Homework 3 - Solution

Eris Azizaj

December 5, 2018

Problem 1. Explain how you explored and cleaned your data. Did you decide to eliminate any features? Did you create any new ones? Why?

Answer: In this dataset, the majority of variables give the same information. For instance, the first variable *age upon outcome* gives the same information as *outcome age days* and *outcome age years* which tells the age of the cat in days and years. So there is no new information in keeping all of them so instead use only one. The same logic applies to other variables. I removed variables: *age upon outcome, animal id, sex upon outcome, outcome age years, cat kitten outcome, sex age outcome, dob month year*. Moreover I think that *dob_month* variable is wrong in this dataset since it is not the combination of *dob_month* and *dob_year* but the combination of *outcome_month* and *outcome_year*. I kept the variable name because it might be cases when the one who will adopt a cat might like the name of it and this is a reason for adoption. However, instead of it I created and used a dummy variable *name.fix* which has a value of 1 if the outcome has a name and 0 otherwise. I want to see if having a name effect the adoption probability or not. I also decided to keep *periods* and *period range* because it might tell some information for the category of the cats that were applied euthanasia. Also I filtered the given training dataset to have only four categories of *outcome_type*: Transfer, Adoption, Return to Owner, Euthanasia. Another important variable is *coat_pattern* which has a lot of missing values. Instead of eliminating these rows I decided to create the category "empty" instead of NAs. At the end, the dataset that I will use to train the model has 19 variables and 20292 observations. As I will explain in the following questions below, for different methods we have to convert the variables in different classes, otherwise the chosen method will not work.

Problem 2. Explain how you chose to evaluate your model (metrics, training/validation set partitioning, cross-validation).

Answer: As the major metrics for the evaluation of the model I chose to use "Accuracy". Depending on the methods used I partitioned the data differently. I used two different methods: Support Vector Machine(SVM) and Random Forest.

(a) SVM

For this method I had to change the class of each input variable to either numerical or logical. An important step in SVM is to scale the data. I run the model in both unscaled and scaled data. There is improvement in the accuracy of scaled sample but not much. For SVM I used two different types of kernel: *linear* and *radial*. For this method, in both kernel types I used 10 fold cross validation.

(b) Random Forest

For this method I had to change the class of all input variables that were character to factor, so we can have categories. Another important step when using Random Forest is to partition the data at hand in 80% training set and 20% validation set.

For both methods training the model took a LOT OF TIME. So, in order to save some time when playing with the model to find which variables give better results first I applied the method to only 10% of the training set. Whenever found the best model, then I would apply it to the whole training set.

Problem 3. List the models that you tried on the training set and their initial performance. Which model did you select to continue your analysis?

Answer: As it was mentioned before, I used two different methods: SVM and Random Forest.

(a) SVM

First I will show the initial results of SVM for different kernels and for both unscaled and scaled data. Before scaling the data the initial performance of SVM using linear kernel and having all variables as input was as in Fig 1. So we have a 74.3% accuracy level for this case and control C=1.

```
Support Vector Machines with Linear Kernel

20292 samples
   18 predictor
    4 classes: 'Adoption', 'Euthanasia', 'Return to Owner', 'Transfer'

No pre-processing
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 18262, 18262, 18264, 18263, 18262, 18263, ...
Resampling results:

  Accuracy   Kappa
  0.7427559  0.5329047

Tuning parameter 'C' was held constant at a value of 1
```

Figure 1: Performance of linear kernel in unscaled data

The performance of SVM with radial kernel for the unscaled data using all the variables as input was as in Fig 2, So we have a 77.2% accuracy level with best tuning parameter $sigma = 0.048$ and control C=1. We see an improvement when using radial instead of linear kernel keeping everything else the same.

```
Support Vector Machines with Radial Basis Function Kernel

20292 samples
   18 predictor
    4 classes: 'Adoption', 'Euthanasia', 'Return to Owner', 'Transfer'

No pre-processing
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 18263, 18262, 18262, 18263, 18262, 18262, ...
Resampling results across tuning parameters:

  C     Accuracy   Kappa
  0.25  0.7560124  0.5572718
  0.50  0.7611860  0.5672134
  1.00  0.7723725  0.5886526

Tuning parameter 'sigma' was held constant at a value of 0.0481136
Accuracy was used to select the optimal model using the largest value.
The final values used for the model were sigma = 0.0481136 and C = 1.
```

Figure 2: Performance of radial kernel in unscaled data

Now lets see how these models perform on scaled data. Performance of linear and radial kernels for scaled data(all variables included) is as in Fig 3 and Fig 4 respectively.

```
Support Vector Machines with Linear Kernel

20292 samples
   18 predictor
    4 classes: 'Adoption', 'Euthanasia', 'Return to Owner', 'Transfer'

No pre-processing
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 18263, 18262, 18264, 18264, 18262, 18262, ...
Resampling results:

  Accuracy   Kappa
  0.7428055  0.5330002

Tuning parameter 'C' was held constant at a value of 1
```

Figure 3: Performance of linear kernel in scaled data

```
Support Vector Machines with Radial Basis Function Kernel

20292 samples
   18 predictor
    4 classes: 'Adoption', 'Euthanasia', 'Return to Owner', 'Transfer'

No pre-processing
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 18263, 18263, 18262, 18263, 18263, 18264, ...
Resampling results across tuning parameters:

  C     Accuracy   Kappa
  0.25  0.7565559  0.5583025
  0.50  0.7612866  0.5673871
  1.00  0.7716849  0.5872545

Tuning parameter 'sigma' was held constant at a value of 0.04779694
Accuracy was used to select the optimal model using the largest value.
The final values used for the model were sigma = 0.04779694 and C = 1.
```

Figure 4: Performance of radial kernel in scaled data

As we can see from the above results there is a very small performance improvement for the linear case but a performance decrease for case. However, when using radial kernel we will have always better performance than both linear cases.

(b) Random Forest

Performance of the Random Forest in 80% of the whole data is as in Fig 5. We have a 79.7% accuracy which is the highest compared to all other results seen in other methods(SVM).

```
Random Forest

16235 samples
   18 predictor
    4 classes: 'Adoption', 'Euthanasia', 'Return to Owner', 'Transfer'

No pre-processing
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 14611, 14612, 14612, 14611, 14613, 14611, ...
Resampling results across tuning parameters:

  mtry  Accuracy   Kappa
    2   0.5302765  0.1313859
   27   0.7976600  0.6364781
   52   0.8008012  0.6458105
   78   0.8019095  0.6492997
  103   0.8006778  0.6478782
  128   0.8001849  0.6475682
  154   0.7999383  0.6475473
  179   0.7993225  0.6466168
  204   0.7974130  0.6437372
  230   0.7969818  0.6431066

Accuracy was used to select the optimal model using the largest value.
The final value used for the model was mtry = 78.
```

Figure 5: Performance of Random Forest on 80% of training data

This traing model perform very well in the validation set giving us 95.8% accuracy. Based on these results I will continue my analysis with the Random Forest method.

```
Confusion Matrix and Statistics

                Reference
Prediction       Adoption Euthanasia Return to Owner Transfer
  Adoption           1740          5              20       62
  Euthanasia            0        170               1        3
  Return to Owner       1          1             168        1
  Transfer             32         34              12     1807

Overall Statistics

               Accuracy : 0.9576
                 95% CI : (0.9509, 0.9636)
    No Information Rate : 0.4617
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.9275
 Mcnemar's Test P-Value : 1.647e-12

Statistics by Class:

                     Class: Adoption Class: Euthanasia Class: Return to Owner Class: Transfer
Sensitivity                   0.9814           0.80952               0.83582          0.9648
Specificity                   0.9619           0.99896               0.99922          0.9643
Pos Pred Value                0.9524           0.97701               0.98246          0.9586
Neg Pred Value                0.9852           0.98970               0.99151          0.9696
Prevalence                    0.4370           0.05176               0.04954          0.4617
Detection Rate                0.4289           0.04190               0.04141          0.4454
Detection Prevalence          0.4503           0.04289               0.04215          0.4646
Balanced Accuracy             0.9716           0.90424               0.91752          0.9645
```

Figure 6: Performance of Random Forest model on the validation data

Problem 4. Explain how you improved (fine-tuned) the model selected at point 3.

Answer: To increase the performance of the model I tried to select the variables that give the best split and highest performance. I started removing a variable at a time from the full model and see how the performance changed. If there is an increase of the metric I removed it from the "ideal" model otherwise keep it. Doing this exercise I realized that the increase in the performance of Random Forest model improved just a little. Since this improvement was so small I decided to continue with the model that uses all features.

Problem 5. Explain how you tested your model. What is the expected performance?

Answer: For the method I chose to continue my analysis, I partition the whole data into 80% training and 20% validation. I trained the model on training set and tested the model on validation set. As seen in the above results, my model has an accuracy level of 95.8% when applied to validation. So the expected performance on the new testing set is about 95.8%. We expect our model to classify the outcome type right in 95.8% of the cases.

Problem 6. Write a conclusion (at most 18 sentences!) summarizing the most important findings of the assignmentwhat did we learn about the dataset?

Answer: In this assignment we learned how to apply supervised learning methods of SVM and Random Forest on a given dataset. Before applying these methods it is important to clean the data and bring the variables in the right format(class). For SVM we can not use variables of class character or factor so we had to convert every variable into logical or numeric. Choosing the right kernel will guarantee us the highest performance. In theory we

know that scaling the data for SVM might improve the model significantly but this was not the case in this assignment as the results improved just a bit for the linear kernel but did worse for radial. Random Forest performed better compared to SVM for this dataset. Still it was important to bring the variables used in the format this method can work. Testing this model in the validation set gave 95.8% accuracy. For obvious reasons I chose to continue my analysis with Random Forest. Applying the model to the unseen data(testing set) is expected to give the right class 95.8% of the time.