Pet Adoption Analysis and Speed Prediction

Rajasvi Vinayak Sharma (A59012988)[1] and Preyaa Patel (A59005037)[2]

***Abstract*— Pet adoption is a very crucial step towards saving lives of stray animals and giving them new home. With new upcoming online pet adoption websites, we have enough data to analyze the factors responsible for early adoption. In this project, we have explored data from pet profiles on PetFinder.my to assess the factors contributing to adoption speed. Furthermore, we enrich and correlate these profile features with sentiments extracted from profile description to find patterns and predict pet adoption speed.**

I. INTRODUCTION

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.my has been Malaysia's leading animal welfare platform since 2008, with a database of more than 150,000 animals. 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 pet characteristics like age, condition, breed etc. In this project we want to understand how quickly pets get adopted and what factors appeal to viewers.

II. DATASET

We have taken PetFinder.my Adoption profiles data from Kaggle.[1] This database contains text, tabular, and image data; specifically, we will be carrying out an analysis on tabular data which contains data from more than 14000 profiles. We will perform our primary analysis on tabular data, which has various features from the Petfinder.my online pet profiles like pet's condition, location, gender, age, breed, etc., which contribute to the adoption speed.

Primary data fields considered are Type, Name, Age, Breed, Gender, Color, Health, Quantity, Fee. Furthermore we will be using sentiment data which contains key entities and sentiments scores extracted from each profile description.

Our target field is Adoption Speed which takes following values based on how quick the adoption was for a profile:
0 - Adopted on the same day as it was listed.
1 - Adopted between 1 and 7 days (1st week) after being listed.
2 - Pet was adopted between 8 and 30 days (1st month) after being listed.
3 - Adopted between 31 and 90 days (2nd 3rd month) after being listed.
4 - No adoption after 100 days of being listed.
(There are no pets in this dataset that waited between 90 and 100 days).

III. DATA ANALYSIS & VISUALIZATION

We aim to analyze the dependence of adoption speed with some of the features in a probabilistic manner to understand the data better and come up with meaningful observations. Some relevant results of our exploratory data analysis are as follows:

A. Age vs Adoption Speed

1) Overall: It can be seen from Fig 1. that the pets with adoption speed 4 are older as compared to other speeds. Younger pets have a low chance of not getting adopted after 100 days (Adoption Speed 4).

2) Cats vs Dogs: We cut into the subspace of Type: **Dogs, Cats** and visualized their age data separately to make comparisons. As seen in the

[1]Rajasvi Vinayak Sharma, A59012988, Department of Electrical and Computer Engineering

[2]Preyaa Patel, A59005037, Department of Electrical and Computer Engineering

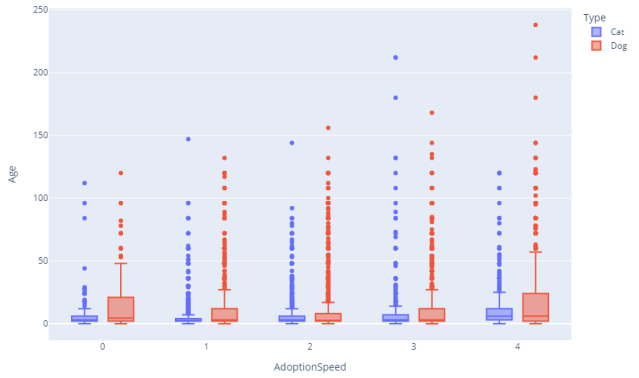

Fig. 1. Box plot showing Age distribution across various Adoption speed for Cats (Blue) & Dogs (Red).

Fig. 2. Histogram distribution of Cats and Dogs with color hue as Adoption Speed. Marginal box plot distribution displayed above each.

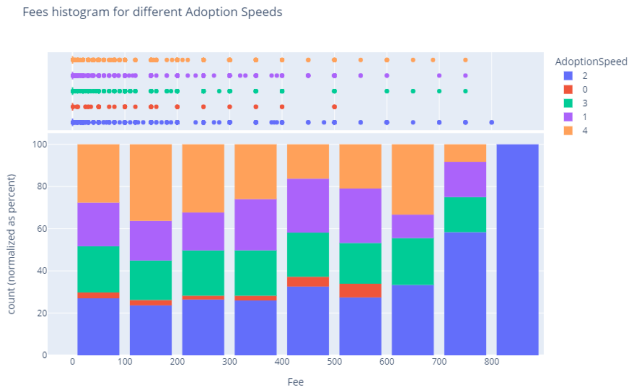

Fig. 3. Fees histogram for different Adoption Speeds

graph in Fig. 2, for faster rates of adoption, the age spread for dogs is more than that of cats. This implies that Age factor affects Dog adoption less than Cats adoptions, especially for quicker adoption speeds.

B. Fees vs Adoption Speed

We observed from the Fig. 3 that there is less difference in the graphs across lower fee ranges. As count bar is normalized by percent, it can be clearly said that the portion taken up by each adoption speed almost similar across smaller fee-ranges. Therefore, we can say that fees or no fee doesn't matter in adoption influencing adoption speed.

C. Does health profile matter towards adoption speed?

From Fig 4. it can be seen that surprisingly those pets which are unvaccinated, they have a higher chances of getting adopted across all adoption speed categories (Prob. of unvaccinated = 0.48; Prob. of vaccinated adoption = 0.39). For instance, Adoption speed 2 has probability of 0.5 for unvaccinated whereas it's just 0.35 for vaccinated.
Although, when we analyze adoption speed based on Dewormed status, it is seen that users prefer pets which are dewormed across all adoption speed categories (Prob. of dewormed = 0.56; Prob. of non-dewormed adoption = 0.32). Moreover, those pets for which dewormed status is unsure, they have drastically low probability of getting adopted.

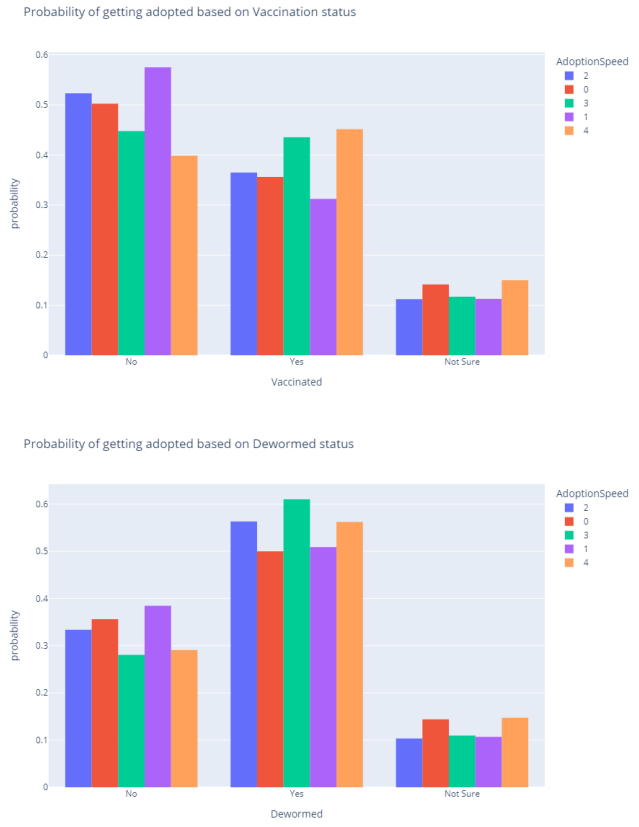

Fig. 4. Probability of getting adopted based on Vaccination Status

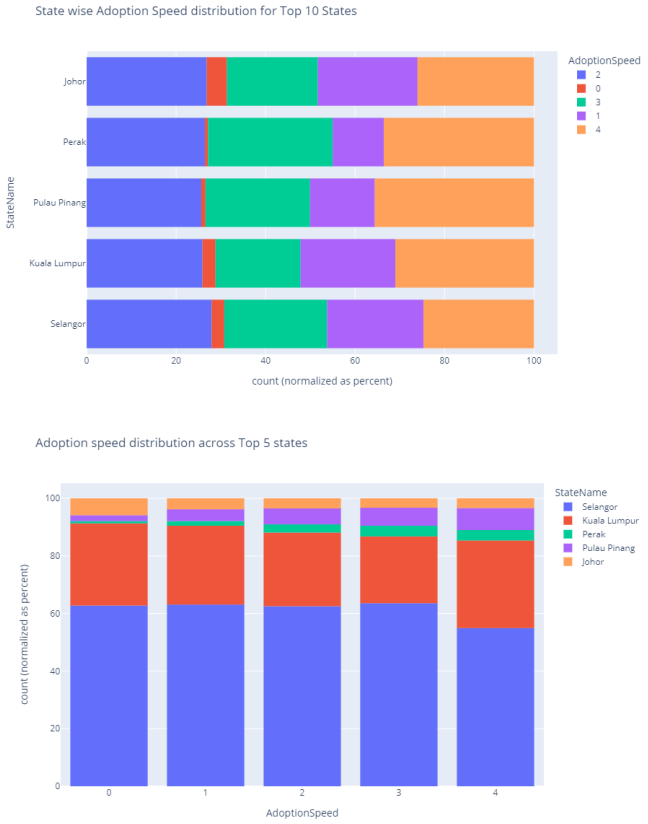

Fig. 5. Adoption stats for top 5 states

D. Which region contributes most to adoption

Fig. 5 are plots of the top 5 states by number of pet listings. Selangor has the highest number of listings and the fastest adoption speeds as compared to the other states. The pets not adopted after 100 days of being listed is the lowest in Selangor. Kuala Lumpur follows closely behind. Even though Kuala Lumpur is the capital of Malaysia, Selangor has the best adoption speeds.

E. Is there any notable correlation for adoption speed?

It can be said, that as such no strong correlation is visible which can single-handedly bias the adoption speed. We can say confidently that it's not a simple problem, and is governed by combination of multiple features. Also, Dewormed and Vaccinated seem to be correlated with correlation coefficient 0.63 which makes sense since they all fall under health factors. According to the below equation of Pearson correlation coefficient (r), we got the correlation matrix.

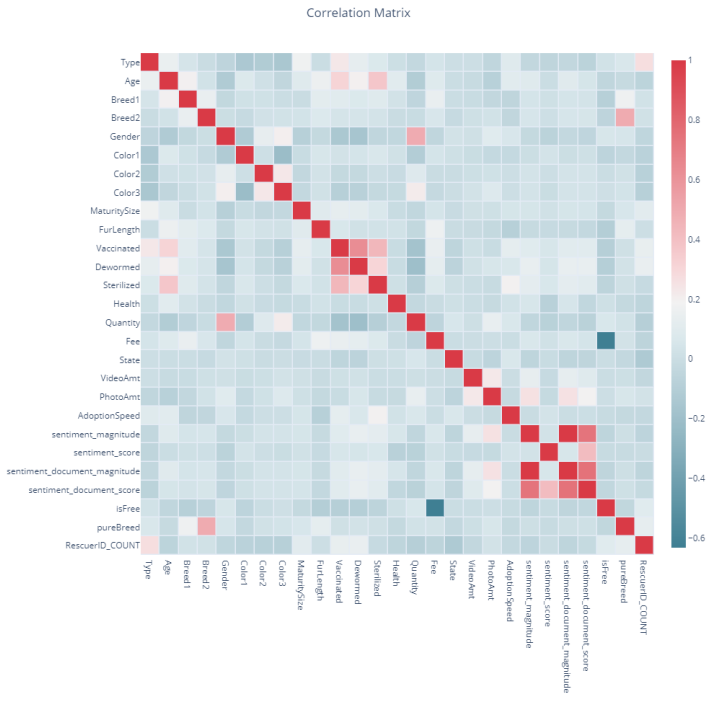

Fig. 6. Pearson correlation matrix across all features with each other.

$$r = \frac{\sum(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum(x_i - \bar{x})^2 \sum(y_i - \bar{y})^2}}$$

IV. Adoption Speed Prediction

In this project, we are trying to predict the pet adoption speed based on user profile features i.e. Age, Gender, Breed, StateName, Color etc.

A. Feature engineering

Apart from provided features from user profile data, we have also engineered certain features based on our data analysis which can be relevant in terms of predicting adoption speed.

RescuerID Count: For each PetID, there is a Rescuer ID mentioned which corresponds to a particular rescuer. We aggregated the number of Pets rescued by each RescuerID to get overall contribution of rescuer towards adoption speed.

Sentiment Features: We added sentiment data for each PetID by parsing JSON which contains sentiments for each sentence of profile description. We calculated following main features from sentence and document level per PetID i.e. **sentiment magnitude & score, document sentiment magnitude & score.** [2]

Finally we have used total of 27 features including sentiment, user profile details for training models. (Refer Fig7. for features)

B. Evaluation Techniques & Metrics

Root Mean Squared Error: In order to evaluate the performance of our model we are using Root Mean Squared Error.

$$RMSE = \sqrt{\frac{1}{n}\Sigma_{i=1}^{n}\big(Predicted_i - Actual_i\big)^2}$$

Stratified K-fold Cross-Validation:[3] The most used validation technique is K-Fold Cross-Validation which involves splitting the training dataset into k folds. The first k-1 folds are used for training, and the remaining fold is held for testing, which is repeated for K-folds. A total of K folds are fit and evaluated, and the mean RMSE for all these folds is returned. The stratified k fold cross-validation is an extension of the cross-validation technique used for classification problems. It maintains the same class ratio throughout the K folds as the ratio in the original dataset. For our use-case, **we have used stratified 5-fold validation technique.**

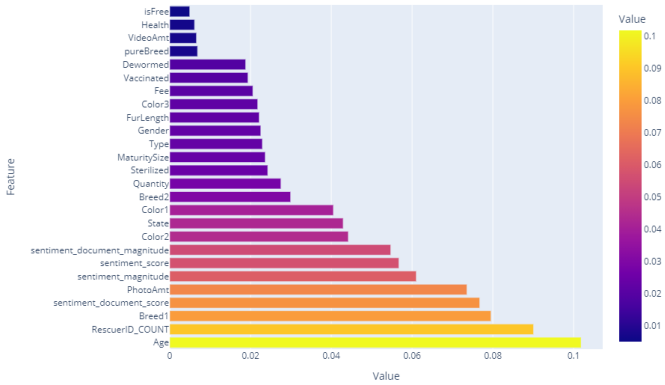

Fig. 7. Importance of different features for adoption speed prediction by Random Forest model.

V. Model Descriptions & Result

For our project, we considered this problem as a regression problem where adoption speed is continuous target variable. We implemented three regression models: **Linear Regression, Random Forest, and Light Gradient Boosting Machine (LGBM).**

TABLE I
RMSE score for Different Models across Training and Out-Of-Fold (OOF) set

Model	Train RMSE	OOF RMSE
Linear Regression	1.13	1.21
Random Forest	0.88	0.43
LGBM	0.91	1.04

A. Baseline Model: Linear Regression

This model considers that the output is a linear function of the inputs. Each input has a weight and a bias associated with it, which are the parameters of the model to be learnt while training.[4]

For our problem this is the most basic approach where we assume the adoption speed to be a linear

combination of the inputs. So, we considered the basic linear regression model as our baseline.

It can be seen that Linear Regression performs poorly and has the Highest RMSE score for both training as well as validation/OOF set. It makes sense since there was no strong correlation of a particular feature with adoption speed, and therefore linear models might not be a good choice given the number of features are very high.

B. Light Gradient Boosting Machine (LGBM)

LGBM is based on decision trees but implemented to increase the efficiency of the model and to reduce memory usage. It uses two methods Gradient-based One Side Sampling and Exclusive Feature Bundling which fulfills the limitations of histogram-based algorithm that is primarily used in all Gradient Boosted Decision Trees (GBDT) frameworks [5] which is why we used this model over other GBDT frameworks.

Although LGBM model offers pretty great performance but it requires lot of parameter tuning to actually beat other models. In our use-case, it can be seen that LGBM offers good performance with greater RMSE than baseline, it is still not able to beat Random Forest. But LGBM is faster in terms of training which reflects that first-version of model which is above baseline can be generated using LGBM.

C. Random Forest

Individual decision trees have high variance but when combined in parallel, the variance decreases as now the final output depends on a combined result of the decision trees. In the random forest technique, we do random row and feature sampling from the dataset to form datasets for the individual trees followed by ensembling the results.[6]

From results, Random Forest offers the lowest RMSE error with Out-Of-Fold set without overfitting over training dataset. It takes time to run but needs lesser hyperparameter tuning. Hence, As expected the random forest gives better results than the LGBM model.

VI. Summary

From data visualization, we observed that Age is a vital factor in adoption which is also reflected from our Random Forest model's feature importance list.

Furthermore, it can be said from data analysis that fee is not at all considered while adopting. In addition to this, pets which are unvaccinated have higher probability of getting adopted across all adoption speeds.

In our predictive task we observed that baseline model of Linear Regression is significantly outperformed by LGBM and Random Forest model. Therefore, it can be said that pet adoption is not a trivial problem which is governed by linearly varying parameters, rather requires much more sophisticated models and analysis.

References

[1] https://www.kaggle.com/c/petfinder-adoption-prediction/data
[2] https://cloud.google.com/natural-language/docs/basics
[3] https://analyticsindiamag.com/hands-on-tutorial-on-performance-measure-of-stratified-k-fold-cross-validation/
[4] https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LinearRegression.html/
[5] https://www.geeksforgeeks.org/lightgbm-light-gradient-boosting-machine/
[6] https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html