelter Pet Images

CS354 : Computational Intelligence Lab
Code can be found at:
https://github.com/rohanjha04/Pawpularity-CI/

Group 16

Jha Rohan	210002041
Nishkarsh Luthra	210001045
Hrishesh Sharma	210003037

Course Instructor : Dr. Aruna Tiwari

Contents

Section 1 : Understanding the Contest

1.1 : The Challenge

PROBLEM STATEMENT

Predicting the popularity of shelter pet photos based on image and metadata is a significant challenge in the realm of Computer Vision and Machine Learning.

IMPORTANCE

This tool can significantly impact adoption rates and contribute to reducing the number of stray animals.

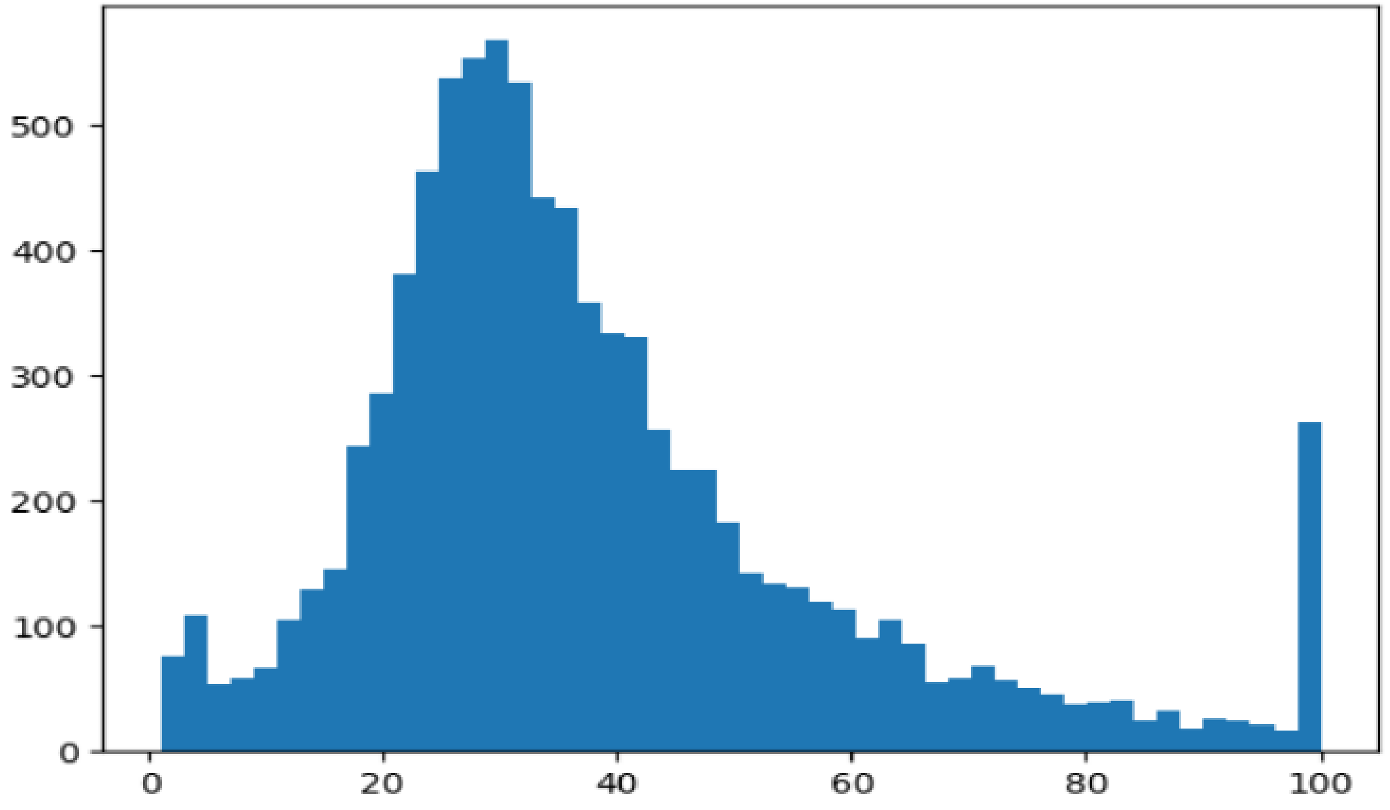
1.2 : Dataset Overview

SKEWED DATA

More than 7000 values have a score of less than 50, only 1807 samples have score more than 50 in the training dataset. For less than 50, maximum samples lie in the range of 20-40.

DATASET SPLIT

The dataset of 9912 images was split into training, validation, and testing sets for model development.



Input Data Skewness

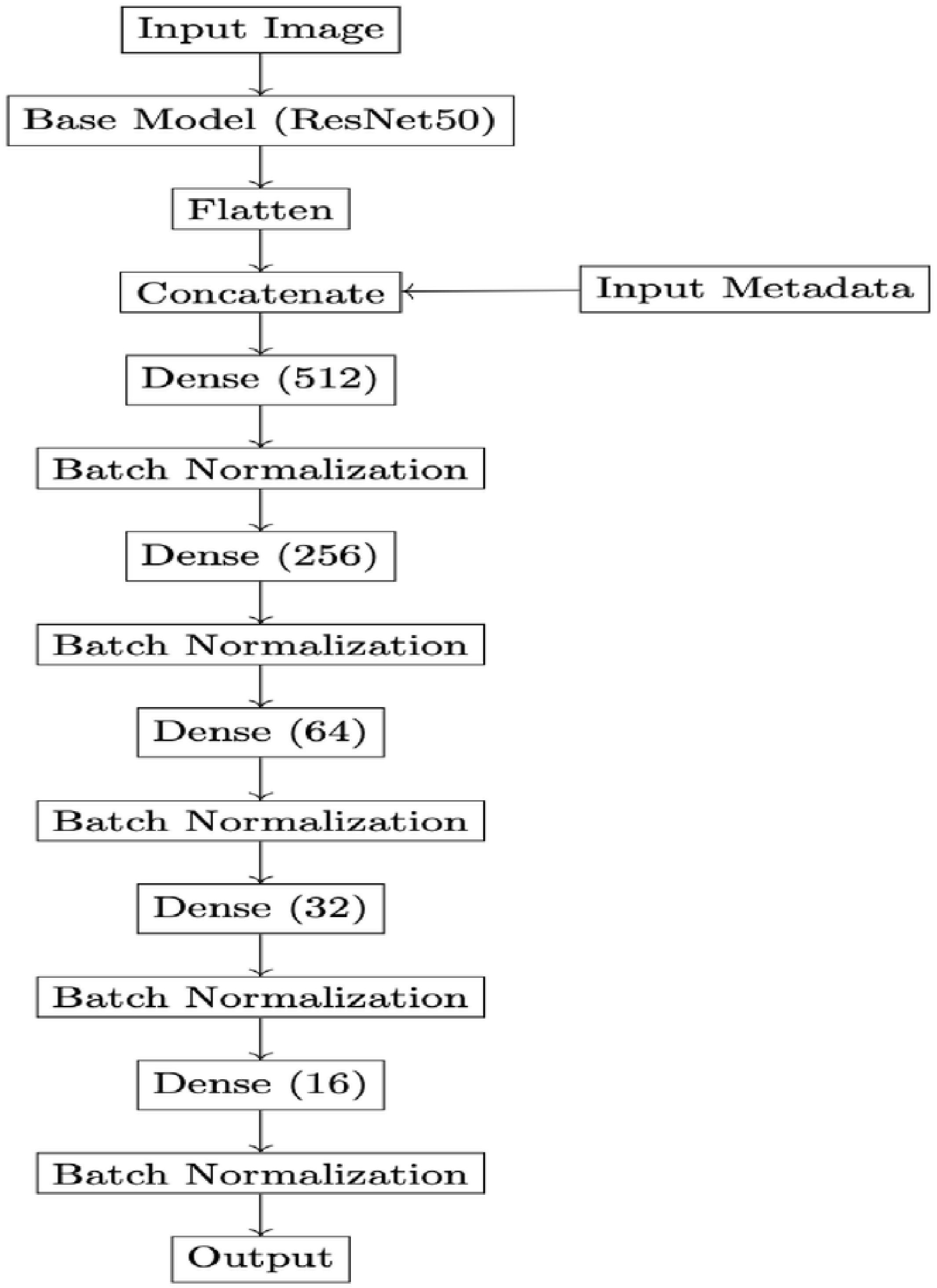
1.3 : Architecture Overview

RESNET50

We passed images through the data loader, then utilised the ResNet50 architecture for feature extraction and subsequent processing with fully connected dense layers. Then, use the sigmoid activation function to get output between 0-and 1.

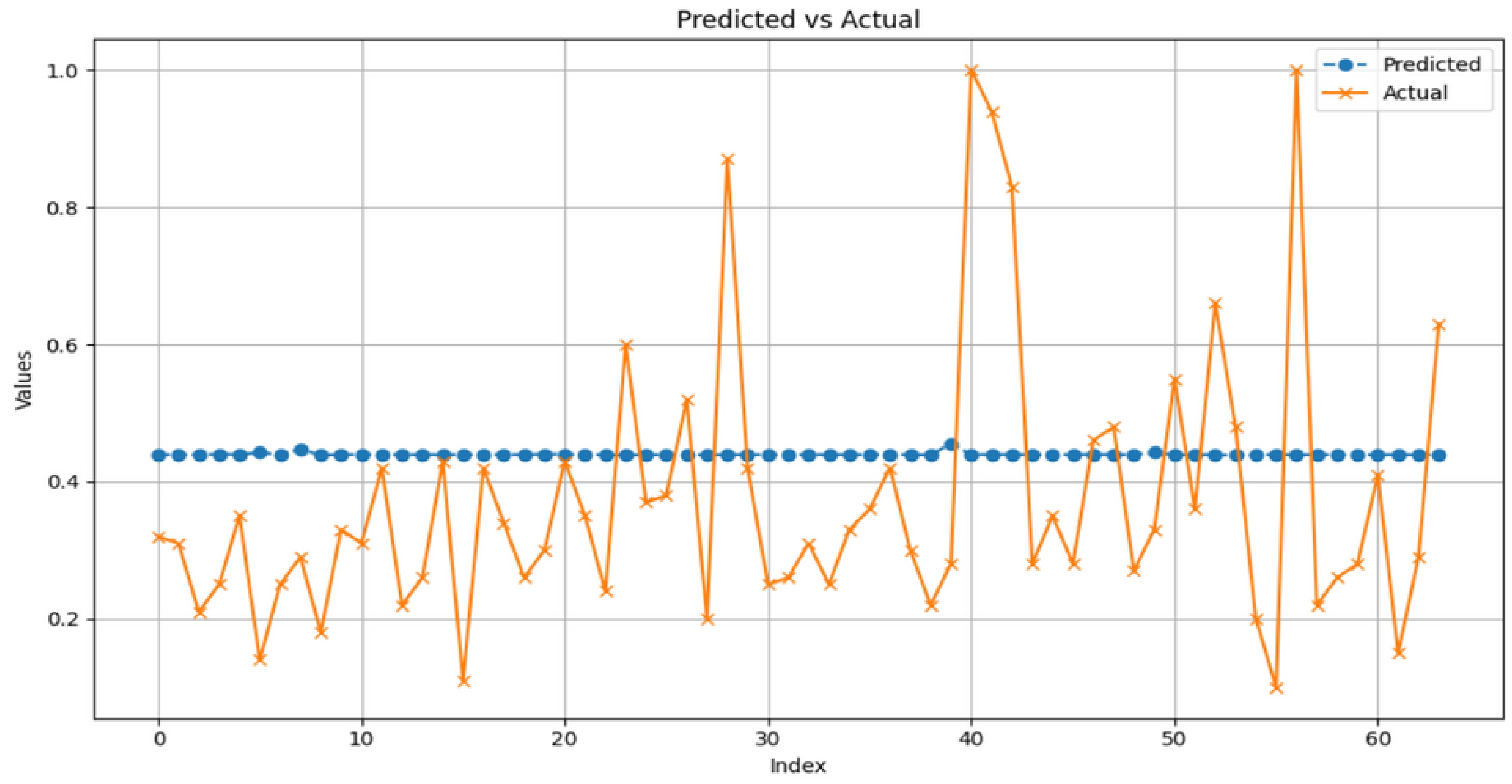
Metadata & Performance Measure

The metadata provided in the dataset was concatenated with output features from Resnet50. Mean sqaured error was used to measure performance. RMSE score was used for prediction accuracy.



Architecture

2. Experimentation & Optimisation



Predicted vs Actual

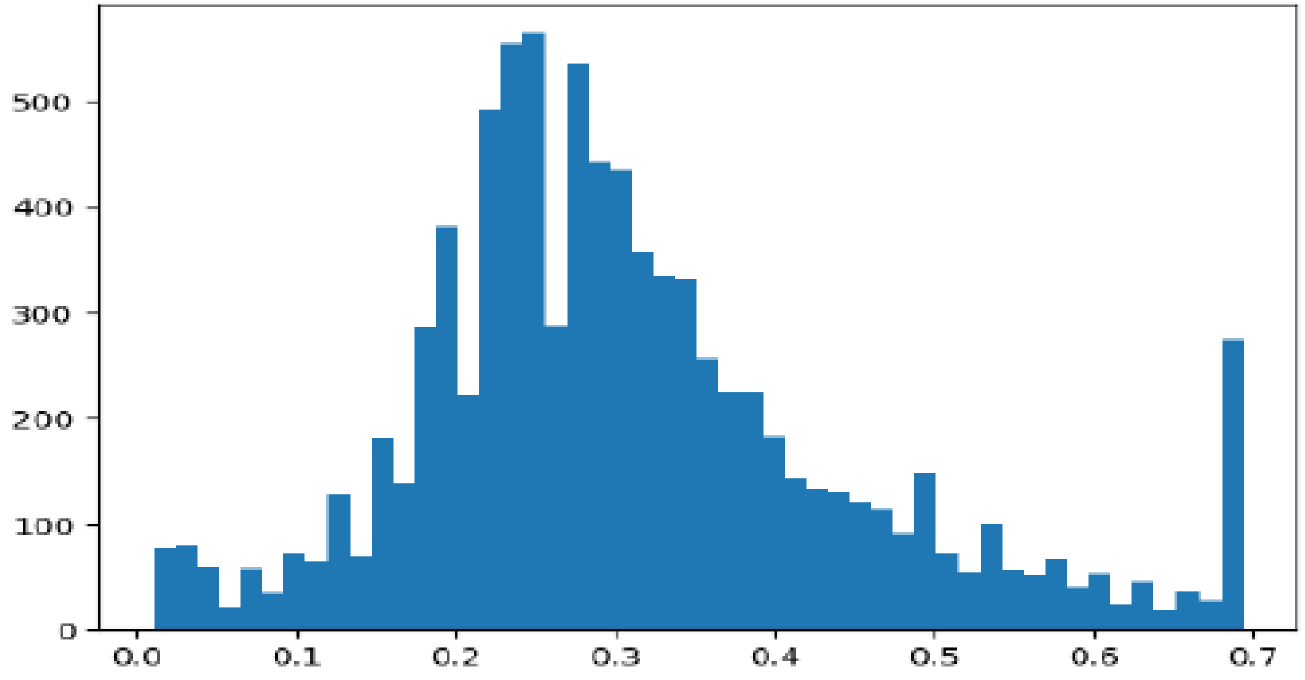
1. Try to train Resnet

Initially we made all layers trainable that were 50 of resnet and 10 deep layers but due to GPU and RAM limitations could only train 10 epochs. we got good RMSE but model as whole was useless as it was predicting values in range 0.43 - 0.45.

2. Experimentation & Optimisation

2. Using Log1p for scaling

We scaled the data using log1p, but even after scaling, most data was in the range of 0.2 to 0.4. Thus, the results were similar to Expt 1.



Data after scaling

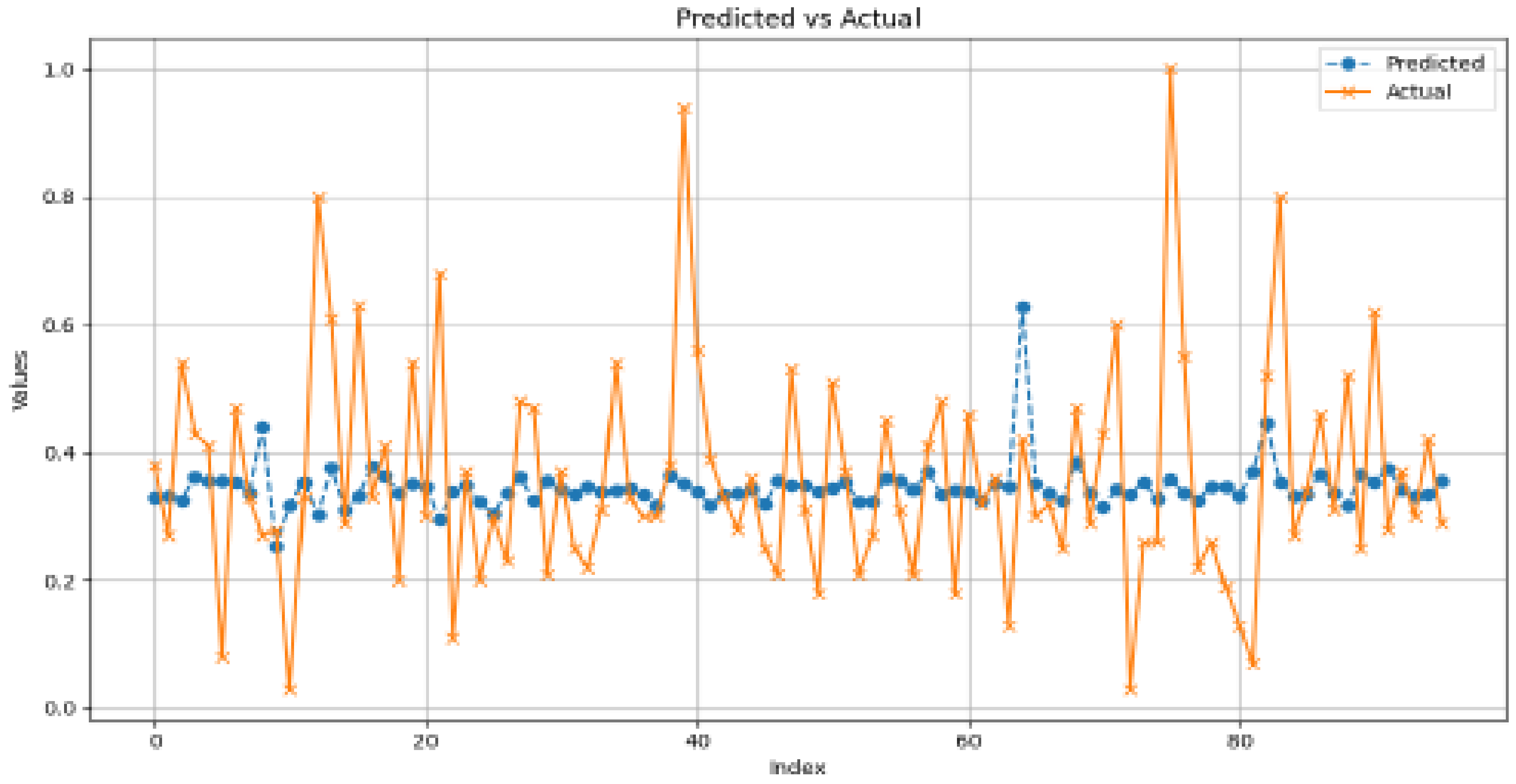
3. Using ML Methods

Tried Using ML approaches like XG Regressor, SVRegressor, etc. However, the main problem was that these models did not support partial fit, and our data was massive. So, all data could not be loaded into memory to perform "fit" at once. So, we did not get good results.

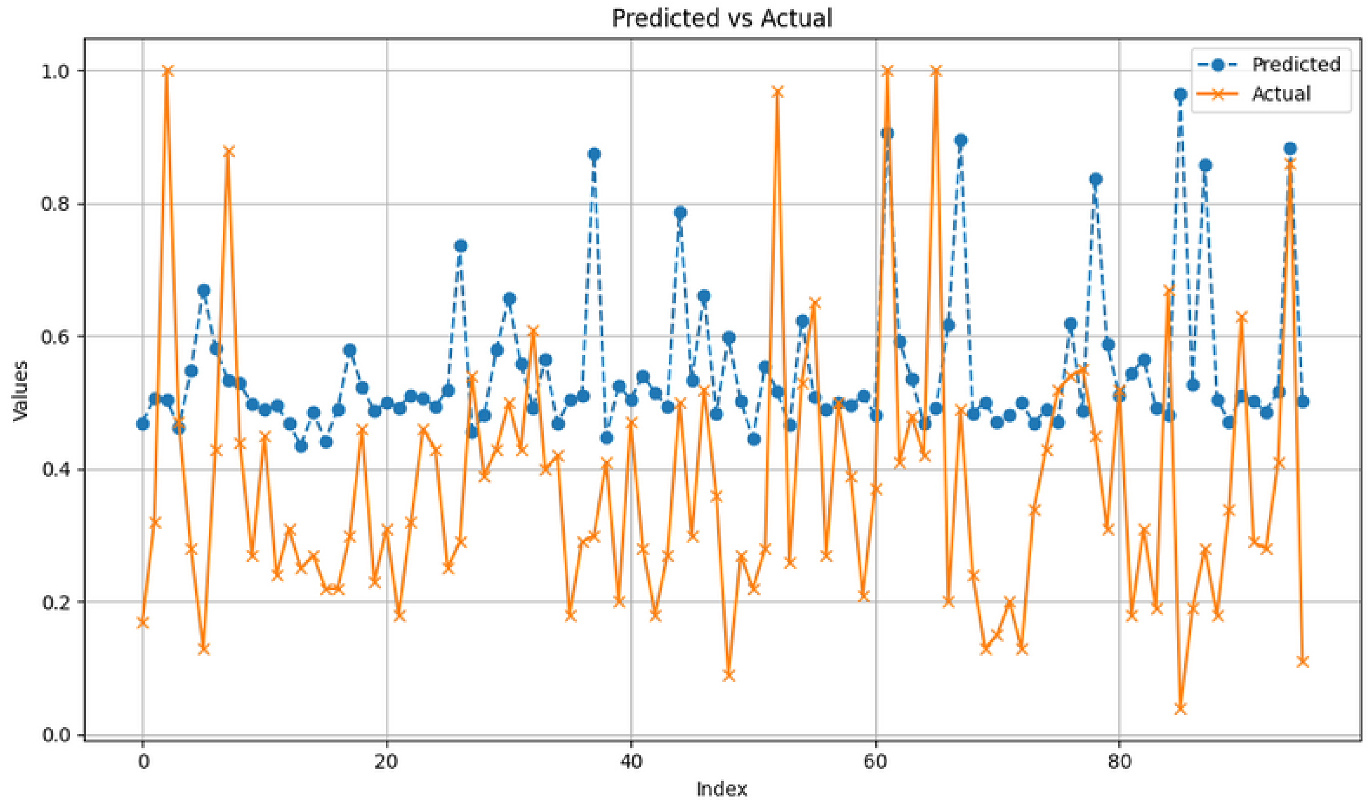
2.4 : Using weighted back prop.

Using exponential loss function for minority class

Inspired by the concept of DLINEX, we decided to multiply an exponential function of e^(5*y) while backpropagating the losses. However, this made the model biased towards higher values. To fix this we decided to lower the factor to 3*y instead of 5*y. However, this, too, provided a poor fit, and the model still struggled to predict the extreme values correctly.



Using e^3*y for labels>0.5

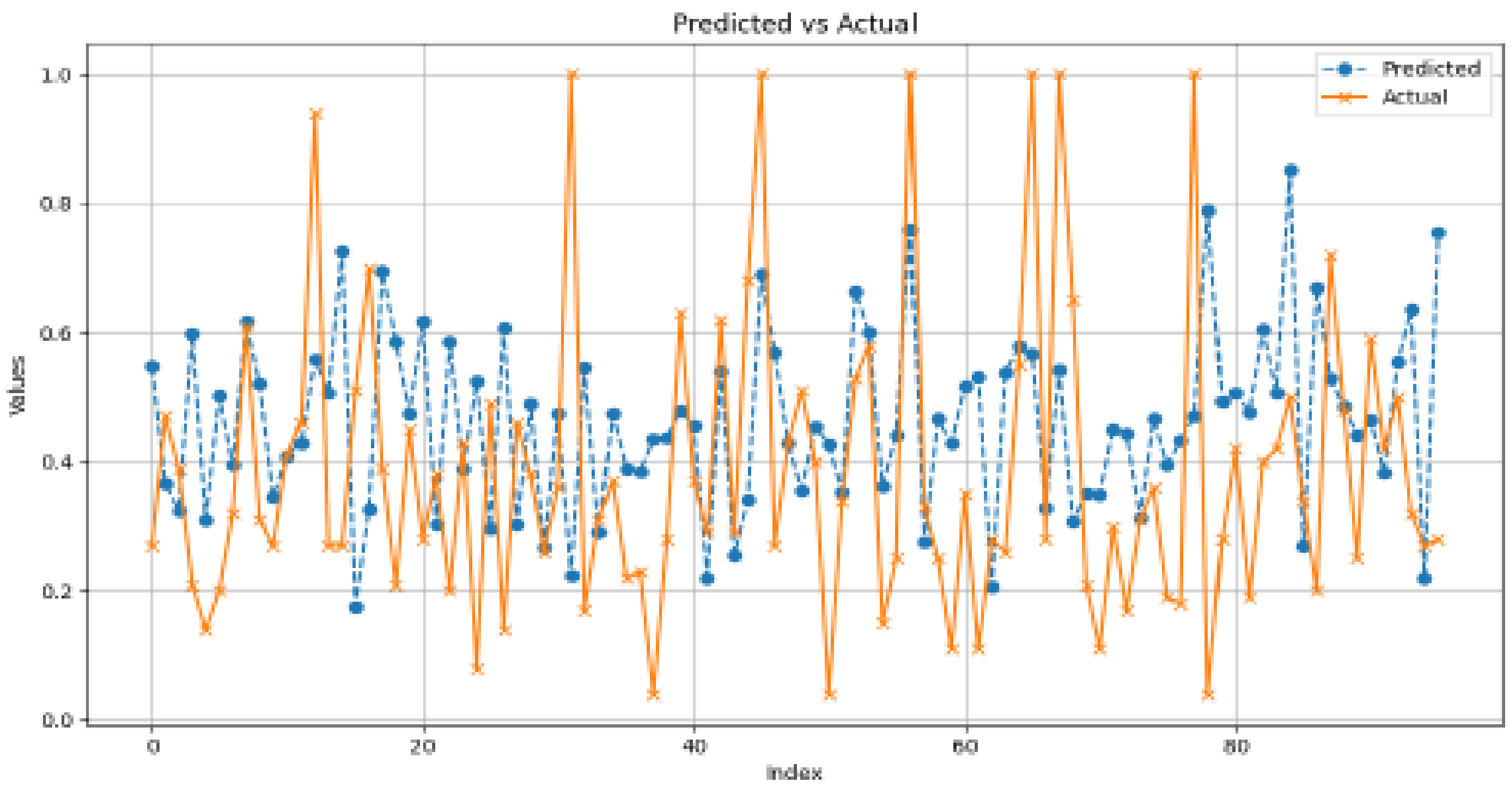


Using e^5*y for labels>0.5

2.4 : Using weighted back prop.

Using Discontinuos Weighting Mechanism

We experimented with a special way of giving importance to different parts of the data because it wasn't evenly distributed. Most of the data fell between 0.2 and 0.4, so we split that range into three parts: 0 to 0.2, 0.2 to 0.5, and 0.5 to 1.0. We assigned different weights to each part: 1 for the first two ranges and 2.5 for the third. Then, we tweaked the weights a bit: 1 for ranges 1 and 2, 2.5 for the rest, and 2.5 for range 1, 1 for range 2, and 2 for range 3. After trying these adjustments, the third setup produced the best results, with RMSE scores of **0.19746 on the training dataset and 0.19759 on the testing dataset.**



Using 2.5 weight for 0 to 0.2, 1 for 0.2 - 0.5 and 2 for rest

CONCLUSION

- **Diverse Experimentation:** We performed various experiments to address the challenge of predicting the "pawpularity" score of images of cats and dogs. From training ResNet50 directly to employing machine learning methods and weighted backpropagation, the team explored multiple avenues to optimize the model and generalised performance.

- **Data Analysis and Preprocessing:** An in-depth dataset analysis revealed its highly skewed nature, with most samples scoring below 50. This insight led to exploring techniques such as data scaling using log1p to normalize the distribution of labels and enhance model generalization.

- **Architecture Design:** We employed a sophisticated architecture involving feature extraction using ResNet50, concatenation with metadata, and dense layers for prediction. The design was tailored to accommodate the unique characteristics of the dataset and maximize model performance within resource constraints.

- **Dynamic Learning Strategies:** Dynamic learning rate scheduling with exponential decay was employed to prevent overfitting and enhance model adaptability. This strategy allowed for fine-tuning the learning process and optimization of model convergence.

THANK YOU

Group 16