Animal Shelter Outcome Prediction

Yvonne Kirschler

2025-04-15

Contents

1 Introduction

This report is part of the **HarvardX Data Science Capstone – Choose Your Own** module. It explores the challenge of predicting animal outcomes (e.g. Adoption, Transfer, Euthanasia) at the **Austin Animal Center**, using available intake data.

The dataset contains over 79,000 records and includes features such as:

- animal type
- intake type
- intake condition
- sex upon intake

Goal: Predict the `outcome_type` using machine learning
Main challenge: High class imbalance (e.g., ~42% Adoption)

2 Methods

2.1 Data loading

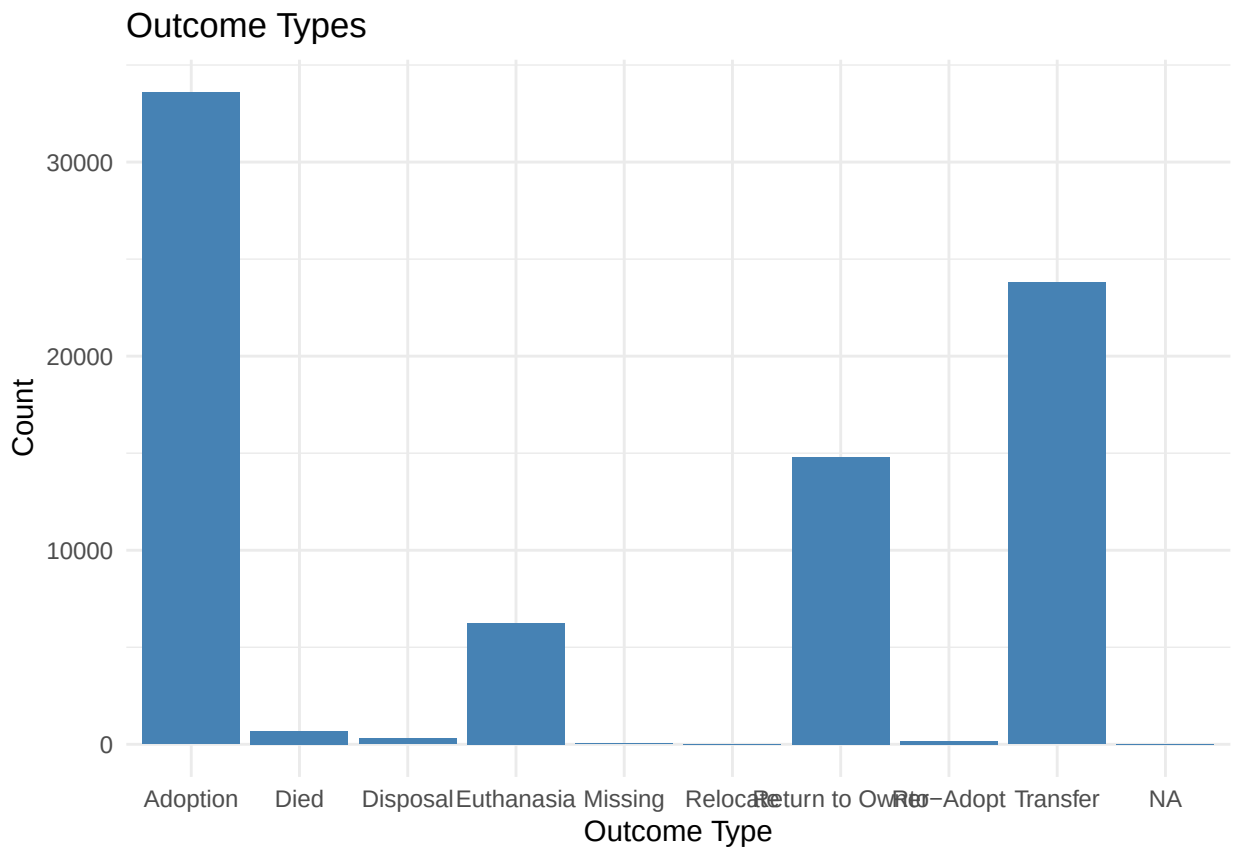
```
data <- read_csv("data/archive/aac_intakes_outcomes.csv")
glimpse(data)
```

```
## Rows: 79,672
## Columns: 41
## $ age_upon_outcome           <chr> "10 years", "7 years", "6 years", "10 years~
## $ animal_id_outcome          <chr> "A006100", "A006100", "A006100", "A047759",~
## $ date_of_birth              <dttm> 2007-07-09, 2007-07-09, 2007-07-09, 2004-0~
## $ outcome_subtype            <chr> NA, NA, NA, "Partner", NA, NA, NA, NA, NA, ~
## $ outcome_type               <chr> "Return to Owner", "Return to Owner", "Retu~
## $ sex_upon_outcome           <chr> "Neutered Male", "Neutered Male", "Neutered~
## $ `age_upon_outcome_(days)`  <dbl> 3650, 2555, 2190, 3650, 5840, 5475, 5475, 5~
## $ `age_upon_outcome_(years)` <dbl> 10, 7, 6, 10, 16, 15, 15, 15, 15, 18, 16, 1~
## $ age_upon_outcome_age_group <chr> "(7.5, 10.0]", "(5.0, 7.5]", "(5.0, 7.5]", ~
## $ outcome_datetime           <dttm> 2017-12-07 14:07:00, 2014-12-20 16:35:00, ~
## $ outcome_month              <dbl> 12, 12, 3, 4, 11, 11, 11, 9, 3, 9, 11, 12, ~
## $ outcome_year               <dbl> 2017, 2014, 2014, 2014, 2013, 2013, 2014, 2~
## $ outcome_monthyear          <chr> "2017-12", "2014-12", "2014-03", "2014-04",~
## $ outcome_weekday            <chr> "Thursday", "Saturday", "Saturday", "Monday~
## $ outcome_hour               <dbl> 0, 16, 17, 15, 11, 11, 19, 16, 15, 19, 13, ~
## $ outcome_number             <dbl> 1, 2, 3, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1~
## $ dob_year                   <dbl> 2007, 2007, 2007, 2004, 1997, 1998, 1999, 1~
## $ dob_month                  <dbl> 7, 7, 7, 4, 10, 6, 10, 8, 3, 8, 8, 1, 10, 4~
## $ dob_monthyear              <chr> "2017-12", "2014-12", "2014-03", "2014-04",~
## $ age_upon_intake            <chr> "10 years", "7 years", "6 years", "10 years~
## $ animal_id_intake           <chr> "A006100", "A006100", "A006100", "A047759",~
## $ animal_type                <chr> "Dog", "Dog", "Dog", "Dog", "Dog", "Dog", "~
```

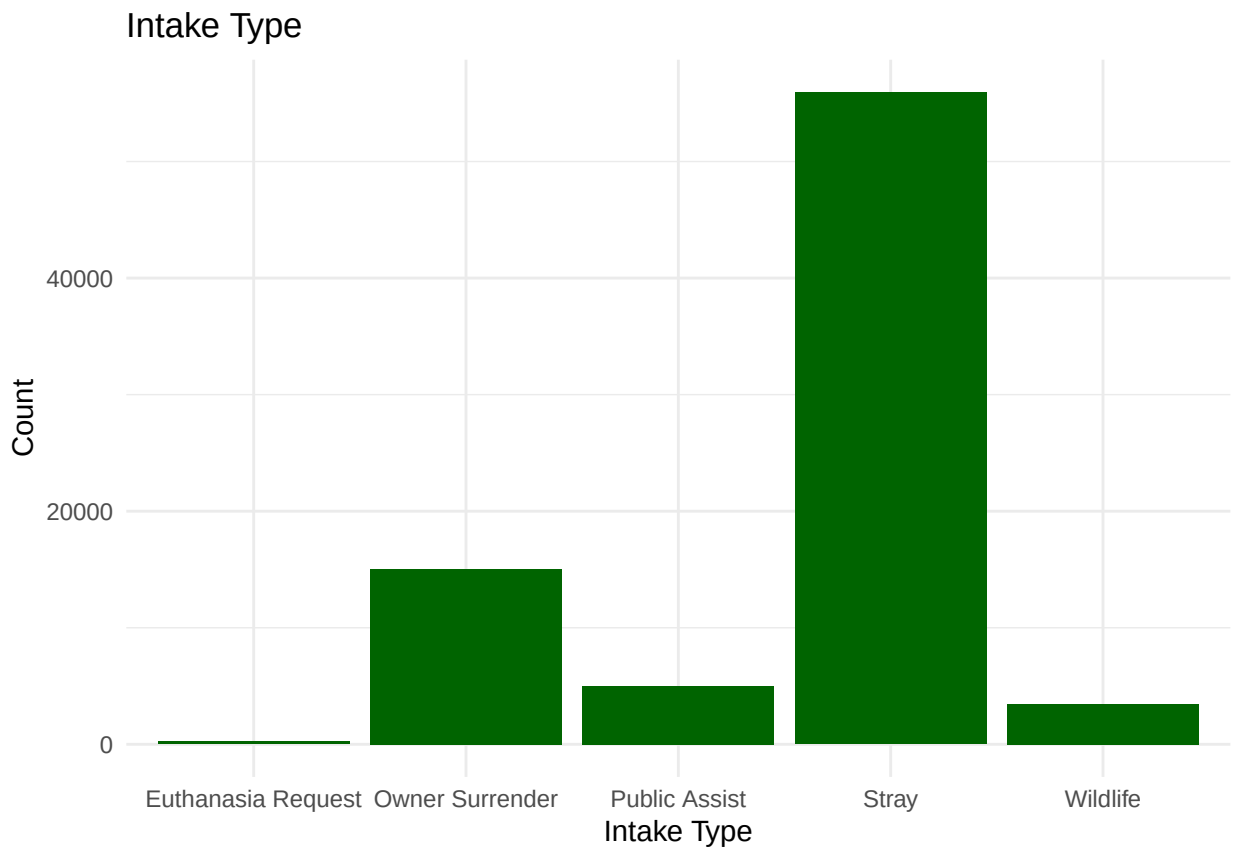
```
## $ breed                      <chr> "Spinone Italiano Mix", "Spinone Italiano M~
## $ color                      <chr> "Yellow/White", "Yellow/White", "Yellow/Whi~
## $ found_location             <chr> "Colony Creek And Hunters Trace in Austin (~
## $ intake_condition           <chr> "Normal", "Normal", "Normal", "Normal", "In~
## $ intake_type                <chr> "Stray", "Public Assist", "Public Assist", ~
## $ sex_upon_intake            <chr> "Neutered Male", "Neutered Male", "Neutered~
## $ count                      <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1~
## $ `age_upon_intake_(days)`   <dbl> 3650, 2555, 2190, 3650, 5840, 5475, 5475, 5~
## $ `age_upon_intake_(years)`  <dbl> 10, 7, 6, 10, 16, 15, 15, 15, 15, 18, 16, 1~
## $ age_upon_intake_age_group  <chr> "(7.5, 10.0]", "(5.0, 7.5]", "(5.0, 7.5]", ~
## $ intake_datetime            <dttm> 2017-12-07 00:00:00, 2014-12-19 10:21:00, ~
## $ intake_month               <dbl> 12, 12, 3, 4, 11, 11, 11, 9, 3, 9, 11, 12, ~
## $ intake_year                <dbl> 2017, 2014, 2014, 2014, 2013, 2013, 2014, 2~
## $ intake_monthyear           <chr> "2017-12", "2014-12", "2014-03", "2014-04",~
## $ intake_weekday             <chr> "Thursday", "Friday", "Friday", "Wednesday"~
## $ intake_hour                <dbl> 14, 10, 14, 15, 9, 14, 15, 11, 9, 17, 15, 1~
## $ intake_number              <dbl> 1, 2, 3, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1~
## $ time_in_shelter            <chr> "0 days 14:07:00.000000000", "1 days 06:14:~
## $ time_in_shelter_days       <dbl> 0.58819444, 1.25972222, 1.11388889, 4.97013~
```

2.2 Exploratory data analysis

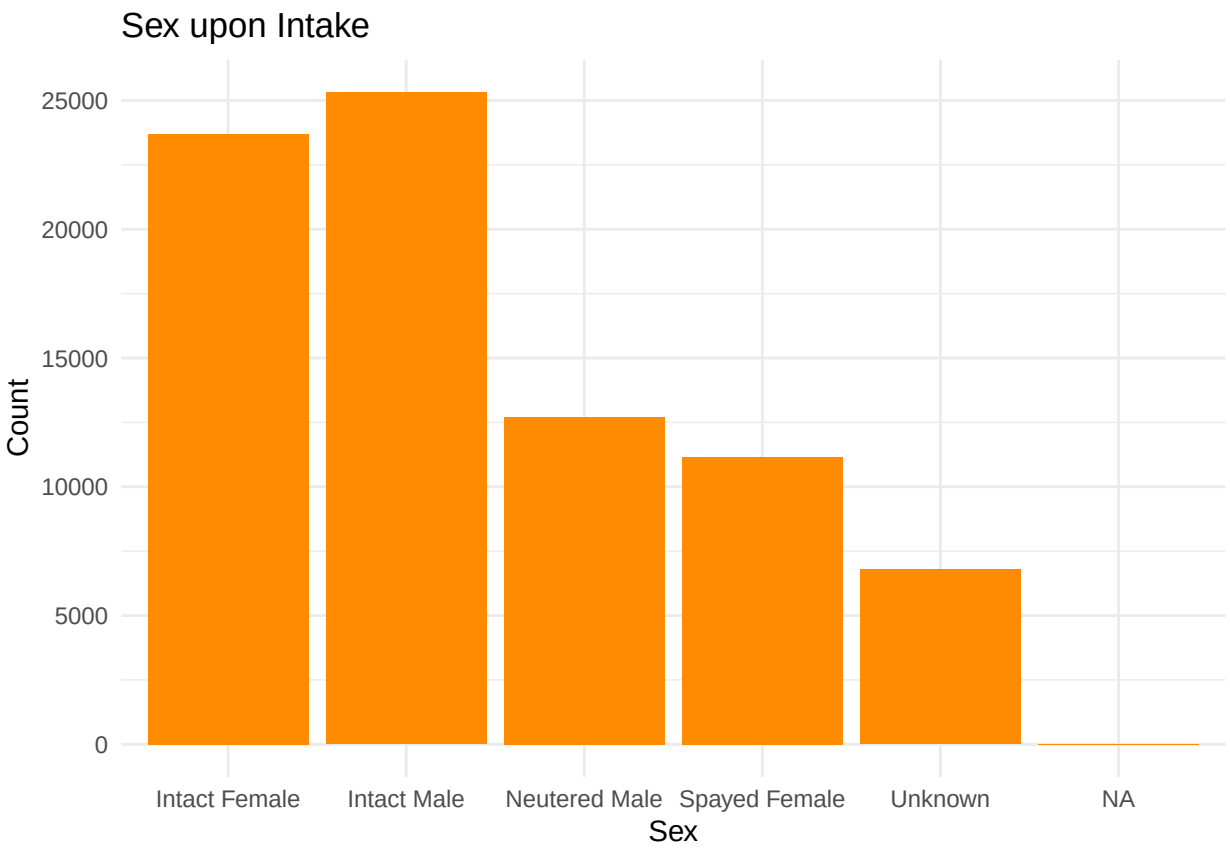
```r
# Outcome distribution
data %>% count(outcome_type) %>%
  ggplot(aes(x = fct_infreq(outcome_type), y = n)) +
  geom_col(fill = "steelblue") +
  labs(title = "Outcome Types", x = "Outcome Type", y = "Count") +
  theme_minimal()
```




```
# Intake type distribution
data %>% count(intake_type) %>%
  ggplot(aes(x = fct_infreq(intake_type), y = n)) +
  geom_col(fill = "darkgreen") +
  labs(title = "Intake Type", x = "Intake Type", y = "Count") +
  theme_minimal()
```




```
# Sex upon intake
data %>% count(sex_upon_intake) %>%
  ggplot(aes(x = fct_infreq(sex_upon_intake), y = n)) +
  geom_col(fill = "darkorange") +
  labs(title = "Sex upon Intake", x = "Sex", y = "Count") +
  theme_minimal()
```



2.3 Data preprocessing

Only variables that are known at the time of intake were selected,
to simulate a real-time decision support scenario.

```
data <- data %>%
  filter(!is.na(outcome_type)) %>%
  drop_na(animal_type, intake_type, sex_upon_intake, intake_condition)
```

2.4 Train/test split

```
set.seed(42)
train_index <- createDataPartition(data$outcome_type, p = 0.8, list = FALSE)
train_set <- data[train_index, ]
test_set  <- data[-train_index, ]
```

3 Modeling

3.1 Baseline model

```
most_common <- train_set %>%
  count(outcome_type) %>%
  slice_max(n, n = 1) %>%
  pull(outcome_type)

baseline_predictions <- rep(most_common, nrow(test_set))
baseline_accuracy <- mean(baseline_predictions == test_set$outcome_type)
baseline_accuracy
```

```
## [1] 0.421773
```

3.2 Random Forest model (5-fold CV)

```
rf_data <- train_set %>%
  mutate(outcome_type = as.factor(outcome_type)) %>%
  select(outcome_type, animal_type, intake_type, sex_upon_intake, intake_condition)

rf_control <- trainControl(method = "cv", number = 5)

set.seed(42)
rf_model <- train(
  outcome_type ~ .,
  data = rf_data,
  method = "rf",
  trControl = rf_control,
  importance = TRUE
)

rf_predictions <- predict(rf_model, newdata = test_set)

# Align factor levels
combined_levels <- union(levels(rf_predictions), levels(test_set$outcome_type))
rf_predictions <- factor(rf_predictions, levels = combined_levels)
test_set$outcome_type <- factor(test_set$outcome_type, levels = combined_levels)

rf_accuracy <- mean(rf_predictions == test_set$outcome_type)
rf_confusion <- confusionMatrix(rf_predictions, test_set$outcome_type)

rf_accuracy
```

```
## [1] 0.5803616
```

```
rf_confusion
```

```
## Confusion Matrix and Statistics
```

```
##
##                  Reference
## Prediction        Adoption Died Disposal Euthanasia Missing Relocate
##   Adoption            5688   62        4        287       7        0
##   Died                   0    1        0          0       0        0
##   Disposal               0    0        0          0       0        0
##   Euthanasia            90   42       50        773       0        3
##   Missing                0    0        0          0       0        0
##   Relocate               0    0        0          0       0        0
##   Return to Owner      619    6        0         59       1        0
##   Rto-Adopt              0    0        0          0       0        0
##   Transfer             321   27        6        129       1        0
##                  Reference
## Prediction        Return to Owner Rto-Adopt Transfer
##   Adoption                   1027        17     3265
##   Died                          0         0        0
##   Disposal                      0         0        0
##   Euthanasia                   34         0      117
##   Missing                       0         0        0
##   Relocate                      0         0        0
##   Return to Owner            1727        13      322
##   Rto-Adopt                     0         0        0
##   Transfer                    170         5     1055
##
## Overall Statistics
##
##                Accuracy : 0.5804
##                  95% CI : (0.5727, 0.588)
##     No Information Rate : 0.4218
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.3604
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: Adoption Class: Died Class: Disposal
## Sensitivity                   0.8467   7.246e-03        0.000000
## Specificity                   0.4931   1.000e+00        1.000000
## Pos Pred Value                0.5492   1.000e+00             NaN
## Neg Pred Value                0.8151   9.914e-01        0.996233
## Prevalence                    0.4218   8.664e-03        0.003767
## Detection Rate                0.3571   6.278e-05        0.000000
## Detection Prevalence          0.6502   6.278e-05        0.000000
## Balanced Accuracy             0.6699   5.036e-01        0.500000
##                      Class: Euthanasia Class: Missing Class: Relocate
## Sensitivity                    0.61939       0.000000       0.0000000
## Specificity                    0.97711       1.000000       1.0000000
## Pos Pred Value                 0.69702            NaN             NaN
## Neg Pred Value                 0.96795       0.999435       0.9998117
## Prevalence                     0.07835       0.000565       0.0001883
## Detection Rate                 0.04853       0.000000       0.0000000
## Detection Prevalence           0.06963       0.000000       0.0000000
```

```
## Balanced Accuracy              0.79825       0.500000       0.5000000
##                       Class: Return to Owner Class: Rto-Adopt Class: Transfer
## Sensitivity                           0.5838         0.000000          0.22169
## Specificity                           0.9214         1.000000          0.94100
## Pos Pred Value                        0.6287              NaN          0.61552
## Neg Pred Value                        0.9066         0.997803          0.73941
## Prevalence                            0.1857         0.002197          0.29878
## Detection Rate                        0.1084         0.000000          0.06624
## Detection Prevalence                  0.1725         0.000000          0.10761
## Balanced Accuracy                     0.7526         0.500000          0.58134
```

4 Results

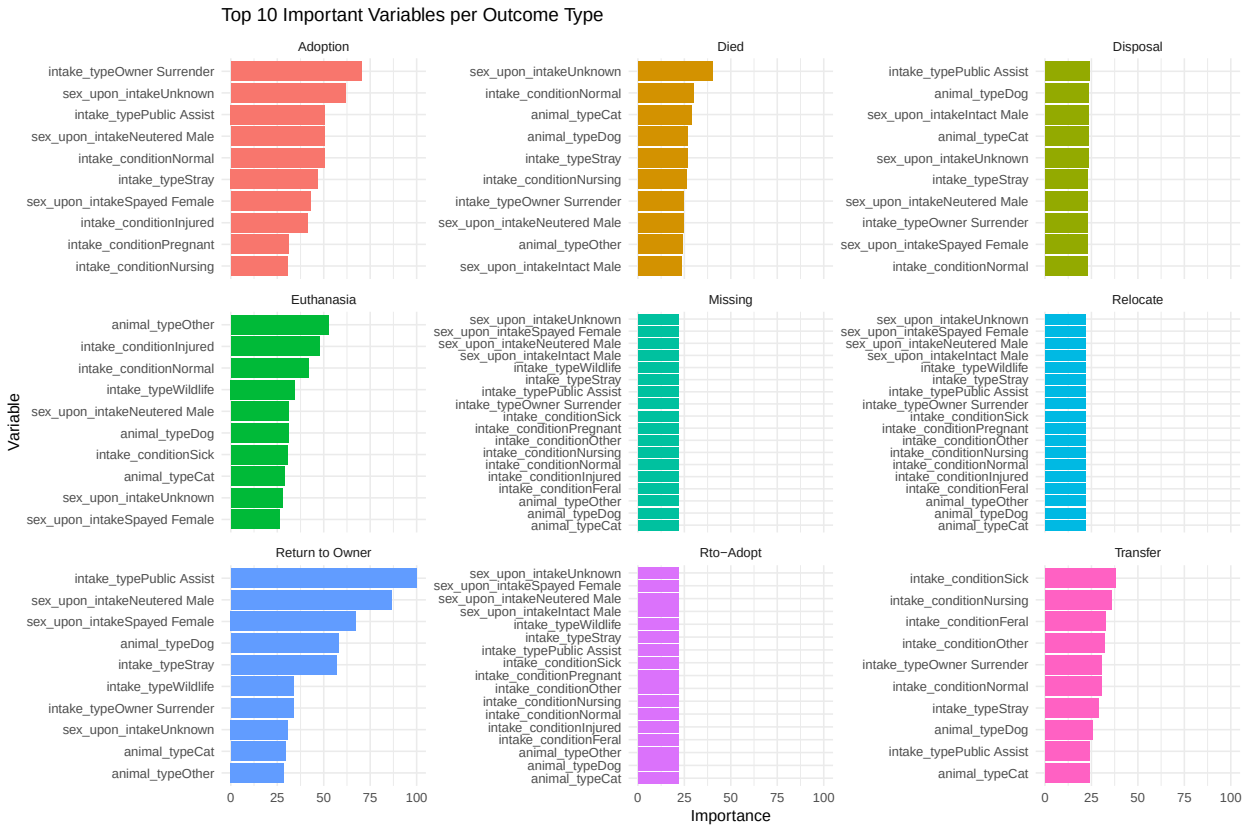
4.1 Accuracy comparison

```
absolute_improvement <- rf_accuracy - baseline_accuracy
relative_improvement <- absolute_improvement / baseline_accuracy

tibble(
  Baseline = baseline_accuracy,
  RandomForest = rf_accuracy,
  AbsoluteImprovement = absolute_improvement,
  RelativeImprovement = percent(relative_improvement, accuracy = 0.1)
)
```

```
## # A tibble: 1 x 4
##   Baseline RandomForest AbsoluteImprovement RelativeImprovement
##      <dbl>        <dbl>               <dbl> <chr>
## 1    0.422        0.580               0.159 37.6%
```

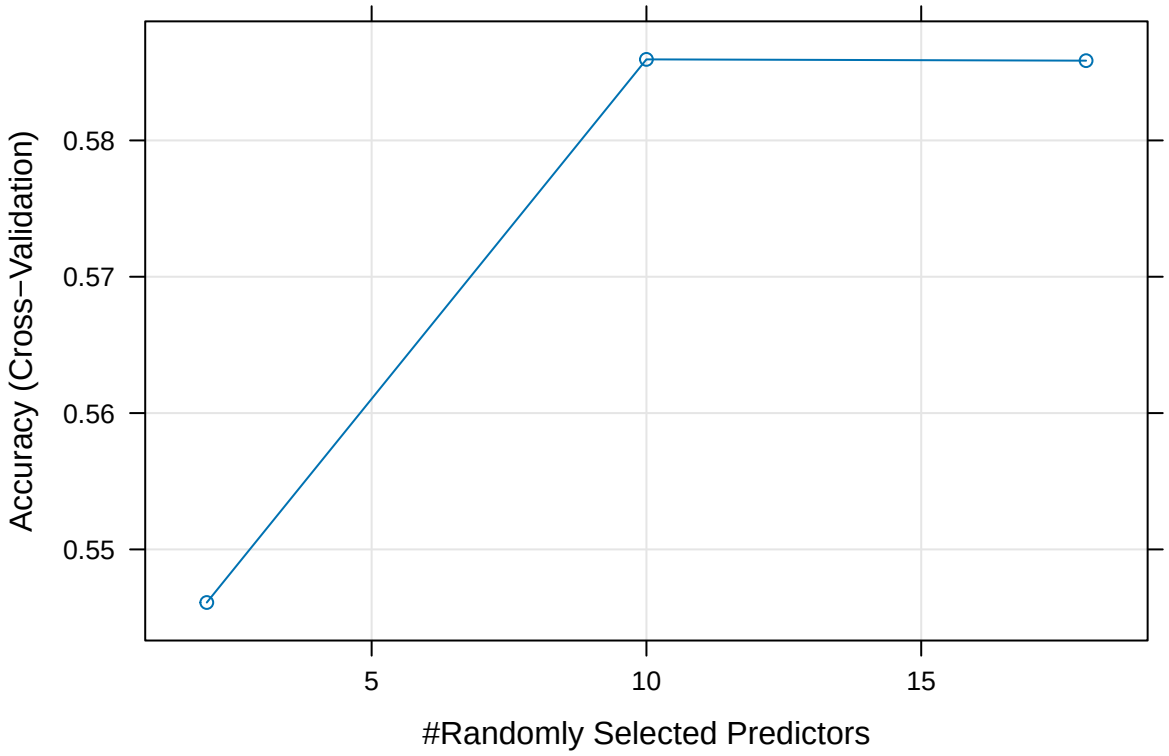
4.2 Variable importance

```
importance_df <- varImp(rf_model)$importance %>%
  rownames_to_column("Variable") %>%
  pivot_longer(-Variable, names_to = "Outcome", values_to = "Importance") %>%
  group_by(Outcome) %>%
  slice_max(order_by = Importance, n = 10)

ggplot(importance_df, aes(x = reorder_within(Variable, Importance, Outcome), y = Importance, fill = Out
  geom_col(show.legend = FALSE) +
  facet_wrap(~ Outcome, scales = "free_y") +
  coord_flip() +
  scale_x_reordered() +
  labs(
    title = "Top 10 Important Variables per Outcome Type",
    x = "Variable",
    y = "Importance"
  ) +
  theme_minimal()
```



4.3 Model performance visualization

```
plot(rf_model)
```



5 Conclusion

This project explored how to predict animal shelter outcomes using features available at the time of intake. Two models were compared:

- **Baseline (majority class):** 0.4218
- **Random Forest (5-fold CV):** 0.5804
- **Relative improvement:** 37.6%

The Random Forest model significantly outperformed the baseline,
with a relative accuracy improvement of over 37%.
Top predictors were `intake_type`, `sex_upon_intake`, and `intake_condition`.

6 Interpretation

- Stray animals tend to be adopted more often
- Owner-surrendered animals show different outcome trends
- Health-related intake conditions are linked to higher euthanasia or transfer rates

- Neutered/spayed animals may be more adoptable

7 Limitations

- Strong class imbalance affects minority classes
- No temporal or behavioral history used
- Model not tuned for hyperparameters beyond `mtry`

8 References

- Austin Animal Center Dataset: https://www.kaggle.com/datasets/aaronschlegel/austin-animal-center-shelter-intakes-and-outcomes
- caret R package: https://topepo.github.io/caret/
- randomForest package: https://cran.r-project.org/web/packages/randomForest
- OpenAI (ChatGPT): Supported structure and phrasing; modeling and decisions by Yvonne Kirschler

9 Appendix

```
## R version 4.4.3 (2025-02-28)
## Platform: aarch64-apple-darwin20
## Running under: macOS Sequoia 15.4
##
## Matrix products: default
## BLAS:   /Library/Frameworks/R.framework/Versions/4.4-arm64/Resources/lib/libRblas.0.dylib
## LAPACK: /Library/Frameworks/R.framework/Versions/4.4-arm64/Resources/lib/libRlapack.dylib;  LAPACK v
##
## locale:
## [1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
##
## time zone: Europe/Zurich
## tzcode source: internal
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
##  [1] tidytext_0.4.2       scales_1.3.0         randomForest_4.7-1.2
##  [4] caret_7.0-1          lattice_0.22-6       lubridate_1.9.4
##  [7] forcats_1.0.0        stringr_1.5.1        dplyr_1.1.4
## [10] purrr_1.0.2          readr_2.1.5          tidyr_1.3.1
## [13] tibble_3.2.1         ggplot2_3.5.1        tidyverse_2.0.0
##
## loaded via a namespace (and not attached):
##  [1] tidyselect_1.2.1     timeDate_4041.110    farver_2.1.2
```

```
##  [4] fastmap_1.2.0        janeaustenr_1.0.0    pROC_1.18.5
##  [7] digest_0.6.37        rpart_4.1.24         timechange_0.3.0
## [10] lifecycle_1.0.4      tokenizers_0.3.0     survival_3.8-3
## [13] magrittr_2.0.3       compiler_4.4.3       rlang_1.1.5
## [16] tools_4.4.3          utf8_1.2.4           yaml_2.3.10
## [19] data.table_1.16.4    knitr_1.49           labeling_0.4.3
## [22] bit_4.5.0.1          plyr_1.8.9           withr_3.0.2
## [25] nnet_7.3-20          grid_4.4.3           stats4_4.4.3
## [28] fansi_1.0.6          e1071_1.7-16         colorspace_2.1-1
## [31] future_1.34.0        globals_0.16.3       iterators_1.0.14
## [34] MASS_7.3-64          cli_3.6.4            rmarkdown_2.29
## [37] crayon_1.5.3         generics_0.1.3       rstudioapi_0.17.1
## [40] future.apply_1.11.3  reshape2_1.4.4       tzdb_0.5.0
## [43] proxy_0.4-27         splines_4.4.3        parallel_4.4.3
## [46] vctrs_0.6.5          hardhat_1.4.1        Matrix_1.7-2
## [49] hms_1.1.3            bit64_4.5.2          listenv_0.9.1
## [52] foreach_1.5.2        gower_1.0.2          recipes_1.2.1
## [55] glue_1.8.0           parallelly_1.43.0    codetools_0.2-20
## [58] stringi_1.8.4        gtable_0.3.6         munsell_0.5.1
## [61] pillar_1.9.0         htmltools_0.5.8.1    ipred_0.9-15
## [64] lava_1.8.1           R6_2.5.1             vroom_1.6.5
## [67] evaluate_1.0.1       SnowballC_0.7.1      class_7.3-23
## [70] Rcpp_1.0.14          nlme_3.1-167         prodlim_2024.06.25
## [73] xfun_0.49            pkgconfig_2.0.3      ModelMetrics_1.2.2.2
```