Genre Classification

Alexandria Simms

2025-07-14

Contents


```
features <- fread("fma_metadata/features.csv", data.table = FALSE) %>%
  janitor::clean_names()

tracks_raw <- read_csv("fma_metadata/tracks.csv", col_names = FALSE)
```

```
## Warning: One or more parsing issues, call `problems()` on your data frame for details
## e.g.:
##   dat <- vroom(...)
##   problems(dat)

## Rows: 106577 Columns: 53
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (52): X2, X3, X4, X5, X6, X7, X8, X9, X10, X11, X12, X13, X14, X15, X16,...
## dbl  (1): X1
##
## 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.
```

```r
header1 <- tracks_raw[1, ] %>% unlist(use.names = FALSE)
header2 <- tracks_raw[2, ] %>% unlist(use.names = FALSE)
column_names <- make.names(paste0(header1, ".", header2), unique = TRUE)
tracks <- tracks_raw[-c(1, 2), ]
colnames(tracks) <- column_names
tracks$track_id <- as.integer(rownames(tracks))
```

```r
tracks_clean <- tracks %>%
  select(track_id, genre_top = `track.genre_top`, subset = `set.subset`) %>%
  filter(!is.na(genre_top), subset == "small") %>%
  mutate(genre_top = as.factor(genre_top))
```

```r
colnames(features)[1] <- "track_id"
features <- features[-1, ]
features$track_id <- as.integer(features$track_id)
```

```
## Warning: NAs introduced by coercion
```

```r
combined_data <- inner_join(tracks_clean, features, by = "track_id")
```

```r
colnames(features)[1] <- "track_id"
features <- features[-1, ]
features$track_id <- as.integer(features$track_id)
combined_data <- inner_join(tracks_clean, features, by = "track_id")
```

"'{r. Normalize Features}

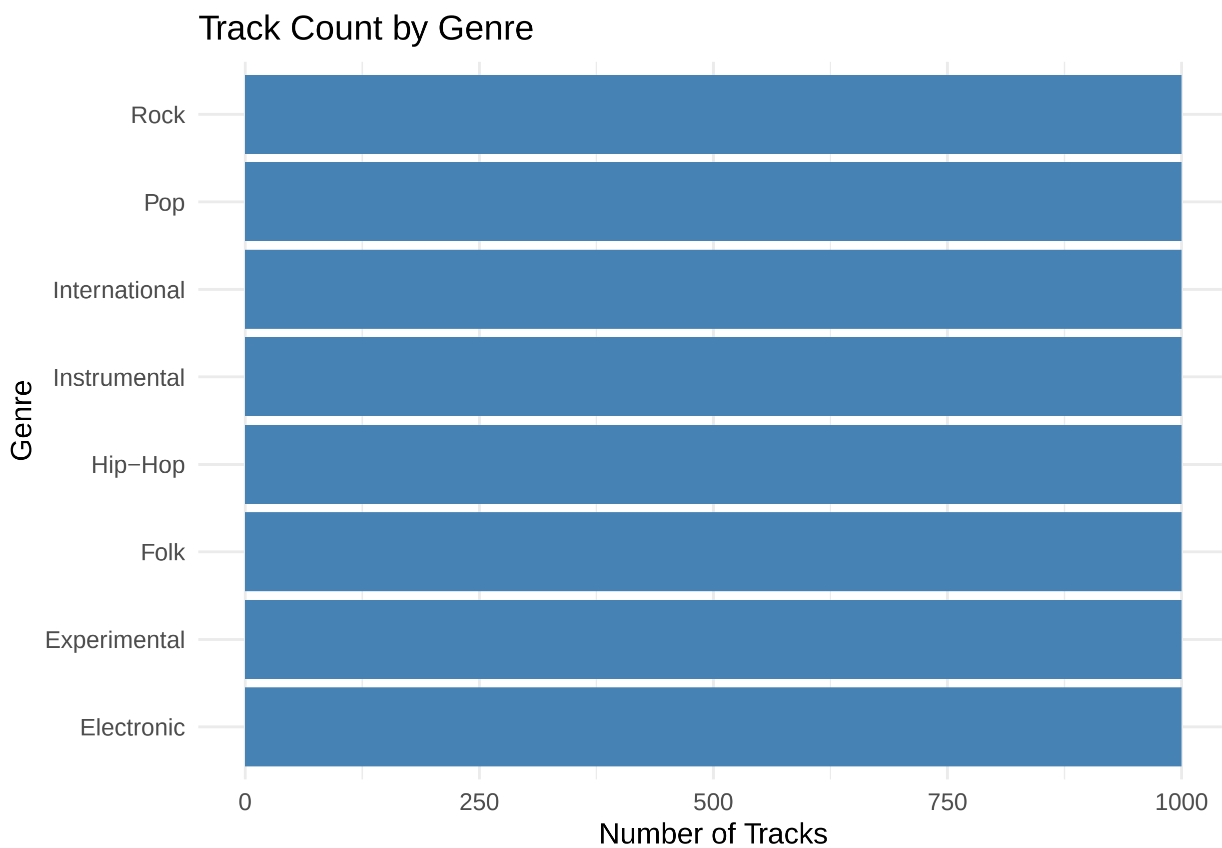
features <- features[-1,] features$track_i d < -as.integer(features$track_id) combined_data <- inner_join(tracks_clean, features, by = "track_id") numeric_features <- combined_data %>% select(-track_id, -subset, -genre_top) %>% select(where(is.numeric))

feature_matrix <- scale(numeric_features)

model_data <- data.frame(genre_top = combined_data$genre_top, feature_matrix)

```
\newpage

# **Exploratory Analysis Alexandria Simms**


''' r
genre_counts <- tracks_clean %>%
  count(genre_top, sort = TRUE)
```

```r
ggplot(genre_counts, aes(x = reorder(genre_top, n), y = n)) +
  geom_bar(stat = "identity", fill = "steelblue") +
  coord_flip() +
  labs(title = "Track Count by Genre",
       x = "Genre", y = "Number of Tracks") +
  theme_minimal()
```




```r
# Double-check subset distribution
table(tracks_clean$subset)
```

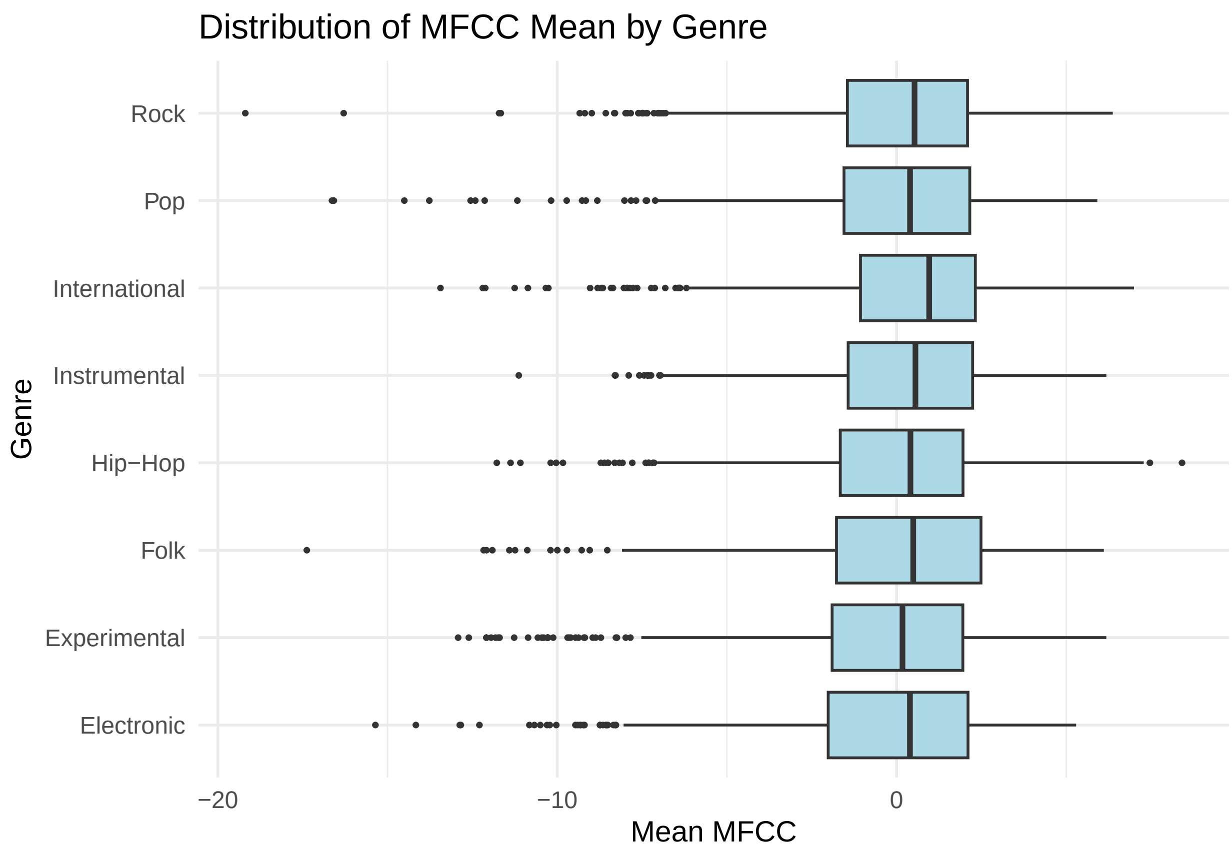
```
##
## small
##  8000
```

```r
# Identify MFCC, chroma, and spectral columns
mfcc_cols <- grep("^mfcc", names(combined_data), value = TRUE)
chroma_cols <- grep("^chroma", names(combined_data), value = TRUE)
spectral_cols <- grep("^spectral", names(combined_data), value = TRUE)

# Convert relevant columns to numeric safely
```

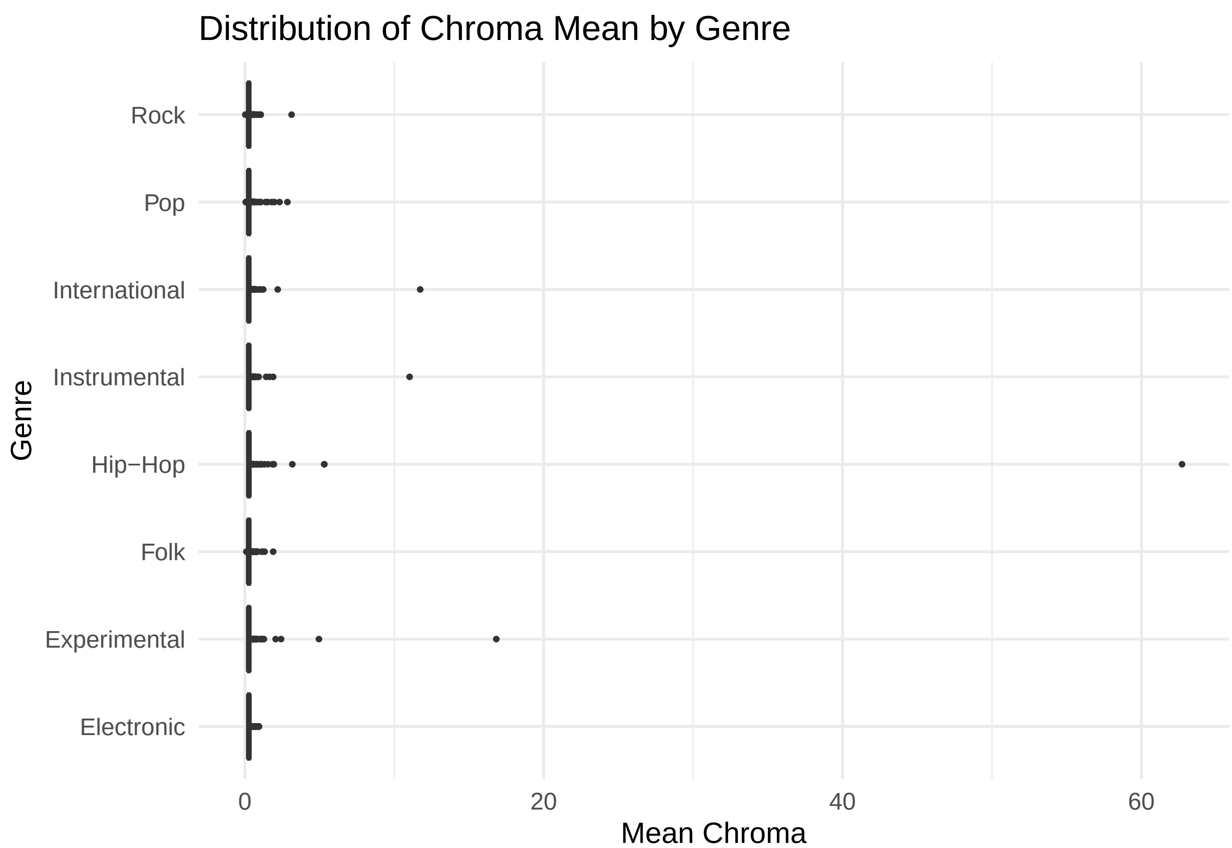
```r
combined_data <- combined_data %>%
  mutate(across(all_of(c(mfcc_cols, chroma_cols, spectral_cols)), ~as.numeric(.)))

# Calculate per-track means
audio_means <- combined_data %>%
  mutate(
    mfcc_mean = rowMeans(select(., all_of(mfcc_cols)), na.rm = TRUE),
    chroma_mean = rowMeans(select(., all_of(chroma_cols)), na.rm = TRUE),
    spectral_mean = rowMeans(select(., all_of(spectral_cols)), na.rm = TRUE)
  ) %>%
  select(genre_top, mfcc_mean, chroma_mean, spectral_mean)

# Make sure mfcc_cols is defined
mfcc_cols <- grep("^mfcc", colnames(combined_data), value = TRUE)

# Convert MFCC columns to numeric if needed
combined_data <- combined_data %>%
  mutate(across(all_of(mfcc_cols), ~as.numeric(.)))

# Create mean column
combined_data <- combined_data %>%
  mutate(mfcc_mean = rowMeans(select(., all_of(mfcc_cols)), na.rm = TRUE))

# Get column names that contain chroma
chroma_cols <- grep("^chroma", colnames(combined_data), value = TRUE)

# Convert to numeric and calculate the mean
combined_data <- combined_data %>%
  mutate(across(all_of(chroma_cols), ~as.numeric(.))) %>%
  mutate(chroma_mean = rowMeans(select(., all_of(chroma_cols)), na.rm = TRUE))

# Get column names that contain spectral
spectral_cols <- grep("^spectral", colnames(combined_data), value = TRUE)

# Convert to numeric and calculate the mean
combined_data <- combined_data %>%
  mutate(across(all_of(spectral_cols), ~as.numeric(.))) %>%
  mutate(spectral_mean = rowMeans(select(., all_of(spectral_cols)), na.rm = TRUE))
```

```r
ggplot(combined_data, aes(x = genre_top, y = mfcc_mean)) +
  geom_boxplot(outlier.size = 0.5, fill = "lightblue") +
  coord_flip() +
  labs(title = "Distribution of MFCC Mean by Genre",
       x = "Genre", y = "Mean MFCC") +
```

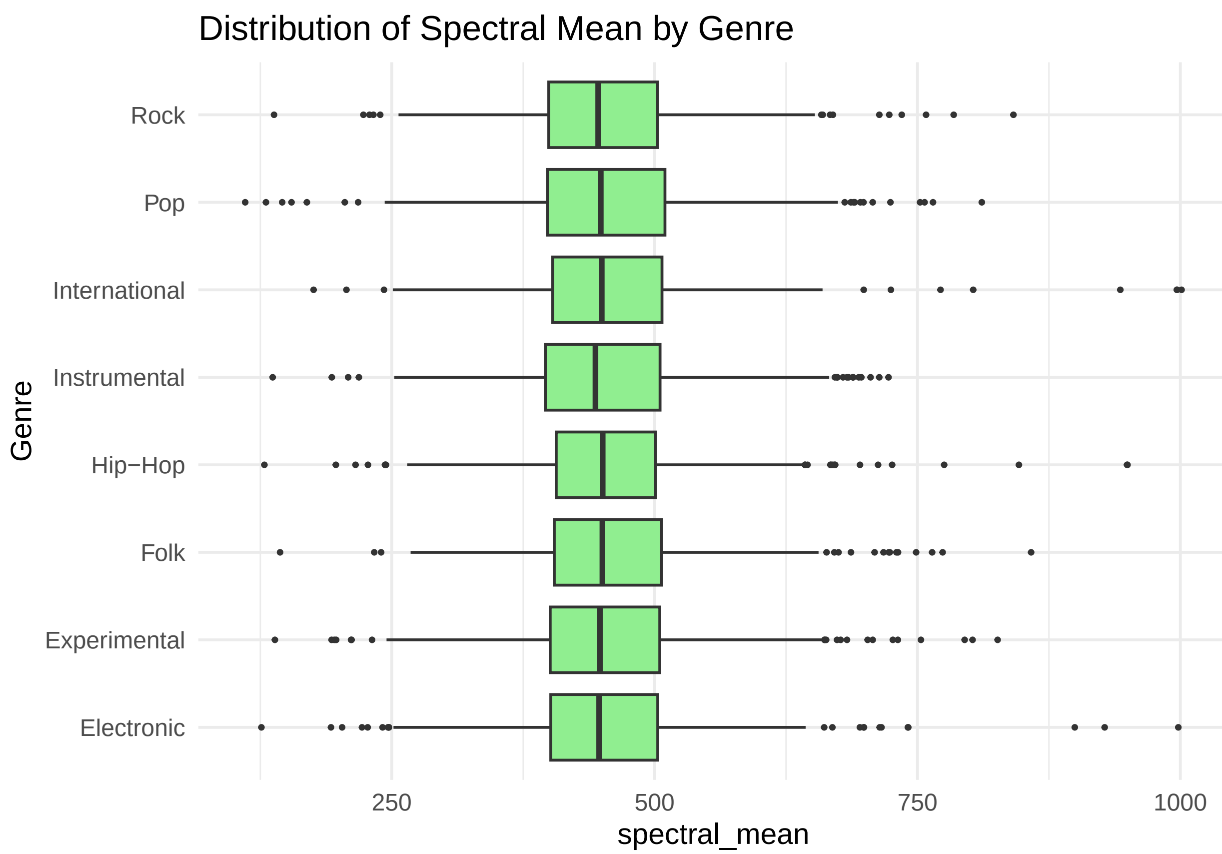
```
  theme_minimal()
```




```
ggplot(combined_data, aes(x = genre_top, y = chroma_mean)) +
  geom_boxplot(outlier.size = 0.5, fill = "lightgreen") +
  coord_flip() +
  labs(title = "Distribution of Chroma Mean by Genre",
       x = "Genre", y = "Mean Chroma") +
  theme_minimal()
```




```r
ggplot(combined_data, aes(x = genre_top, y = spectral_mean)) +
  geom_boxplot(outlier.size = 0.5, fill = "lightgreen") +
  coord_flip() +
  labs(title = "Distribution of Spectral Mean by Genre",
       x = "Genre", y = "spectral_mean") +
  theme_minimal()
```

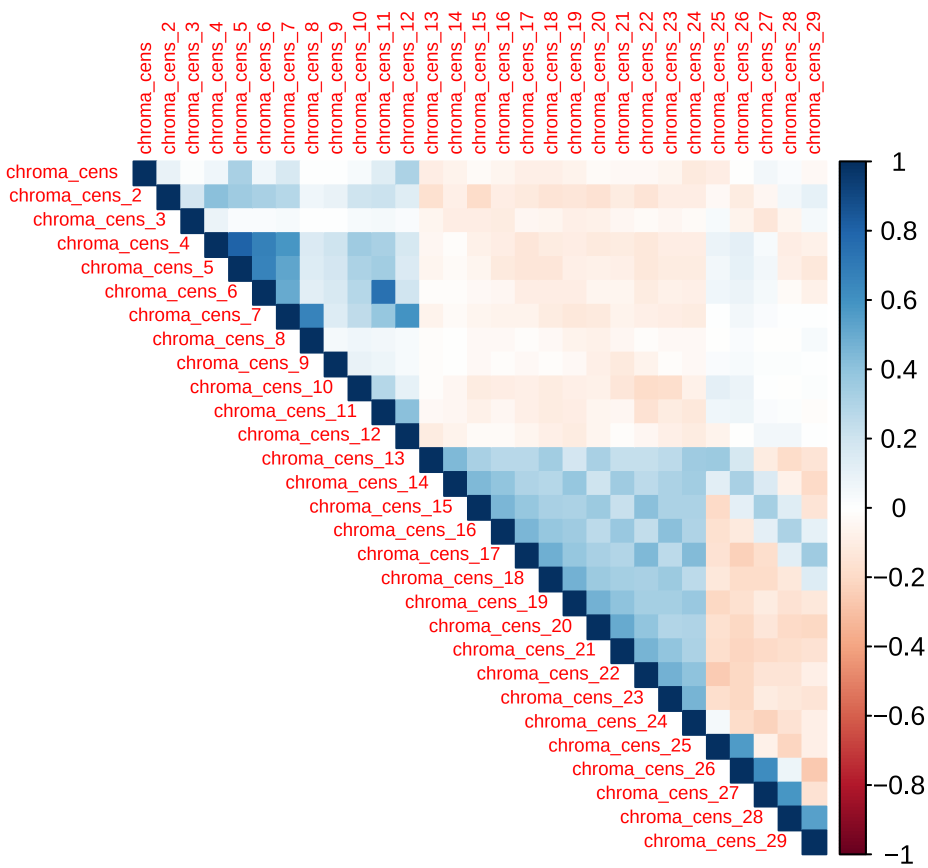



```r
# Drop first row of features
features <- features[-1, ]
features$track_id <- as.integer(features$track_id)

# Merge features with cleaned tracks
combined_data <- inner_join(tracks_clean, features, by = "track_id")

# Convert relevant columns to numeric
mfcc_cols <- grep("^mfcc", names(combined_data), value = TRUE)
chroma_cols <- grep("^chroma", names(combined_data), value = TRUE)
spectral_cols <- grep("^spectral", names(combined_data), value = TRUE)

combined_data <- combined_data %>%
  mutate(across(all_of(c(mfcc_cols, chroma_cols, spectral_cols)), ~as.numeric(.)))

# Create model_data with scaled numeric features
numeric_features <- combined_data %>%
  select(-track_id, -subset, -genre_top) %>%
  select(where(is.numeric))

feature_matrix <- scale(numeric_features)
```

```r
model_data <- data.frame(genre_top = combined_data$genre_top, feature_matrix)
```

```r
set.seed(123)
train_index <- createDataPartition(model_data$genre_top, p = 0.8, list = FALSE)
train_set <- model_data[train_index, ]
test_set <- model_data[-train_index, ]
```

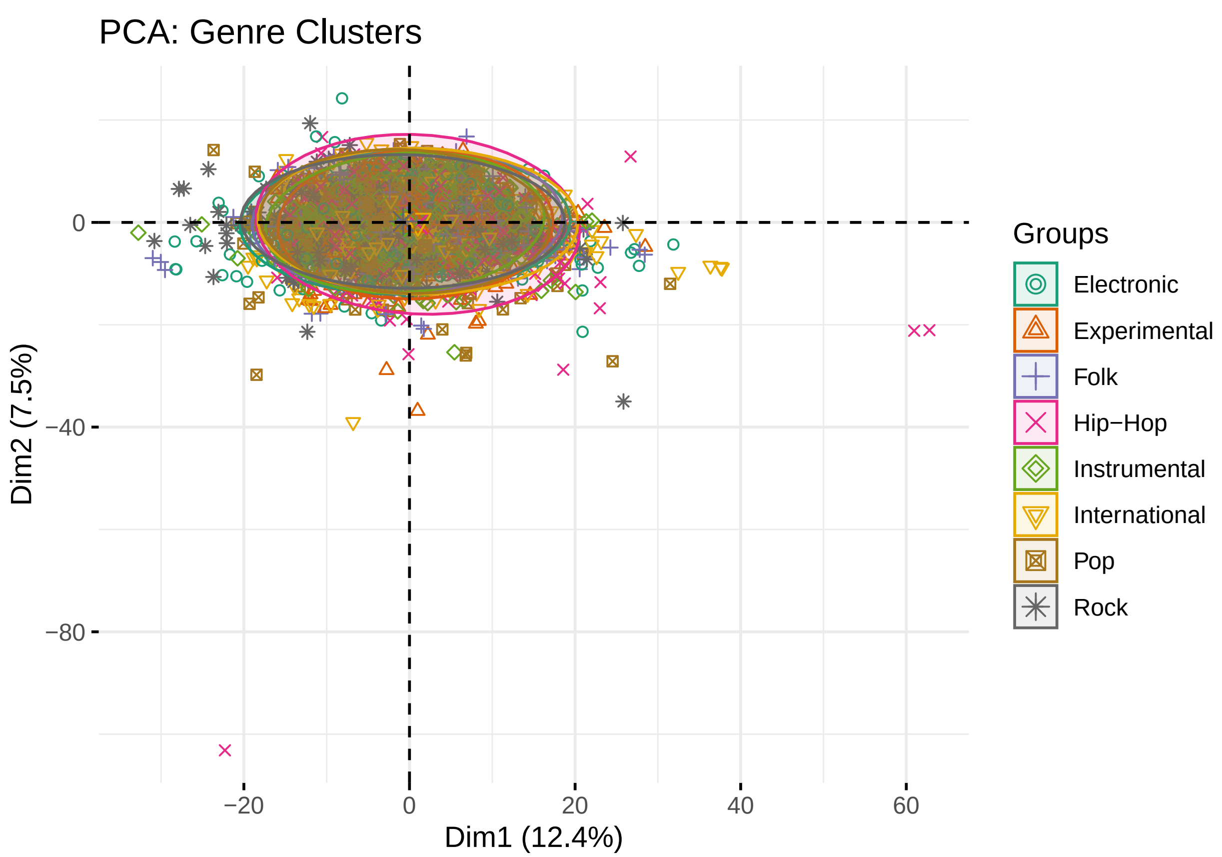
```r
library(corrplot)

# Compute correlation matrix on a subset to avoid overload
corr_matrix <- cor(model_data[ , 2:30])  # First 30 features
corrplot(corr_matrix, method = "color", type = "upper", tl.cex = 0.6)
```




```r
#PCA for Visualization
library(FactoMineR)
library(factoextra)
```

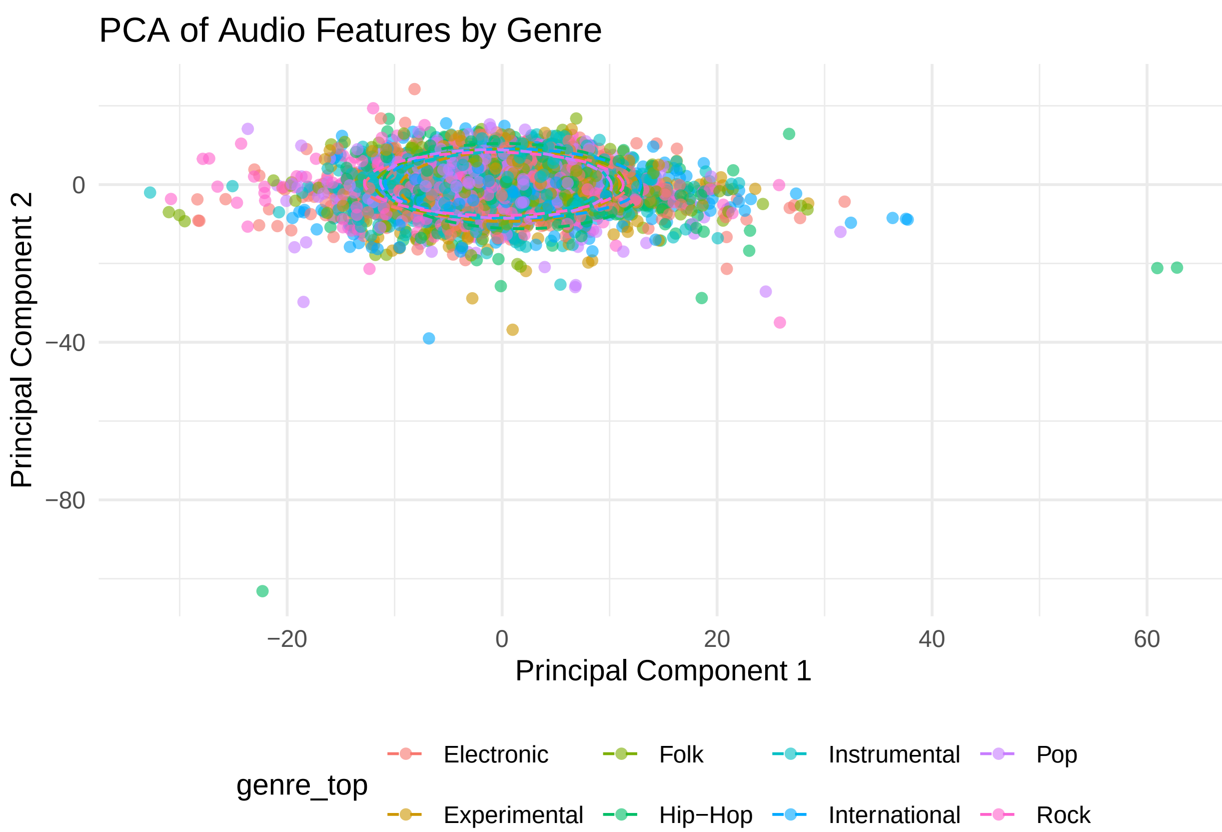
```
## Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WB
```

```r
# Run PCA on scaled features
pca_res <- prcomp(model_data[ , -1], center = TRUE, scale. = TRUE)

# Plot first 2 PCs, colored by genre
fviz_pca_ind(pca_res,
             geom.ind = "point",
             habillage = model_data$genre_top,
             addEllipses = TRUE,
             palette = "Dark2",
             title = "PCA: Genre Clusters")
```



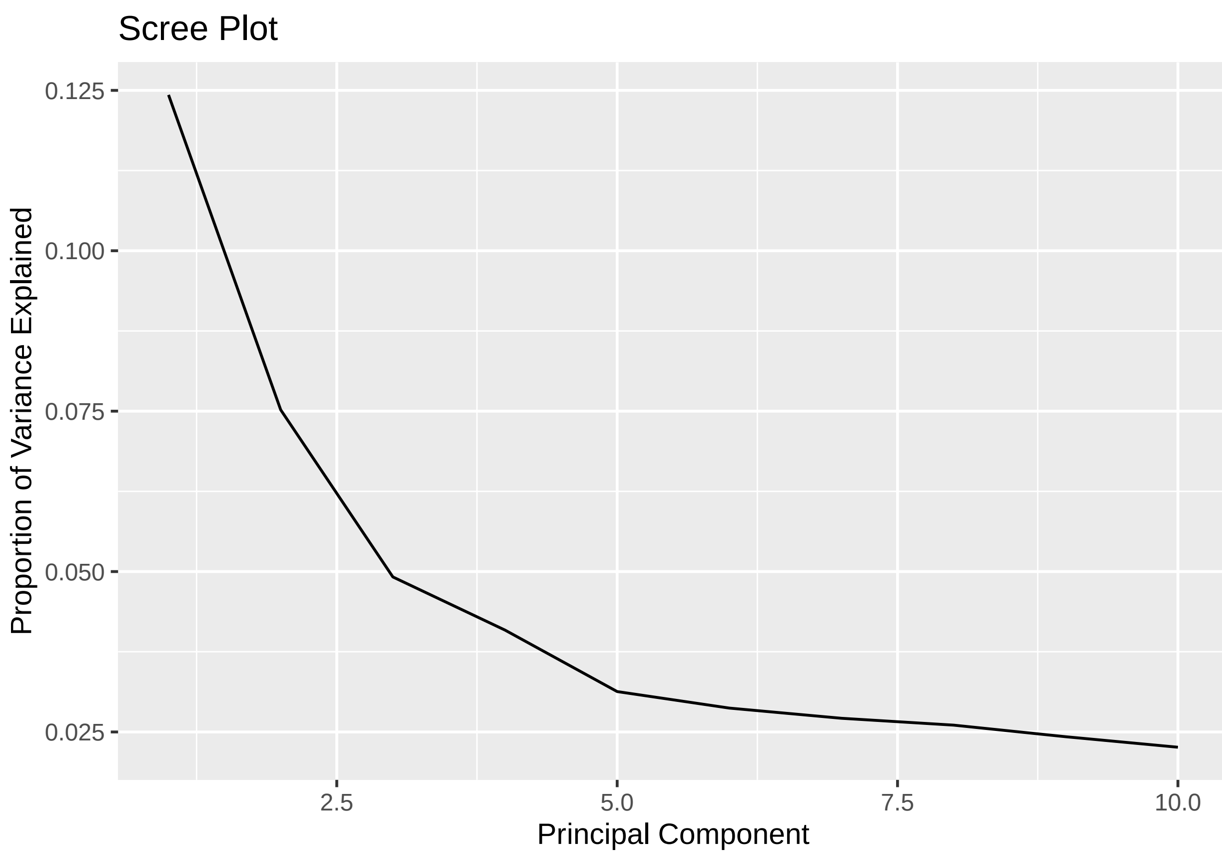

```r
# Create pca_df
pca_result <- prcomp(model_data[, -1], center = TRUE, scale. = TRUE)
# Extract the first two principal components
pca_scores <- as.data.frame(pca_result$x[, 1:2])

# Add genre labels back in from model_data
pca_df <- cbind(pca_scores, genre_top = model_data$genre_top)


# PCA of Audio Features by Genre
```

```r
ggplot(pca_df, aes(x = PC1, y = PC2, color = genre_top)) +
  geom_point(alpha = 0.6) +
  stat_ellipse(type = "norm", level = 0.68, linetype = "dashed") +
  labs(title = "PCA of Audio Features by Genre",
       x = "Principal Component 1",
       y = "Principal Component 2") +
  theme_minimal() +
  theme(legend.position = "bottom")
```




```r
pca_variance <- pca_result$sdev^2
pca_prop_var <- pca_variance / sum(pca_variance)
qplot(y = pca_prop_var[1:10], x = 1:10, geom = "line") +
  labs(title = "Scree Plot", x = "Principal Component", y = "Proportion of Variance Expl
```

```
## Warning: 'qplot()' was deprecated in ggplot2 3.4.0.
## This warning is displayed once every 8 hours.
## Call 'lifecycle::last_lifecycle_warnings()' to see where this warning was
## generated.
```


Scree Plot
Proportion of Variance Explained
0.125
0.100
0.075
0.050
0.025
2.5
5.0
7.5
10.0
Principal Component

1 Methods: Alexandria Simms

2 K Nearest Neighbors Model

```r
knn_model <- train(
  genre_top ~ ., data = train_set,
  method = "knn",
  tuneLength = 5,
  trControl = trainControl(method = "cv", number = 5)
)
knn_preds <- predict(knn_model, newdata = test_set)
confusionMatrix(knn_preds, test_set$genre_top)
```

```
## Confusion Matrix and Statistics
##
##                Reference
## Prediction      Electronic Experimental Folk Hip-Hop Instrumental International
##   Electronic            24           19   13      11           19            12
##   Experimental          22           26   15      24           24            23
##   Folk                  14           13   20       9           11             8
##   Hip-Hop               12           11   13      28           12            16
##   Instrumental          15           11    9      14           22            15
##   International         18           19   16      19           14            33
##   Pop                   17           24   16      13           24            18
##   Rock                  11           13    9      10           18            11
##                Reference
## Prediction      Pop Rock
##   Electronic     12   16
##   Experimental   18   22
##   Folk           10   12
##   Hip-Hop        22   15
##   Instrumental   16   14
##   International  16   18
##   Pop            37   12
##   Rock           11   23
##
## Overall Statistics
##
##                Accuracy : 0.2006
##                  95% CI : (0.1769, 0.2259)
##     No Information Rate : 0.1356
##     P-Value [Acc > NIR] : 3.153e-09
```

```
##
##                   Kappa : 0.0854
##
##  Mcnemar's Test P-Value : 0.3524
##
## Statistics by Class:
##
##                      Class: Electronic Class: Experimental Class: Folk
## Sensitivity                     0.1805              0.19118     0.18018
## Specificity                     0.8902              0.84017     0.91903
## Pos Pred Value                  0.1905              0.14943     0.20619
## Neg Pred Value                  0.8835              0.87613     0.90570
## Prevalence                      0.1252              0.12806     0.10452
## Detection Rate                  0.0226              0.02448     0.01883
## Detection Prevalence            0.1186              0.16384     0.09134
## Balanced Accuracy               0.5353              0.51567     0.54961
##                      Class: Hip-Hop Class: Instrumental Class: International
## Sensitivity                 0.21875             0.15278              0.24265
## Specificity                 0.89186             0.89760              0.87041
## Pos Pred Value              0.21705             0.18966              0.21569
## Neg Pred Value              0.89282             0.87104              0.88669
## Prevalence                  0.12053             0.13559              0.12806
## Detection Rate              0.02637             0.02072              0.03107
## Detection Prevalence        0.12147             0.10923              0.14407
## Balanced Accuracy           0.55531             0.52519              0.55653
##                      Class: Pop Class: Rock
## Sensitivity             0.26056     0.17424
## Specificity             0.86522     0.91075
## Pos Pred Value          0.22981     0.21698
## Neg Pred Value          0.88346     0.88598
## Prevalence              0.13371     0.12429
## Detection Rate          0.03484     0.02166
## Detection Prevalence    0.15160     0.09981
## Balanced Accuracy       0.56289     0.54250
```

3 Random Forest Model

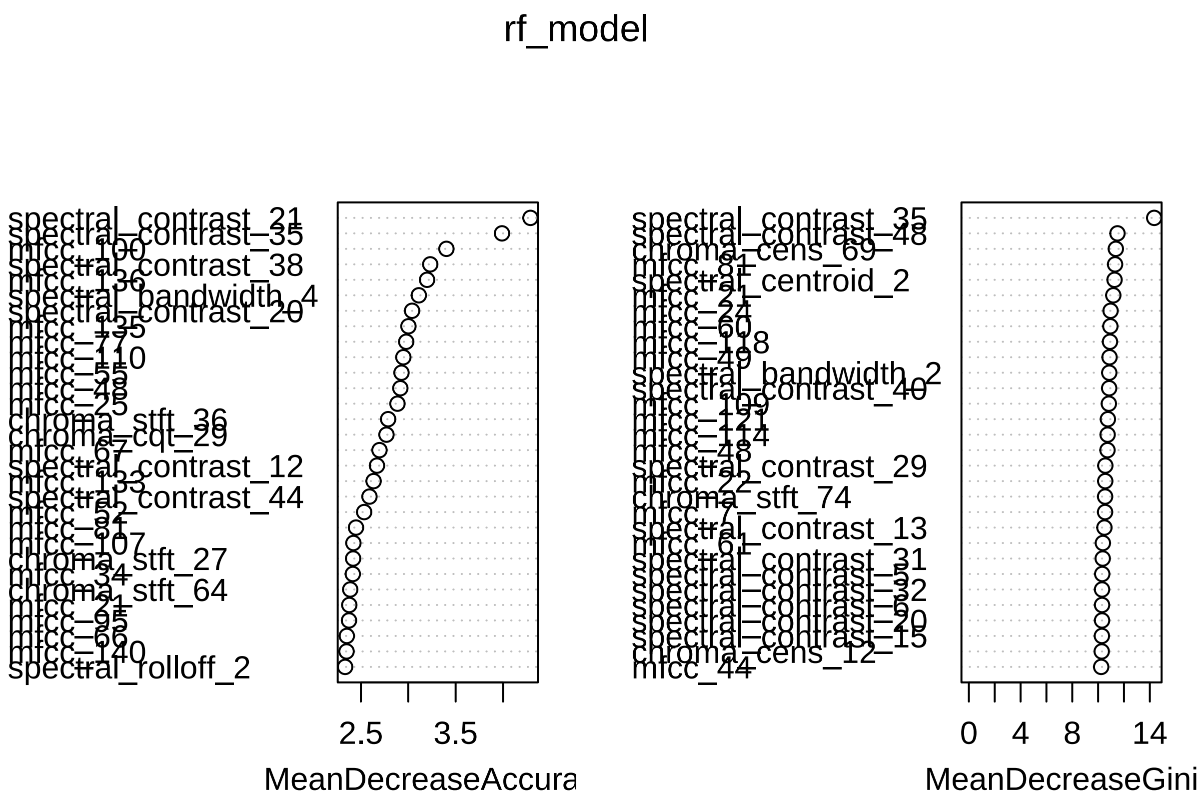
```r
rf_model <- randomForest(
  genre_top ~ ., data = train_set,
  ntree = 100, importance = TRUE
)
rf_preds <- predict(rf_model, newdata = test_set)
```

```r
confusionMatrix(rf_preds, test_set$genre_top)
```

```
## Confusion Matrix and Statistics
##
##                Reference
## Prediction      Electronic Experimental Folk Hip-Hop Instrumental International
##   Electronic            20           18   17      15           21            18
##   Experimental          18           28   16      20            8            15
##   Folk                   6            6    2       2            8             6
##   Hip-Hop               18           15   13      24           13            14
##   Instrumental          20           21   21      18           40            23
##   International         14           22   12      23           17            31
##   Pop                   29           14   22      19           24            16
##   Rock                   8           12    8       7           13            13
##                Reference
## Prediction      Pop Rock
##   Electronic     17   26
##   Experimental   16   17
##   Folk            8    6
##   Hip-Hop        11    6
##   Instrumental   23   15
##   International  16   18
##   Pop            36   23
##   Rock           15   21
##
## Overall Statistics
##
##                Accuracy : 0.1902
##                  95% CI : (0.167, 0.2151)
##     No Information Rate : 0.1356
##     P-Value [Acc > NIR] : 4.396e-07
##
##                   Kappa : 0.0716
##
##  Mcnemar's Test P-Value : 0.0001385
##
## Statistics by Class:
##
##                      Class: Electronic Class: Experimental Class: Folk
## Sensitivity                    0.15038             0.20588    0.018018
## Specificity                    0.85791             0.88121    0.955836
## Pos Pred Value                 0.13158             0.20290    0.045455
## Neg Pred Value                 0.87582             0.88312    0.892927
## Prevalence                     0.12524             0.12806    0.104520
```

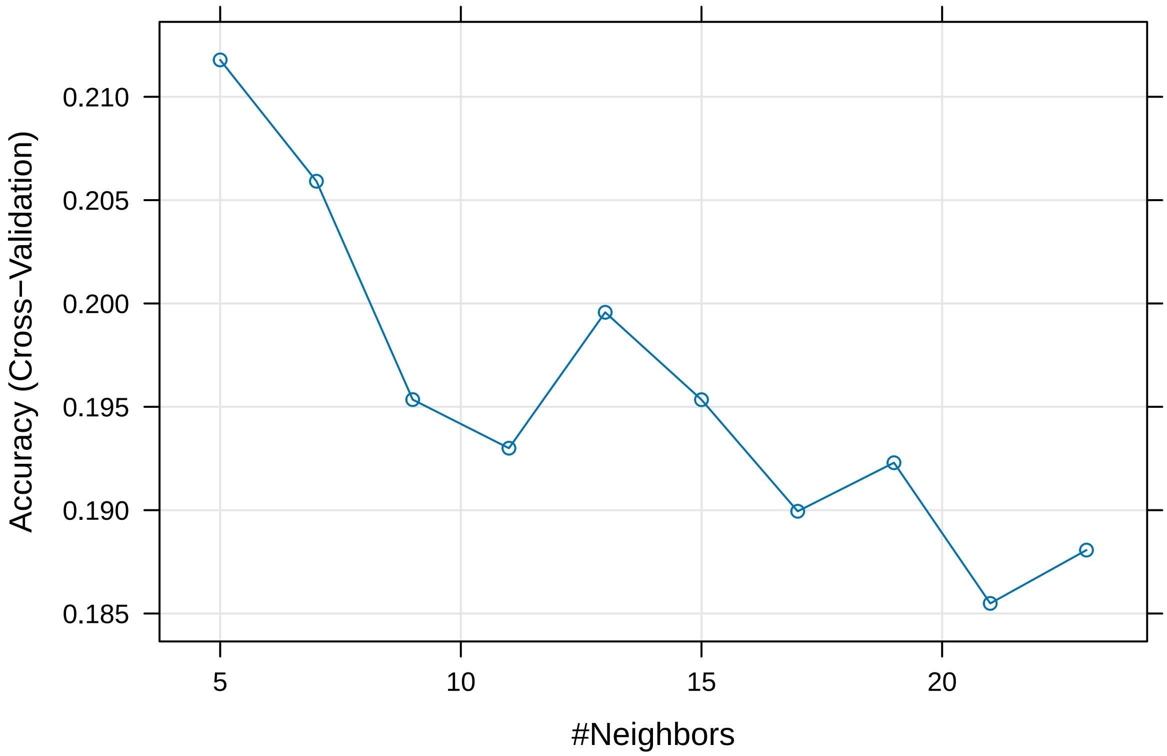
```
## Detection Rate                     0.01883             0.02637    0.001883
## Detection Prevalence               0.14313             0.12994    0.041431
## Balanced Accuracy                  0.50414             0.54355    0.486927
##                      Class: Hip-Hop Class: Instrumental Class: International
## Sensitivity                  0.1875             0.27778              0.22794
## Specificity                  0.9036             0.84641              0.86825
## Pos Pred Value               0.2105             0.22099              0.20261
## Neg Pred Value               0.8903             0.88195              0.88449
## Prevalence                   0.1205             0.13559              0.12806
## Detection Rate               0.0226             0.03766              0.02919
## Detection Prevalence         0.1073             0.17043              0.14407
## Balanced Accuracy            0.5456             0.56209              0.54810
##                      Class: Pop Class: Rock
## Sensitivity              0.2535     0.15909
## Specificity              0.8402     0.91828
## Pos Pred Value           0.1967     0.21649
## Neg Pred Value           0.8794     0.88497
## Prevalence               0.1337     0.12429
## Detection Rate           0.0339     0.01977
## Detection Prevalence     0.1723     0.09134
## Balanced Accuracy        0.5469     0.53869
```

```r
varImpPlot(rf_model)
```



4 Model Tuning

```r
set.seed(123)
holdout_index <- createDataPartition(model_data$genre_top, p = 0.8, list = FALSE)
training_set <- model_data[holdout_index, ]
holdout_set  <- model_data[-holdout_index, ]

training_set$genre_top <- as.factor(training_set$genre_top)
holdout_set$genre_top  <- as.factor(holdout_set$genre_top)

# KNN Tuning
tuned_knn <- train(
  genre_top ~ ., data = training_set,
  method = "knn", tuneLength = 10,
  trControl = trainControl(method = "cv", number = 5)
)
plot(tuned_knn)
```

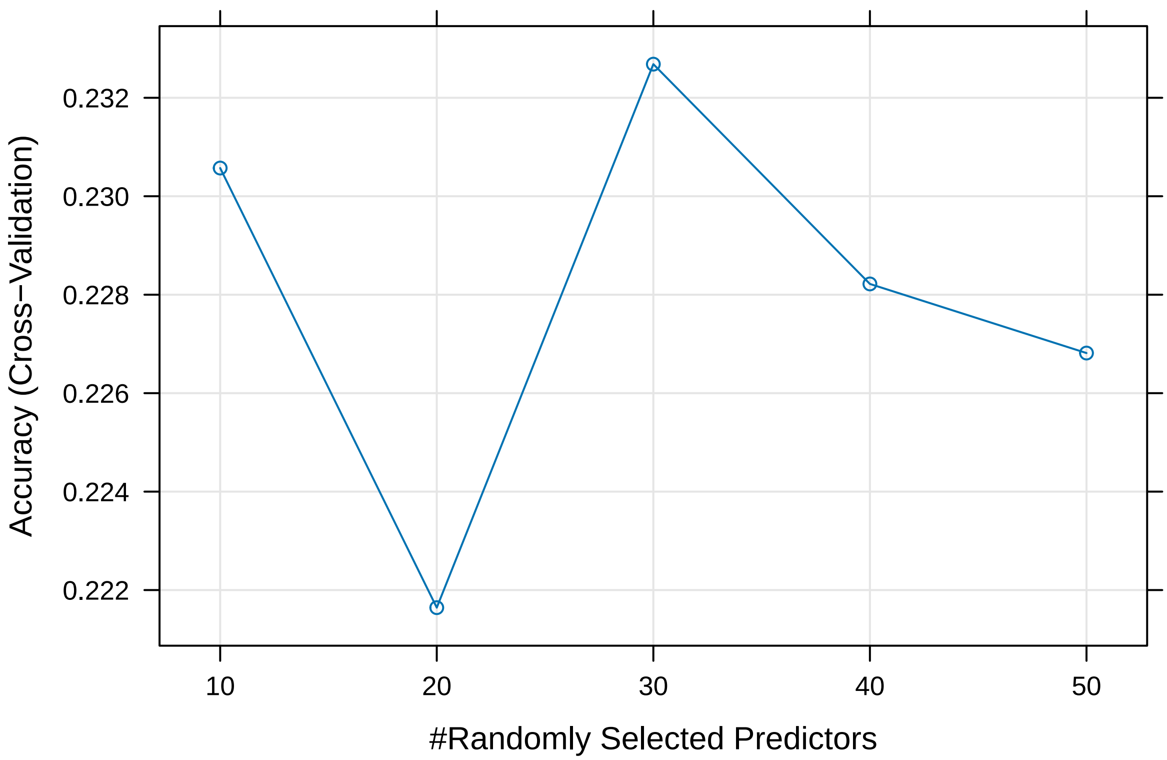



```r
knn_final_preds <- predict(tuned_knn, newdata = holdout_set)
confusionMatrix(knn_final_preds, holdout_set$genre_top)
```

```
## Confusion Matrix and Statistics
##
##                Reference
## Prediction      Electronic Experimental Folk Hip-Hop Instrumental International
##   Electronic            25           13   12      11           21            15
##   Experimental          17           30   17      21           25            23
##   Folk                  14           14   18       7           10             6
##   Hip-Hop               14           12   14      29           11            20
##   Instrumental          14            9   11      13           22            11
##   International         15           20   18      20           17            32
##   Pop                   20           22   14      17           23            16
##   Rock                  14           16    7      10           15            13
##                Reference
## Prediction      Pop Rock
##   Electronic     14   12
##   Experimental   18   23
##   Folk           13   13
##   Hip-Hop        22   13
```

```
##   Instrumental   17   14
##   International  18   24
##   Pop            30   12
##   Rock           10   21
##
## Overall Statistics
##
##                Accuracy : 0.1949
##                  95% CI : (0.1715, 0.22)
##     No Information Rate : 0.1356
##     P-Value [Acc > NIR] : 5.083e-08
##
##                   Kappa : 0.079
##
##  Mcnemar's Test P-Value : 0.2355
##
## Statistics by Class:
##
##                      Class: Electronic Class: Experimental Class: Folk
## Sensitivity                    0.18797             0.22059     0.16216
## Specificity                    0.89451             0.84449     0.91903
## Pos Pred Value                 0.20325             0.17241     0.18947
## Neg Pred Value                 0.88498             0.88063     0.90383
## Prevalence                     0.12524             0.12806     0.10452
## Detection Rate                 0.02354             0.02825     0.01695
## Detection Prevalence           0.11582             0.16384     0.08945
## Balanced Accuracy              0.54124             0.53254     0.54060
##                      Class: Hip-Hop Class: Instrumental Class: International
## Sensitivity                 0.22656             0.15278              0.23529
## Specificity                 0.88651             0.90305              0.85745
## Pos Pred Value              0.21481             0.19820              0.19512
## Neg Pred Value              0.89320             0.87171              0.88419
## Prevalence                  0.12053             0.13559              0.12806
## Detection Rate              0.02731             0.02072              0.03013
## Detection Prevalence        0.12712             0.10452              0.15443
## Balanced Accuracy           0.55654             0.52791              0.54637
##                      Class: Pop Class: Rock
## Sensitivity             0.21127     0.15909
## Specificity             0.86522     0.90860
## Pos Pred Value          0.19481     0.19811
## Neg Pred Value          0.87665     0.88389
## Prevalence              0.13371     0.12429
## Detection Rate          0.02825     0.01977
## Detection Prevalence    0.14501     0.09981
## Balanced Accuracy       0.53824     0.53385
```

```r
# RF Tuning
rf_grid <- expand.grid(mtry = c(10, 20, 30, 40, 50))
rf_tuned <- train(
  genre_top ~ ., data = training_set,
  method = "rf",
  trControl = trainControl(method = "cv", number = 5),
  tuneGrid = rf_grid,
  importance = TRUE
)
plot(rf_tuned)
```




```r
rf_final_preds <- predict(rf_tuned, newdata = holdout_set)
cm_rf <- confusionMatrix(rf_final_preds, holdout_set$genre_top)
print(cm_rf)
```

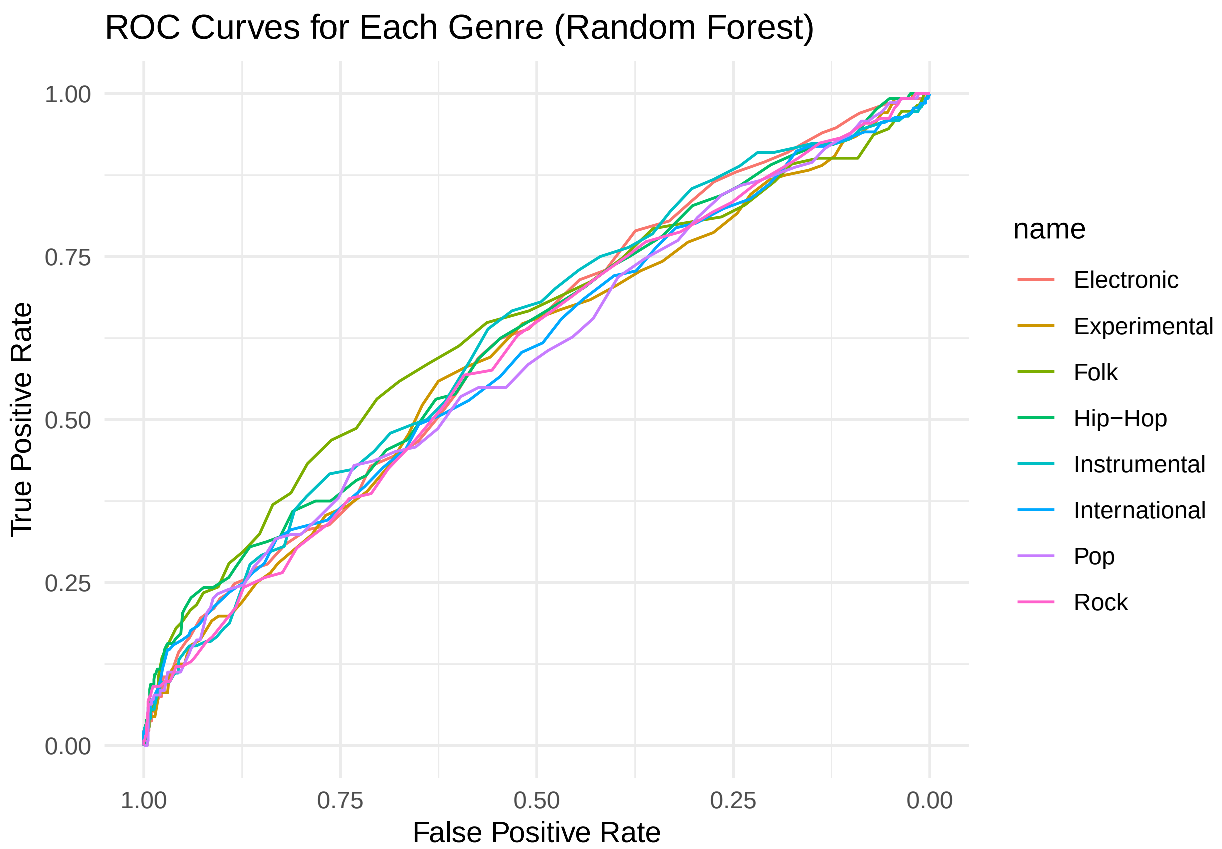
```
## Confusion Matrix and Statistics
##
##                 Reference
## Prediction        Electronic Experimental Folk Hip-Hop Instrumental International
##   Electronic              29            9   10      11           11            12
```

```
##   Experimental          11            35   16       20            16            21
##   Folk                   4             1    9        1             0             2
##   Hip-Hop               13            13    9       26             7            10
##   Instrumental          20            24   23       18            43            26
##   International         16            28   13       22            24            34
##   Pop                   29            18   20       23            27            22
##   Rock                  11             8   11        7            16             9
##                Reference
## Prediction      Pop Rock
##   Electronic     17   11
##   Experimental   15   29
##   Folk            3    1
##   Hip-Hop        10    8
##   Instrumental   25   22
##   International  22   14
##   Pop            41   22
##   Rock            9   25
##
## Overall Statistics
##
##                Accuracy : 0.2279
##                  95% CI : (0.203, 0.2543)
##     No Information Rate : 0.1356
##     P-Value [Acc > NIR] : 2.713e-16
##
##                   Kappa : 0.1137
##
##  Mcnemar's Test P-Value : 3.070e-13
##
## Statistics by Class:
##
##                      Class: Electronic Class: Experimental Class: Folk
## Sensitivity                    0.21805             0.25735    0.081081
## Specificity                    0.91281             0.86177    0.987382
## Pos Pred Value                 0.26364             0.21472    0.428571
## Neg Pred Value                 0.89076             0.88765    0.902017
## Prevalence                     0.12524             0.12806    0.104520
## Detection Rate                 0.02731             0.03296    0.008475
## Detection Prevalence           0.10358             0.15348    0.019774
## Balanced Accuracy              0.56543             0.55956    0.534231
##                      Class: Hip-Hop Class: Instrumental Class: International
## Sensitivity                 0.20312             0.29861              0.25000
## Specificity                 0.92505             0.82789              0.84989
## Pos Pred Value              0.27083             0.21393              0.19653
## Neg Pred Value              0.89441             0.88269              0.88526
```

```
## Prevalence                 0.12053               0.13559               0.12806
## Detection Rate             0.02448               0.04049               0.03202
## Detection Prevalence       0.09040               0.18927               0.16290
## Balanced Accuracy          0.56409               0.56325               0.54995
##                      Class: Pop Class: Rock
## Sensitivity             0.28873     0.18939
## Specificity             0.82500     0.92366
## Pos Pred Value          0.20297     0.26042
## Neg Pred Value          0.88256     0.88923
## Prevalence              0.13371     0.12429
## Detection Rate          0.03861     0.02354
## Detection Prevalence    0.19021     0.09040
## Balanced Accuracy       0.55687     0.55652
```

5 Results

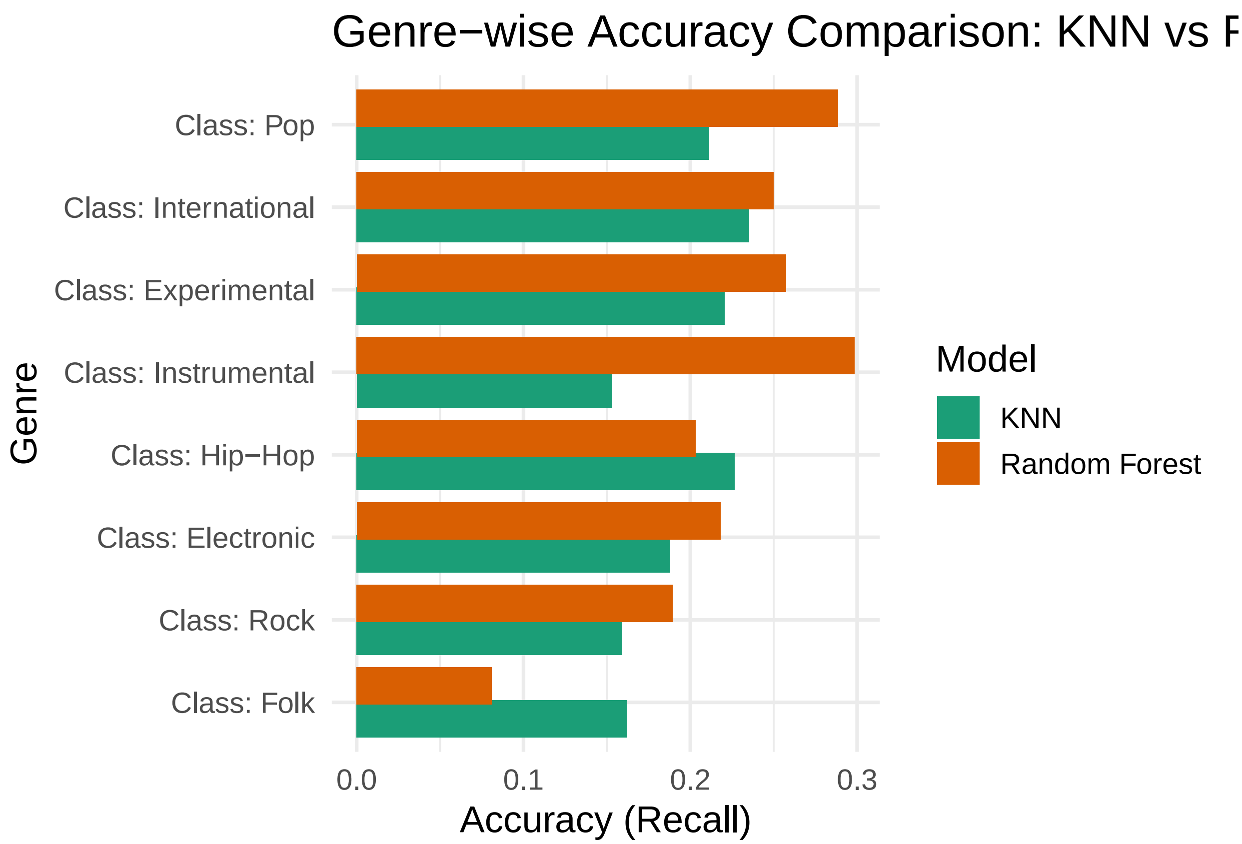
```r
rf_probabilities <- predict(rf_tuned, newdata = holdout_set, type = "prob")
roc_list <- list()
true_labels <- holdout_set$genre_top
classes <- colnames(rf_probabilities)

for (class in classes) {
  binary_labels <- ifelse(true_labels == class, 1, 0)
  prob <- rf_probabilities[[class]]
  roc_obj <- roc(binary_labels, prob, levels = c(0, 1), direction = "<")
  roc_list[[class]] <- roc_obj
}

ggroc(roc_list) +
  labs(title = "ROC Curves for Each Genre (Random Forest)",
       x = "False Positive Rate", y = "True Positive Rate") +
  theme_minimal()
```



```r
sapply(roc_list, auc)
```

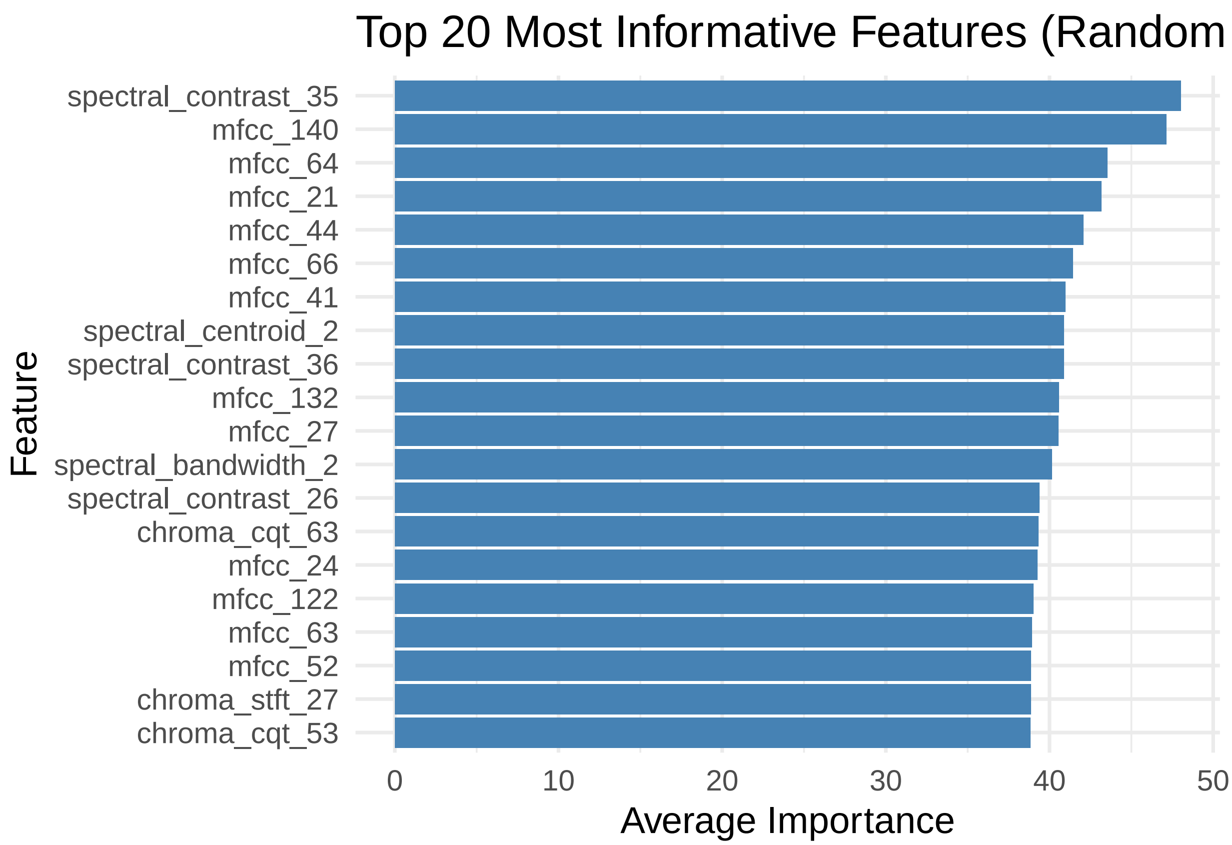
```
##    Electronic  Experimental          Folk       Hip-Hop  Instrumental
##     0.6161569     0.5954096     0.6364140     0.6216667     0.6245991
## International           Pop          Rock
##     0.5979744     0.5958550     0.5995194
```

```r
cm_knn <- confusionMatrix(knn_final_preds, holdout_set$genre_top)
knn_accuracy <- as.data.frame(cm_knn$byClass) %>%
  mutate(Genre = rownames(.), Model = "KNN") %>%
  select(Genre, Sensitivity, Model)

rf_accuracy <- as.data.frame(cm_rf$byClass) %>%
  mutate(Genre = rownames(.), Model = "Random Forest") %>%
  select(Genre, Sensitivity, Model)

combined_accuracy <- bind_rows(knn_accuracy, rf_accuracy)

ggplot(combined_accuracy, aes(x = reorder(Genre, Sensitivity), y = Sensitivity, fill = M
  geom_bar(stat = "identity", position = position_dodge(width = 0.8)) +
  coord_flip() +
  labs(title = "Genre-wise Accuracy Comparison: KNN vs Random Forest",
       x = "Genre", y = "Accuracy (Recall)") +
  theme_minimal(base_size = 14) +
  scale_fill_brewer(palette = "Dark2")
```



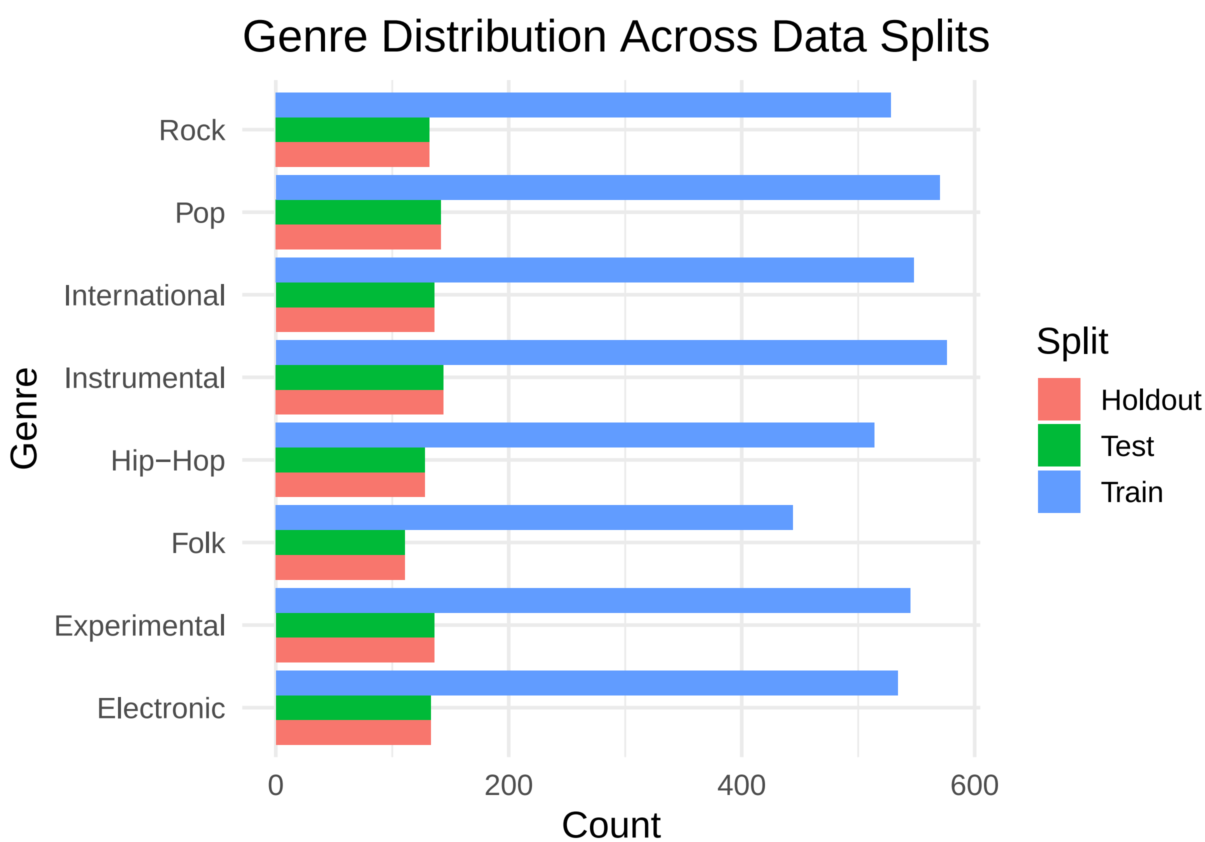

```r
# Get variable importance from rf_tuned
rf_importance <- varImp(rf_tuned, scale = TRUE)

# Calculate average importance across genres for each feature
rf_importance_overall <- rf_importance$importance %>%
  rowMeans() %>%
  sort(decreasing = TRUE)

# Convert to data frame
rf_top_features <- data.frame(
  Feature = names(rf_importance_overall),
  Importance = rf_importance_overall
)

# Plot Top 20 features
rf_top_features %>%
  slice_max(order_by = Importance, n = 20) %>%
  ggplot(aes(x = reorder(Feature, Importance), y = Importance)) +
  geom_bar(stat = "identity", fill = "steelblue") +
  coord_flip() +
  labs(title = "Top 20 Most Informative Features (Random Forest)",
```

```r
       x = "Feature", y = "Average Importance") +
  theme_minimal(base_size = 14)
```




```r
# Genre Frequency in Train vs Test vs Holdout sets

library(ggplot2)
library(dplyr)

# Add labels for dataset splits
genre_distribution <- bind_rows(
  train_set %>% mutate(Split = "Train"),
  test_set %>% mutate(Split = "Test"),
  holdout_set %>% mutate(Split = "Holdout")
)

# Plot distribution
ggplot(genre_distribution, aes(x = genre_top, fill = Split)) +
  geom_bar(position = "dodge") +
  labs(title = "Genre Distribution Across Data Splits",
       x = "Genre", y = "Count") +
```

```r
  coord_flip() +
  theme_minimal(base_size = 14)
```




```r
save(train_set, test_set, rf_model, knn_model, file = "models_and_data.RData")
```