Data Mining Project 3

Olivia Hofmann, Michael Perkins

2024-11-19

Introduction

The objective of this project is to classify counties into risk levels (**high**, **medium**, or **low**) for future pandemics using COVID-19 data. This report follows the CRISP-DM framework, focusing on **Data Preparation**, **Modeling**, **Evaluation**, and **Deployment**. The results can help stakeholders prepare for and mitigate the impact of future pandemics.

---

1. Data Preparation

```r
# Load required libraries
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.1     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.1
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```r
library(lubridate)
library(caret)
```

```
## Warning: package 'caret' was built under R version 4.4.2
```

```
## Loading required package: lattice
##
## Attaching package: 'caret'
##
## The following object is masked from 'package:purrr':
##
##     lift
```

```r
library(rpart)  # Decision Tree
library(randomForest)  # Random Forest
```

```
## randomForest 4.7-1.2
## Type rfNews() to see new features/changes/bug fixes.
##
## Attaching package: 'randomForest'
##
## The following object is masked from 'package:dplyr':
##
##     combine
##
## The following object is masked from 'package:ggplot2':
##
##     margin
```

```r
library(e1071)  # SVM
library(DMwR2)
```

```
## Warning: package 'DMwR2' was built under R version 4.4.2
```

```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```r
library(PRROC)
```

```
## Warning: package 'PRROC' was built under R version 4.4.2
```

```r
library(ggplot2)
library(gridExtra)
```

```
##
## Attaching package: 'gridExtra'
##
## The following object is masked from 'package:randomForest':
##
##     combine
##
## The following object is masked from 'package:dplyr':
##
##     combine
```

Define Classes

- The classes are based on confirmed COVID-19 cases per 10,000 population per week:
  - **High Risk:**   50 cases.
  - **Medium Risk:** 10–49 cases.
  - **Low Risk:** $<$ 10 cases.

- These thresholds were chosen based on observed patterns in case severity and the need to trigger timely interventions.

```r
# Load mobility data
final_merged_dataset <- read_csv("data/final_merged_dataset.csv")
```

```
## Rows: 6892452 Columns: 21
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr   (7): country_region_code, country_region, sub_region_1, sub_region_2, ...
## dbl  (10): census_fips_code, retail_and_recreation_percent_change_from_basel...
## lgl   (2): metro_area, iso_3166_2_code
## date  (2): mobility_date, week
##
## 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.
```

Data Preparation Steps

1. **Merge and Clean Data**: Ensure a single dataset with a class attribute.
2. **Select Predictive Features**: Extract features with potential predictive power.
3. **Handle Missing Data**: Use imputation or remove incomplete rows for models that cannot handle missing data.

```r
# Define risk levels
final_data <- final_merged_dataset %>%
  mutate(risk_level = case_when(
    count >= 50 ~ "high",
    count >= 10 ~ "medium",
    TRUE ~ "low"
  )) %>%
  mutate(risk_level = factor(risk_level, levels = c("low", "medium", "high")))

# Select relevant features
classification_data <- final_data %>%
  select(retail_and_recreation_percent_change_from_baseline,
         grocery_and_pharmacy_percent_change_from_baseline,
         workplaces_percent_change_from_baseline,
         PC1, PC2, week, risk_level) %>%
  drop_na()

# Split data into training and testing
set.seed(123)
training_index <- sample(1:nrow(classification_data), 0.7 * nrow(classification_data))
training_data <- classification_data[training_index, ]
testing_data <- classification_data[-training_index, ]

# Balance training data
min_class_size <- min(table(training_data$risk_level))
balanced_training_data <- training_data %>%
  group_by(risk_level) %>%
  sample_n(min_class_size) %>%
  ungroup()
```

---

2. Modeling

Model 1: Decision Tree

- **Advantages**: Simple and interpretable.

```
dt_model <- rpart(risk_level ~ ., data = balanced_training_data, method = "class",
                  control = rpart.control(cp = 0.05, maxdepth = 3))
```

Model 2: Random Forest

- **Advantages**: Handles large datasets and captures feature interactions.

```
rf_model <- randomForest(risk_level ~ ., data = balanced_training_data, ntree = 100, mtry = 2)
```

Model 3: Support Vector Machine (SVM)

- **Advantages**: Effective for high-dimensional spaces.

```
set.seed(123)
subsample_index <- sample(1:nrow(balanced_training_data), 0.01 * nrow(balanced_training_data))
subsample_data <- balanced_training_data[subsample_index, ]

svm_model <- svm(risk_level ~ ., data = subsample_data, cost = 0.1, gamma = 0.01, kernel = 'linear')
```

---

3. Evaluation

Confusion Matrices and Metrics

```
# Decision Tree
predictions_dt <- predict(dt_model, testing_data, type = "class")
cat("Confusion Matrix for Decision Tree:\n")
```

```
## Confusion Matrix for Decision Tree:
```

```
dt_conf <- confusionMatrix(predictions_dt, testing_data$risk_level)

# Random Forest
predictions_rf <- predict(rf_model, testing_data, type = "response")
cat("Confusion Matrix for Random Forest:\n")
```

```
## Confusion Matrix for Random Forest:
```

```r
rf_conf <- confusionMatrix(predictions_rf, testing_data$risk_level)

# SVM
predictions_svm <- predict(svm_model, testing_data)
cat("Confusion Matrix for SVM:\n")
```

```
## Confusion Matrix for SVM:
```

```r
confusionMatrix(predictions_svm, testing_data$risk_level)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    low medium   high
##     low    235999  19177  29397
##     medium  50280  59524 179208
##     high     2077  12137 545456
##
## Overall Statistics
##
##                Accuracy : 0.7421
##                  95% CI : (0.7413, 0.7429)
##     No Information Rate : 0.6654
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.5607
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
## Statistics by Class:
##
##                      Class: low Class: medium Class: high
## Sensitivity              0.8184       0.65528      0.7234
## Specificity              0.9425       0.77985      0.9625
## Pos Pred Value           0.8293       0.20596      0.9746
## Neg Pred Value           0.9383       0.96291      0.6363
## Prevalence               0.2544       0.08016      0.6654
## Detection Rate           0.2082       0.05252      0.4813
## Detection Prevalence     0.2511       0.25503      0.4939
## Balanced Accuracy        0.8805       0.71756      0.8429
```

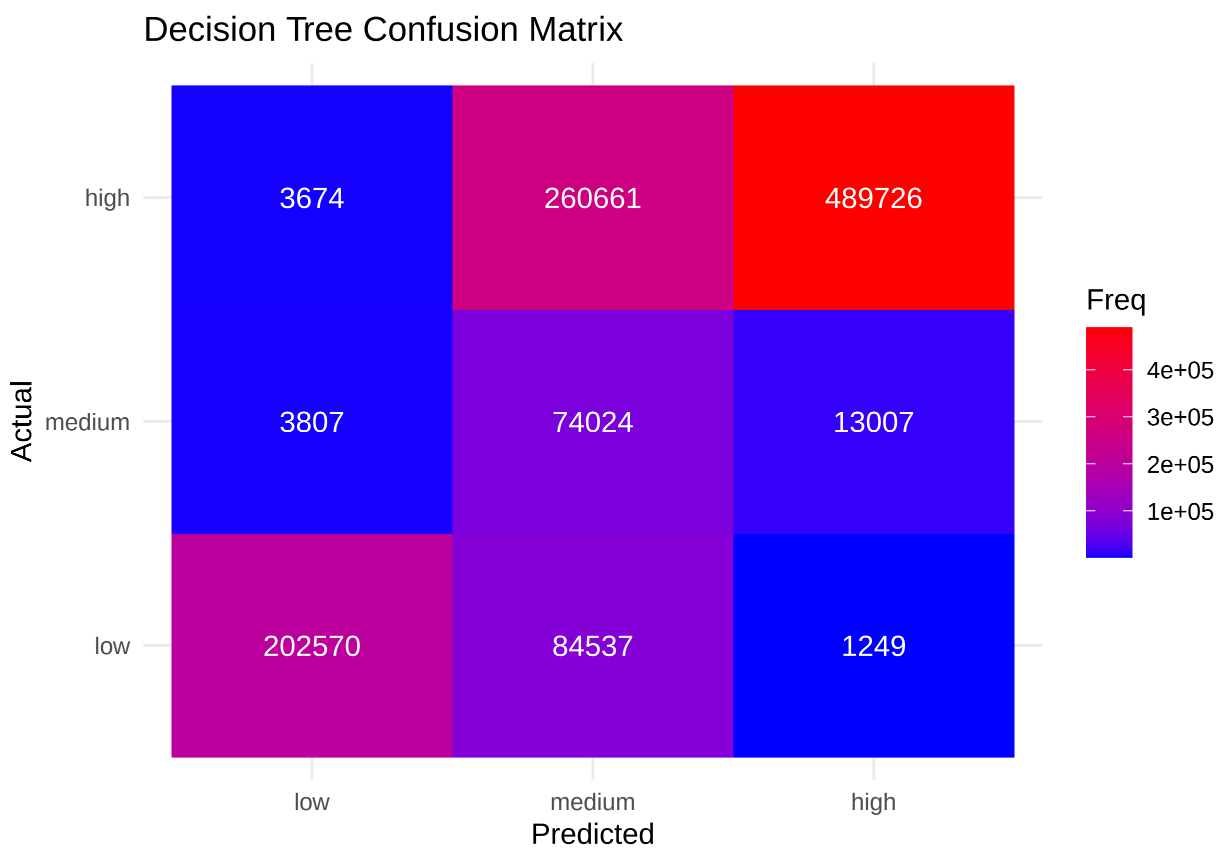
Model Performance Summary

- **Decision Tree**:
  - Accuracy: 0.6762
- **Random Forest**:
  - Accuracy: 0.4338
- **SVM**:
  - Accuracy: 0.7556

Model Performance Visualization

---

1. Confusion Matrix Heatmaps



---

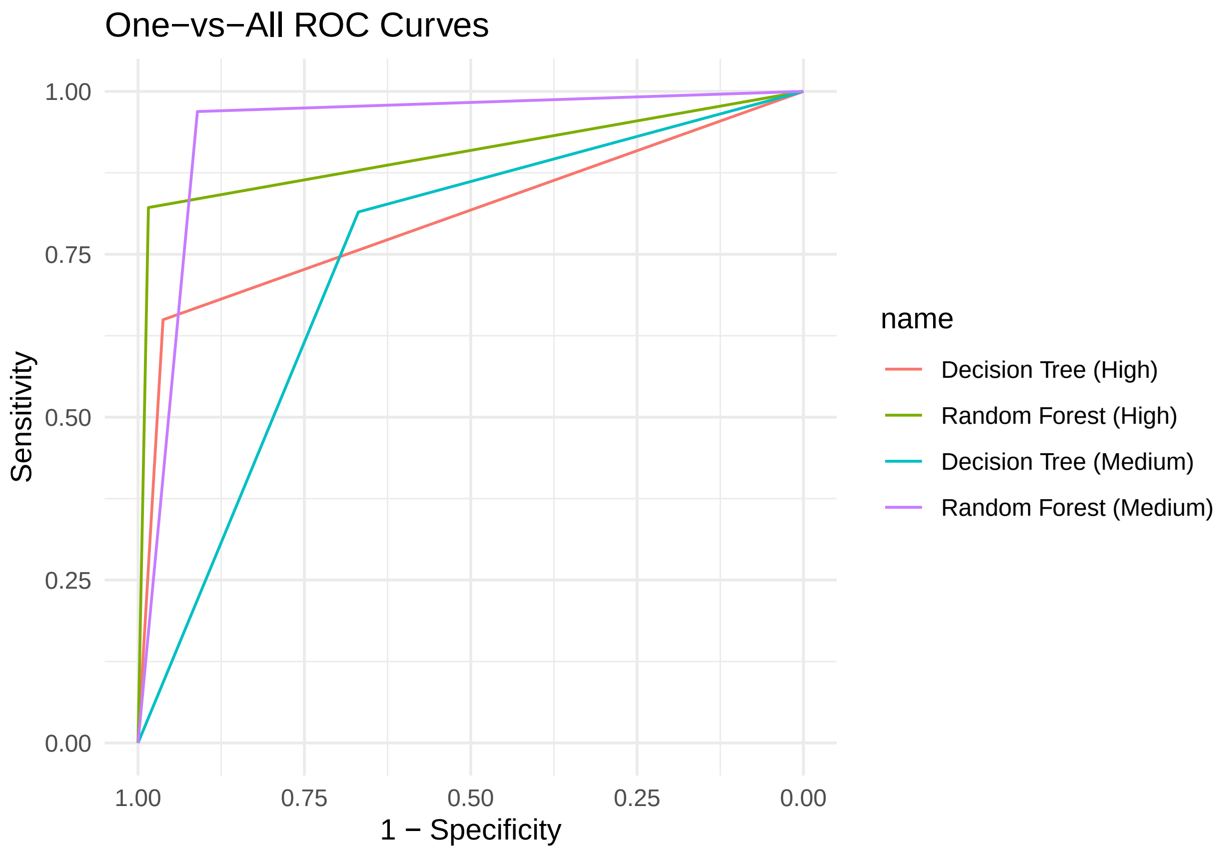
2. ROC Curves

```
## Setting levels: control = 0, case = 1

## Setting direction: controls < cases

## Setting levels: control = 0, case = 1

## Setting direction: controls < cases

## Setting levels: control = 0, case = 1
```

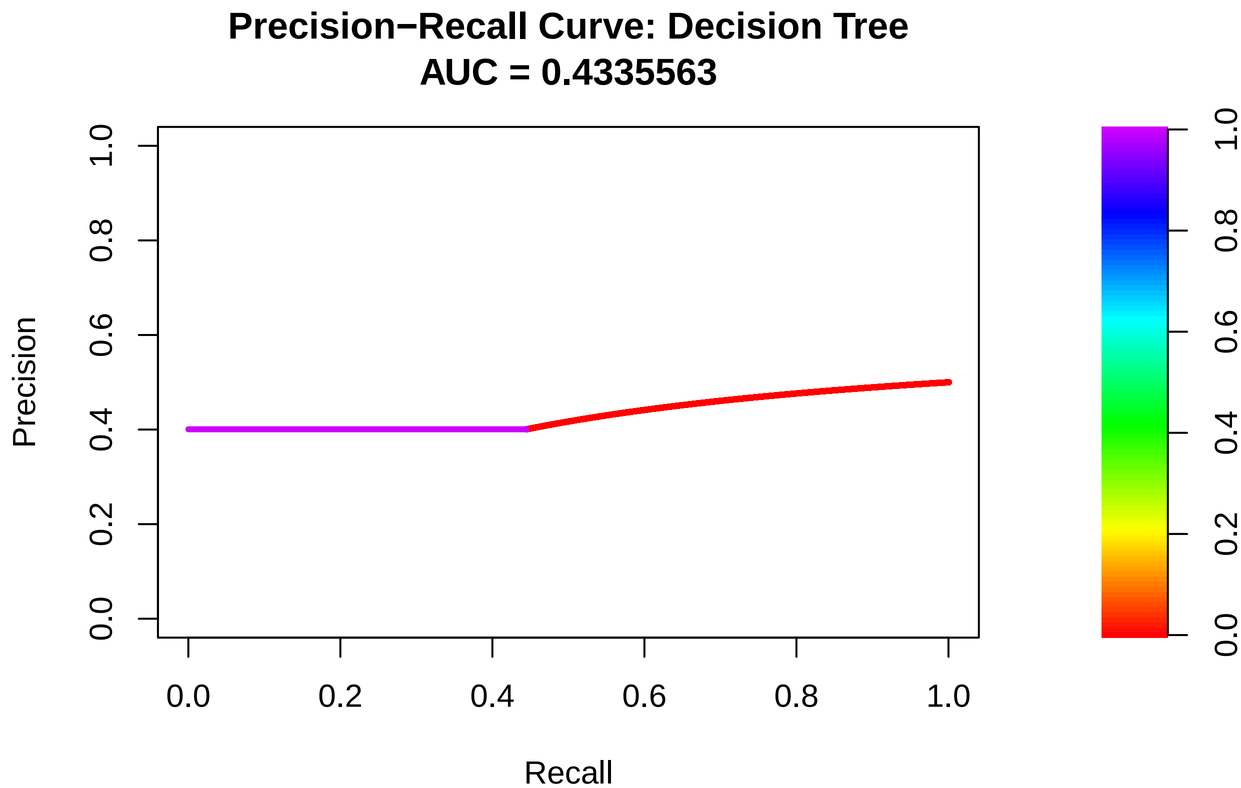
```
## Setting direction: controls < cases
```

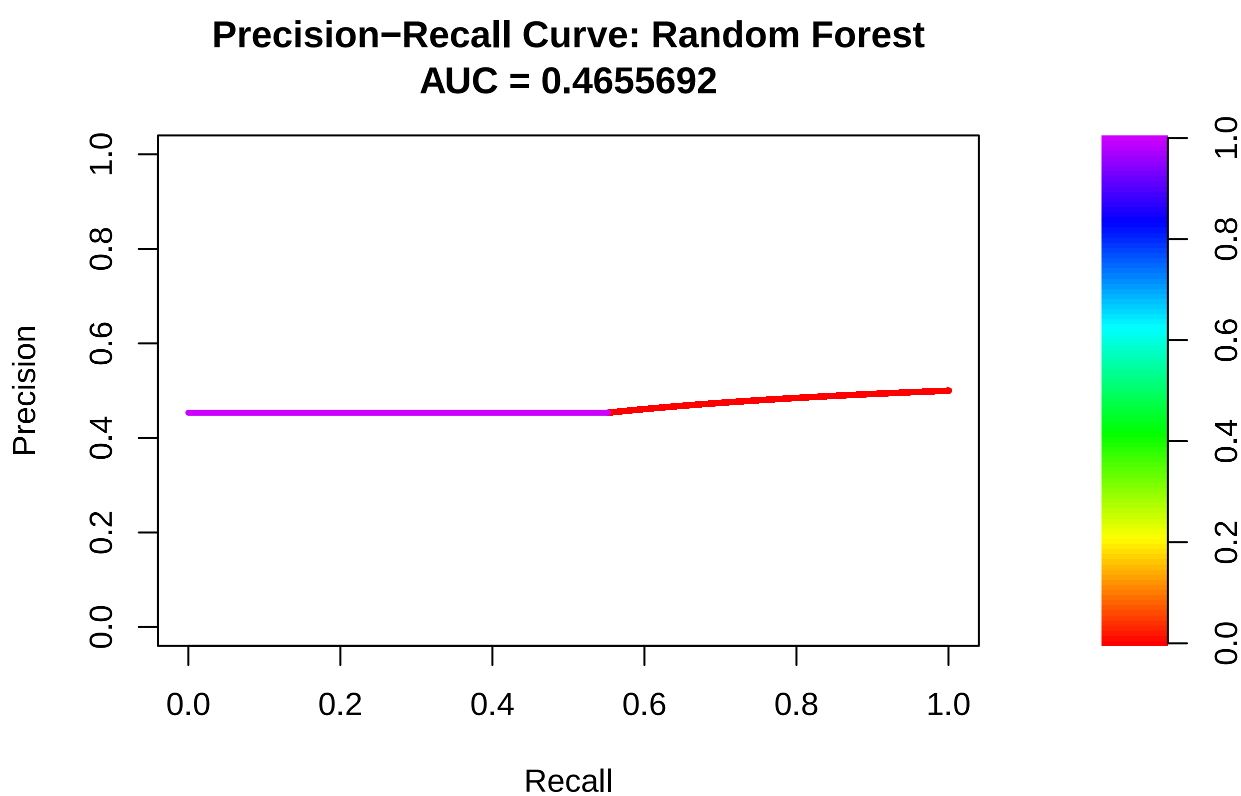
```
## Setting levels: control = 0, case = 1
```

```
## Setting direction: controls < cases
```

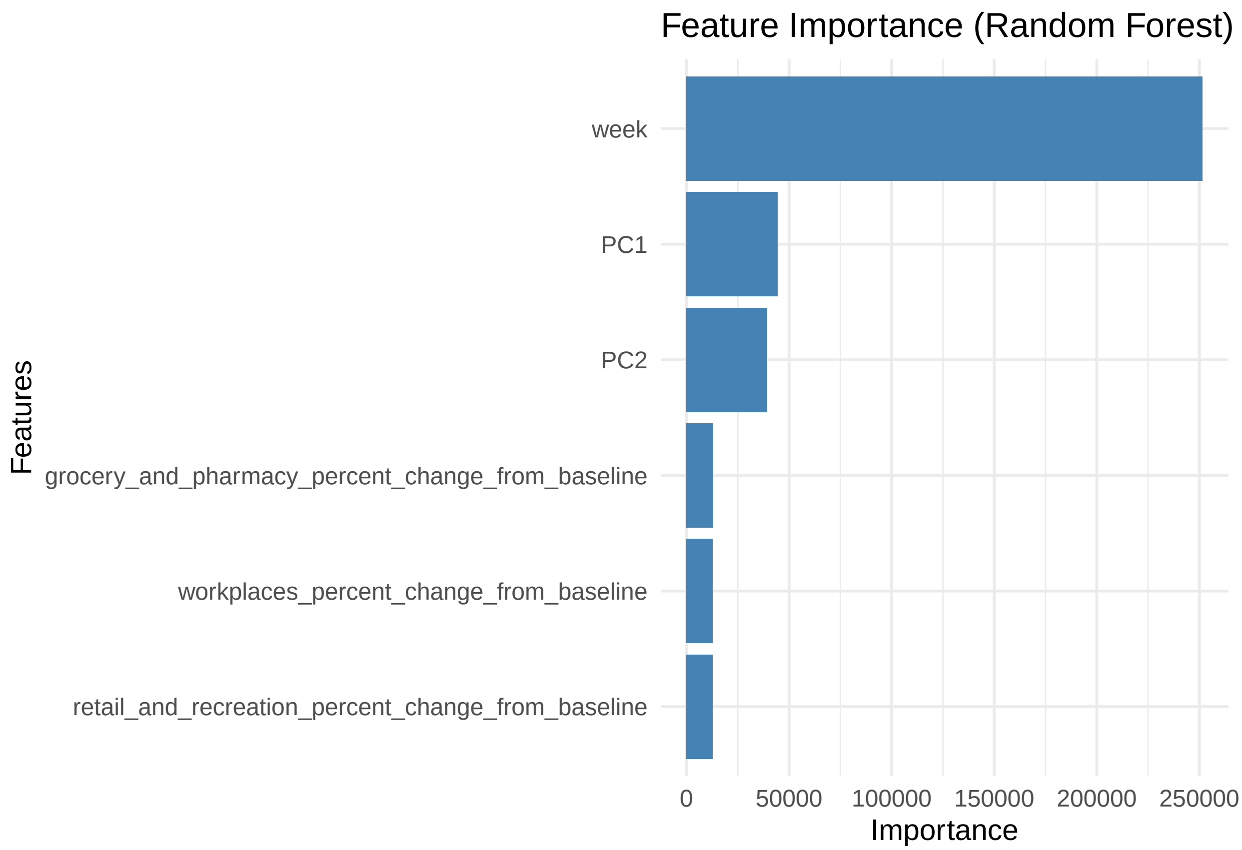


3. Precision-Recall Curves

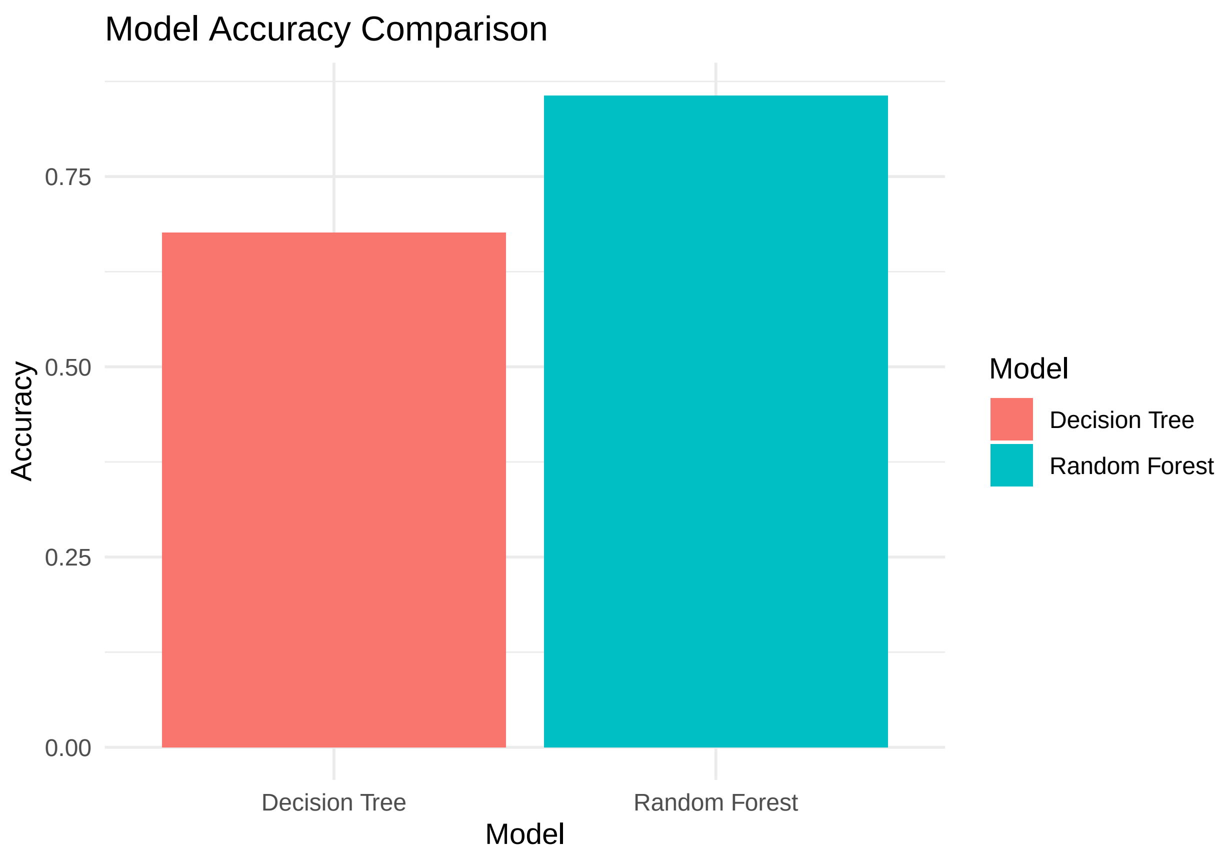




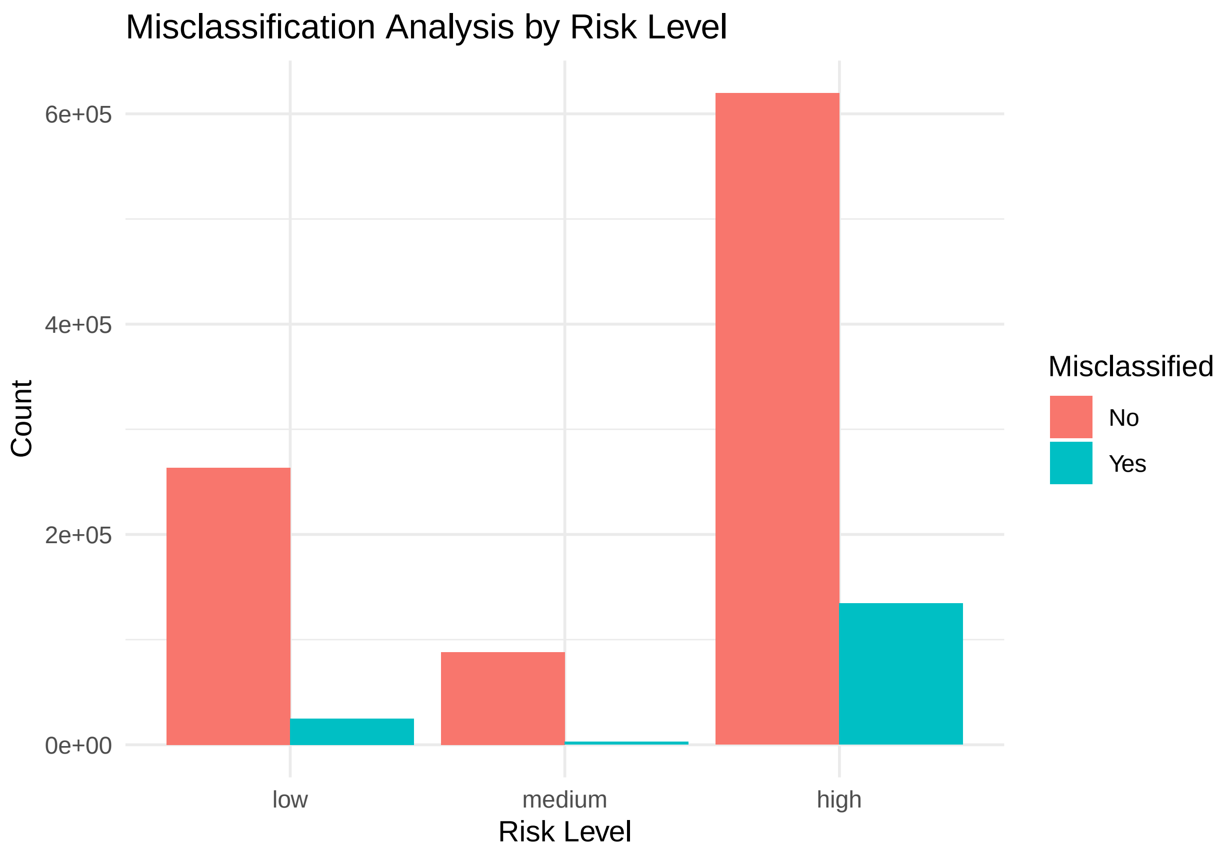
4. Feature Importance for Random Forest



5. Bar Chart Comparison of Model Metrics



6. Misclassification Analysis



Conclusion

These visuals provide insights into the models' performance and help stakeholders understand the trade-offs between accuracy, precision, and recall for each classification method. They also highlight areas for improvement, such as addressing misclassifications.

---

4. Deployment

- **Practical Use**: The model can guide early interventions (e.g., mask mandates, closures).
- **Update Frequency**: Weekly updates based on new data.
- **Integration**: Stakeholders can incorporate model predictions into decision-making frameworks.

```r
# Save Random Forest model
saveRDS(rf_model, file = "rf_model_balanced.rds")

# Load the model
loaded_model <- readRDS("rf_model_balanced.rds")
```

Appendix

- **Team Contributions**:
  - Olivia Hofmann: Lead on data preparation and feature engineering.
  - Michael Perkins: Lead on modeling and evaluation.
- **Graduate Work**:
  - Additional models: Gradient Boosting and k-Nearest Neighbors (to be implemented).