Animal Shelter Adoption Rate Analysis

In-depth Analysis

Summary

The goal of this project was to gain a better understanding of what makes a specific animal more likely to have a longer adoption time and propose strategies for resource optimization for a shelter. Throughout this analysis, I have focused specifically on dogs, so my goal here was to develop a model to determine time in shelter based on available data at intake.

I tested numerous regression models and was able to achieve a maximum R^2 value of % and therefore only explain about 1/10th of the variance in shelter times. However, I think this model could still be useful for shelters in gaining a general idea of animals at a high risk of having a longer adoption time upon intake.

Data Preparation

To prepare the data for building and optimizing a model, I used all data that would be available at animal intake and transformed it as needed.

1. Breed (categorical). To transform this variable to a usable, numerical form, I used the get dummies function to create a binary column for each breed.
2. Purebreed v. mixed breed (binary). I created a binary column that returns 1 or 0 if the animal is mixed breed or purebreed, respectively.
3. Intake condition (categorical). I used the same transformation as I used for breed for this column.
4. Intake type (categorical). I used the same transformation as I used for breed for this column.
5. Intake sex (categorical). I used the same transformation as I used for breed for this column.
6. Coat color (categorical). I used the same transformation as I used for breed for this column.
7. Multicolor v. single color coat (binary). I created a binary column that returns 1 or 0 if the animal is multicolor or single colored, respectively.
8. Age at intake (days). Although this column is presented in days, it is actually years -

Additionally, time in shelter is exponentially distributed, so I used a log transformation of that for the model's target.

Ordinary Least Squares

I used a linear regression using Ordinary Least Squares to develop a model to predict shelter times. This method was not particularly effective, and I was only able to obtain a model with an R^2 of only 10.86%. I used both scikit-learn and statsmodels to try to optimize the regression.

Feature Selection

For feature selection, I searched for the model that would explain the most variance in the data (highest R^2). I examined root mean squared error as well (standard deviation of the residuals) to select the optimal model.

First, I examined the top correlated variables to see if R^2 will improve. I looped through all possible numbers n of top correlated variables and determined that the highest R^2 value, 10.67%, was obtained with the top 114 correlated features. The root mean squared error was relatively high, at ~1.7; however, this model could still be useful in getting a general estimate of shelter time for a given dog.

Ridge Regression

I created another model using Ridge Regression. I thought this model would be particularly appropriate for this dataset since there is so much variation and not a lot of available data: by introducing a bias, we might see an improvement in ability to predict new data instead of overfitting to the available data.

I included all features in this model since Ridge regression will shrink features that do not contribute to the model's predictive power.

I found this model performed similarly well to OLS, with a maximum R^2 of 10.55% and an rmse of 1.7.

Parameter Optimization

scikit-learn has a built-in parameter optimization for alpha for the Ridge regression model, so I used that to optimize alpha.

Support Vector Regression

As a result of the data being sparse with many features, I chose to use a Support Vector Regression (SVR). SVM models are known to be effective in high dimensional spaces. Additionally, only certain training data are used in the decision function, which I hoped would help with the stochastic nature of this dataset.

I continued to use R^2 to evaluate the efficacy of this model, in addition to RMSE. After scaling the data and optimizing parameters, I was able to obtain an R^2 of 14.06 and an RMSE of ~0.94, the best values yet.

Data Preprocessing

I used scikit-learn's preprocessor to scale the data, as recommended by the documentation, because SVM algorithms are not scale invariant.

Of note, one of my goals is to develop a model that can be used on new data. This requires that new data be scaled using the StandardScalar and that results be “inverse transformed” to get an interpretable outcome.

Feature Selection

Because SVR is more computationally expensive, I only examined the performance of the model on the top 114 features compared to all features. I found that the model performed best using the top 114 features.

Parameter Optimization

I used GridSearchCV[1] to test values of C and epsilon from 0.01 to 100. The results found that the default values of 1.0 and 0.1, respectively, were the optimal parameters.

Ensemble Methods

After working to optimize several different models, I decided to try a few ensemble methods to see if combining model results would improve overall predictive power and model performance.

Averaging

I used averaging of the OLS, Ridge, and SVR models (and combinations of the three) to see if the predictive power could be improved. I found that using only Ridge and SVR resulted in a slightly higher R^2 but also a higher error rate.

None of the models seemed extraordinarily better than the SVR model on its own; OLS and SVR together performed moderately.

Gradient Boosting Regression

I also included a gradient boosting model to see if this strategy could help improve the model. Boosting is the development of many sequential models, where each model attempts to correct errors made in the previous model. The final model is the weighted mean of many weaker models. Because all of my model attempts have been mediocre, I hoped that this type of

[1] Note that this takes an extraordinarily long time to run - recommend running overnight.

ensemble method could help improve predictive power. Gradient Boosting is highly resistant to overfitting, so I used 1,000 estimators.

This model had a much higher R^2 value of 18.26% and a similar RMSE to those in previous models (~1.63).

Model Testing

I created several test animals to predict their shelter times and demonstrate the potential use of these models.

Challenges

One of the main challenges with developing a model with this data was that the vast majority of the available data were categorical; age at intake was the only numerical variable available, and even that was not purely linear: it appears that the ages are age in integer years or half-years multiplied by 365 to generate a loose estimate of age in days. I created dummy columns using categorical data, e.g., for each breed, there is a column that returns 1 if that dog is that breed or 0 if not. This resulted in a sparse, high-dimensional dataset.

Additionally, SVR was highly computationally expensive and very slow to run. It made it difficult to optimize and test different options.

Finally, I believe this data has a lot of unknowns, including why people adopt certain animals. Most animals at this shelter are adopted within a week, which does not really lend itself to a lot of predictable variation. In the end, I think these models could be useful for indicating a trend for adoption times but are likely not dependable for operational decisions.