Final Project Presentation

—

CS-UY 4563

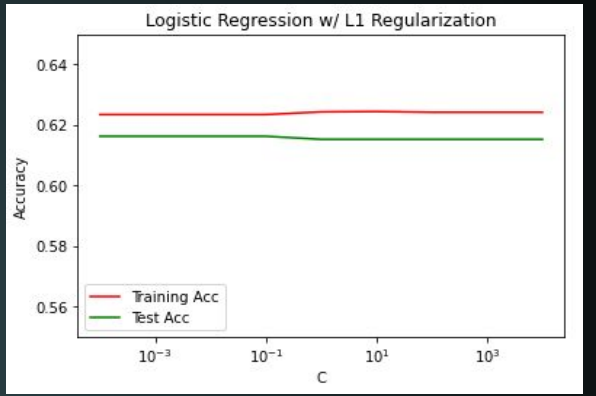
Nick Broome & Dan Be’eri Longman

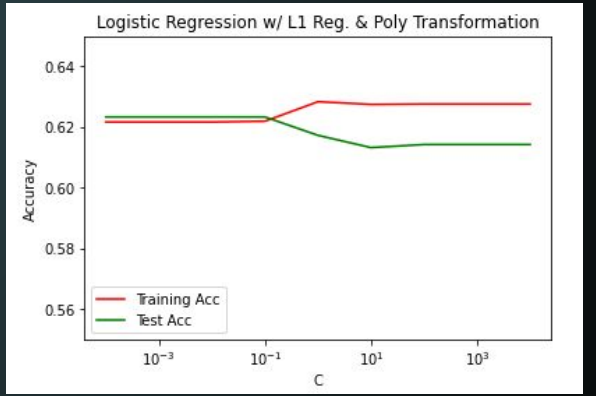
Introduction

- We utilized PetFinder.my's dataset containing 9912 adoption photos of various pets with metadata containing some descriptive feature of each image.
  - This includes thing such as how many eyes are visible, if the animal is wearing an accessory, etc.
- We implemented preprocessing for our Logistic Regression and SVM models.
  - This preprocessing mainly consisted of taking the "pawpularity" feature and changing into from a continuous value to categorical.
  - We measured the mean and all pets that were above were assigned "1" and all below it were assigned "-1".

Logistic Regression

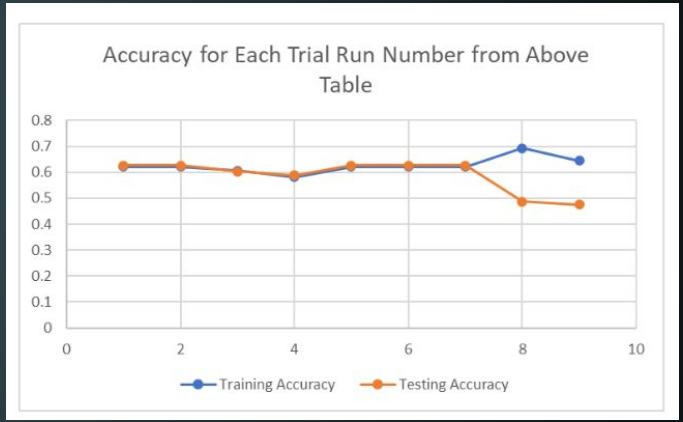
- 13 different features were utilized, classified as either 1 or 0 except for ‘pawpularity’, which was 1 or -1
- Ran a dry test with no penalty terms, and then incorporated L1 and L2 regression.
  - Utilized 9 C-values.
- After running those tests, we applied polynomial transformation to our features.
- Our tests with L1 and L2 regression were slightly more accurate and consistent than the initial test.
- Our tests with polynomial transformation applied yielded high accuracy at first, but also diverged and indicated that there was an issue with bias or variance at higher C-values.
  - Our highest accuracy was at C-values 0.0001 - 0.1 with L1 regularization and polynomial transformation.





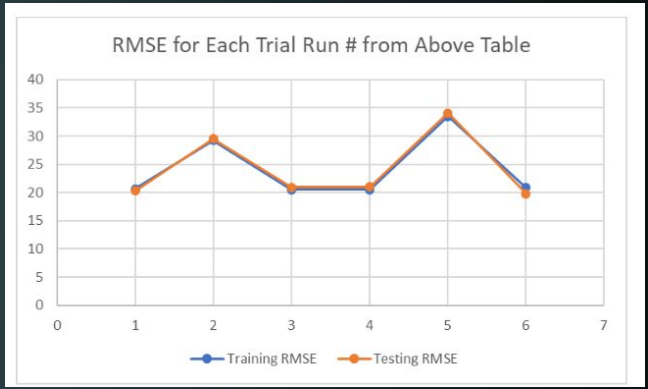
SVM

- Initially we examined the p-values, but we ended up using the Histogram of Oriented Gradients (HOG) algorithm for analysis.
  - Its major benefit is that it examines images and extracts features that may not be visible to us as humans.
- Our training set accuracy rose 10% when C=10,000, but our test set accuracy dropped below 50% after implementing the HOG algorithm.
- This might be due to the HOG algorithm working “too well” and extracting noise from the image, leading to an overfitting due to the irrelevant data.



Neural Network

- No significant overfitting occurred as our training and testing RMSE were extremely close in value
- However, there seems to be underfitting as our RMSE scores seem high.
- Inconsistent results and overflow after a few iterations suggest either image metadata is not detailed enough to perform prediction task, or neural network is not a fitting algorithm for the given problem.



Conclusion

- For Logistic Regression, trying different feature transformations or increasing the sample size are ways to potentially get accuracy.
- Our SVM implementation did yield a relatively high accuracy for the one situation for our training dataset, but also yielded a vastly lower accuracy for the testing dataset. We could try other regularization techniques potentially to achieve this.
- Neural networks seemed to perform the worse, most likely due to how manually entered metadata seemed to be inflexible for this model.
- As such, we would advise staying away from Neural Networks, but Logistic Regression and SVM models can both be suitable, but will only be about 62% accurate.

Works Cited

1. https://towardsdatascience.com/an-introduction-to-neural-networks-with-implementation-from-scratch-using-python-da4b6a45c05b
2. https://towardsdatascience.com/svm-implementation-from-scratch-python-2db2fc52e5c2
3. https://www.kaggle.com/competitions/petfinder-pawpularity-score