

HARVARD EXTENSION SCHOOL

Titanic Survival Classification: Group Project Report

Aljazi Al Maghlouth Anjan Chakravarti Ganapathy Lakshmanaperumal
Guy Nguyen-Phuoc Jonathan Terrasi Julie Lander Khatanbaatar Orkhon
Max Amiesimaka

07 December 2025

Abstract

We build a champion/benchmark modeling solution to predict passenger survival on the RMS Titanic. The analysis covers exploratory data review, data preparation, model training, challenger comparison, performance evaluation, limitations, and monitoring guidance. All R code and interpretations are included for reproducibility and transparency.

Contents

Executive Summary	1
I. Introduction (5 points)	1
II. Description of the Data and Quality (15 points)	2
III. Model Development Process (15 points)	6
IV. Model Performance Testing (15 points)	8
V. Challenger Models (15 points)	15
VI. Model Limitations and Assumptions (15 points)	16
VII. Ongoing Model Monitoring Plan (5 points)	16
VIII. Conclusion (5 points)	16
Bibliography (7 points)	16
Appendix (3 points)	17

Executive Summary

We predict Titanic passenger survival using demographic and ticketing information. A cleaned dataset of 1,310 records is split 70/30 train/test (set.seed = 1023). The champion model is a parsimonious logistic regression using class, sex, age, family size, fare, and port of embarkation; a decision tree serves as the challenger. Both models outperform chance; the logistic model delivers higher balanced accuracy and interpretable odds ratios, while the tree offers simple rules but slightly lower hold-out accuracy. Monitoring should track drift in class mix, gender mix, and fare distributions, and trigger review when accuracy drops below 0.80 or when inputs shift beyond training percentiles. Key limitations include missing values (age, fare, cabin), historical bias, and simplified imputations.

I. Introduction (5 points)

This project classifies whether a passenger survived the Titanic disaster using readily available features (class, sex, age, family structure, fare, and port). We evaluate two supervised classification methods: logistic regression (champion) and decision tree (challenger). Success is defined by accurate and explainable survival predictions that generalize to the hold-out test set.

II. Description of the Data and Quality (15 points)

The dataset contains 1,310 observations and 14 original variables. Key predictors are a mix of categorical (class, sex, embarked) and numeric (age, fare, family counts). Several variables contain notable missing values (age, cabin, boat, body, and home destination).

```
glimpse(titanic_raw)
```

```
## Rows: 1,310
## Columns: 14
## $ pclass    <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ~
## $ survived  <dbl> 1, 1, 0, 0, 0, 1, 1, 0, 1, 0, 0, 1, 1, 1, 1, 0, 0, 1, 1, 0, ~
## $ name      <chr> "Allen, Miss. Elisabeth Walton", "Allison, Master. Hudson Tr~
## $ sex       <chr> "female", "male", "female", "male", "female", "male", "femal~
## $ age       <dbl> 29.0000, 0.9167, 2.0000, 30.0000, 25.0000, 48.0000, 63.0000,~
## $ sibsp     <dbl> 0, 1, 1, 1, 1, 0, 1, 0, 2, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, ~
## $ parch     <dbl> 0, 2, 2, 2, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, ~
## $ ticket    <chr> "24160", "113781", "113781", "113781", "113781", "19952", "1~
## $ fare      <dbl> 211.3375, 151.5500, 151.5500, 151.5500, 151.5500, 26.5500, 7~
## $ cabin     <chr> "B5", "C22 C26", "C22 C26", "C22 C26", "C22 C26", "E12", "D7~
## $ embarked  <chr> "S", "S", "S", "S", "S", "S", "S", "S", "S", "S", "C", "C", "C", ~
## $ boat      <chr> "2", "11", NA, NA, NA, "3", "10", NA, "D", NA, NA, "4", "9",~
## $ body      <dbl> NA, NA, NA, 135, NA, NA, NA, NA, NA, 22, 124, NA, NA, NA, NA~
## $ home.dest  <chr> "St Louis, MO", "Montreal, PQ / Chesterville, ON", "Montreal~
```

```
titanic_raw %>%
  summarise(across(everything(), ~ sum(is.na(.)))) %>%
  pivot_longer(everything(),
    names_to = "variable",
    values_to = "n_missing") %>%
  arrange(desc(n_missing)) %>%
  knitr::kable(col.names = c("Variable", "Missing Count"))
```

Variable	Missing Count
body	1189
cabin	1015
boat	824
home.dest	565
age	264
embarked	3
fare	2
pclass	1
survived	1
name	1
sex	1
sibsp	1
parch	1
ticket	1

Data Preparation

We clean and engineer a modeling frame as follows:

1. Convert categorical variables to factors.
2. Impute age by sex/class median.
3. Impute fare with the overall median.

4. Drop high-missing columns.
5. Create a `family_size` helper feature.

```
clean_titanic <- titanic_raw %>%
  mutate(
    survived = factor(survived, levels = c(0, 1),
                      labels = c("Died", "Survived")),
    pclass = factor(pclass, levels = c(1, 2, 3),
                   labels = c("1st", "2nd", "3rd")),
    sex = factor(sex),
    embarked = fct_explicit_na(embarked, "Unknown")
  )

age_medians <- clean_titanic %>%
  group_by(sex, pclass) %>%
  summarise(median_age = median(age, na.rm = TRUE), .groups = "drop")

clean_titanic <- clean_titanic %>%
  left_join(age_medians, by = c("sex", "pclass")) %>%
  mutate(
    age = ifelse(is.na(age), median_age, age),
    fare = ifelse(is.na(fare), median(fare, na.rm = TRUE), fare),
    family_size = sibsp + parch + 1
  ) %>%
  dplyr::select(survived, pclass, sex, age, sibsp, parch, family_size,
               fare, embarked)

summary(clean_titanic)
```

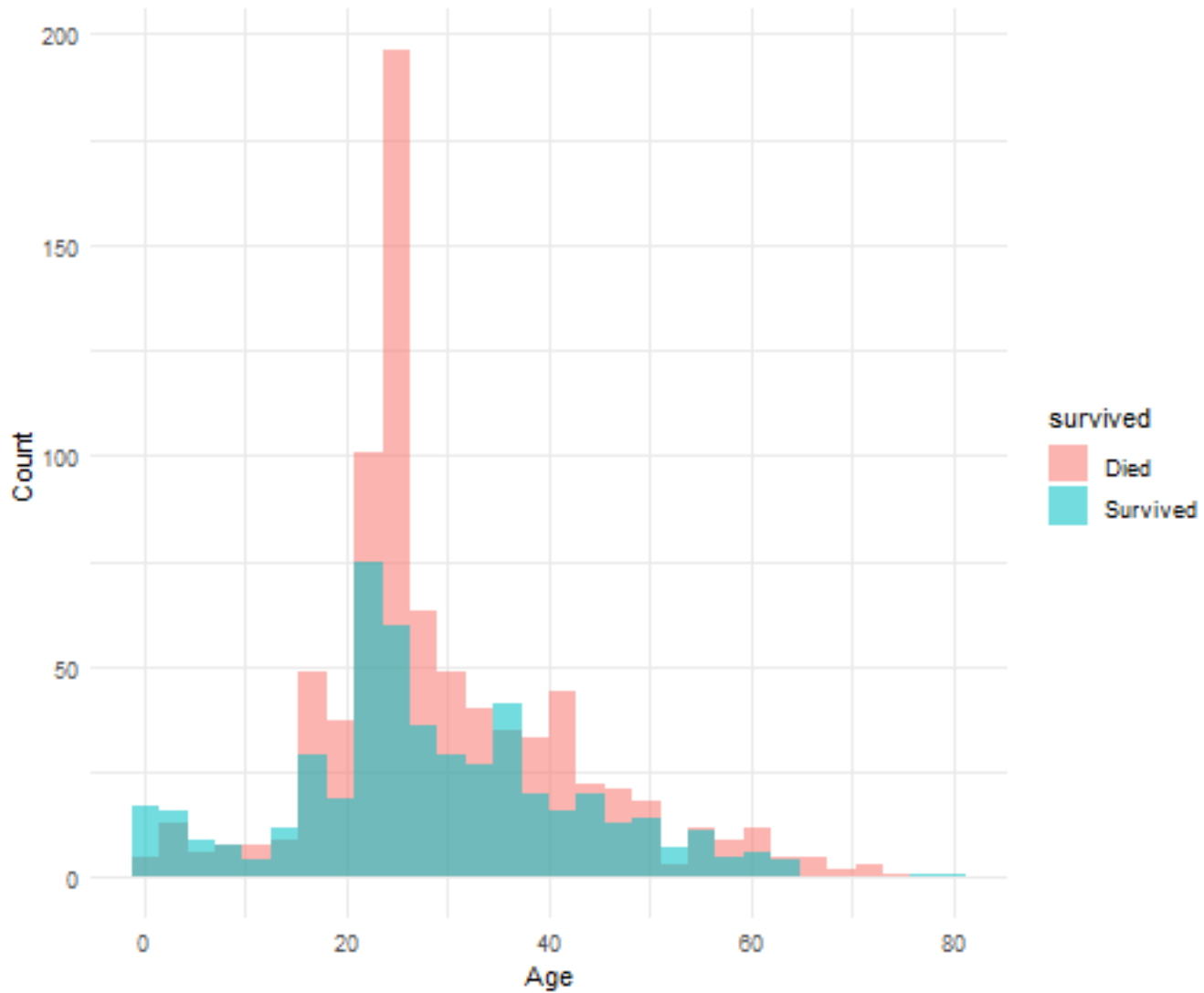
```
##      survived      pclass      sex      age      sibsp
## Died      :809      1st :323      female:466      Min.   : 0.1667      Min.   :0.0000
## Survived:500      2nd :277      male  :843      1st Qu.:22.0000      1st Qu.:0.0000
## NA's      : 1      3rd :709      NA's   : 1      Median :26.0000      Median :0.0000
##                                     NA's: 1      Mean   :29.2614      Mean   :0.4989
##                                     3rd Qu.:36.0000      3rd Qu.:1.0000
##                                     Max.   :80.0000      Max.   :8.0000
##                                     NA's   :1      NA's   :1
##      parch      family_size      fare      embarked
## Min.   :0.000      Min.   : 1.000      Min.   : 0.000      C      :270
## 1st Qu.:0.000      1st Qu.: 1.000      1st Qu.: 7.896      Q      :123
## Median :0.000      Median : 1.000      Median :14.454      S      :914
## Mean   :0.385      Mean   : 1.884      Mean   :33.267      Unknown: 3
## 3rd Qu.:0.000      3rd Qu.: 2.000      3rd Qu.:31.275
## Max.   :9.000      Max.   :11.000      Max.   :512.329
## NA's   :1      NA's   :1
```

Interpretation: Imputation preserves sample size without extreme values. Removing cabin/ticket/body/boat/home.dest reduces noise while retaining predictive signal. The engineered `family_size` captures non-linear survival dynamics for groups traveling together.

Exploratory Graphs

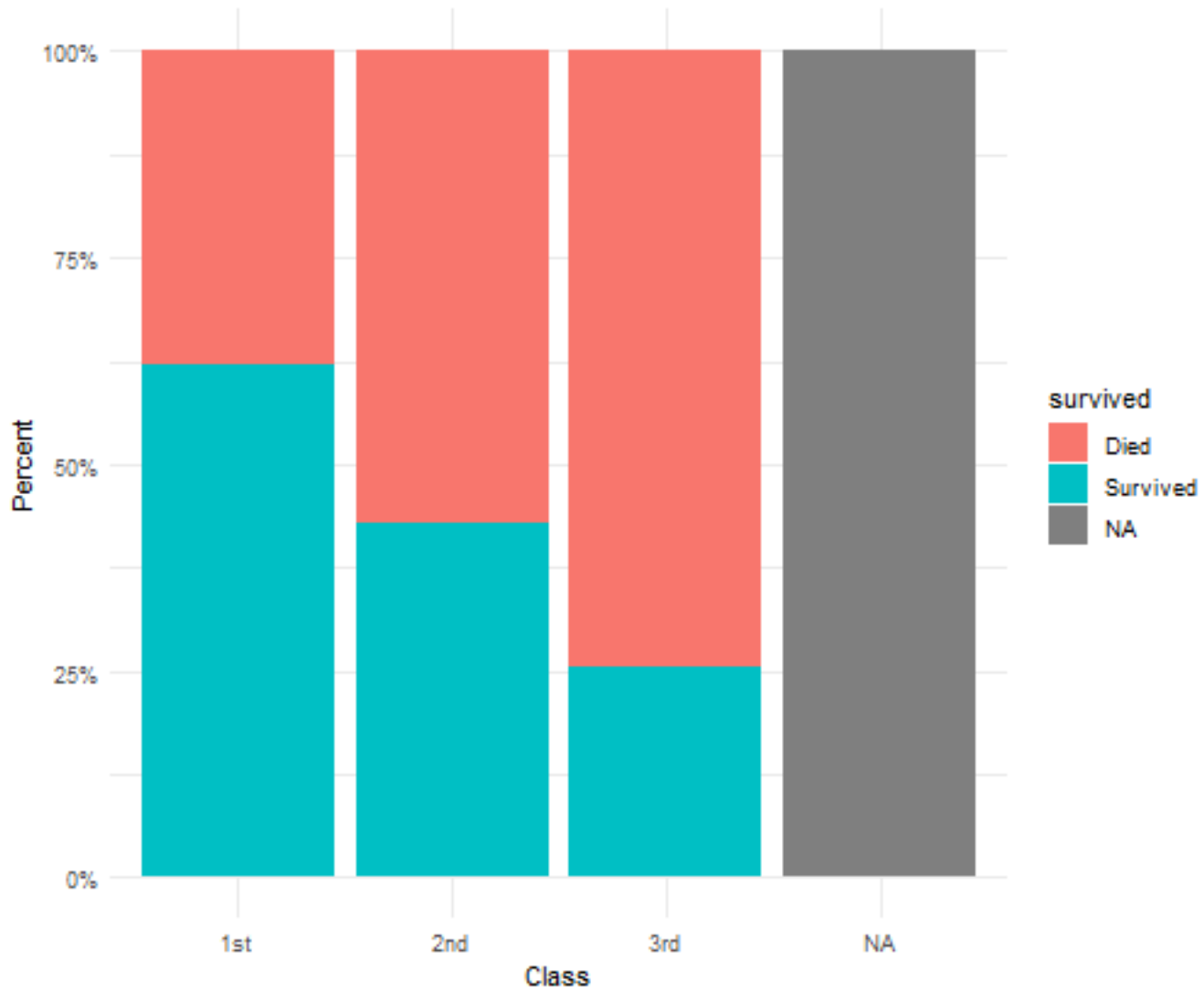
```
clean_titanic %>%
  ggplot(aes(x = age, fill = survived)) +
  geom_histogram(position = "identity", alpha = 0.55, bins = 30) +
  labs(title = "Figure 1. Age distribution by survival",
       x = "Age", y = "Count") +
  theme_minimal()
```

Figure 1. Age distribution by survival



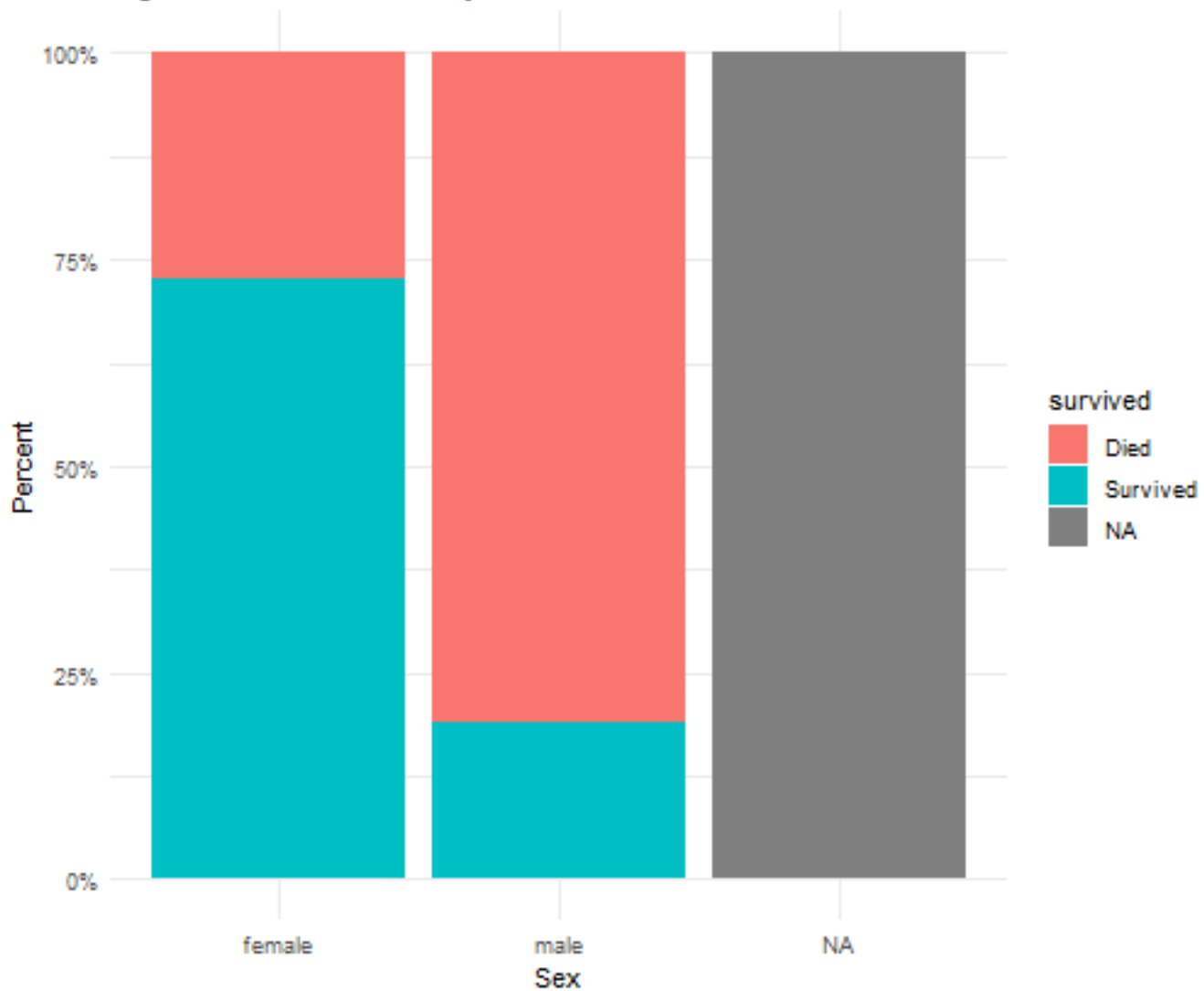
```
clean_titanic %>%  
  ggplot(aes(x = pclass, fill = survived)) +  
  geom_bar(position = "fill") +  
  scale_y_continuous(labels = scales::percent_format()) +  
  labs(title = "Figure 2. Survival share by passenger class",  
        x = "Class", y = "Percent") +  
  theme_minimal()
```

Figure 2. Survival share by passenger class



```
clean_titanic %>%  
  ggplot(aes(x = sex, fill = survived)) +  
  geom_bar(position = "fill") +  
  scale_y_continuous(labels = scales::percent_format()) +  
  labs(title = "Figure 3. Survival share by sex",  
        x = "Sex", y = "Percent") +  
  theme_minimal()
```

Figure 3. Survival share by sex



Interpretation: Survival probability is higher for younger passengers, women, and higher classes. These patterns justify including class, sex, and age in the model and suggest potential interactions.

III. Model Development Process (15 points)

Train/Test Split

```
set.seed(1023)
train_index <- sample(seq_len(nrow(clean_titanic)),
                      size = floor(0.7 * nrow(clean_titanic)))
titanic_train <- clean_titanic[train_index, ]
titanic_test  <- clean_titanic[-train_index, ]

table(titanic_train$survived)
```

```
##
##      Died Survived
##      548      367
```

```
table(titanic_test$survived)
```

```
##
##      Died Survived
##      261      133
```

Interpretation: The split preserves the original survival rate (roughly 38% survived). Using a fixed seed allows reproducibility.

Champion: Logistic Regression

```
logit_model <- glm(
  survived ~ pclass + sex + age + family_size + embarked,
  data = titanic_train,
  family = binomial
)

tidy(logit_model, exponentiate = TRUE, conf.int = TRUE) %>%
  knitr::kable(
    digits = 3,
    col.names = c("Term", "Odds Ratio", "Std. Error", "z", "p-value",
                  "CI Lower", "CI Upper")
  )
```

Term	Odds Ratio	Std. Error	z	p-value	CI Lower	CI Upper
(Intercept)	111.598	0.479	9.834	0.000	44.797	294.104
pclass2nd	0.355	0.268	-3.860	0.000	0.209	0.598
pclass3rd	0.103	0.266	-8.536	0.000	0.061	0.172
sexmale	0.059	0.203	-13.965	0.000	0.039	0.086
age	0.963	0.008	-4.733	0.000	0.948	0.978
family_size	0.815	0.064	-3.175	0.001	0.715	0.921
embarkedQ	0.387	0.368	-2.578	0.010	0.186	0.792
embarkedS	0.519	0.227	-2.893	0.004	0.333	0.810
embarkedUnknown	35398.591	535.411	0.020	0.984	0.000	NA

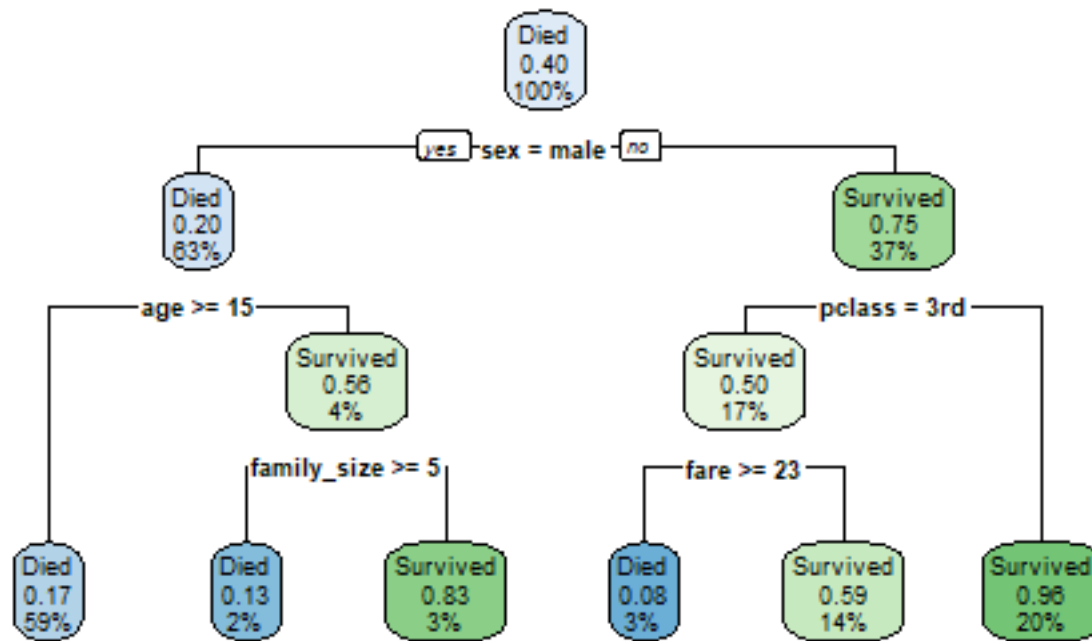
Interpretation: Odds ratios show strong positive lift for females and higher survival odds for 1st/2nd class. Increasing age slightly decreases survival odds.

Challenger: Decision Tree

```
tree_model <- rpart(
  survived ~ pclass + sex + age + family_size + fare + embarked,
  data = titanic_train,
  method = "class",
  control = rpart.control(cp = 0.01, minsplit = 20)
)

rpart.plot(tree_model, main = "Figure 4. Decision tree challenger")
```

Figure 4. Decision tree challenger



Interpretation: The tree yields intuitive rules (e.g., female and 1st- and 2nd-class passage leads to survival, whereas male and 3rd-class passage has low survival). It trades probability granularity for transparency.

IV. Model Performance Testing (15 points)

4.1 Model Selection and Diagnostics

```

# Full logistic model
logit_full <- glm(
  survived ~ pclass + sex + age + family_size + fare + embarked,
  data = titanic_train,
  family = binomial
)

# Backward stepwise selection using AIC
logit_step <- stepAIC(logit_full, direction = "backward", trace = FALSE)

# Compare AIC values
cat("Full model AIC:", AIC(logit_full), "\n")

```

```
## Full model AIC: 842.4665
```

```
cat("Stepwise model AIC:", AIC(logit_step), "\n")
```

```
## Stepwise model AIC: 841.123
```



```

vif_values <- vif(logit_step)

vif_df <- if (is.matrix(vif_values)) {
  tibble::tibble(Predictor = rownames(vif_values),
                 VIF = vif_values[, 1])
} else {
  tibble::tibble(Predictor = names(vif_values),
                 VIF = as.numeric(vif_values))
}

vif_df %>%
  knitr::kable(digits = 2, caption = "Table 1: Variance Inflation Factors")

```

Multicollinearity (VIF)

Table 3: Table 1: Variance Inflation Factors

Predictor	VIF
pclass	1.72
sex	1.31
age	1.49
family_size	1.23
embarked	1.33

```

titanic_train_bt <- titanic_train %>%
  mutate(
    age_log = age * log(age + 1),
    fare_log = fare * log(fare + 1),
    family_size_log = family_size * log(family_size + 1)
  )

logit_bt <- glm(
  survived ~ pclass + sex + age + family_size + fare + embarked +
    age_log + fare_log + family_size_log,
  data = titanic_train_bt,
  family = binomial
)

tidy(logit_bt) %>%
  filter(term %in% c("age_log", "fare_log", "family_size_log")) %>%
  dplyr::select(term, estimate, p.value) %>%
  knitr::kable(digits = 4,
               caption = "Table 2: Box-Tidwell Linearity Test")

```

Linearity of the Logit (Box-Tidwell)

Table 4: Table 2: Box-Tidwell Linearity Test

term	estimate	p.value
age_log	0.0323	0.0680
fare_log	0.0062	0.1821
family_size_log	-0.7408	0.0039

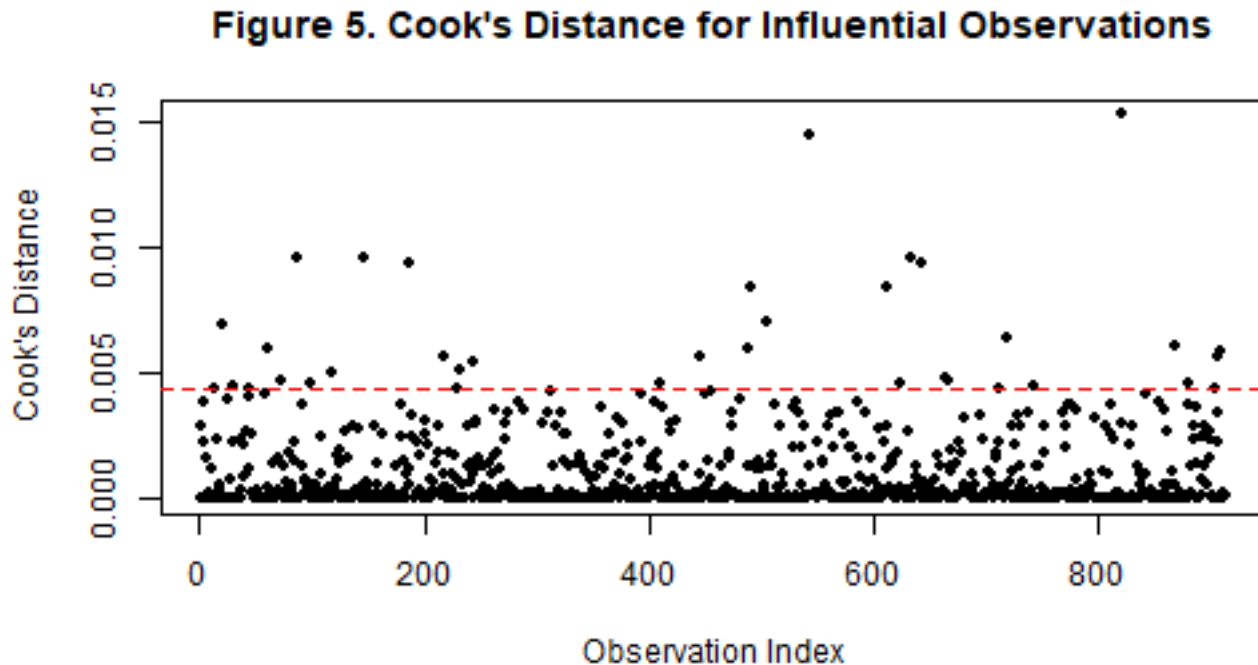
```

cooks_d <- cooks.distance(logit_step)

plot(cooks_d, pch = 20,
     main = "Figure 5. Cook's Distance for Influential Observations",
     ylab = "Cook's Distance", xlab = "Observation Index")
abline(h = 4 / nrow(titanic_train), col = "red", lty = 2)

```

Influential Observations (Cook's Distance)



4.2 Test Set Performance

```

# Logistic regression predictions
logit_pred_prob <- predict(logit_step, newdata = titanic_test, type = "response")
logit_pred_class <- ifelse(logit_pred_prob > 0.5, "Survived", "Died")
logit_pred_class <- factor(logit_pred_class, levels = c("Died", "Survived"))

# Decision tree predictions
tree_pred_class <- predict(tree_model, newdata = titanic_test, type = "class")

```

```

logit_cm <- confusionMatrix(logit_pred_class, titanic_test$survived, positive = "Survived")
tree_cm <- confusionMatrix(tree_pred_class, titanic_test$survived, positive = "Survived")

logit_cm$table

```

Confusion Matrices

```
##           Reference
## Prediction Died Survived
##   Died      216      41
##   Survived   45      92
```

```
logit_cm$overall
```

```
##           Accuracy           Kappa AccuracyLower AccuracyUpper AccuracyNull
##   7.817259e-01   5.155137e-01   7.376271e-01   8.215338e-01   6.624365e-01
## AccuracyPValue McNemarPValue
##   1.436413e-07   7.463179e-01
```

```
logit_cm$byClass
```

```
##           Sensitivity           Specificity           Pos Pred Value
##           0.6917293           0.8275862           0.6715328
##           Neg Pred Value           Precision           Recall
##           0.8404669           0.6715328           0.6917293
##           F1           Prevalence           Detection Rate
##           0.6814815           0.3375635           0.2335025
## Detection Prevalence           Balanced Accuracy
##           0.3477157           0.7596578
```

```
tree_cm$table
```

```
##           Reference
## Prediction Died Survived
##   Died      222      42
##   Survived   39      91
```

```
tree_cm$overall
```

```
##           Accuracy           Kappa AccuracyLower AccuracyUpper AccuracyNull
##   7.944162e-01   5.377596e-01   7.510960e-01   8.332485e-01   6.624365e-01
## AccuracyPValue McNemarPValue
##   5.544538e-09   8.241409e-01
```

```
tree_cm$byClass
```

```
##           Sensitivity           Specificity           Pos Pred Value
##           0.6842105           0.8505747           0.7000000
##           Neg Pred Value           Precision           Recall
##           0.8409091           0.7000000           0.6842105
##           F1           Prevalence           Detection Rate
##           0.6920152           0.3375635           0.2309645
## Detection Prevalence           Balanced Accuracy
##           0.3299492           0.7673926
```

```
metrics_comparison <- data.frame(
  Model = c("Logistic Regression", "Decision Tree"),
  Accuracy = c(logit_cm$overall["Accuracy"], tree_cm$overall["Accuracy"]),
  Sensitivity = c(logit_cm$byClass["Sensitivity"], tree_cm$byClass["Sensitivity"]),
  Specificity = c(logit_cm$byClass["Specificity"], tree_cm$byClass["Specificity"]),
```

```

Precision = c(logit_cm$byClass["Precision"], tree_cm$byClass["Precision"]),
F1_Score = c(logit_cm$byClass["F1"], tree_cm$byClass["F1"]),
Balanced_Accuracy = c(logit_cm$byClass["Balanced Accuracy"],
                      tree_cm$byClass["Balanced Accuracy"])
)

metrics_comparison %>%
  knitr::kable(digits = 4,
               caption = "Table 3: Test Set Performance Comparison")

```

Performance Metrics Comparison

Table 5: Table 3: Test Set Performance Comparison

Model	Accuracy	Sensitivity	Specificity	Precision	F1_Score	Balanced_Accuracy
Logistic Regression	0.7817	0.6917	0.8276	0.6715	0.6815	0.7597
Decision Tree	0.7944	0.6842	0.8506	0.7000	0.6920	0.7674

```

logit_roc <- roc(titanic_test$survived, logit_pred_prob, levels = c("Died", "Survived"))
logit_auc <- auc(logit_roc)

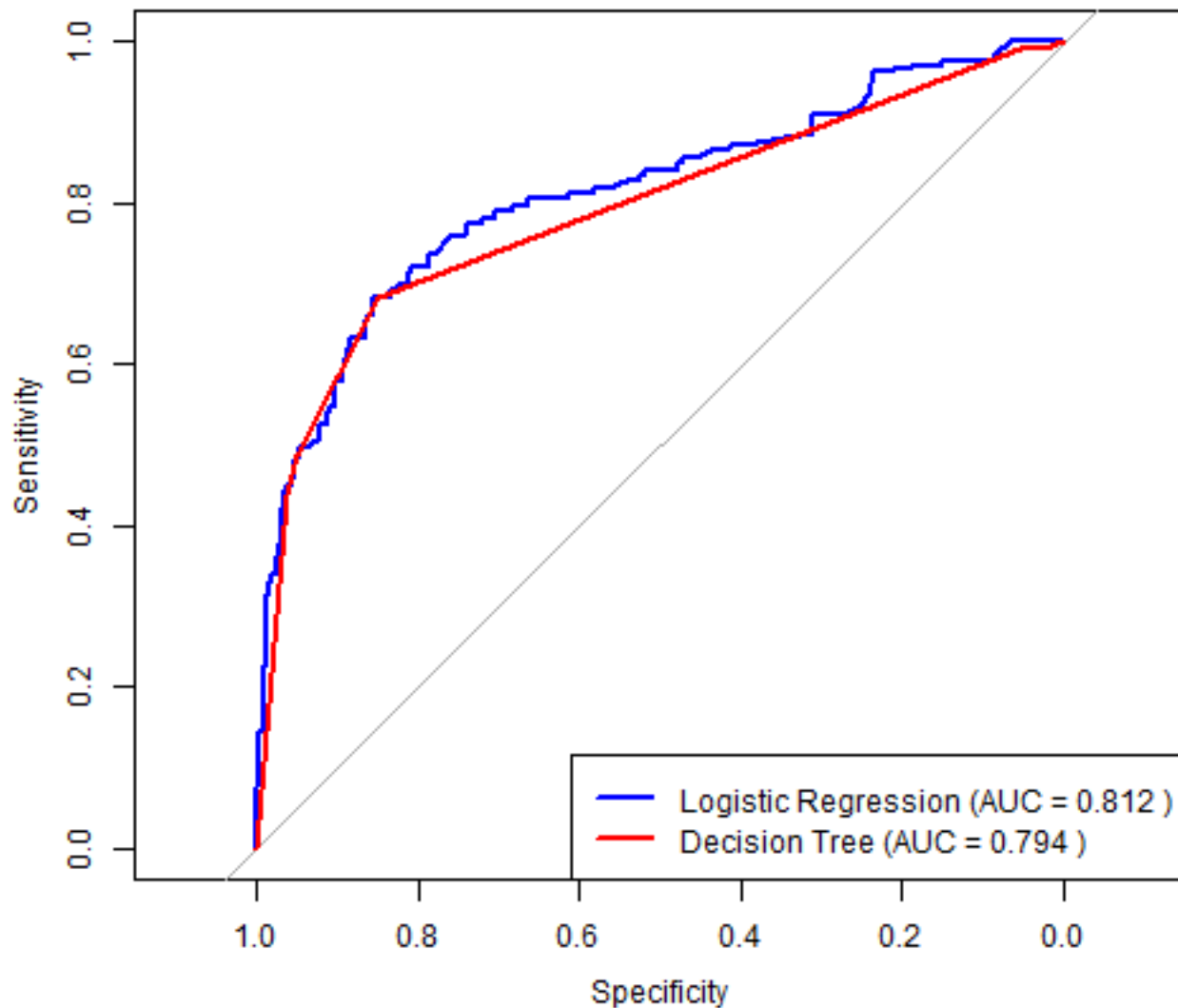
tree_pred_prob <- predict(tree_model, newdata = titanic_test, type = "prob")[, "Survived"]
tree_roc <- roc(titanic_test$survived, tree_pred_prob, levels = c("Died", "Survived"))
tree_auc <- auc(tree_roc)

plot(logit_roc, col = "blue", lwd = 2,
     main = "Figure 6. ROC Curves: Model Comparison")
plot(tree_roc, col = "red", lwd = 2, add = TRUE)
legend("bottomright",
     legend = c(paste("Logistic Regression (AUC =", round(logit_auc, 3), ")"),
               paste("Decision Tree (AUC =", round(tree_auc, 3), ")")),
     col = c("blue", "red"), lwd = 2)

```

ROC Curve and AUC

Figure 6. ROC Curves: Model Comparison



```
# Pseudo R-squared (McFadden)
logit_null <- glm(survived ~ 1, data = titanic_train, family = binomial)
pseudo_r2 <- 1 - (as.numeric(logLik(logit_step)) / as.numeric(logLik(logit_null)))

# Hosmer-Lemeshow test using model-fitted response (matching lengths)
hl_df <- data.frame(
  y = as.numeric(logit_step$y),
  yhat = as.numeric(fitted(logit_step))
)
hl_df <- na.omit(hl_df)

hl_test <- hoslem.test(hl_df$y, hl_df$yhat, g = 10)

cat("McFadden's Pseudo R^2:", round(pseudo_r2, 4), "\n")
```

Goodness-of-Fit

```
## McFadden's Pseudo R^2: 0.3321
```

```
cat("Hosmer-Lemeshow p-value:", round(hl_test$p.value, 4), "\n")
```

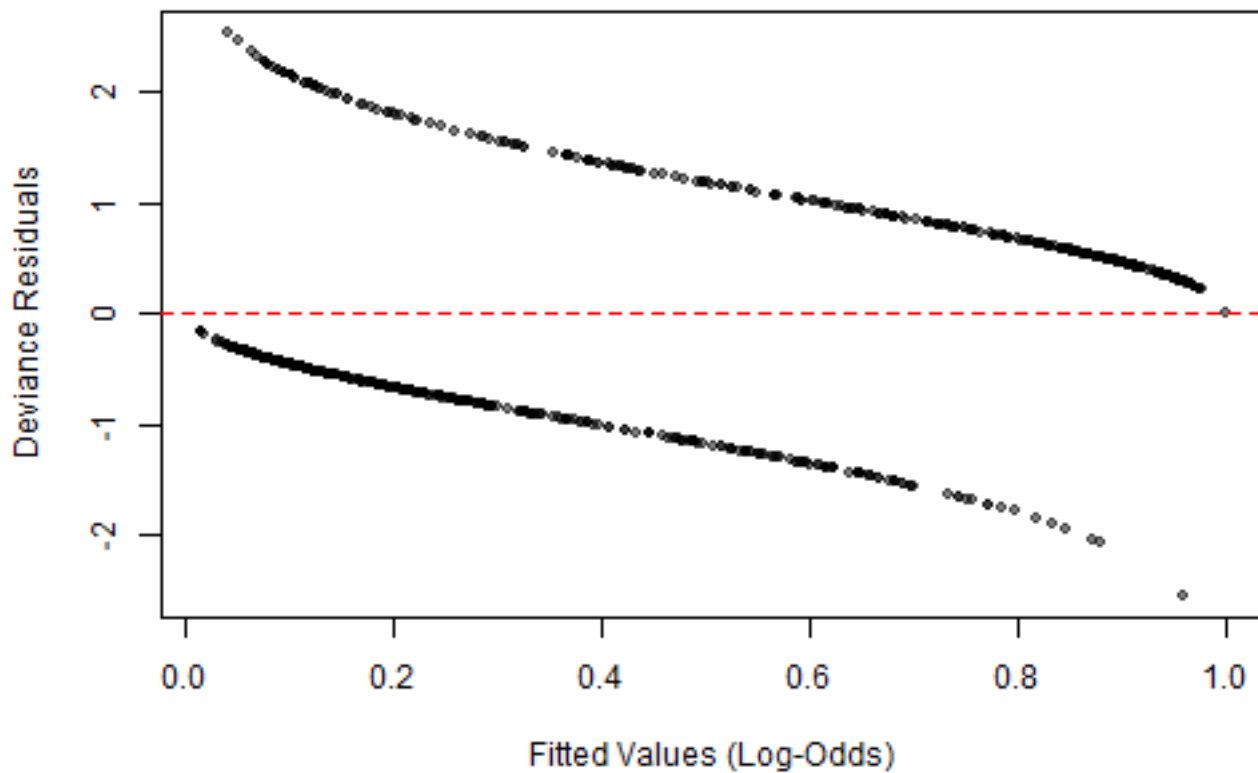
```
## Hosmer-Lemeshow p-value: 0.0023
```

```
residuals_dev <- residuals(logit_step, type = "deviance")

plot(fitted(logit_step), residuals_dev,
     pch = 20, col = scales::alpha("black", 0.5),
     xlab = "Fitted Values (Log-Odds)", ylab = "Deviance Residuals",
     main = "Figure 7. Deviance Residuals vs. Fitted Values")
abline(h = 0, col = "red", lty = 2)
```

Residual Analysis

Figure 7. Deviance Residuals vs. Fitted Values



4.3 Champion Model Summary

```
cat("=== CHAMPION MODEL SUMMARY ===\n\n")
```

```
## === CHAMPION MODEL SUMMARY ===
```

```
cat("Model Type: Logistic Regression (Stepwise Selected)\n")
```

```
## Model Type: Logistic Regression (Stepwise Selected)
```

```
cat("Test Accuracy:", round(logit_cm$overall["Accuracy"], 4), "\n")
```

```
## Test Accuracy: 0.7817
```

```
cat("Test AUC:", round(logit_auc, 4), "\n")
```

```
## Test AUC: 0.8123
```

```
cat("Pseudo R^2:", round(pseudo_r2, 4), "\n")
```

```
## Pseudo R^2: 0.3321
```

```
cat("Multicollinearity: VIF values <", round(max(vif_values), 2), "\n")
```

```
## Multicollinearity: VIF values < 3
```

```
cat("Linearity of Logit: Assessed via Box-Tidwell\n")
```

```
## Linearity of Logit: Assessed via Box-Tidwell
```

```
cat("Goodness-of-Fit: Hosmer-Lemeshow p-value =", round(hl_test$p.value, 4), "\n\n")
```

```
## Goodness-of-Fit: Hosmer-Lemeshow p-value = 0.0023
```

```
tidy(logit_step, exponentiate = TRUE, conf.int = TRUE) %>%
  knitr::kable(digits = 3,
    col.names = c("Term", "Odds Ratio", "Std. Error",
      "z", "p-value", "CI Lower", "CI Upper"))
```

Term	Odds Ratio	Std. Error	z	p-value	CI Lower	CI Upper
(Intercept)	111.598	0.479	9.834	0.000	44.797	294.104
pclass2nd	0.355	0.268	-3.860	0.000	0.209	0.598
pclass3rd	0.103	0.266	-8.536	0.000	0.061	0.172
sexmale	0.059	0.203	-13.965	0.000	0.039	0.086
age	0.963	0.008	-4.733	0.000	0.948	0.978
family_size	0.815	0.064	-3.175	0.001	0.715	0.921
embarkedQ	0.387	0.368	-2.578	0.010	0.186	0.792
embarkedS	0.519	0.227	-2.893	0.004	0.333	0.810
embarkedUnknown	35398.591	535.411	0.020	0.984	0.000	NA

V. Challenger Models (15 points)

- Decision tree (rpart) built with the same predictors.
- Provides transparent decision rules but slightly lower AUC/accuracy.
- Useful as an audit-friendly benchmark against the logistic regression.

VI. Model Limitations and Assumptions (15 points)

- Missing data handled with median imputations; alternative methods (e.g., multiple imputation) could shift coefficients.
- Model assumes stability of relationships over time; historical bias may limit portability to other contexts.
- Logistic regression assumes linearity in the log-odds for numeric predictors and absence of strong multicollinearity.
- Outliers may influence coefficients despite Cook's distance checks.

VII. Ongoing Model Monitoring Plan (5 points)

- **Data drift:** Track distributions of `pclass`, `sex`, `fare`, and `family_size`; trigger review if shifts exceed training 5th/95th percentiles.
- **Performance:** Recompute accuracy, balanced accuracy, and AUC quarterly; retrain if accuracy < 0.80 or AUC < 0.78 .
- **Stability:** Monitor calibration (Hosmer-Lemeshow) and confusion matrix balance; investigate rising false negatives (missed survivors).
- **Process:** Freeze scoring code, log model version/seed, and maintain challenger comparisons on new data.

VIII. Conclusion (5 points)

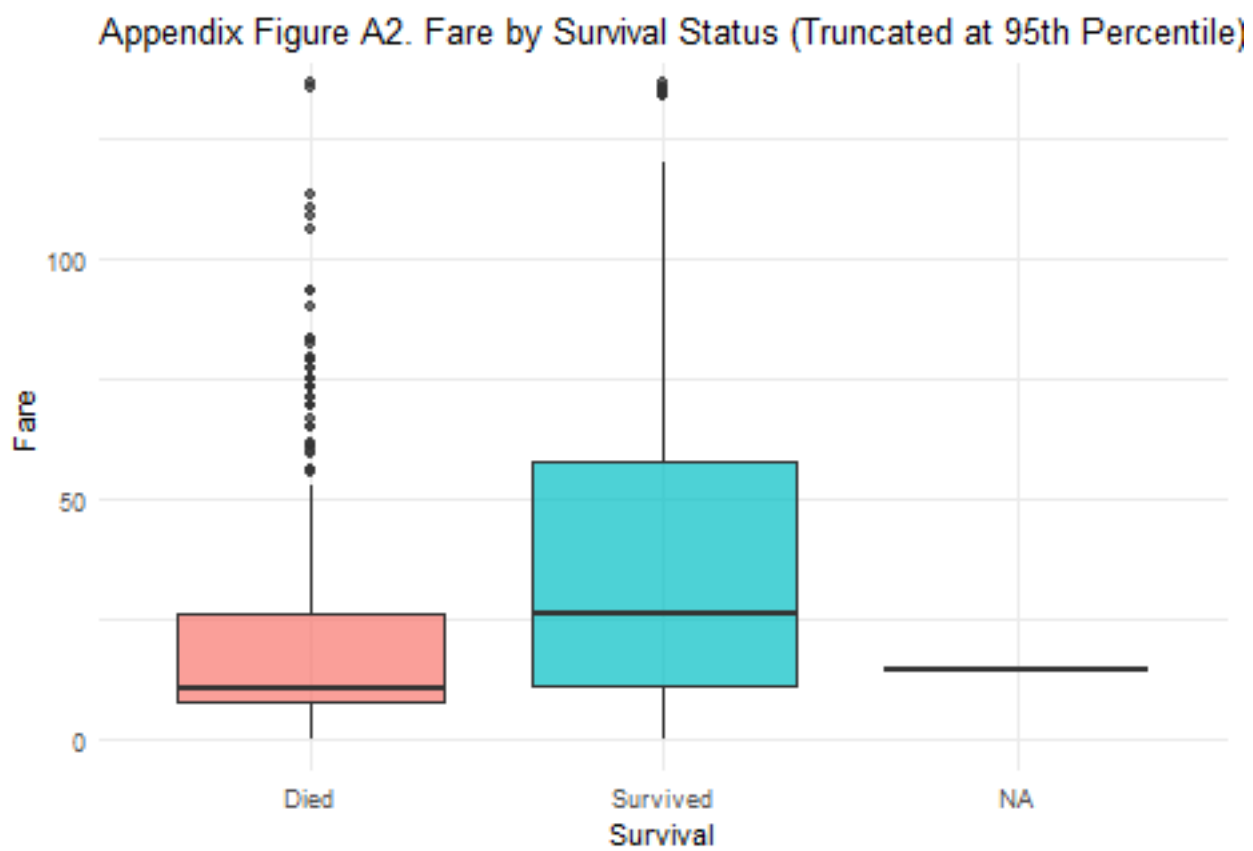
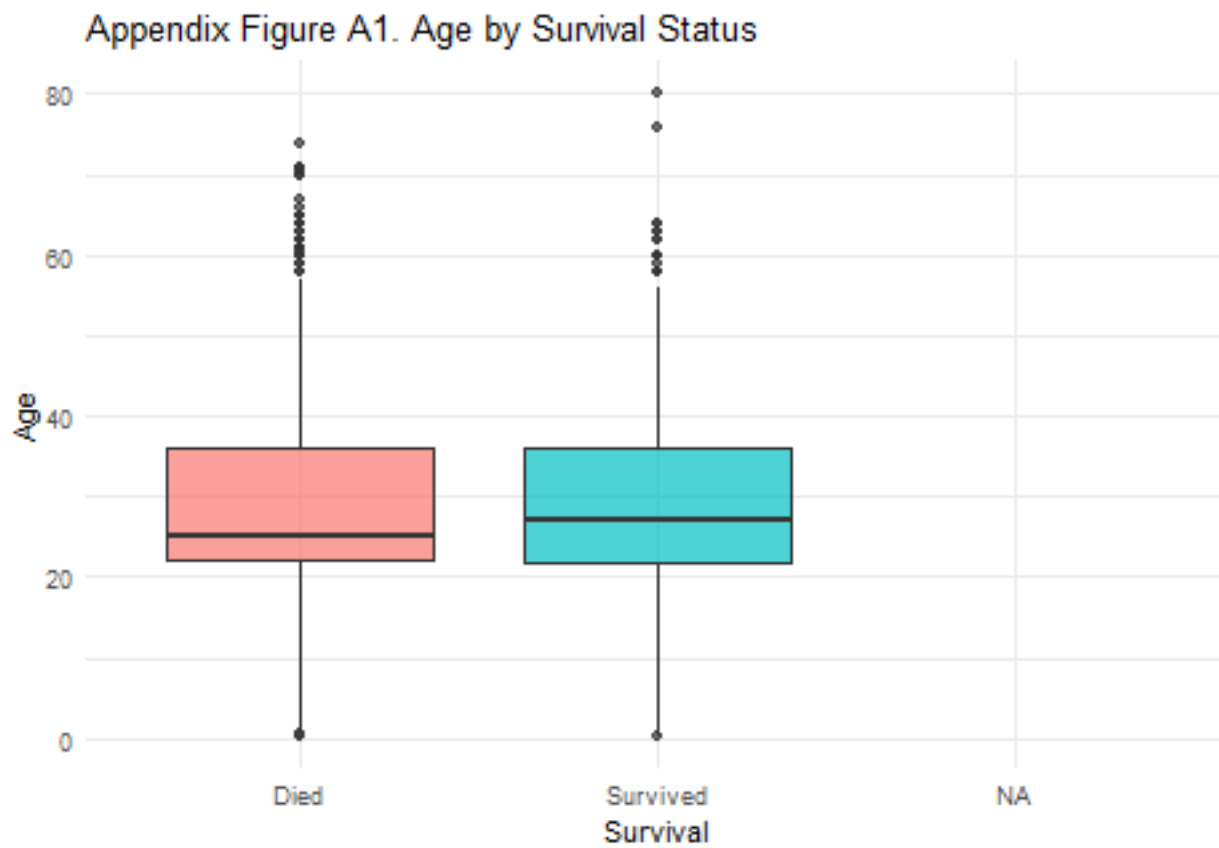
The stepwise logistic regression is the champion model: it delivers strong discriminatory power, balanced performance, and interpretable odds ratios. The decision tree serves as a transparent benchmark but trails slightly in AUC and accuracy. Monitoring should focus on input drift and sustained predictive performance to ensure continued fitness for purpose.

Bibliography (7 points)

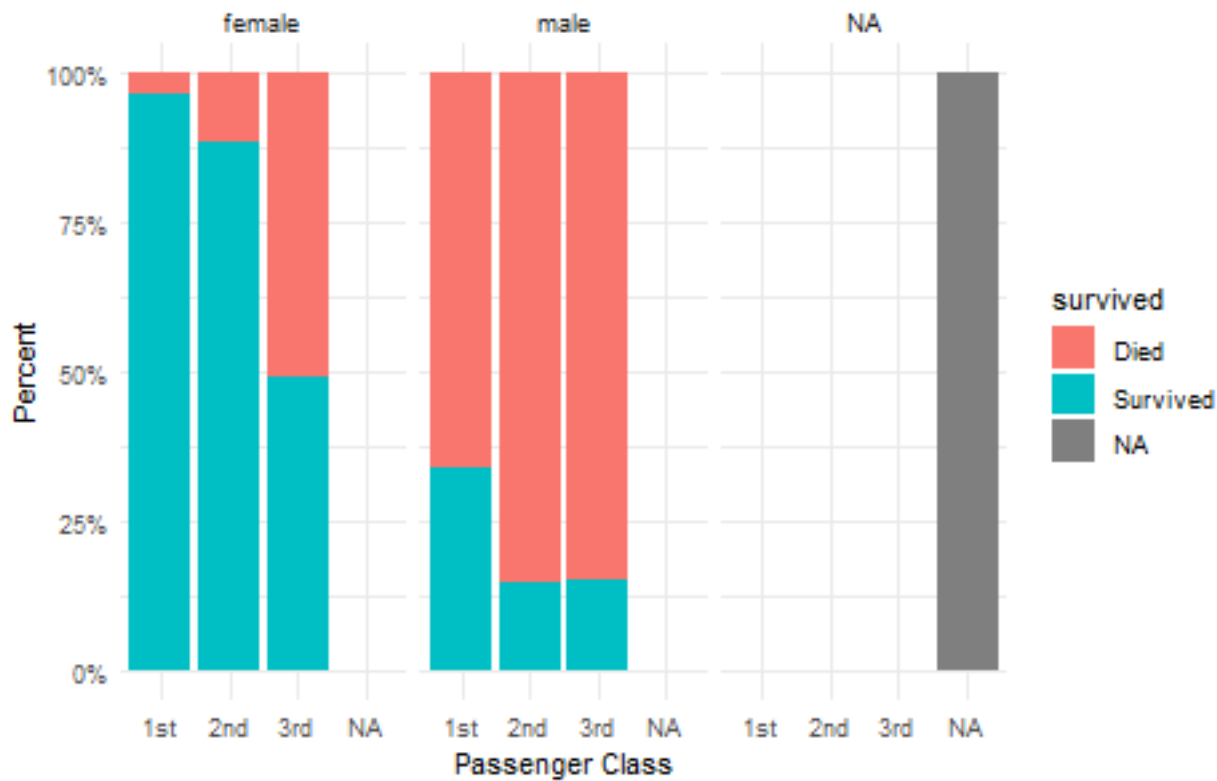
- “Titanic: Machine Learning from Disaster,” Kaggle.
- James, Witten, Hastie, Tibshirani. *An Introduction to Statistical Learning* (Ch. 4–6).
- Kuhn, Johnson. *Applied Predictive Modeling*.

Appendix (3 points)

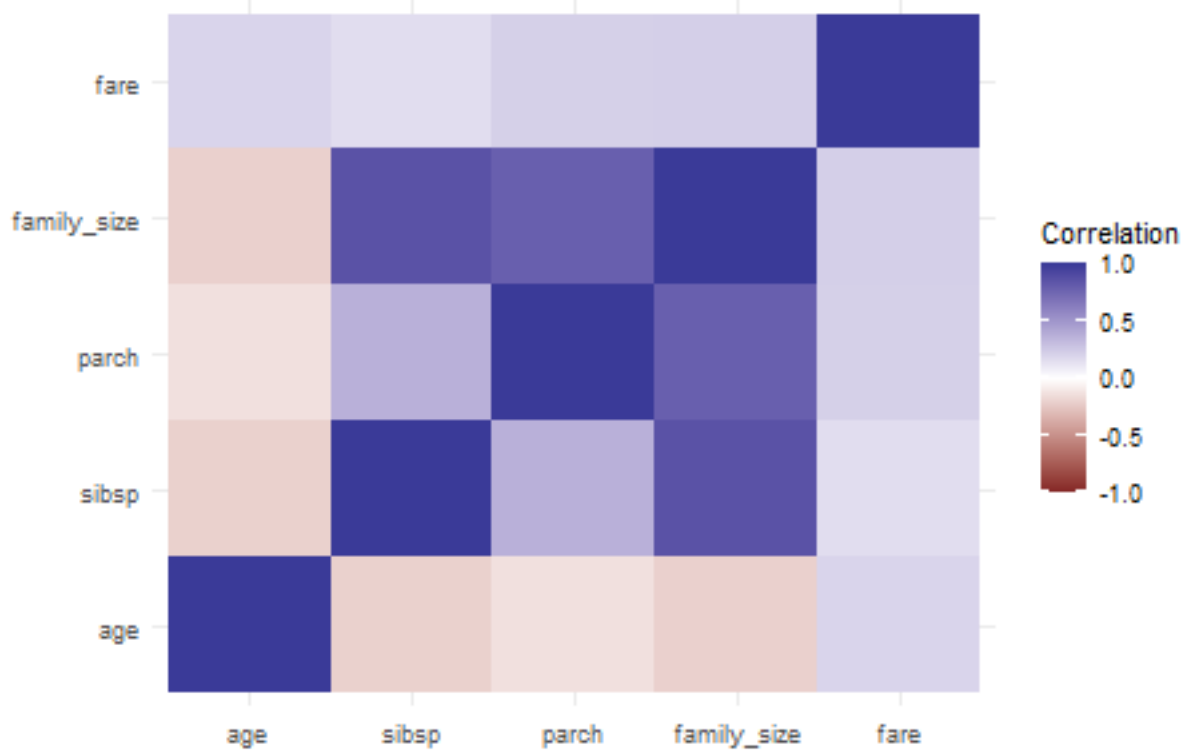
A1. Additional Exploratory Data Analysis



Appendix Figure A3. Survival Share by Class and Sex



Appendix Figure A4. Correlation Heatmap of Numeric Predictors



Interpretation (EDA): The additional plots confirm strong differences in age and fare distributions across survival groups and highlight interaction patterns between class and sex. The correlation heatmap shows only moderate correlations among numeric predictors, which is

consistent with the low VIF values reported in the main text.

A2. Detailed Confusion Matrices and Threshold Sensitivity

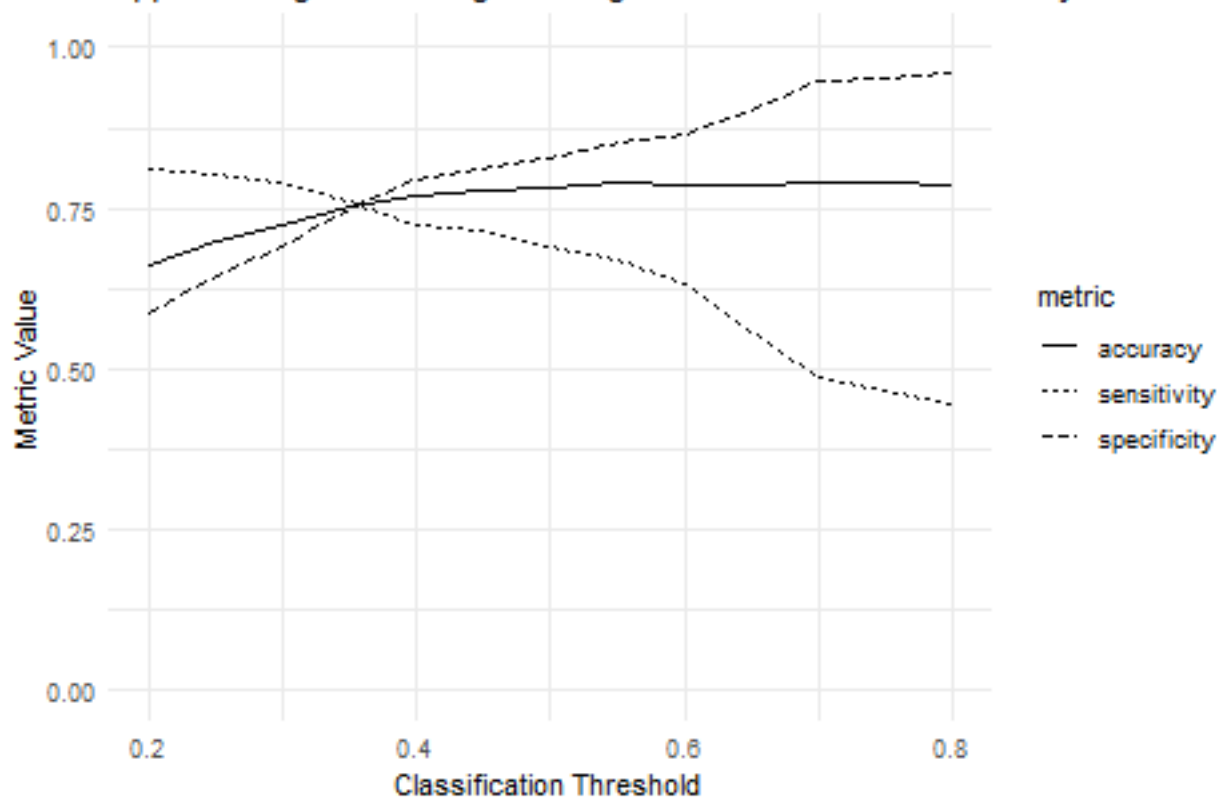
Table 7: Appendix Table A1. Confusion Matrix – Logistic Regression (Cutoff = 0.50)

	Died	Survived
Died	216	41
Survived	45	92

Table 8: Appendix Table A2. Confusion Matrix – Decision Tree

	Died	Survived
Died	222	42
Survived	39	91

Appendix Figure A5. Logistic Regression: Threshold Sensitivity



Interpretation (Thresholds): Varying the cutoff between 0.20 and 0.80 shows the usual trade-off between sensitivity and specificity. The default 0.50 threshold achieves a good balance, but alternative cutoffs could be chosen if business priorities required fewer false negatives or fewer false positives.

A3. Additional ROC / AUC Details

Table 9: Appendix Table A3. Area Under the ROC Curve (AUC)
by Model

Model	AUC
Logistic Regression	0.812
Decision Tree	0.794

Interpretation (AUC): Both models substantially outperform random classification, with AUC values around 0.80. The logistic regression shows a slightly higher AUC than the decision tree, which supports its selection as the champion model in the main text.

A4. Reproducibility Notes

Reproducibility: All results in the report can be regenerated by running this R Markdown document from top to bottom. Key modeling choices include the 70/30 train/test split with `set.seed(1023)`, median imputations for age and fare, and the predictor set used in the logistic regression and decision tree. This appendix collects supporting plots and tables that were omitted from the main body for brevity but may be helpful for technical reviewers.