Machine Learning Engineer Nanodegree

Capstone Project

Rishabh Srivastava

January 6th, 2019

I. Definition

Project Overview

I propose to work on the Petfinder.my Adoption Prediction challenge on kaggle (https://www.kaggle.com/c/petfinder-adoption-prediction). PetFinder.my has been Malaysia's leading animal welfare platform since 2008, with a database of more than 150,000 animals.

Millions of stray animals suffer on the streets or are euthanized in shelters every day around the world. If homes can be found for them, many precious lives can be saved — and more happy families created. PetFinder collaborates closely with animal lovers, media, corporations, and global organizations to improve animal welfare.

Animal adoption rates are strongly correlated to the metadata associated with their online profiles, such as descriptive text and photo characteristics. As one example, PetFinder is currently experimenting with a simple AI tool called the Cuteness Meter, which ranks how cute a pet is based on qualities present in their photos.

In this competition you will be developing algorithms to predict the adoptability of pets - specifically, how quickly is a pet adopted? If successful, they will be adapted into AI tools that will guide shelters and rescuers around the world on improving their pet profiles' appeal, reducing animal suffering and euthanization.

Top participants may be invited to collaborate on implementing their solutions into AI tools for assessing and improving pet adoption performance, which will benefit global animal welfare.

In this competition I will be developing algorithms to predict the adoptability of pets - specifically, how quickly is a pet adopted? If successful, they will be adapted into AI tools that will guide shelters and rescuers around the world on improving their pet profiles' appeal, reducing animal suffering and euthanization. *Very Deep Convolutional Networks for Large-Scale Image Recognition* - Karen Simonyan, Andrew Zisserman (2014) is a paper where deep neural networks is used to classify images. My personal motivation comes from the fact that, if homes can be found for the pets, many precious lives can be saved — and more happy families created.

The Dataset consists of 2 sets of files. Training and Testing set. Every set primarily of 4 files.

1. Csv files: It has the following details. It has in all 24 parameters. PetID is the index of these files for cross reference for the next set of files. This also contains the AdoptionSpeed column containing values of 0, 1, 2, 3 and 4 each representing a day, week, month, 3 months and more than 3 months respectively. The training set contains 2.74%, 20.61%, 26.92%, 21.74% and 28% respectively for the 5 labels of AdoptionSpeed.
2. Images files: For every PetID, images are provided containing that pet image.
3. Metadata files: For every PetID, it contains google image api annotations.
4. Sentiment files: For every PetID, it contains textual sentiments attached.

The training set consists of 14994 samples of pets. The testing set doesn't contain the AdoptionSpeed label and comes with 23 labels. I'll split the training data into training and testing sets with a 80-20 split. For tuning the parameters I'll split the splitted training data into training and validation. I might also consider 5-fold cross validation. As we can see that the label 0 is less than 10% of the rest of the data, I'll use stratified sampling to split the data at every step to balance the labels.

Problem Statement

The problem at hand is to predict the rate at which a pet shall be adopted. There a 5 categories of adoption rates in which the prediction has to be done. For every pet, there are various parameters along with image and textual data are provided as training parameters. These parameters include details like color, breed, vaccinated status, health, etc. of the pet. The problem is a multi-class classification problem where the predicted class is the adoption speed for each pet.

I propose to use Multi-input Deep learning model. The 2 inputs I am definitely exploring include images and pet features. I am still exploring and if possible I'll try to integrate the metadata nd sentiments as the 3rd and the 4th inputs. The output will be a 5 output sigmoid layer providing probabilities for every class. For the images I'll be using a transfer learning model with Inception V3.

Metrics

We'll evaluate the models on these metrics: Confusion matrix, precision, recall and f1-score. These metrics will give us a fair idea of how the algorithm is performing on the datasets. We can directly compare these metrics with the benchmark implementations. F1 is a balanced metrics that takes both accuracy and precision into account.

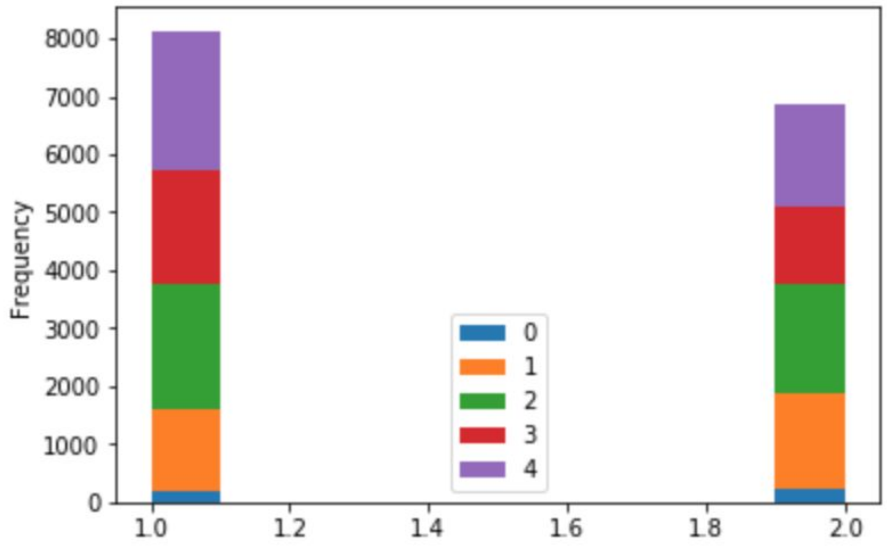
I have not added accuracy here as one of the labels is less than 10% of the rest of the labels individually. Hence, the classes are imbalanced and so accuracy as a metrics won't be correct to use.

II. Analysis

Data Exploration

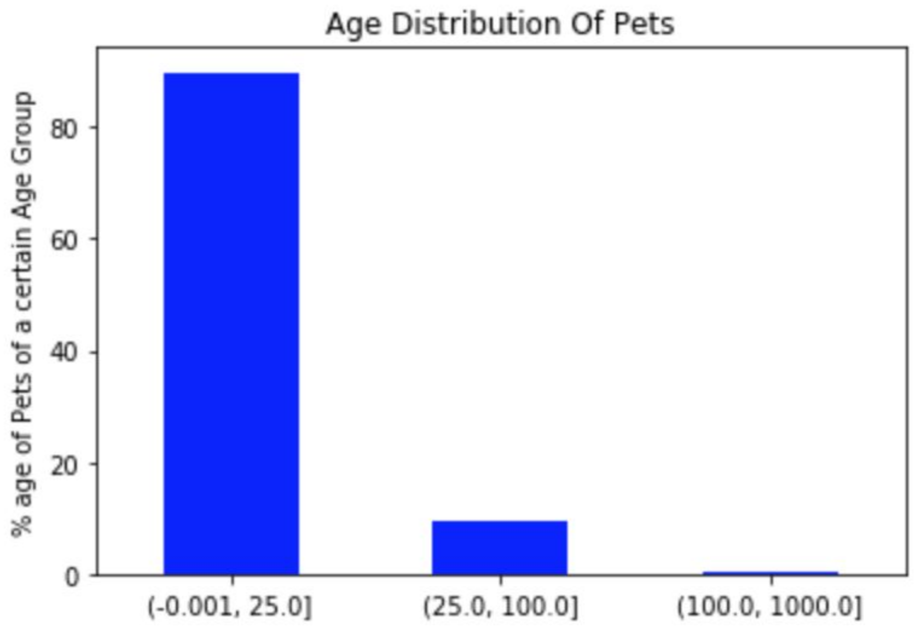
We start by analyzing the training data. The training data contains 14993 data points (rows). It has 23 parameters (columns) include target variable *“AdoptionSpeed”*. There are varied parameters including:

- **Categorical variables**:
  - *PetID*: It is the unique index for every pet entry. It is used to refer the images and sentiment from other data sources.
  - Personal Parameters: They determine the personal characteristics of a pet.
    - *Type:* Cat or Dog. There is no clear distinction as to if a cat or dog makes a difference in adoption pattern. Similar patterns were observed for other categorical parameters also.

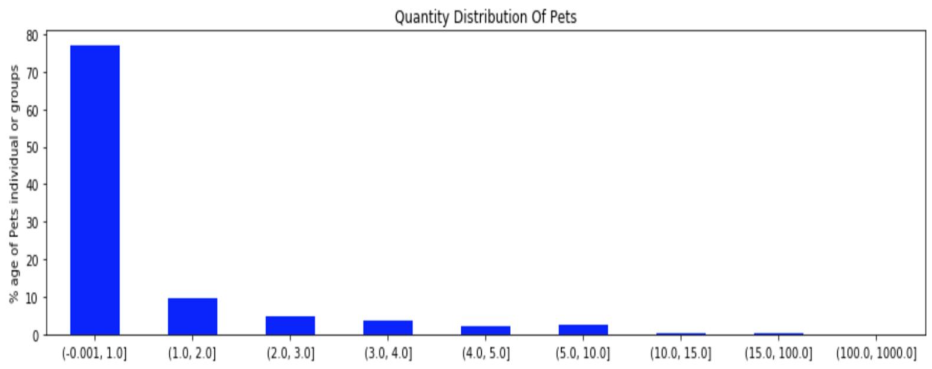


    - *Breed:* Breed1 (Primary Breed), Breed2 (Secondary Breed)
    - *Gender:* Male, Female, Mixed, if profile represents group of pets
  - Visual Parameters: They define the visual characteristic of a pet.
    - *Color:* Color1 (Primary Color), Color2 (Secondary Color), Color3 (Tertiary Color)

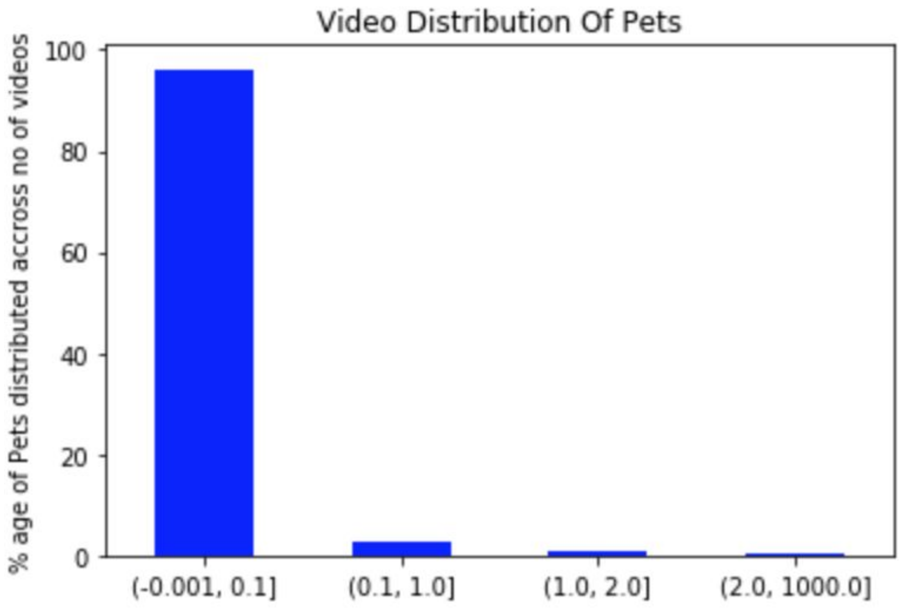
        - *MaturitySize:* Small, Medium, Large, Extra Large, Not Specified
        - *FurLength:* Short, Medium, Long, Not Specified
    - Health Parameters: These determine, how healthy the pet is or how safe the pet is.
        - *Vaccinated:* Yes, No, Not Sure
        - *Dewormed:* Yes, No, Not Sure
        - *Sterilized:* Yes, No, Not Sure
        - *Health:* Healthy, Minor Injury, Serious Injury, Not Specified
    - Other Parameters.
        - *State:* The state at which this pet is located.
        - *RescuerID:* The rescuer who found out/enlisted this pet.
- **Numeric parameters**:
    - Integer Parameters:
        - *Age:* Age of pet when listed, in months. Some analysis clearly shows that more than 92% of pets fall within 25 years age bracket. Very rarely cats and dogs live more than that.



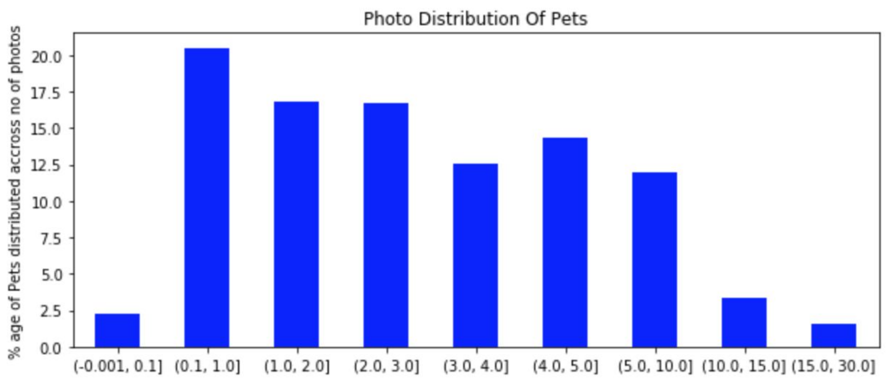
        - *Quantity:* Number of pets represented in profile. Profiles having 1 pet is 75% while profiles having less than 5 pets comprise nearly 99% of the data points.



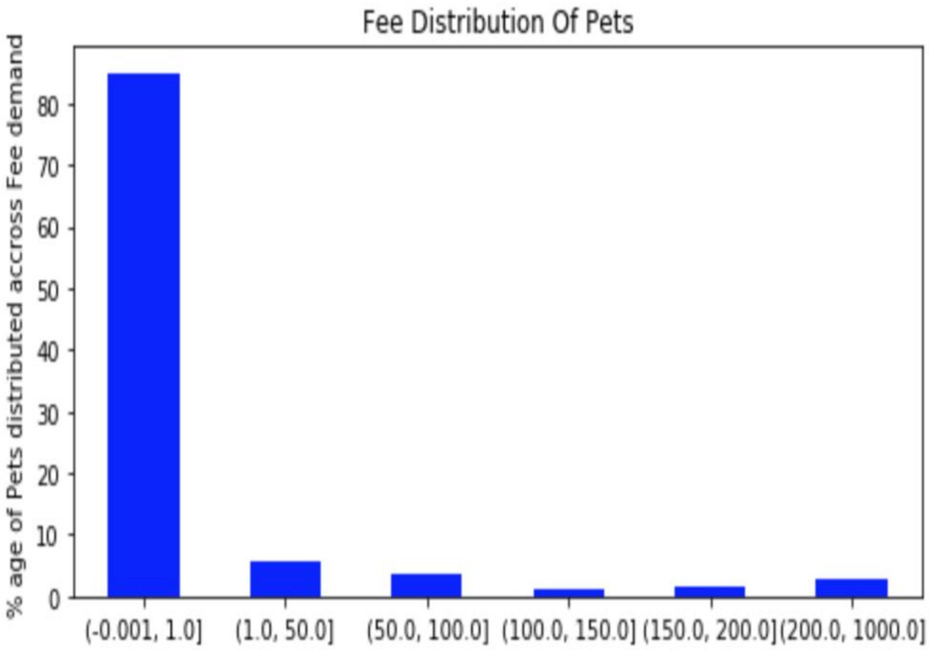
- *VideoAmt:* Total uploaded videos for this pet. Nearly no videos are uploaded for any pet.



- *PhotoAmt:* Total uploaded photos for this pet. This is a more balanced graph which shows that number of photos have varied throughout.

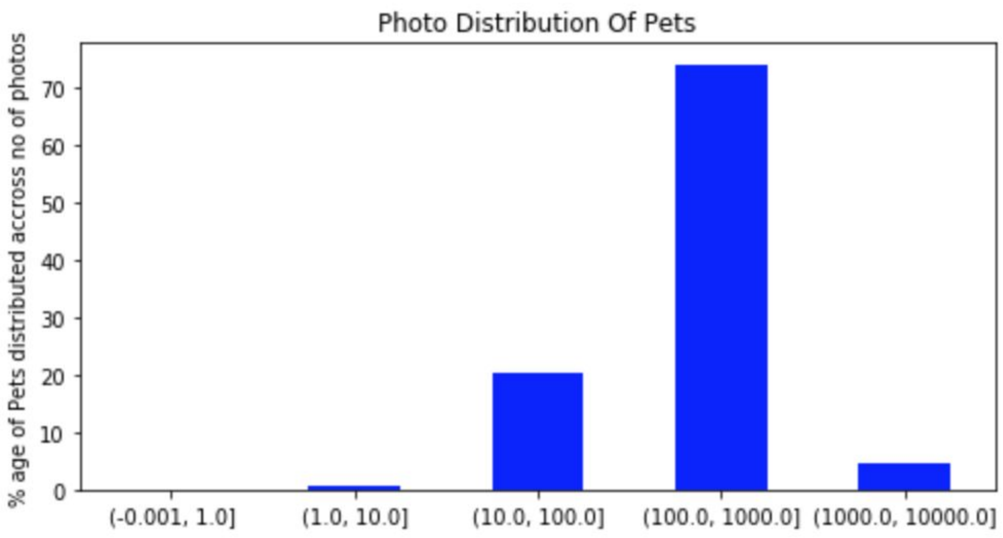


- Real Valued Parameters:
  - *Fee:* Adoption fee (0 = Free). This graph shows that nearly 90% of the pet adoptions are free of cost. Rest of the 10% is distributed evenly.



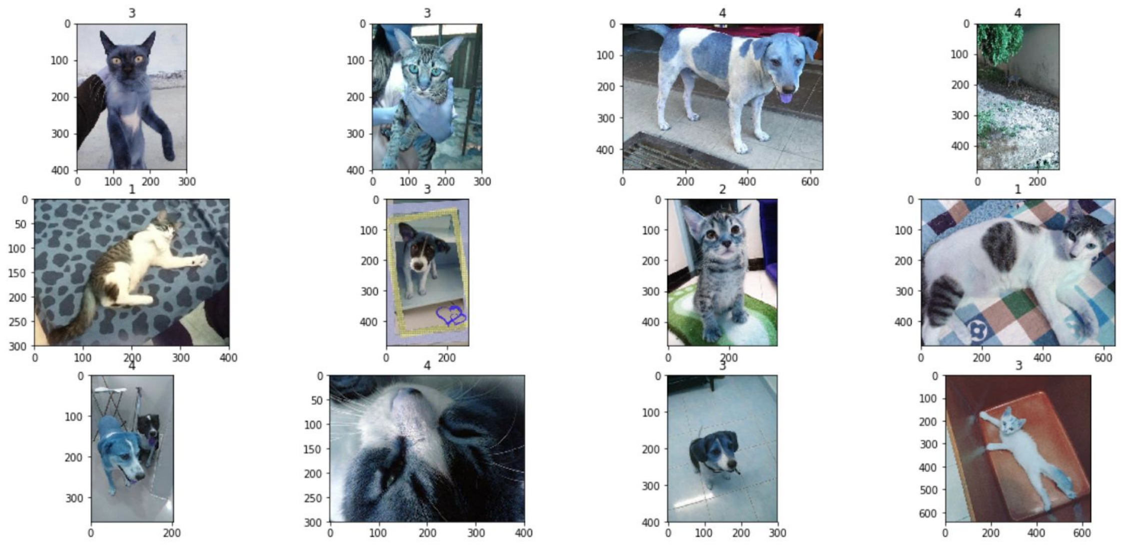
- **Textual Parameters**:
  - *Name:* Name of pet (NaN if not named). *Explanations of data given in the next section*.

- *Description:* Profile write-up for this pet. For quantifying this parameter, I tool the number of characters for every description. Some of the pets were really well described.



Apart from the csv file, there are other data sources as well:

- *Images:* 58311 images. 1 pet can have 0, 1 or more number of images. The dimensions of the images is not fixed



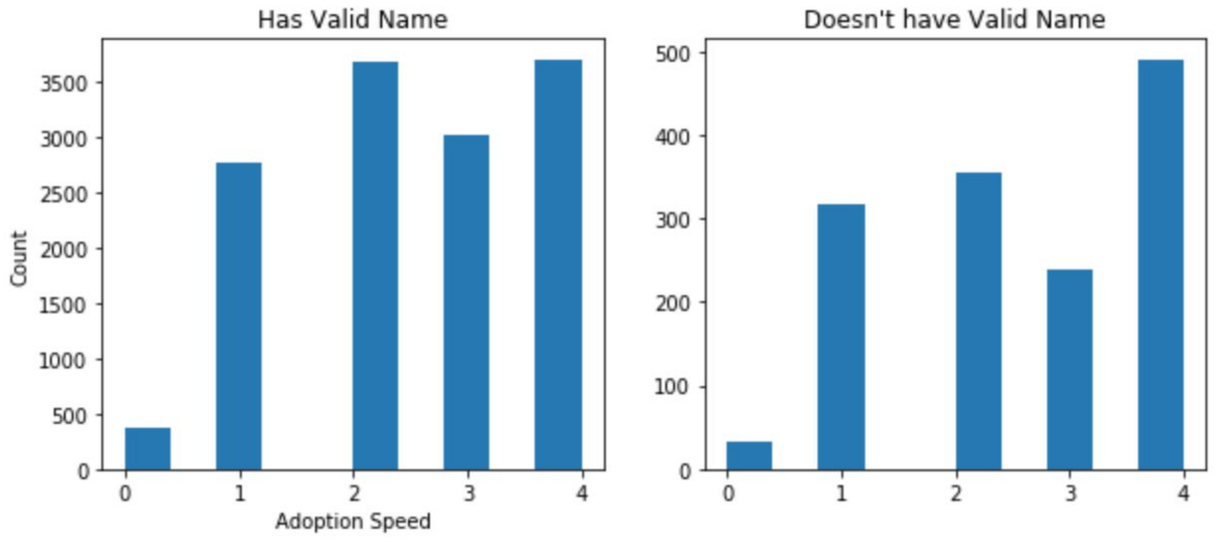
- *Image metadata:* They have run the images through Google's Vision API, providing analysis on Face Annotation, Label Annotation, Text Annotation and

Image Properties. We may optionally utilize this supplementary information for your image analysis.

- *Description Sentiment:* They have run each pet profile's description through Google's Natural Language API, providing analysis on sentiment and key entities.

Exploratory Visualization

I have taken the name data as a key data point. The visualization shows that pets who don’t have a valid name have a lower probability of being adopted than the ones with a valid name. To get this I first found if the text contained a lot of text “no” and listed all the data. I then studied the table and found out that instead of the text “no”, text “name” makes more sense to find out. Some examples include, “Not named yet”, “Unnamed”, “To be named by owner”. I also found that a lot of entries in this table were that of NaN.



In retrospection, both these observations, make a lot of sense. Both these were substantially backed by data as well.

- Filtering on “name”.
- And the importance of name for the adoptability as it gives a personalization to the pet.

Algorithms and Techniques

This is a multi-label classification problem. I have decided to use *GridSearch* and *Neural Networks* on csv and images as 4 separate models. All these work well on multi-class classification. All of these have a different approach towards a solution. There are a lot of categorical values, which gives me a strong institution that this type of dataset should be easily classifiable by these algorithms

Benchmark

For the benchmarking, I have taken a default DecisionTreeClassifier, trained it on the data set whose preprocessing will be explained later. This gives an F1 score of **37.71**%. All our models will be evaluated against this benchmark.

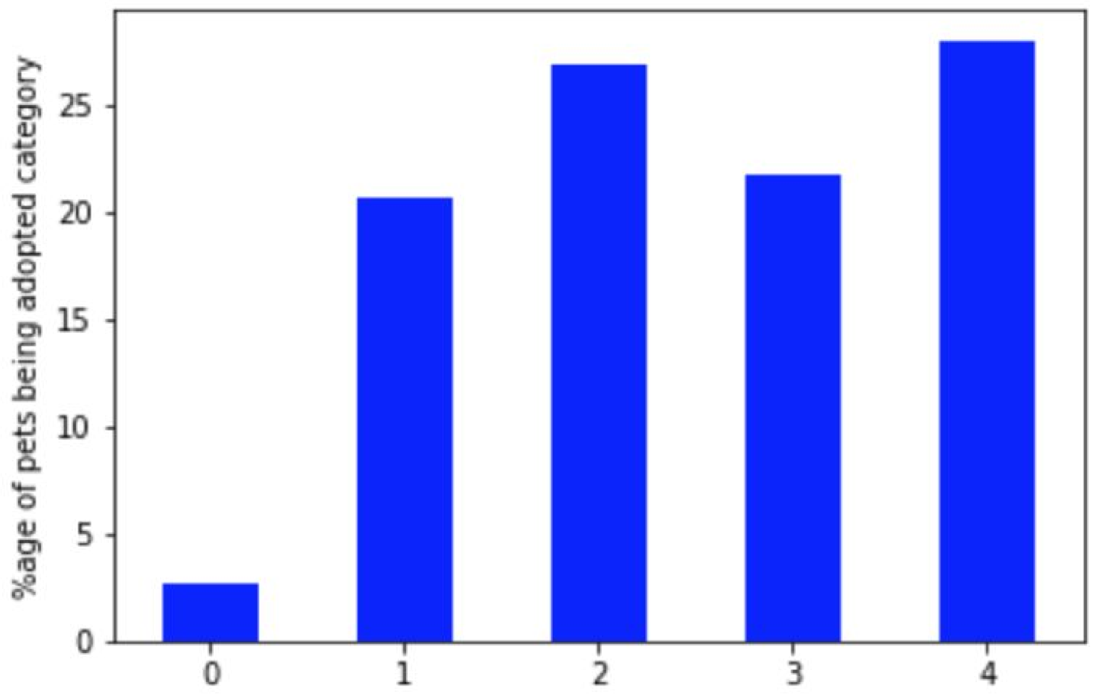
III. Methodology

Data Preprocessing

Whatever pre-processing I am doing, I have modularize the code in a way that it can be directly used by the test data as well. Preprocessing is divided into following sub-sections:

- **Reducing textual information to be used as numeric data**. I am not doing any detailed text analysis.
    - *Name:*
        - Intuition: A pet with a valid name has a higher rate of adoptability than one who does not have, as it gives more personal touch to the pet.
        - Process: I have extracted the information of whether a pet has a valid name or not and stored it in a separate column “HasValidName”. I have successfully changed this text field to a categorical counterpart. Then I remove the “Name column”
    - *Description:*
        - Intuition: A pet with a greater detail of description has a higher chance of adoptability, as the buyer can know the pet in a great detail.
        - Process: I have extracted the number of characters in the description text and stored it in a separate column “DescriptionCharLength”. I have successfully changed text field to a numerical counterpart.
- **Reducing categorical data to numeric data.** I have applied one hot encoding using get_dummies of pandas library. I am dropping the first column_value for every category.
- **Image Pre-processing.** This consists of 2 steps:
    - Scaling all images to 128x128 pixel size. This ensures a uniformity.
    - Image Augmentation as on an average only 4 images are available per pet. This doesn’t give us enough data.

- **Extracting Target values**: I found out the percentages of various classes. It clearly shows a non-even function. Hence, splitting the data needed to be stratified. For Neural network training I did a one hot encoding on the target varaibles as well. For DecisionTree and Grid Search, it was not needed.



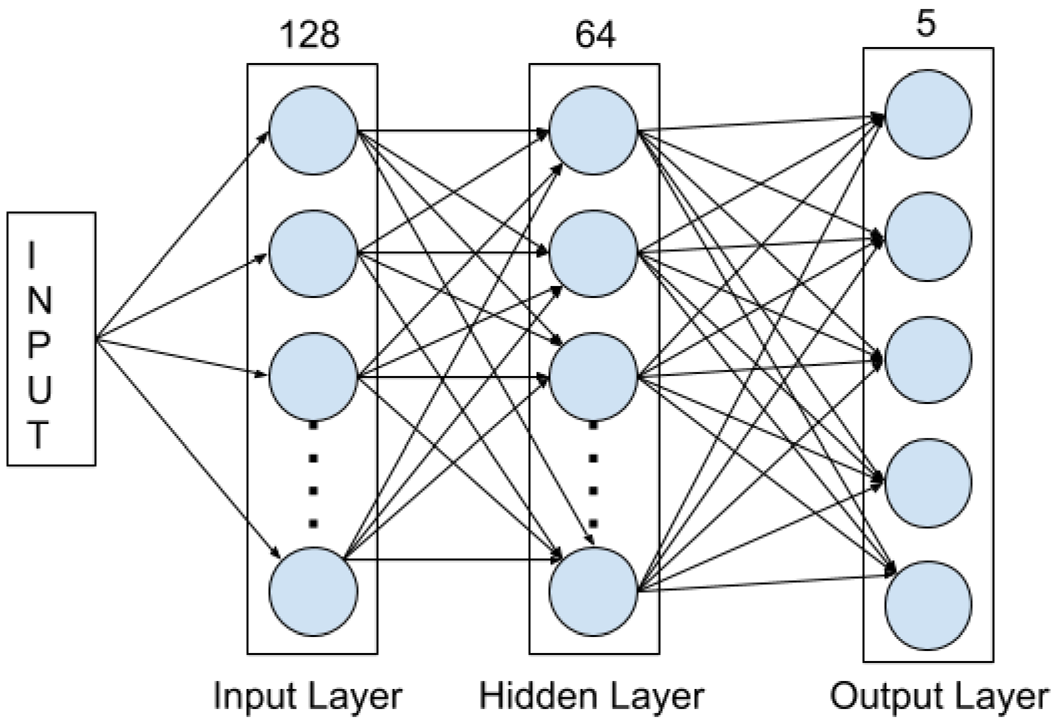
- **Splitting Data:** I have split the entire data using sklearn split method using the parameter stratify = true on a split size of 80% and 20% for training and testing data.

Implementation

- **GridSearch**: I have optimized AdaBoostClassifier with DecisionTreeClassifier in the grid search. I have taken the score as f_beta score where beta = 0.5 as scorer. I have taken the optimization of max_depth and min_samples_split for the DecisionTreeClassifier. I have taken the cross_validation as 3 as default. Post training, I have used the best prediction to predict the best classes for the test cases extracted above.

  I have started with 1, 10, 100 and 1000 as values for max_depth and 1, 10, 100 for min_samples_split.

- **ANN**: For training this artificial neural network, first I have defined a f1 score function which can be passed metrics to the neural network. I have then created the Neural Network Architecture with 1 hidden layer. And Input, and the hidden layer having a dropout of 0.1 and 0.3 respectively. At the end I have used softmax layer with 5 outputs for predicting the 5 classes. I have used categorical_crossentropy and adam as loss function and optimizer respectively,



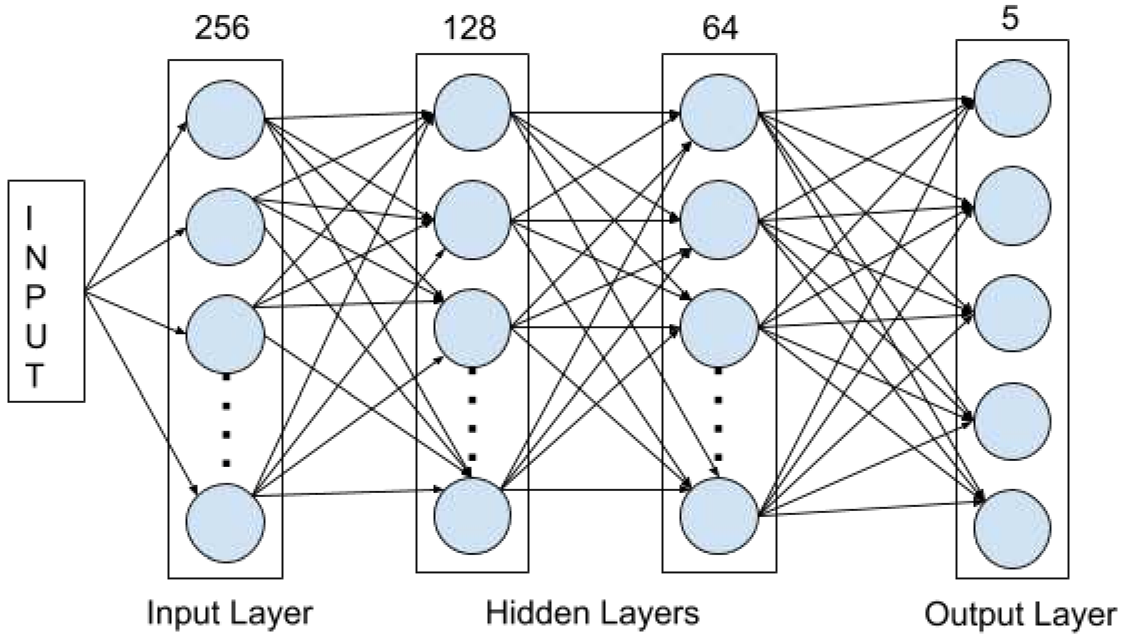
- **CNN:** I take *InceptionV3* pre trained model and train my images on them. Validation data is separated out for fitting the data. I use ModelCheckPoint to store the best model trained.

Refinement

- To optimize the performance, I realized that as I was not using any linear algorithm for classification, hence not dropping off any of the column category while doing the one hot encoding of a column might actually increase the performance of the algorithms. So I corrected that for all of the implementation.
- Once I trained the GridSearch on the aforementioned values, I realized that the best model actually chose 1000 as max depth which was my max value. So I

probed what the best values of GridSearch should be. After running the GridSearch for 2 more times, I realized that the best parameters to probe are in the range 1100-1300 for max depth and 15-30 for min_samples_split.

- The f1 score on the ANN is 0.39 whereas that of GridSearch was 0.41, so I tried inserting another input layer at the beginning, to come up with a better performing model.



IV. Results

Model Evaluation and Validation

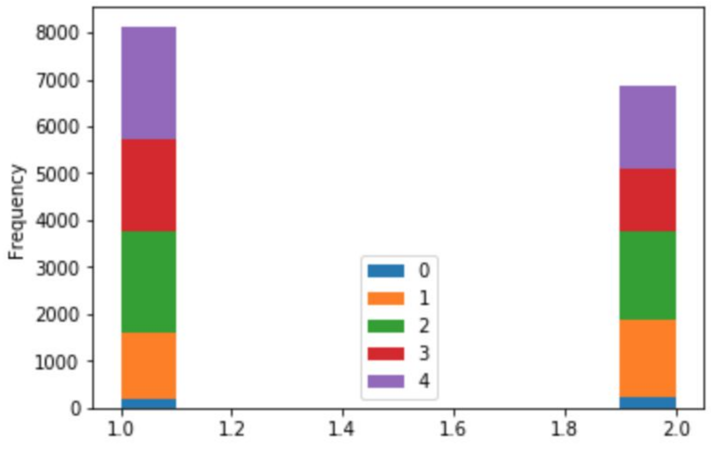
- GridSearch gives a F1 score of 44.08% on the test set that we created from the data split.
- ANN gives F1 score of 85.60% on the same test set. The precision and recall are 90.71 and 81.16% respectively.
- CNN gives F1 score of less than 1%. There is definitely some problem with the implementation of this algorithm.

Justification

DecisionTreeClassifier gives an F1 score of nearly 38%. GridSearch performs well, ANN performs extremely well and CNN performs poorly than the benchmark model. CNN model for some reason isn't able to learn the characteristics of the images and hence unable to train well. I need to dive deeper into what the issue with images might be.

V. Conclusion

Free-Form Visualization



I am picking one of the simplest graphs for this free-form visualization. When I look at this graph, I really cannot identify any correlation between type of pet and whether I can confidently say that any percentage of cats/dogs will be sold more or less than the other type. Yet the algorithms have somehow found a way to find a relation between these seemingly unrelated variables. What I am trying to get to is that its highly impossible for humans to work with such huge datasets and rule based systems might work in certain scenarios but as the problem area increases so will the need for these AI models increase.

Reflection

The project started with a blank slate where I had to start from deciding what problem to pick. I chose a very interesting and relevant problem of deciding the adoption speed of dogs. Then I studied the problem description to know that it was a classification problem. Having studied the course I am able to decide a lot of things on the fly. Post that I studied the data and figured there were multiple data sources. Then I had to decide a metrics on what I would want my results to be based on. I chose the metrics to be F1.

Then I analyzed the data sources 1 by 1. Starting from the csv files and understanding each and every column of that. What data it represents, what type of data it has, what can be the relevance of this data in predicting the result. I then explored the image data. What kind of images were there, were they unique for every pet or multiple photos were present, etc. I also glanced through the sentiment and metadata sources. Then I picked up up how name can affect the overall adoption of a pet. Then I described the benchmark for the project as Decision Tree Classifiers.

I then explained what all data preprocessing steps I have applied to the data columns including the text, categorical and image data. I also explained how I created the output data for various algorithms. Then I went on to explain the implementation of each of the algorithms in detail, including the architecture and initial parameters used. I then explained what tuning and refinement factors I applied to each of the algorithms to arrive at the final models.

At the end I explained which algorithm did better than the benchmark model and by how much. Post the project I think this is a very relevant projects for other industries as well.

The most closely related example is that of inventory management. Given a huge historical data and a need to predict which inventory to prioritize, this algorithm should be used.

Improvement

Frankly, there is a huge huge scope of improvement, that I'll too continue working on even after this project. Few of the areas include:

- Inclusion of sentiment and metadata is an area that is completely left out. I should have included them in the data columns
- There is a scope of adding multiple sources to Neural network trainings. Tabular data and images can go hand in hand as multi-modal neural network.
- Adding a few of our own layers after InceptionV3 model to have the model some space to learn on the training data separately.