Capstone

Neha Deshpande

2025-02-24

```
options(repos = c(CRAN = "https://cloud.r-project.org/"))
install.packages("e1071")
```

```
##
## The downloaded binary packages are in
##  /var/folders/b7/wn07yt7x0z3gmqf1172s33080000gn/T//RtmpOSRjV1/downloaded_packages
```

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(ggplot2)
install.packages("e1071")
```

```
##
## The downloaded binary packages are in
##  /var/folders/b7/wn07yt7x0z3gmqf1172s33080000gn/T//RtmpOSRjV1/downloaded_packages
```

```
install.packages("caret")
```

```
##
## The downloaded binary packages are in
##  /var/folders/b7/wn07yt7x0z3gmqf1172s33080000gn/T//RtmpOSRjV1/downloaded_packages
```

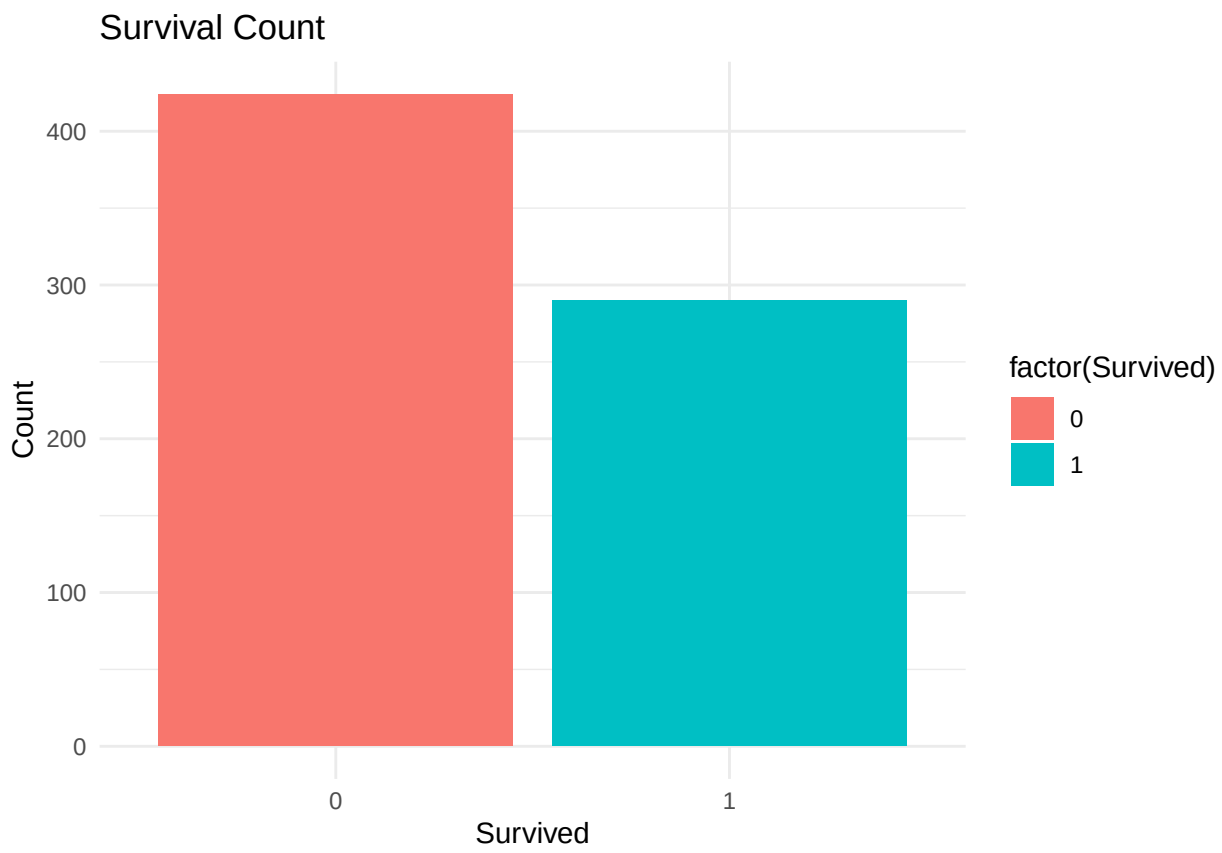
```
library(readr)
clean_test <- read_csv("Downloads/Kaggle/clean_test.csv")
```

```
## Rows: 418 Columns: 11
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (3): Name, Ticket, Cabin
## dbl (8): PassengerId, Pclass, Sex, Age, SibSp, Parch, Fare, Embarked
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

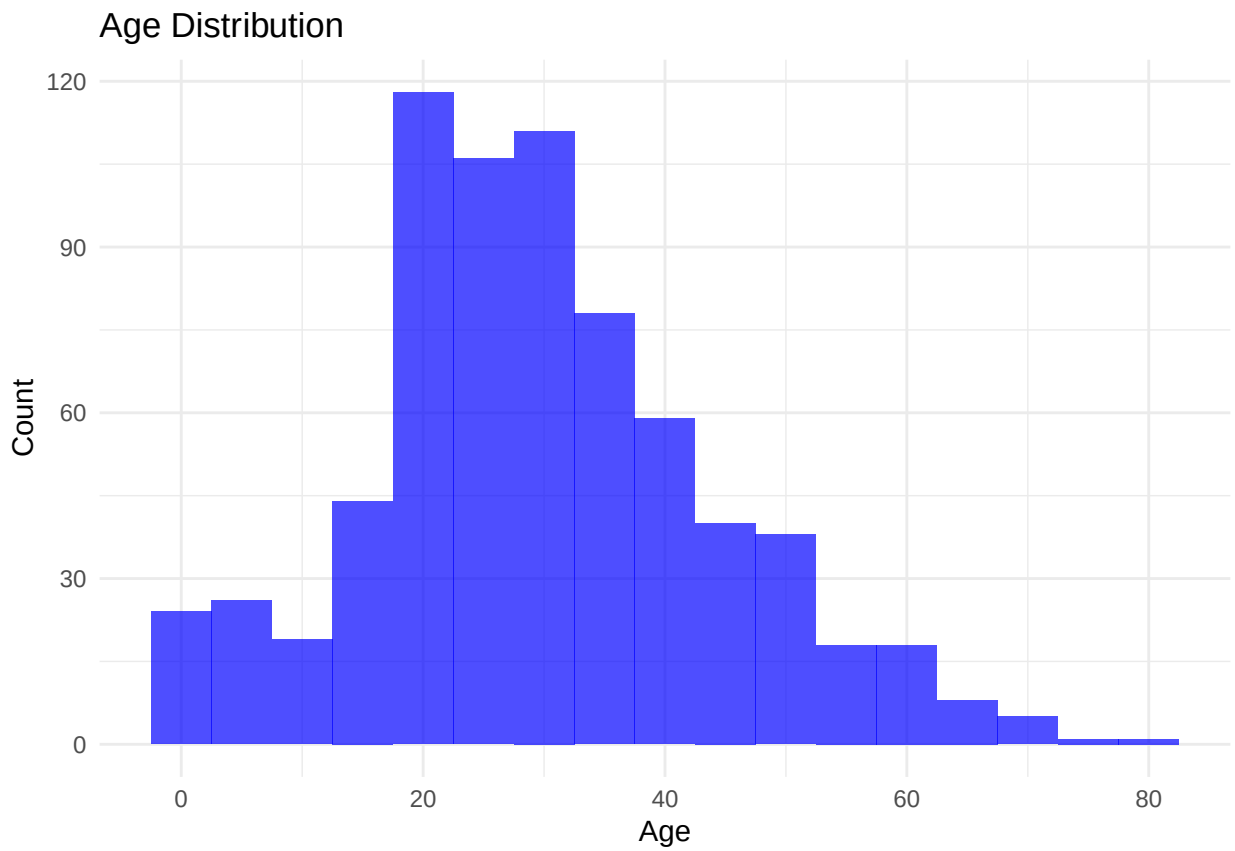
```
clean_train <- read_csv("Downloads/Kaggle/clean_train.csv")
```

```
## Rows: 714 Columns: 12
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (3): Name, Ticket, Cabin
## dbl (9): PassengerId, Survived, Pclass, Sex, Age, SibSp, Parch, Fare, Embarked
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
clean_train %>%
  group_by(Survived) %>%
  summarise(Count = n()) %>%
  ggplot(aes(x = factor(Survived), y = Count, fill = factor(Survived))) +
  geom_bar(stat = "identity") +
  labs(title = "Survival Count", x = "Survived", y = "Count") +
  theme_minimal()
```

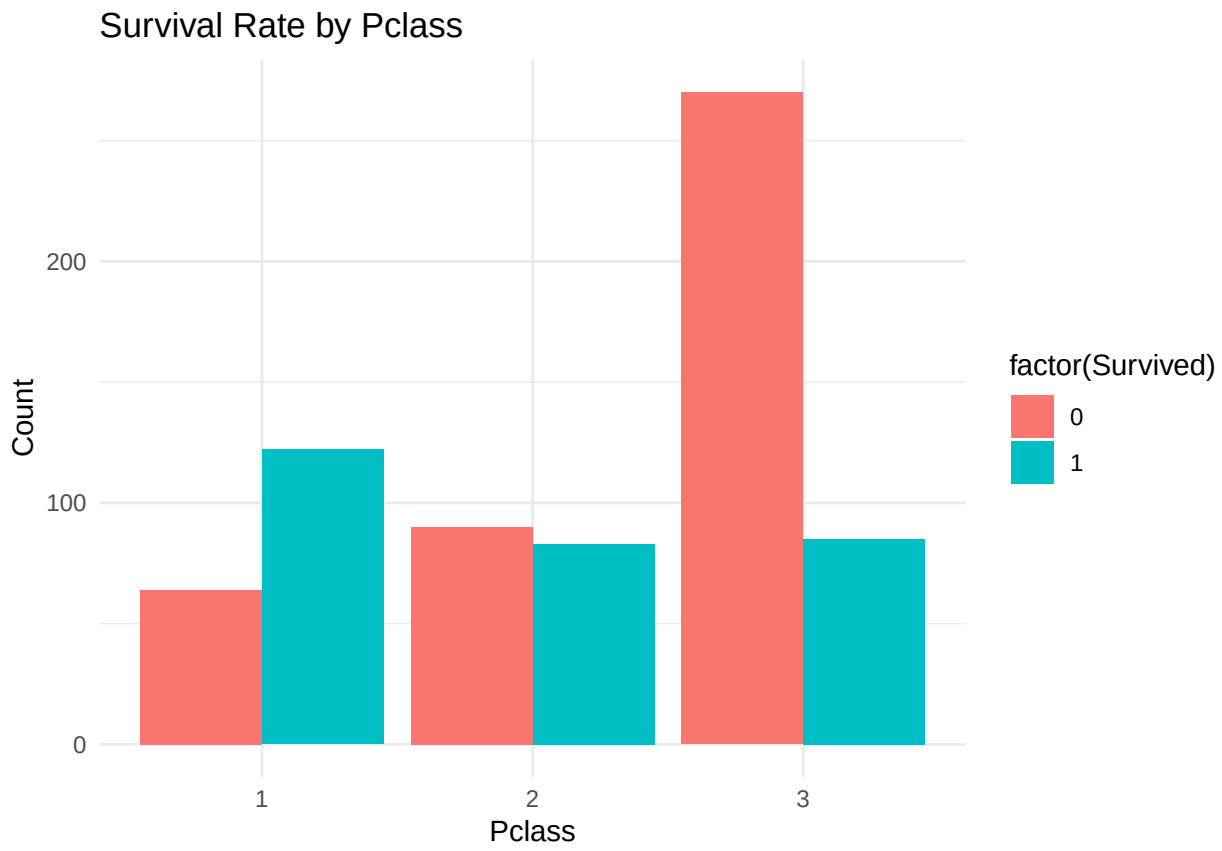



```
ggplot(clean_train, aes(x = Age)) +
  geom_histogram(binwidth = 5, fill = "blue", alpha = 0.7) +
  labs(title = "Age Distribution", x = "Age", y = "Count") +
  theme_minimal()
```



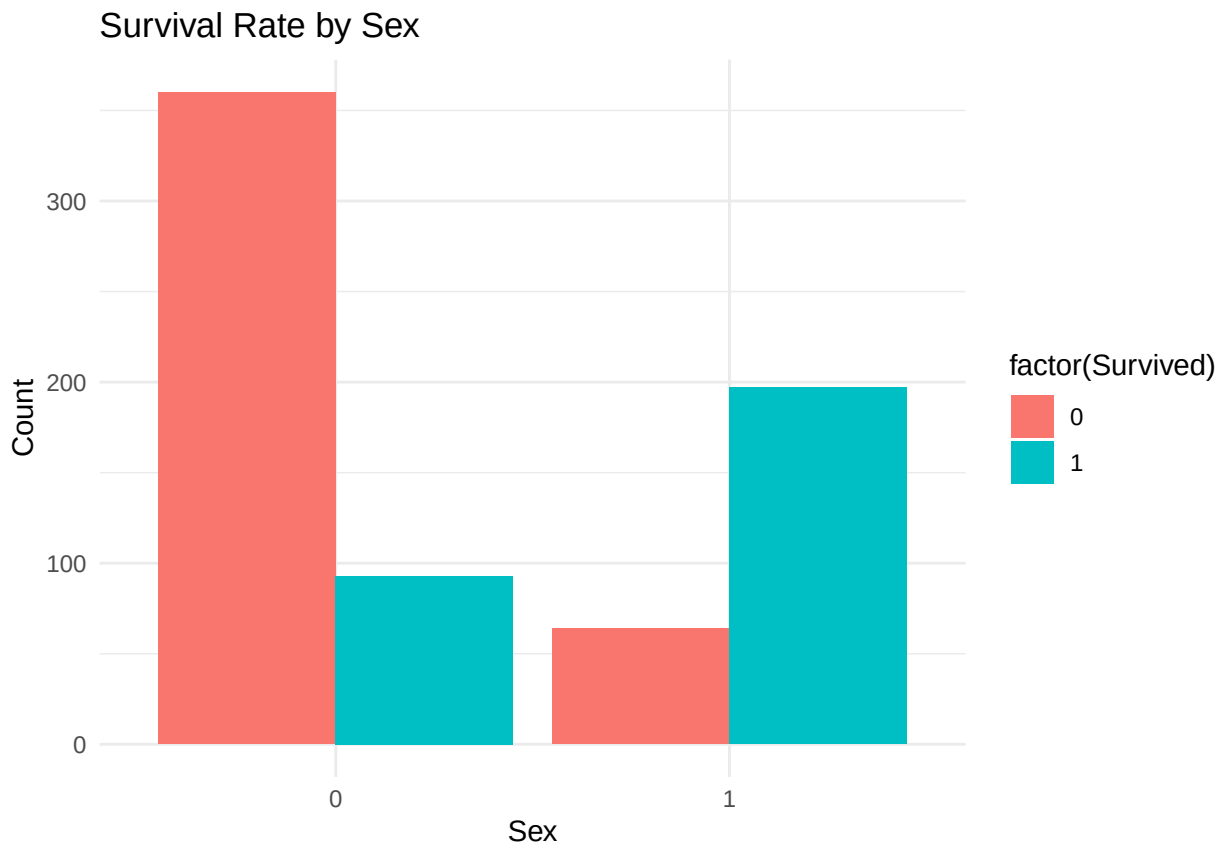

```
clean_train %>%
  group_by(Pclass, Survived) %>%
  summarise(Count = n()) %>%
  ggplot(aes(x = factor(Pclass), y = Count, fill = factor(Survived))) +
  geom_bar(stat = "identity", position = "dodge") +
  labs(title = "Survival Rate by Pclass", x = "Pclass", y = "Count") +
  theme_minimal()
```

```
## `summarise()` has grouped output by 'Pclass'. You can override using the
## `.groups` argument.
```

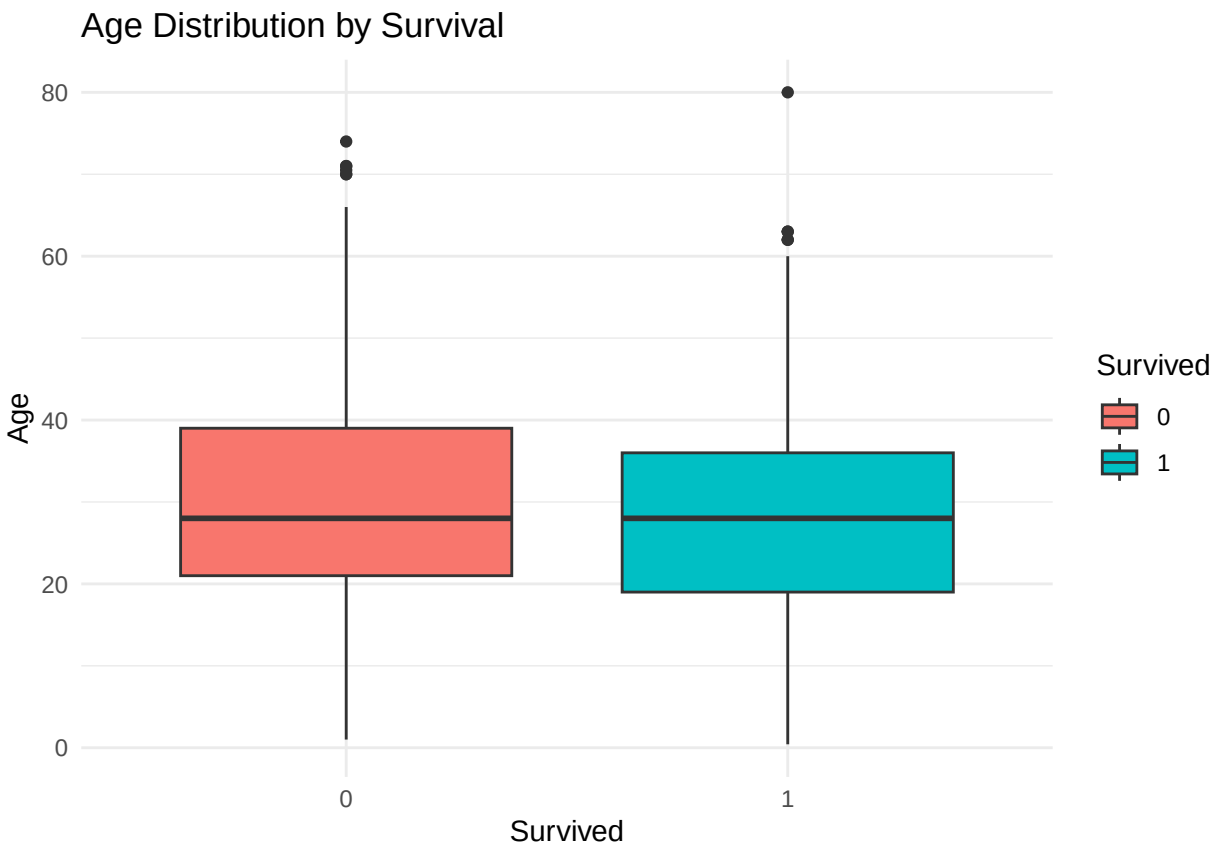



```
clean_train %>%
  group_by(Sex, Survived) %>%
  summarise(Count = n()) %>%
  ggplot(aes(x = factor(Sex), y = Count, fill = factor(Survived))) +
  geom_bar(stat = "identity", position = "dodge") +
  labs(title = "Survival Rate by Sex", x = "Sex", y = "Count") +
  theme_minimal()
```

```
## `summarise()` has grouped output by 'Sex'. You can override using the `.groups`
## argument.
```




```
ggplot(clean_train, aes(x = factor(Survived), y = Age, fill = factor(Survived))) +
  geom_boxplot() +
  labs(title = "Age Distribution by Survival", x = "Survived", y = "Age", fill = "Survived") +
  theme_minimal()
```



First, we started by cleaning the dataset that was provided by Kaggle. Cleaning included the NA, missing, and irrelevant values. Then we printed the summary of the dataset by using summary(clean_train). An important part of data analysis is using exploratory techniques to get an idea of the background of the data. I used several visualizations to show the different patterns and trends that were found in the dataset. The first visualization is a bar plot that shows survival counts. This allows us to see the overall proprtion of surviors and non-survivors. This is followed by a histogram of passenger ages, which helps identify age distributions and possible patterns related to survival. Another grouped bar plot shows survival rates across passenger classes (pclass), revealing that first-class passengers had a higher chance of survival compared to lower classes. Another plot illustrates survival rates by gender, confirming that female passengers had a significantly higher survival probability than males.

```
features <- c("Pclass", "Sex", "Age", "SibSp", "Parch", "Fare", "Embarked")

clean_train <- clean_train %>% select(all_of(features), Survived)

clean_test <- clean_test %>% select(all_of(features), PassengerId)

clean_train$Sex <- as.factor(clean_train$Sex)
clean_train$Embarked <- as.factor(clean_train$Embarked)
```

```r
clean_train$Survived <- as.factor(clean_train$Survived)
clean_test$Sex <- as.factor(clean_test$Sex)
clean_test$Embarked <- as.factor(clean_test$Embarked)

clean_train <- na.omit(clean_train)
clean_test <- na.omit(clean_test)

library(e1071)
svm_model <- svm(Survived ~ ., data = clean_train, kernel = "linear", cost = 1)
svm_model
```

```
##
## Call:
## svm(formula = Survived ~ ., data = clean_train, kernel = "linear",
##     cost = 1)
##
##
## Parameters:
##    SVM-Type:  C-classification
##  SVM-Kernel:  linear
##        cost:  1
##
## Number of Support Vectors:  338
```

```r
predictions <- predict(svm_model, clean_test)

predicted_survival <- as.numeric(predictions) - 1

table(clean_train$Survived)
```

```
##
##   0   1
## 424 290
```

```r
prop.table(table(clean_train$Survived))
```

```
##
##         0         1
## 0.5938375 0.4061625
```

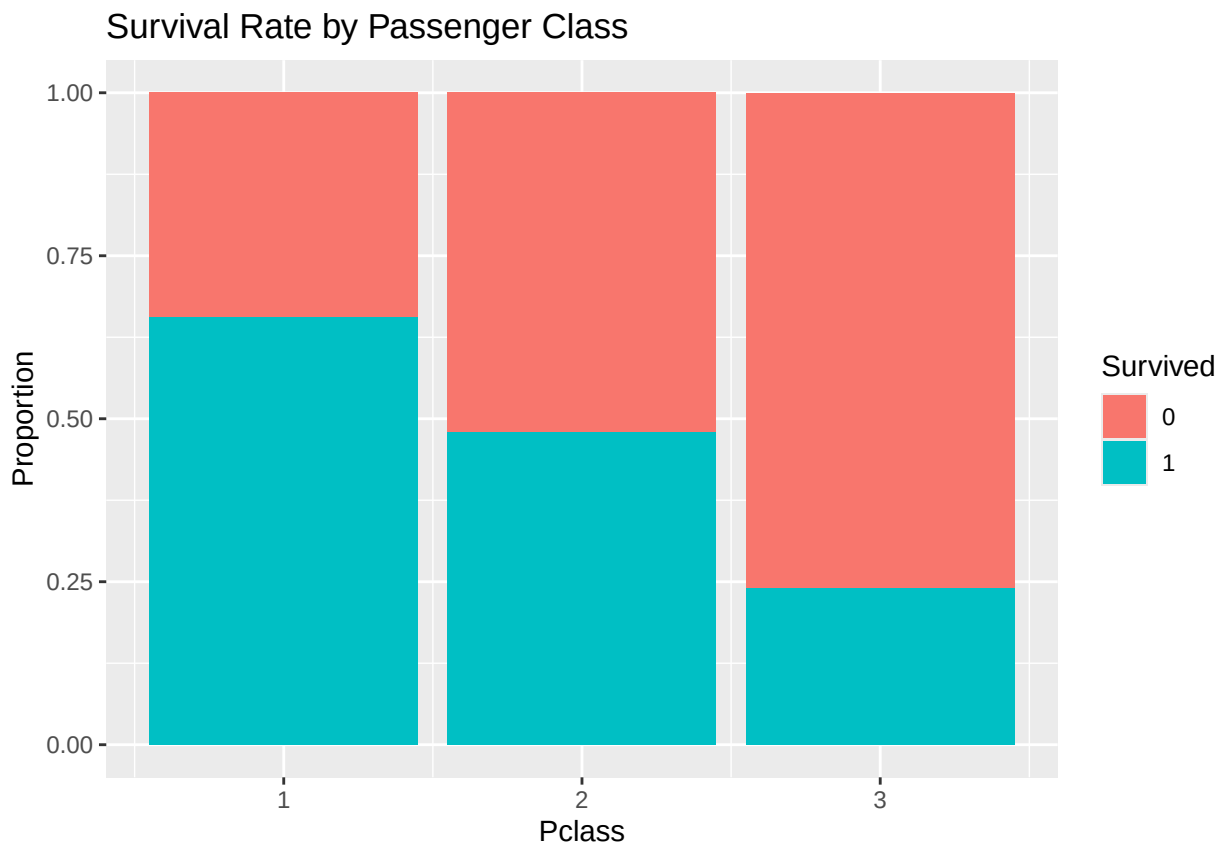
```r
summary(svm_model)
```

```
##
## Call:
## svm(formula = Survived ~ ., data = clean_train, kernel = "linear",
##     cost = 1)
##
##
## Parameters:
##    SVM-Type:  C-classification
##  SVM-Kernel:  linear
```

```
##        cost:  1
##
## Number of Support Vectors:  338
##
##  ( 168 170 )
##
##
## Number of Classes:  2
##
## Levels:
##  0 1
```

```r
library(ggplot2)

ggplot(clean_train, aes(x = Pclass, fill = Survived)) +
  geom_bar(position = "fill") +
  labs(title = "Survival Rate by Passenger Class", y = "Proportion", x = "Pclass")
```




```r
if ("Sex_female" %in% colnames(clean_train)) {
  ggplot(clean_train, aes(x = Sex_female, fill = Survived)) +
    geom_bar(position = "fill") +
    labs(title = "Survival Rate by Gender (Female)", y = "Proportion", x = "Female (1 = Yes, 0 = No)")
}

getwd()
```

```
## [1] "/Users/nehadeshpande"
```

```
submission <- data.frame(PassengerId = clean_test$PassengerId, Survived = predicted_survival)
submission
```

```
##     PassengerId Survived
## 1           892        0
## 2           893        1
## 3           894        0
## 4           895        0
## 5           896        1
## 6           897        0
## 7           898        1
## 8           899        0
## 9           900        1
## 10          901        0
## 11          903        0
## 12          904        1
## 13          905        0
## 14          906        1
## 15          907        1
## 16          908        0
## 17          909        0
## 18          910        1
## 19          911        1
## 20          912        0
## 21          913        0
## 22          915        0
## 23          916        1
## 24          917        0
## 25          918        1
## 26          919        0
## 27          920        0
## 28          922        0
## 29          923        0
## 30          924        1
## 31          926        0
## 32          927        0
## 33          929        1
## 34          930        0
## 35          932        0
## 36          934        0
## 37          935        1
## 38          936        1
## 39          937        0
## 40          938        0
## 41          940        1
## 42          941        1
## 43          942        0
## 44          943        0
## 45          944        1
## 46          945        1
## 47          947        0
## 48          948        0
```

```
## 49          949        0
## 50          951        1
## 51          952        0
## 52          953        0
## 53          954        0
## 54          955        1
## 55          956        0
## 56          958        1
## 57          959        0
## 58          960        0
## 59          961        1
## 60          962        1
## 61          963        0
## 62          964        1
## 63          965        0
## 64          966        1
## 65          967        0
## 66          969        1
## 67          970        0
## 68          971        1
## 69          972        0
## 70          973        0
## 71          974        0
## 72          978        1
## 73          979        1
## 74          981        0
## 75          982        1
## 76          984        1
## 77          986        0
## 78          987        0
## 79          988        1
## 80          989        0
## 81          990        1
## 82          991        0
## 83          992        1
## 84          993        0
## 85          995        0
## 86          996        1
## 87          997        0
## 88          998        0
## 89         1001        0
## 90         1002        0
## 91         1004        1
## 92         1005        1
## 93         1006        1
## 94         1007        0
## 95         1009        1
## 96         1010        0
## 97         1011        1
## 98         1012        1
## 99         1014        1
## 100        1015        0
## 101        1017        1
## 102        1018        0
```

```
## 103        1020        0
## 104        1021        0
## 105        1022        0
## 106        1023        0
## 107        1026        0
## 108        1027        0
## 109        1028        0
## 110        1029        0
## 111        1030        1
## 112        1031        0
## 113        1032        1
## 114        1033        1
## 115        1034        0
## 116        1035        0
## 117        1036        0
## 118        1037        0
## 119        1039        0
## 120        1041        0
## 121        1042        1
## 122        1045        1
## 123        1046        0
## 124        1047        0
## 125        1048        1
## 126        1049        1
## 127        1050        0
## 128        1051        1
## 129        1053        0
## 130        1054        1
## 131        1056        0
## 132        1057        1
## 133        1058        0
## 134        1059        0
## 135        1061        1
## 136        1063        0
## 137        1064        0
## 138        1066        0
## 139        1067        1
## 140        1068        1
## 141        1069        0
## 142        1070        1
## 143        1071        1
## 144        1072        0
## 145        1073        0
## 146        1074        1
## 147        1076        1
## 148        1077        0
## 149        1078        1
## 150        1079        0
## 151        1081        0
## 152        1082        0
## 153        1084        0
## 154        1085        0
## 155        1086        0
## 156        1087        0
```

```
## 157        1088        0
## 158        1089        1
## 159        1090        0
## 160        1093        0
## 161        1094        0
## 162        1095        1
## 163        1096        0
## 164        1098        1
## 165        1099        0
## 166        1100        1
## 167        1101        0
## 168        1102        0
## 169        1104        0
## 170        1105        1
## 171        1106        1
## 172        1107        0
## 173        1109        0
## 174        1110        1
## 175        1112        1
## 176        1113        0
## 177        1114        1
## 178        1115        0
## 179        1116        1
## 180        1118        0
## 181        1120        0
## 182        1121        0
## 183        1122        0
## 184        1123        1
## 185        1124        0
## 186        1126        0
## 187        1127        0
## 188        1128        0
## 189        1129        0
## 190        1130        1
## 191        1131        1
## 192        1132        1
## 193        1133        1
## 194        1134        0
## 195        1137        0
## 196        1138        1
## 197        1139        0
## 198        1140        1
## 199        1142        1
## 200        1143        0
## 201        1144        0
## 202        1145        0
## 203        1146        0
## 204        1149        0
## 205        1150        1
## 206        1151        0
## 207        1152        0
## 208        1153        0
## 209        1154        1
## 210        1155        1
```

```
## 211        1156        0
## 212        1161        0
## 213        1162        0
## 214        1164        1
## 215        1167        1
## 216        1168        0
## 217        1169        0
## 218        1170        0
## 219        1171        0
## 220        1172        1
## 221        1173        0
## 222        1175        1
## 223        1176        1
## 224        1177        0
## 225        1179        0
## 226        1183        1
## 227        1185        0
## 228        1186        0
## 229        1187        0
## 230        1188        1
## 231        1190        0
## 232        1191        0
## 233        1192        0
## 234        1194        0
## 235        1195        0
## 236        1197        1
## 237        1198        0
## 238        1199        0
## 239        1200        0
## 240        1201        1
## 241        1202        0
## 242        1203        0
## 243        1205        1
## 244        1206        1
## 245        1207        1
## 246        1208        0
## 247        1209        0
## 248        1210        0
## 249        1211        0
## 250        1212        0
## 251        1213        0
## 252        1214        0
## 253        1215        0
## 254        1216        1
## 255        1217        0
## 256        1218        1
## 257        1219        0
## 258        1220        0
## 259        1221        0
## 260        1222        1
## 261        1223        0
## 262        1225        1
## 263        1226        0
## 264        1227        0
```

```
## 265        1228        0
## 266        1229        0
## 267        1230        0
## 268        1232        0
## 269        1233        0
## 270        1235        1
## 271        1237        1
## 272        1238        0
## 273        1239        1
## 274        1240        0
## 275        1241        1
## 276        1242        1
## 277        1243        0
## 278        1244        0
## 279        1245        0
## 280        1246        1
## 281        1247        0
## 282        1248        1
## 283        1251        1
## 284        1252        0
## 285        1253        1
## 286        1254        1
## 287        1255        0
## 288        1256        1
## 289        1259        1
## 290        1260        1
## 291        1261        0
## 292        1262        0
## 293        1263        1
## 294        1264        0
## 295        1265        0
## 296        1266        1
## 297        1267        1
## 298        1268        1
## 299        1269        0
## 300        1270        0
## 301        1271        0
## 302        1273        0
## 303        1275        1
## 304        1277        1
## 305        1278        0
## 306        1279        0
## 307        1280        0
## 308        1281        0
## 309        1282        0
## 310        1283        1
## 311        1284        0
## 312        1285        0
## 313        1286        0
## 314        1287        1
## 315        1288        0
## 316        1289        1
## 317        1290        0
## 318        1291        0
```

```
## 319        1292        1
## 320        1293        0
## 321        1294        1
## 322        1295        0
## 323        1296        0
## 324        1297        0
## 325        1298        0
## 326        1299        0
## 327        1301        1
## 328        1303        1
## 329        1304        1
## 330        1306        1
## 331        1307        0
```

```
write.csv(submission, "/Users/nehadeshpande/Documents/submission.csv", row.names = FALSE)
```

```
library(caret)
```

```
## Loading required package: lattice
```

```
train_predictions <- predict(svm_model, clean_train)

train_actual <- as.factor(clean_train$Survived)
train_pred <- as.factor(train_predictions)

conf_matrix <- confusionMatrix(train_pred, train_actual)

print(conf_matrix)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 360  93
##          1  64 197
##
##                Accuracy : 0.7801
##                  95% CI : (0.7479, 0.81)
##     No Information Rate : 0.5938
##     P-Value [Acc > NIR] : < 2e-16
##
##                   Kappa : 0.5369
##
##  Mcnemar's Test P-Value : 0.02544
##
##             Sensitivity : 0.8491
##             Specificity : 0.6793
##          Pos Pred Value : 0.7947
##          Neg Pred Value : 0.7548
##              Prevalence : 0.5938
##          Detection Rate : 0.5042
##    Detection Prevalence : 0.6345
```

```
##       Balanced Accuracy : 0.7642
##
##        'Positive' Class : 0
##
```

```
accuracy <- conf_matrix$overall["Accuracy"]
precision <- conf_matrix$byClass["Precision"]
recall <- conf_matrix$byClass["Recall"]
f1_score <- conf_matrix$byClass["F1"]

cat("Model Performance Metrics:\n")
```

```
## Model Performance Metrics:
```

```
cat("Accuracy:", accuracy, "\n")
```

```
## Accuracy: 0.780112
```

```
cat("Precision:", precision, "\n")
```

```
## Precision: 0.794702
```

```
cat("Recall:", recall, "\n")
```

```
## Recall: 0.8490566
```

```
cat("F1-score:", f1_score, "\n")
```

```
## F1-score: 0.8209806
```

The SVM model code starts by loading and preparing the training data, where selected features are extracted and relevant variables are converted into factors, ensuring that categorical data such as gender and port of embarkation are properly handled. Once the data is cleaned, the SVM model is trained using the svm() function from the e1071 package. The formula Survived ~ . specifies that survival status is the target variable, while all other selected features serve as predictors. After training, predictions are made on the test dataset using predict(), and the results are stored. The survival rate distribution is analyzed using table() and prop.table() to understand class proportions. Titanic survival demonstrates an overall accuracy of 78.01%, meaning it correctly classifies survival outcomes in most cases. The confusion matrix shows that the model successfully identified 360 non-survivors (true negatives) and 197 survivors (true positives). However, it also misclassified 64 passengers as survivors who did not survive (false positives) and 93 passengers as non-survivors who actually survived (false negatives).