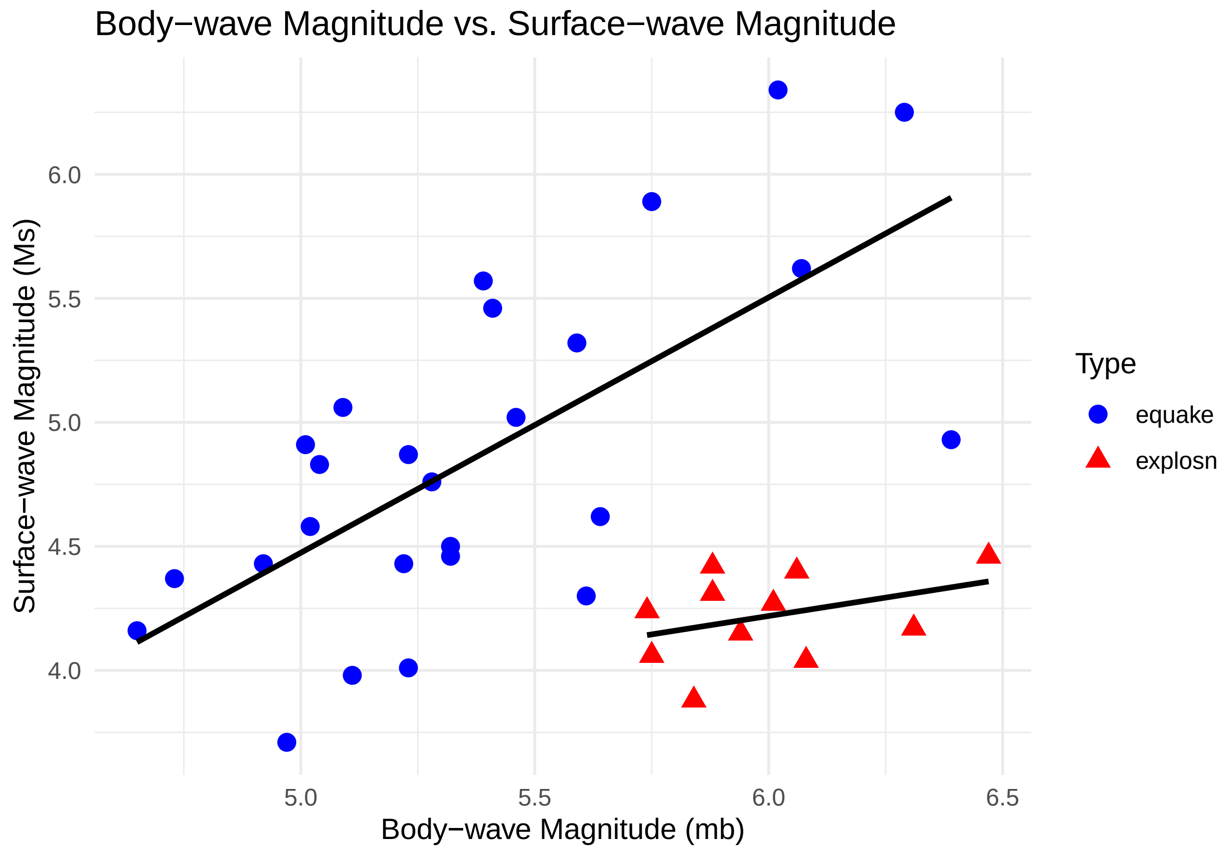
Report_10883408

10883408

2024-04-08

```
## Warning: package 'ggplot2' was built under R version 4.3.3
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



```r
# Load necessary libraries
library(randomForest)
```

```
## Warning: package 'randomForest' was built under R version 4.3.3
```

```
## randomForest 4.7-1.1
```

```
## Type rfNews() to see new features/changes/bug fixes.
```

```
##
## Attaching package: 'randomForest'
```

```
## The following object is masked from 'package:ggplot2':
##
```

```
##     margin
```

```r
library(ggplot2)
library(caret)
```

```
## Warning: package 'caret' was built under R version 4.3.3
```

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

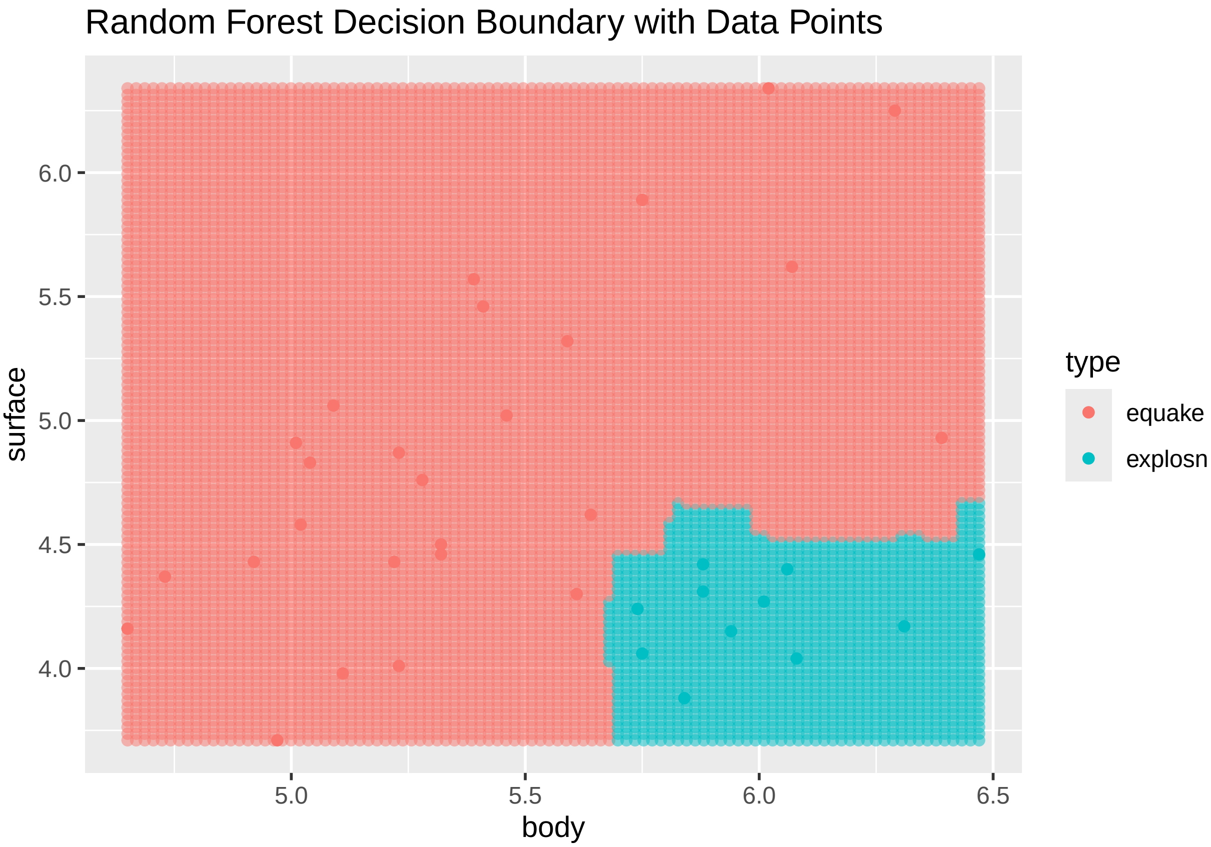
```r
data <- read.table(file = "earthquake.txt", header = TRUE, sep = "", dec = ".")
# 2. Preprocess the data (ensure correct data types)
data$type <- as.factor(data$type)

# 3. For hyperparameter tuning, normally you'd create a train-test split or use cross-validation.
# Since we're evaluating with leave-one-out cross-validation later, we'll skip splitting here.

# 4. Train the Random Forest model
# Perform a grid search for hyperparameter tuning (example: mtry)
# For simplicity, we'll skip the tuning part and use default parameters here
set.seed(123) # For reproducibility
rf_model <- randomForest(type ~ body + surface, data=data)

# 5. Model visualization
# Predict on a grid to visualize decision boundaries
plot_data <- with(data, expand.grid(body=seq(min(body), max(body), length.out=100),
                                    surface=seq(min(surface), max(surface), length.out=100)))
plot_data$type <- predict(rf_model, newdata=plot_data, type="class")

ggplot(data, aes(x=body, y=surface, color=type)) +
  geom_point() +
  geom_point(data=plot_data, aes(x=body, y=surface, color=type), alpha=0.5) +
  labs(title="Random Forest Decision Boundary with Data Points")
```


Random Forest Decision Boundary with Data Points
surface
6.0
5.5
5.0
4.5
4.0
body
5.0
5.5
6.0
6.5
type
equake
explosn