MATH501 COURSEWORK

10880416 10873945 10831920 10819616

2024-03-29

Machine Learning Task

This section contain machine learning tasks used to construct a rule that accurately distinguishes signals generated by nuclear explosions from those generated by earthquakes enables scientists to learn if a foreign power is developing nuclear weapons.

Exploratory Data Analysis

Prior to developing machine learning classifiers, exploratory data analysis offers an initial understanding of the dataset. Table 1 presents overall summary metrics for seismic waves, while Table 2 offers statistics of the waves categorized by type.

Table 1: Summary statistics for body and surface waves

	Mean	Standard Deviation	Minimum	25%	Median	75%	Maximum
Body	5.56	0.49	4.65	5.22	5.59	5.94	6.47
Surface	4.67	0.65	3.71	4.24	4.46	4.93	6.34

Table 2: Summary statistics for body and surface waves based on type

	Type	Mean	Standard Deviation	Minimum	25%	Median	75%	Maximum
Body	Equake	5.38	0.45	4.65	5.05	5.30	5.61	6.39
	Explosn	6.00	0.23	5.74	5.86	5.94	6.07	6.47
Surface	Equake	4.86	0.68	3.71	4.43	4.80	5.26	6.34
	Explosn	4.22	0.18	3.88	4.11	4.24	4.36	4.46

As seen from table 1, the magnitude of body wave is higher than surface wave as indicated by the mean and median value of the two waves.

From table 2, it was observed that the magnitude of body and surface waves vary with the type of signals i.e. earthquake or explosion. Body waves are observed to be higher in magnitude in cases of nuclear explosion as compared to earthquake. In contrary, surface waves generated through earthquake have higher magnitude than those generated via nuclear explosion.

This insight is further explored using a boxplot as shown in figure 1 below:

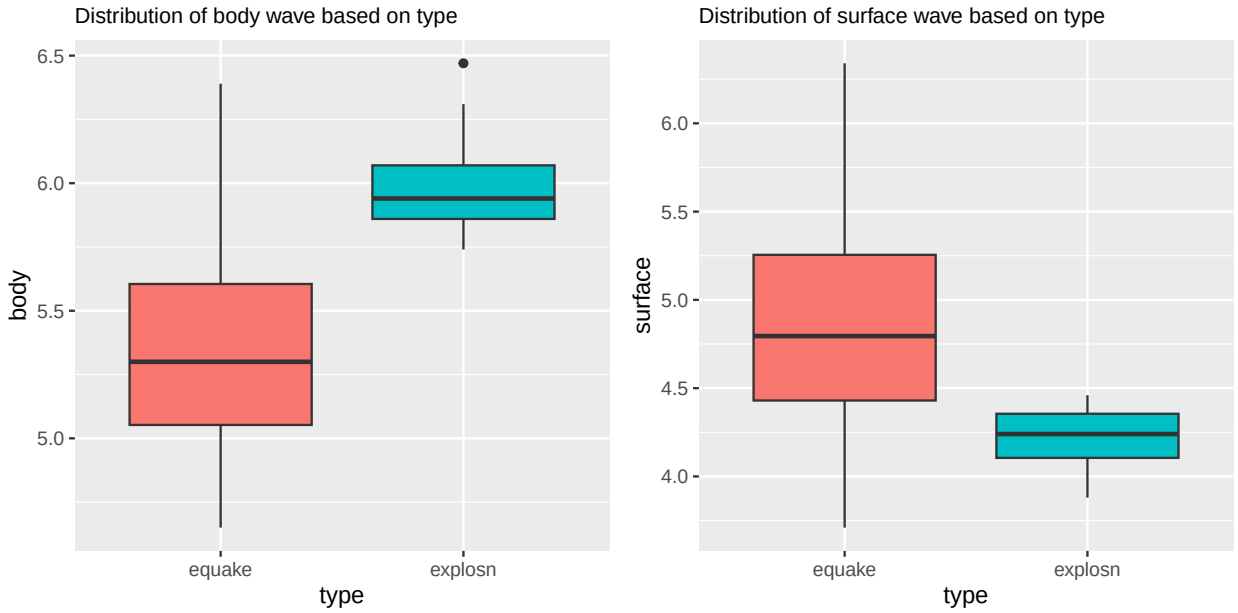

Fig. 1: Box plots showing distribution of body and surface waves, for both signal type.

The boxplot from figure 1 offers an easy comparison between the two variables, body and surface waves based on the type of signals. Based on the graph, the type of signal generated can easily be determined using the observation from the body and surface waves. As seen from the graph, signal generated from nuclear explosion usually have high body wave magnitude (mean = 5.99) than earthquake with a mean body wave of 5.37. However, considering surface wave, signals generated from nuclear explosion have a lower magnitude (mean = 4.21) as compared to those generated from earthquake (mean = 4.86). Therefore, it can be concluded that signals generated from nuclear explosion have high body wave but low surface wave while those generated from earthquake have low body wave but high surface wave.

To further support the claim above, a statistical test (t-test) was conducted to determine if the difference in the observed means for body and surface waves based on the type of signals are significant. The results obtained revealed a very low p-value for both variables. The mean difference in the body wave of earthquake and explosion signal types produced a p-value of **0.00000313**. Similarly, the mean difference in the surface wave between the two type of signals produced a p-value of **0.0000955**. Hence, it can be concluded that these two variables are significant in predicting the type of signals generated.

Furthermore, the relationship between the two predictor variables; body and surface, can be explored using a scatterplot. Using type as factor variable, the relationship between body and surface waves can be visualized as shown in figure 2 below:

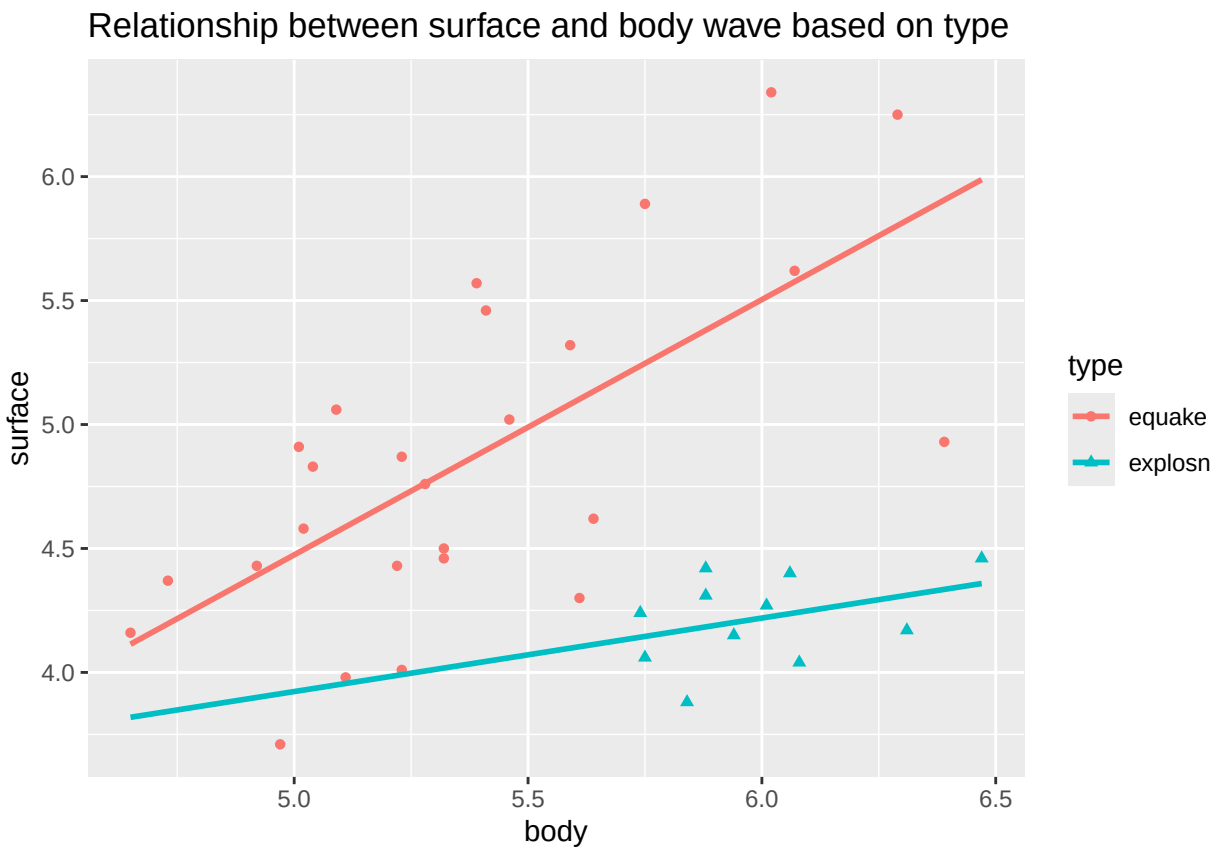

Fig. 2: Scatterplot with regression line showing the relationship between surface and body wave, and the type of signal.

A clear distinction can be seen in the area covered by both type of signals. The earthquake signals occupied mostly the left-upper part of the graph representing a lower body wave (x-axis) and higher surface wave (y-axis). Meanwhile, the signals generated from nuclear explosion are seen to occupy the right-lower part of the graph, indicating a higher body wave and lower surface wave. The distinction in the cluster of the type categories is an indication that the two variables, body and surface waves, are good predictor of the type of signal.

We further explore the relationship between the three variables of the dataset to gain an insight into the correlation between variables and spread of the data. To do this, the variable type was converted from a factor variable to numerical variable with "equake" coded as 0 and "explosn" coded as 1.

```
# convert type category into numeric
data_numeric <- data %>%
  mutate(type = as.numeric(type)) # Convert to numeric

# The type variable is automatically converted to numeric; 1 for equake and 2 for explosn
# It will be reconverted to 0 and 1 for the machine learning purpose
data_numeric <- data_numeric %>%
  mutate(type = case_when(
    type == 2 ~ 1,
    type == 1 ~ 0
  ))
```

The correlation between the three variables including their correlational coefficients and distributions of the data was then visualized using ggpairs as shown below:

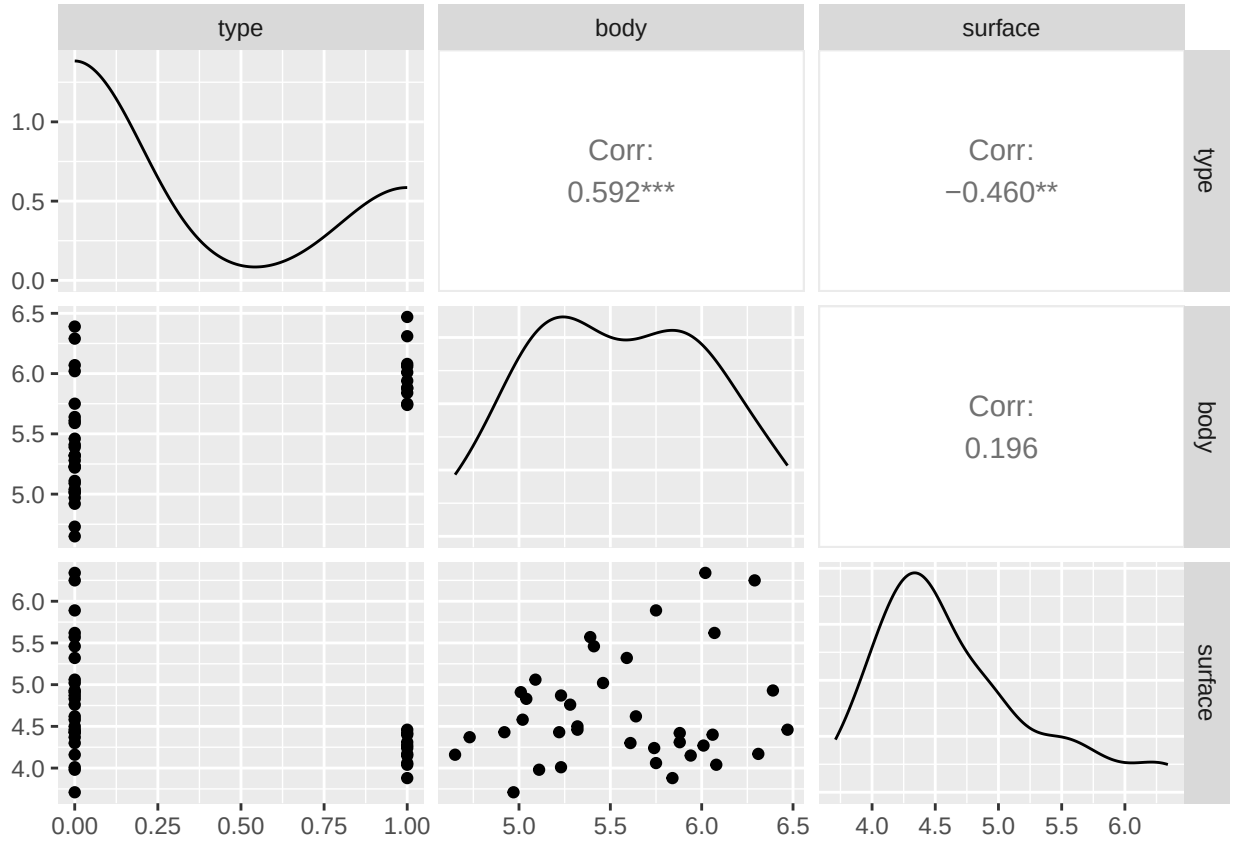

Fig. 3: A Matrix of plot showing the correlation between the three variables of the data and their correlation coefficients.

The plot in figure 3 provides some insights into the relationship between the three variables. The diagonal of the plot represents the distribution of each of the variable type, body and surface. The relationship between type and the two predictor variables body and surface are represented in row 2, column 1 and row 3, column 1 respectively. The scatterplots for the two relationship are not linear and not much information can be gained. The non-linearity is due to the fact that the type variable is not a continuous variable unlike body and surface. In contrast, row 3 column 2 represents the relationship between surface and body. The relationship is linear indicating a positive relationship. This can be interpreted to mean that an increase in surface wave likely to cause an increase in body wave.

The correlation coefficients indicates the magnitude of the relationship between the three variables. As seen from figure 3, body wave and type variable have the highest correlation coefficient of 0.592.This revealed that there is a positive relationship between body waves and type of signal and this relationship is significant (indicated by the *** on the coefficient value). This is similar to our observation in figure 2, that an increase in body wave can be attributed to type 1 (nuclear explosion). In contrast, the correlation coefficient between type and surface wave is -0.460. This represents a negative and significant relationship between these two variables. Hence, an increase in or higher magnitude of surface wave can be attributed to type 0 (earthquake). Lastly, body and surface wave show a positive correlation with a coefficient of 0.196. However, this relationship is not significant and therefore, we can not conclude that an increase in one of the wave will lead to an increase in the other.

Machine Learning Supervised Classification

Two supervised classification methods selected for this machine learning task are K-nearest neighbor and Support Vector Machine. These two methods were chosen due to their simplicity, flexibility, ability to handle non-linear relationships, and robustness against overfitting. Also, due to the small size of the dataset, these methods do not make any assumptions about the underlying data distribution, which can be advantageous for this dataset where assumptions may not hold true.

In addition to the general characteristics of the two classifiers chosen, KNN is simple to understand and implement, making it suitable for quick prototyping and exploration of small datasets. Meanwhile, SVM allows the use of different kernel functions (e.g., linear, polynomial, radial basis function), which can capture complex relationships in the data and adapt to non-linear decision boundaries.

Implementation of Machine Learning Classifiers

The two classifiers will be implemented following the procedures below:

- Train-test split;
- Model Tuning;
- Model Visualization; and
- Model Evaluation

Train-test split

Before implementing the machine learning classifiers, the dataset was first split into two parts; training set and testing set. The training set covers 70% of the data while the remaining 30% was assigned to the testing set. The splitting of the dataset into training and testing set was given below:

```
# Data Partitioning
set.seed(1234) # to make the splitting reproducible
ind <- sample(2, nrow(data), replace = T, prob = c(0.7, 0.3)) # generates a random sample
# of indices (ind) indicating whether each row in the data should belong to the first
# or second group.

training <- data[ind == 1,] # first group for training, 70%
test <- data[ind == 2,] # second group for testing, 30%
```

Due to the small size of the dataset, 26 rows, the splitting was limited to training and testing set with no validation set. However, the train control function from caret package allow us to control parameters for our model by using 3-folds cross validation.

```
# control parameters for train function
library(caret)
trControl <- trainControl(method = "cv", number = 3)

# the resampling method for hyperparameter tuning is set for "cv" - cross validation
# the number = 3; represents the 3-folds for the cross validation
```

KNN Model

The key hyperparameter in KNN is K, which represents the number of neighbors to consider. Choosing an appropriate value for K is crucial, as it affects the bias-variance trade-off of the model. Therefore, careful selection of the hyperparameter K is required. We will print out the output of the model to explore how the KNN model was trained with the value of K been tuned to achieve the highest accuracy.

```
set.seed(123) # to make the model training reproducible
knn_model <- train(type ~ ., data = training,
                   method = "knn", trControl = trControl,
                   tuneLength = 20)
# the training set is used to train the KNN model
# trControl specified the 3-folds cross validation mentioned earlier
# tuneLength of 20 specifies the number of values to be explored for tuning hyperparameter
# of kNN, 'k', which represents the number of nearest neighbors to consider.

print(knn_model) # display the output of the model
#> k-Nearest Neighbors
#>
#> 30 samples
#>  2 predictor
#>  2 classes: 'equake', 'explosn'
#>
#> No pre-processing
#> Resampling: Cross-Validated (3 fold)
#> Summary of sample sizes: 20, 21, 19
#> Resampling results across tuning parameters:
#>
#>   k   Accuracy   Kappa
#>    5  1.0000000  1.0000000
#>    7  1.0000000  1.0000000
#>    9  0.9666667  0.9122807
#>   11  0.8090909  0.3333333
#>   13  0.7350168  0.0000000
#>   15  0.7350168  0.0000000
#>   17  0.7350168  0.0000000
#>   19  0.7350168  0.0000000
#>   21  0.7350168  0.0000000
#>   23  0.7350168  0.0000000
#>   25  0.7350168  0.0000000
#>   27  0.7350168  0.0000000
#>   29  0.7350168  0.0000000
#>   31  0.7350168  0.0000000
#>   33  0.7350168  0.0000000
#>   35  0.7350168  0.0000000
#>   37  0.7350168  0.0000000
#>   39  0.7350168  0.0000000
#>   41  0.7350168  0.0000000
#>   43  0.7350168  0.0000000
#>
#> Accuracy was used to select the optimal model using the largest value.
#> The final value used for the model was k = 7.
```

Let us visualize the change in the accuracy of the model at different values of K.

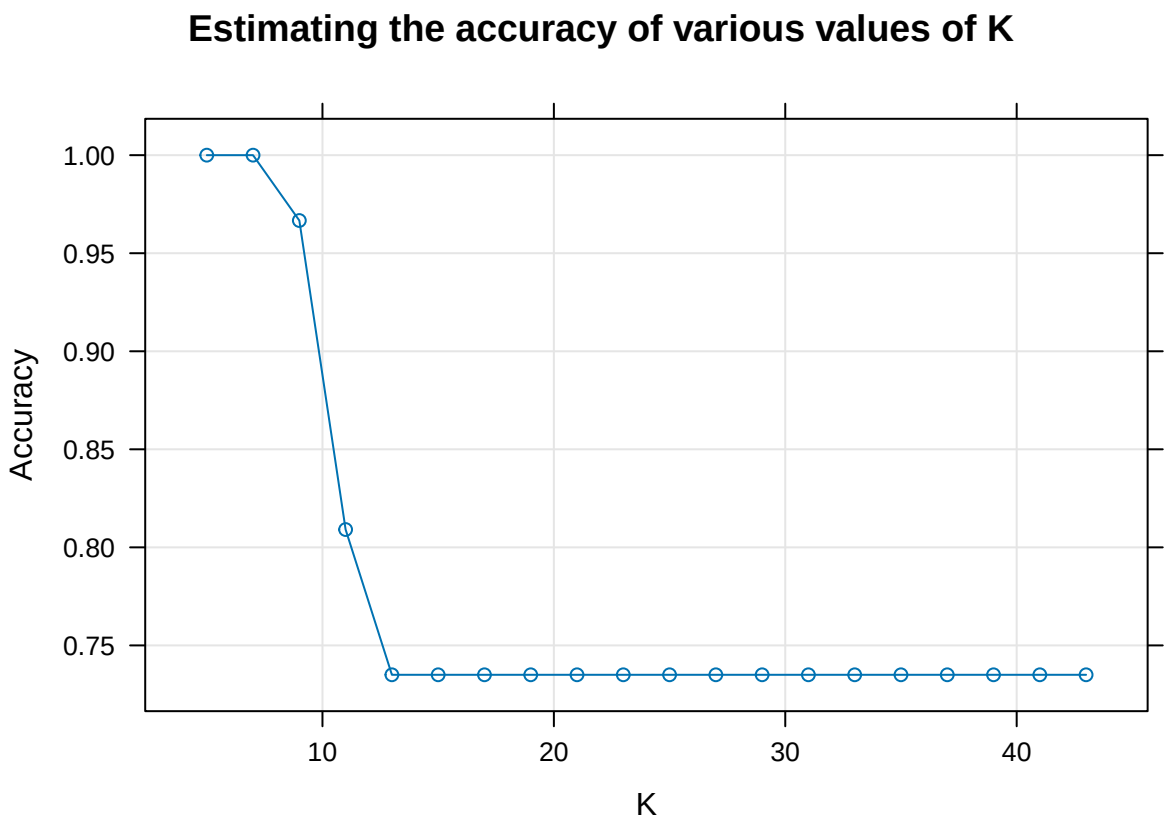

Fig. 4: A graph showing the accuracy of KNN model at different values of K.

It was observed from figure 4 and the model output printed earlier that the value of K that gives the highest accuracy is 7. Reasonably, the train function from the caret library automatically makes use of the optimal k to train the model. Therefore, we do not need to worry about specifying the value of K while training the model.

Visualizing the KNN classifier

Given the optimal K to be 7, the classification boundary of the model at this value of K was visualized and presented in figure 5.

```
# Generate predictions for a grid of points

# Create a sequence of values for the "body" and "surface" feature
x1_range <- seq(min(data$body), max(data$body), length.out = 100)
x2_range <- seq(min(data$surface), max(data$surface), length.out = 100)
# Create a grid of points based on the sequences
grid <- expand.grid(body = x1_range, surface = x2_range)
# Use the trained model to predict the class labels for each point on the grid
predictions_grid <- predict(knn_model, newdata = grid, type = "raw")

# Create a scatter plot of the original data
ggplot(training, aes(x = body, y = surface, color = type)) +
```

```
geom_point() + # Add points to the plot
scale_color_manual(values = c("blue", "darkred"))+ #Specify custom color for the points
theme_classic() + # Set the plot theme to classic
labs(title = "Decision Boundary for KNN Classifier at optimal K",
     x = "body", y = "surface") + # Set plot title and axis labels

# # Overlay predicted regions as tiles
geom_tile(data = grid, aes(x = body, y = surface, fill = factor(predictions_grid)),
          alpha = 0.2, inherit.aes = FALSE) +
scale_fill_manual(values = c("blue", "red")) # Specify custom fill colors for the tiles
```

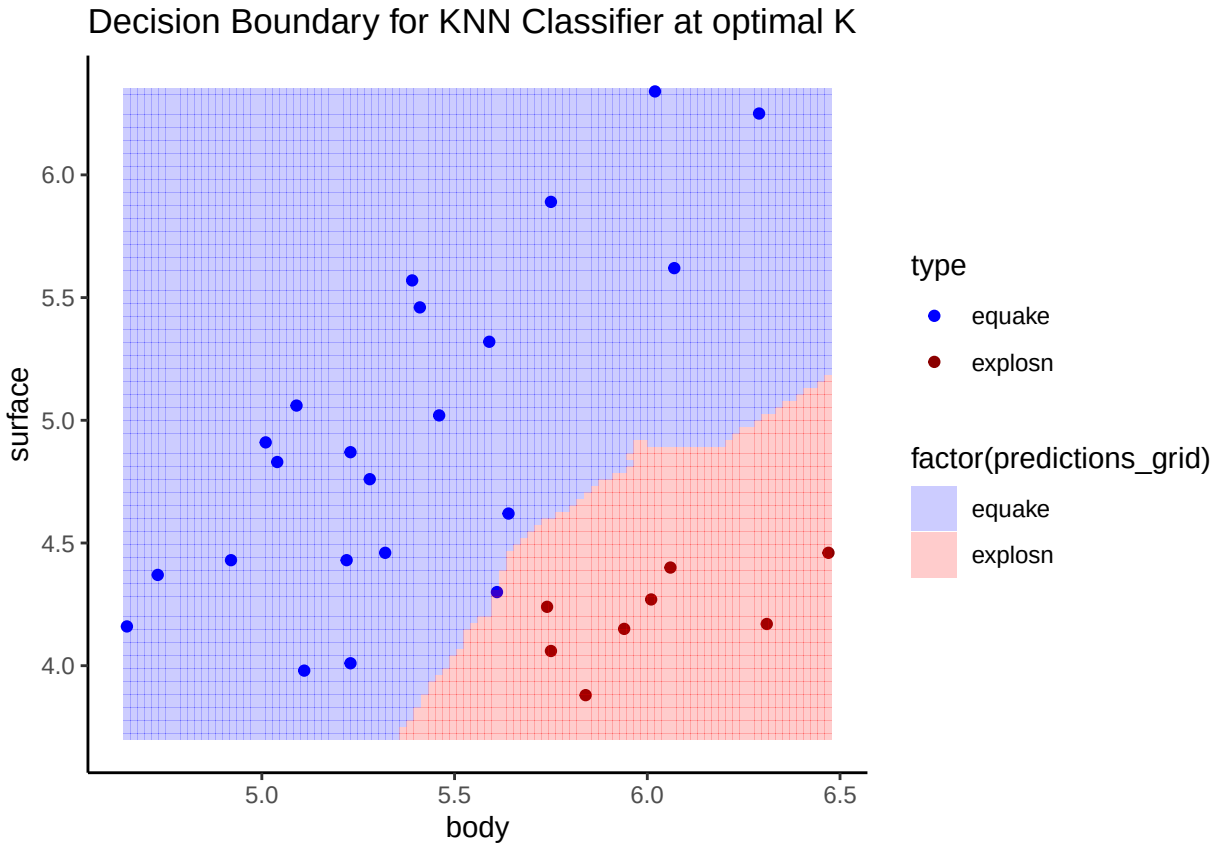

Fig. 5: Classification boundary for KNN classifier at optimal K

The classification boundary shown in figure 5 revealed that the KNN model performed well by accurately predicted a boundary for the test grid. It can be observed that the predicted area for the test grid (blue and red tile) aligned with the original scatterplot of the training data in which blue points represent "equake" and red points represent "explosn". This means that the decision boundary closely follows the training instances.

Model Evaluation

The final performance of the KNN model will be evaluated using leave-one-out cross-validation (LOOVC) to estimate the classification error. In LOOCV, each observation in the dataset is systematically held out as a validation set while the model is trained on the remaining data. During each iteration of LOOCV, the model is trained on n_1 observations and then evaluated on the single observation that was left out. The classification error for each iteration is computed based on the performance of the model on the corresponding

validation set. Finally, the average classification error across all iterations is calculated to estimate the overall performance of the model.

To carry out LOOCV, the resampling method in train function will be specified as LOOCV.

```
# set the train control method to LOOCV
trControl <- trainControl(method = "LOOCV")
# train the KNN model using LOOCV method
set.seed(123)
model_loocv <- train(type ~ ., data = training,
                     method = "knn", trControl = trControl)

print(model_loocv) # display the model output
#> k-Nearest Neighbors
#>
#> 30 samples
#>  2 predictor
#>  2 classes: 'equake', 'explosn'
#>
#> No pre-processing
#> Resampling: Leave-One-Out Cross-Validation
#> Summary of sample sizes: 29, 29, 29, 29, 29, 29, ...
#> Resampling results across tuning parameters:
#>
#>   k  Accuracy   Kappa
#>   5  0.9666667  0.9180328
#>   7  0.9666667  0.9180328
#>   9  0.9666667  0.9180328
#>
#> Accuracy was used to select the optimal model using the largest value.
#> The final value used for the model was k = 9.

# Evaluate the classification error
error <- 1 - model_loocv$results$Accuracy[3] # classification error
# when k is at optimal i.e. k = 9

# Print the classification error
cat("Classification error (LOOCV):", error, "\n")
#> Classification error (LOOCV): 0.03333333
```

The classification error for the KNN model using leave-one-out cross-validation is 0.0333. In contrast to the 3-folds cross validation method used earlier, the LOOCV method gives a lower accuracy. Also, the optimal value of K when LOOCV method is used is 9 while optimal K is 7 when 3-folds cross validation was performed.

SVM Model

SVM works by finding the optimal hyperplane that best separates data points into different classes in a high-dimensional space. SVM can perform linear classification by finding a linear decision boundary. Additionally, through the use of kernel functions (e.g., polynomial, radial basis function), SVM can handle non-linear classification tasks by transforming the feature space into a higher-dimensional space. SVM is suitable for small dataset because it is computationally intensive. In addition, SVM is sensitive to the choice of kernel function and its parameters. Therefore, hyperparameter tuning of the parameters is required to train an optimal model.

The hyperparameters to be tuned for SVM model include the cost and gamma. Using the tune function from e1071 library, we can obtain the best parameters for the SVM model. However, we are going to be tuning parameters for linear, radial and polynomial SVM differently. We will then compare which of the three model performs better before training the final model with optimal parameters.

```
library(e1071)
#>
#> Attaching package: 'e1071'
#> The following objects are masked from 'package:PerformanceAnalytics':
#>
#>     kurtosis, skewness

# find the optimal hyperparameters for SVM linear model
set.seed(1234)
tuned_Lparameters <- tune(svm, type ~ .,
                          data = training,
                          kernel = "linear",
                          ranges = list(cost = c(0.1, 1, 10, 100),
                                        gamma = c(0.1, 1, 10, 100)))
print(tuned_Lparameters)
#>
#> Parameter tuning of 'svm':
#>
#> - sampling method: 10-fold cross validation
#>
#> - best parameters:
#>  cost gamma
#>     1   0.1
#>
#> - best performance: 0
```

As seen from the output, the best parameters for the linear SVM model after 10-folds cross validation are cost = 1; gamma = 0.1. Following the same procedure, the best parameters for radial SVM model was also obtained as cost = 1; gamma = 0.1. In contrast, polynomial SVM gives a value of cost = 1; gamma = 1. However, this method of tuning the hyperparameters is not flexible as it only consider very small value of cost and gamma. However, we will find the accuracy of the model by looping the value of cost and gamma in a range of 1 to 10.

Table 3: Accuracy of Polynomial SVM at different value of cost and gamma

c	g	accuracy
0.1	0.1	0.5714286
0.1	1.0	0.5714286
0.1	10.0	0.8571429
1.0	0.1	0.5714286
1.0	1.0	0.8571429
1.0	10.0	0.8571429
10.0	0.1	0.5714286
10.0	1.0	1.0000000
10.0	10.0	0.8571429

The tuning of the cost and gamma hyperparameters as shown in table 3 revealed that the value of cost and

gamma that yield the best accuracy when considering polynomial kernel is 10 and 1 respectively. Radial kernel also yielded an accuacy of 100% with cost = 10; gamma = 10. However, the linear kernel yields a lower accuracy for all values of cost and gamma as compared to radial and polynomial. Therefore, we will be training our SVM model using polynomial kernel since it yielded a great accuracy with just small value of gamma. It should be noted that higher values of gamma lead to more complex decision boundaries, which can also contribute to overfitting.

```
# Polynomial SVM model
set.seed(123)
svm_Ptuned_model <- svm(type ~ ., data = training,
                        kernel = "polynomial",
                        cost = 10,
                        gamma = 1.0 ) # best tuned parameters specified

# Visualize the model
plot(svm_Ptuned_model, training, color.palette = heat.colors)
```

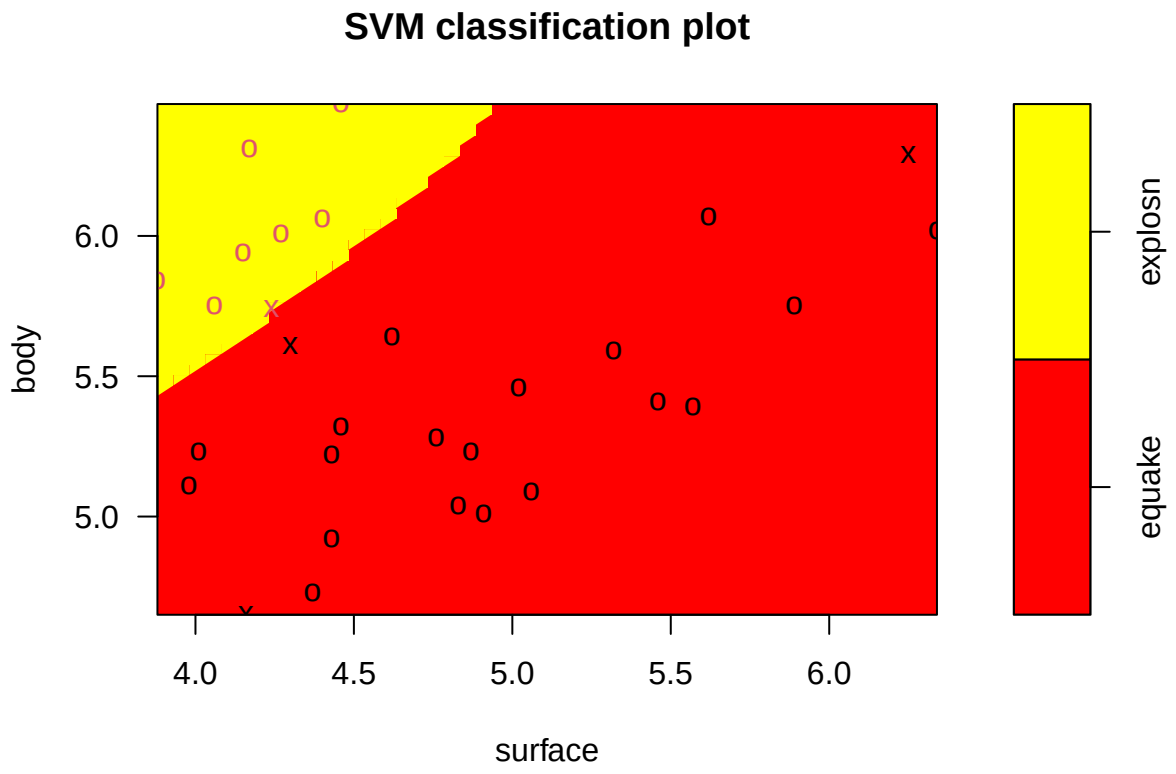

Fig. 6: Classification plot for polynomial SVM model

The figure 6 above shows the classification plot for polynomial kernel SVM model. The red and yellow color palette represent the decision boundary for the two classes “equake” and “explosn” respectively. The “o” symbol represents the data point while the “x” represents the support vectors.

Not much insights can be obtained from the classification plot. To better understand our SVM model, we will use the trained model to predict the class labels for each point on the grid covering the range of values for the ‘body’ and ‘surface’ features in the dataset and visualize the classification boundary as we did earlier with KNN classifier.

Visualization of the SVM Classifier

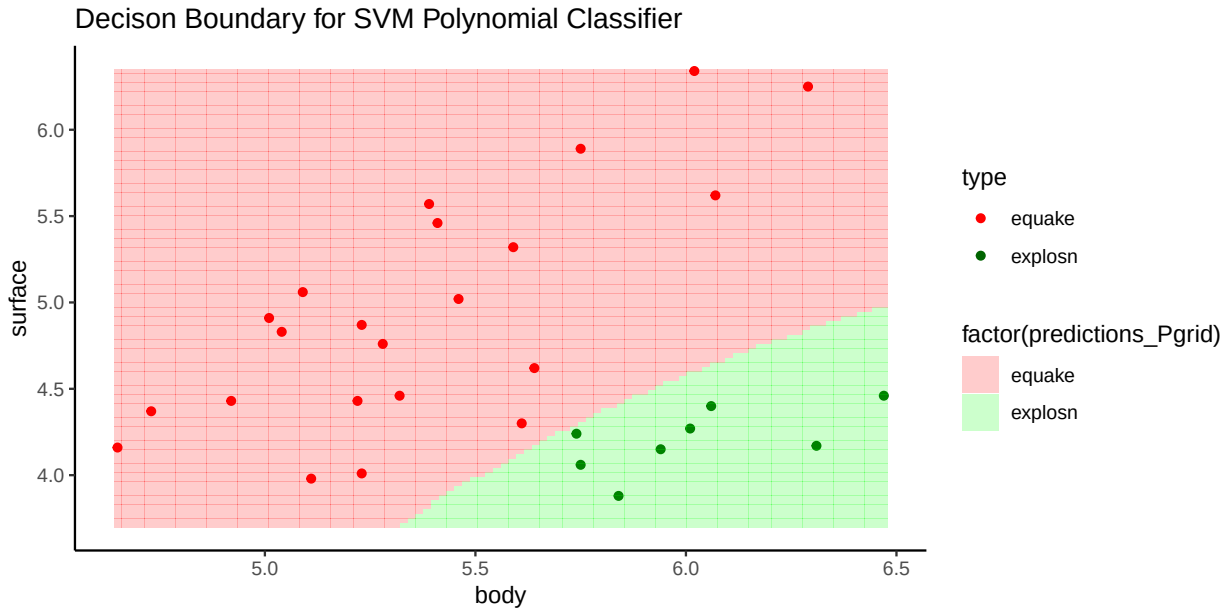

Fig. 7: Classification boundary for the polynomial SVM model.

The classification boundaries for polynomial SVM model shown in figure 7 revealed a great insight about its suitability in classifying accurately signals generated from earthquake as compared to those generated from nuclear explosion. The polynomial SVM model have a distinct decision boundary which separate the two class labels into two different clusters. The decision boundary observed in polynomial SVM is smoother and more distinct than the one obtained in KNN model in figure 5. While in KNN classification boundary, a point belonging to the class label "explosn" was found in the prediction grid for "equake". This suggest that polynomial SVM models is the most suitable classifier for the dataset.

Model Evaluation

We further evaluate the SVM model using leave-One-Out Cross-Validation (LOOCV). This was similar to the procedure used in evaluating KNN, except that in these cases, "SVMPolynomial" is specified as the train method.

```
# set the train control method to LOOCV
trControl <- trainControl(method = "LOOCV")
# train the polynomial SVM model using LOOCV method
set.seed(123)
SVM_loocv <- train(type ~ ., data = training,
                   method = "svmPoly", # SVMPoly is the train method
                  trControl = trControl)
```

Following the LOOCV method, SVM model produced an accuracy of 1.00 at C=1;degree = 3; scale = 0.1. Therefore, the classification error for this model using LOOCV method is 0.

Comparing Classifiers

Confusion matrix of classification models offers metrics for evaluating the performance of the classifiers. Metrics such as accuracy, sensitivity, specificity, precision and ROC allow for comparism between the our models. The confusion matrix for KNN model and SVM are shown in table 3 and 4 respectively.

Table 4: Confusion matrix for KNN

	equake (Observed)	explosn (Observed)
equake (Predicted)	3	0
explosn (Predicted)	1	3

Table 5: Confusion matrix for Radial SVM

	equake (Observed)	explosn (Observed)
equake (Predicted)	4	0
explosn (Predicted)	0	3

The confusion matrices displayed in table 3 and table 4 for KNN and SVM model showed that SVM model is better in classifying signal types appropritely than the KNN model. This indicated that SVM model perform better on our dataset than KNN.

Furthermore, we will compare the metrics of the two model and visualize their ROC before making conclusion about the two models.

Table 6: Performance metrics for KNN and SVM classifiers

	Accuracy (%)	Sensitivity	Specificity	Precision	AUC
KNN	85.714	0.75	1	1	0.875
SVM	100.000	1.00	1	1	1.000

Table 5 further revealed the difference in the two models in terms of accuracy, specificity, sensitivity, precision and AUC. This strongly supports the claim we made earlier that the SVM model is better suitable for classifying signal types based on body and surface wave. SVM model was observed to have an accuracy of 100%, sensitivity (1.00), specificity (1.00), precision (1.00) and AUC (1.00) which are greater than the metric calculated for KNN (accuracy = 85.71%; specificity = 1.00, sensitivity = 0.75, precision = 1.00, AUC = 0.875). KNN model has a sensitivity of 0.75, this means that there is still 25% chance that an earthquake signal (positive class) will be misclassified as nuclear explosion (negative class). Meanwhile, SVM model has a specificity and sensitivity of 1.0, which reflected the ability of the model to accurately classify signals with a zero chance of false positive or false negative.

Figures 8 and 9 display ROC curves, accompanied by their respective areas under the curve (AUC), serving to underscore the performance of both classifiers. Upon visual examination of these figures, it becomes apparent that SVM exhibits superior predictive capabilities for signal types compared to KNN, as indicated by a notably larger AUC. Specifically, the AUC of the SVM classifier (1.0) significantly surpasses that of the KNN classifier (0.875), reinforcing the assertion that SVM stands out as the superior classifier for our dataset.

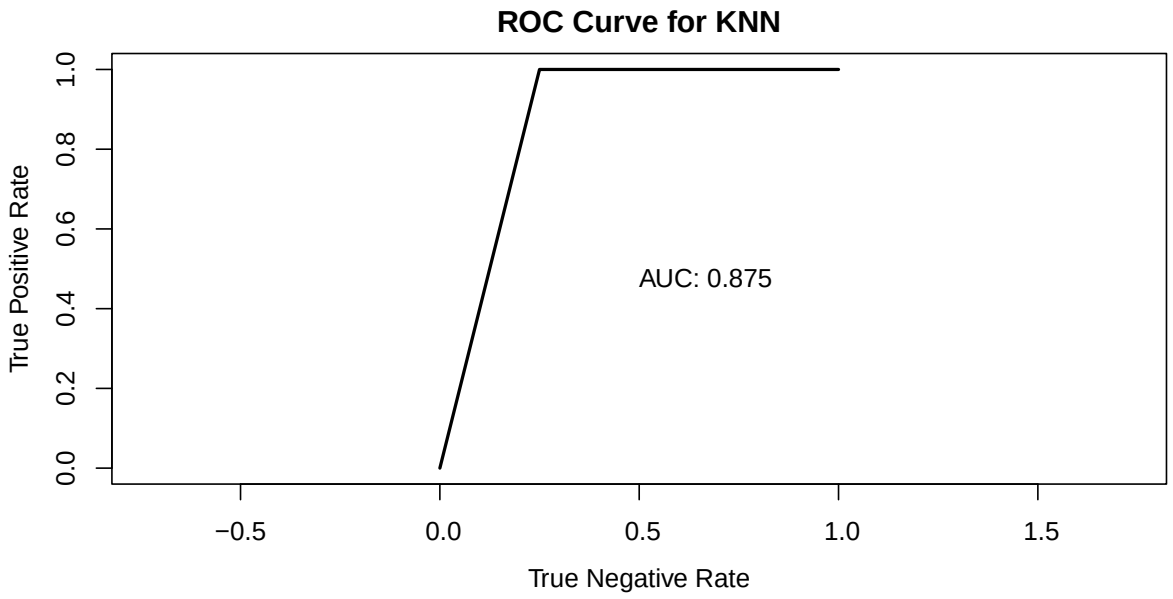

Fig 8: ROC curve for KNN

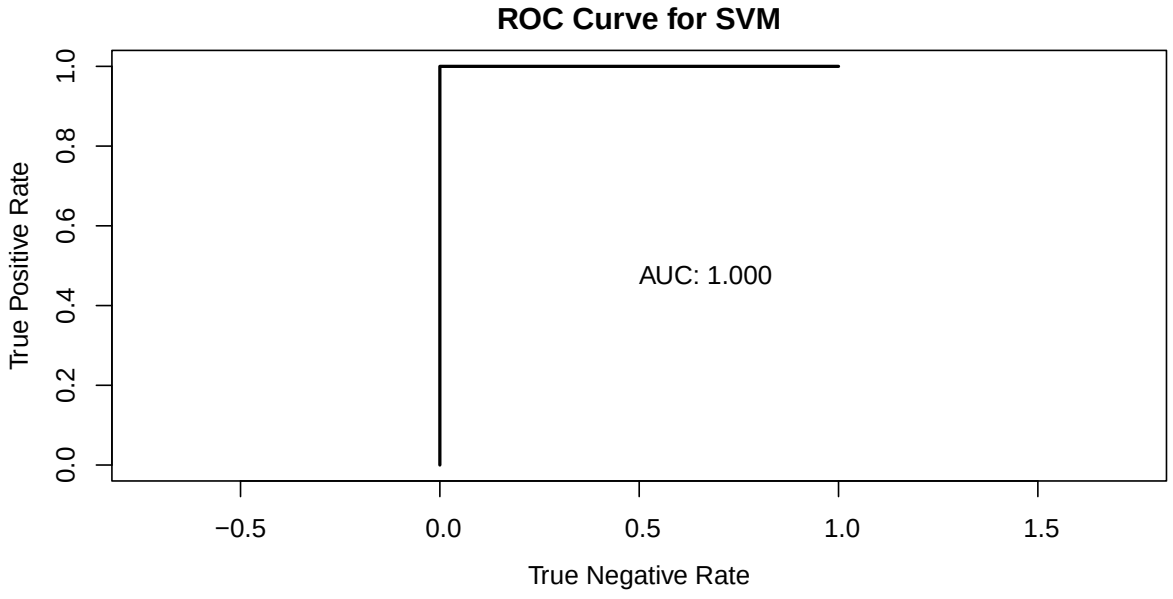

Fig 9: ROC curve for SVM

In terms of predicting whether a signal originates from an earthquake or a nuclear explosion, the SVM with a polynomial kernel, with a cost parameter set to 10 and a gamma value of 1.0, emerges as the superior classifier. It outperforms the KNN classifier across all evaluation metrics, particularly excelling in accurately identifying signal types based on features derived from both body and surface waves. Despite the potential drawback of the SVM's susceptibility to misclassifying signals lacking certain characteristics due to its high cost parameter, it nonetheless demonstrates robust performance in accurately predicting test data, thus establishing its suitability for signal type classification over KNN. Consequently, the SVM stands out as the preferred classifier due to its superior performance across all metrics, especially in scenarios involving the prediction of signal types from body and surface wave features.

KMeans Clustering

K means clustering is an unsupervised machine learning algorithm used for partitioning a dataset into distinct groups, or clusters, based on similarities in the data points' features. The goal of K-means clustering is to group data points into K clusters, where each cluster is represented by its centroid, such that the distances between data points within the same cluster are minimized while maximizing the distances between different clusters.

In this task, we will group the data points in our dataset to distinct group based on body wave and surface wave features only. Ignoring the type variable, we will explore if clustering could be useful to distinguish between earthquakes and explosions.

```
# Prepare data for k means (remove the 'type' variable)
data_kmeans <- data[, c("body", "surface")]

set.seed(123)
# performs k-means clustering
library(cluster)
km <- kmeans(data_kmeans, centers = 2, nstart = 20)

# centers parameter specifies the number of clusters to create, we set it to 2
# nstart specifies the number of times to run the algorithm with different initial centroids.

print(km)
#> K-means clustering with 2 clusters of sizes 29, 8
#>
#> Cluster means:
#>       body  surface
#> 1 5.476207 4.393103
#> 2 5.863750 5.672500
#>
#> Clustering vector:
#>  [1] 2 1 1 2 1 2 1 1 1 2 1 1 1 1 1 1 1 1 1 2 1 2 2 1 2 1 1 1 1 1 1 1 1 1 1
#>
#> Within cluster sum of squares by cluster:
#> [1] 9.783503 2.604137
#>  (between_SS / total_SS =  47.5 %)
#>
#> Available components:
#>
#> [1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss"
#> [6] "betweenss"    "size"         "iter"         "ifault"
```

From the output, we can observe that two different clusters have been found with sizes 29 and 8. For each cluster, the squared distances between the observations to the centroids are calculated. So, each observation will be assigned to one of the two clusters. The mean values for body and surface wave for the clusters are not align with what we observed from the data itself. Cluster 2 with size 8 tends to have a higher mean value for body and surface which is different from what we observed in the data. The best way to find the best model for k means is to try different models with a different number of clusters.

Due to the small size of the dataset, we will look at 6 clusters to select the best cluster for the model. By examining the scree plot in figure 10, it becomes evident that the total within-cluster sum of squares diminishes as the number of clusters increases. The method for determining the optimal number of clusters involves identifying a point of inflection, known as the "elbow," where the rate of decrease in WCSS slows

significantly upon the addition of another cluster. To gain further clarity, horizontal lines are added to the plot to facilitate a more informed decision.

Fig. 10: Scree plot showing the within sum of square (wss) error as the number of cluster increases.

Based on this visualization, it appears that opting for 3 clusters is the most optimal choice. Beyond k=3, the enhancements in the models' performance appear to diminish significantly.

We can then visualize the scatterplot between the body wave and the surface waves and also colour the points based on the cluster id. However, since we are to consider a two groups i.e. earthquake and explosion, we will also visualize the scatterplot between body and surface wave and colour the points based on the cluster id. This means that we will set k=2 as number of clusters.

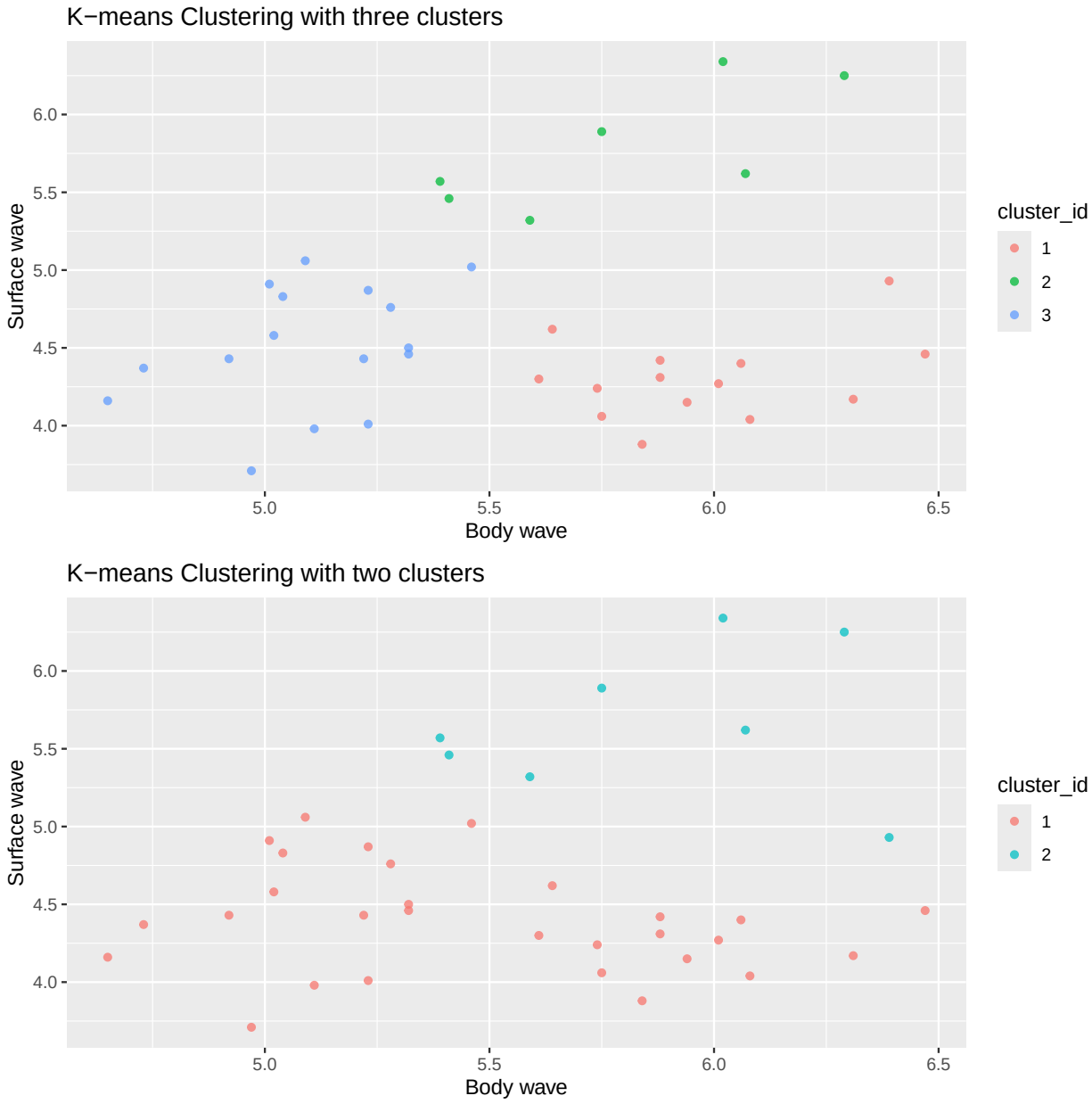

Fig. 11: Clustering of data points into three clusters and two clusters.

It can be deduced from figure 11 that the data points can be distinctly grouped into three clusters such that the distances between data points within the same cluster are minimized while maximizing the distances between different clusters. However, it can be observed that the two cluster K means in figure 11 does not reflect the original data when compared with figure 2. The data points do not follow the same pattern as we have seen in figure 5.

In conclusion, three clusters K means revealed a regular and distinct pattern in the three clusters. Cluster 1 of size 16 is made up of signals that has low body wave and low surface wave. Cluster 2 contains 7 signals point which are characterized with high body wave and high surface wave. The last cluster of size 14, cluster 3 share the same characteristics with the nuclear explosion signals from our original classification. It is characterized by low surface wave and high body wave.

Bayesian Inference Task

First Sub-Task: Frequentist One-way Analysis of Variance

The data used for this task contain the information about four different airlines and their customer rating (satisfaction score).

Visualization of the data

The boxplot produced in figure 12 offers a great insight about the satisfaction of customer with the different airline. From the graph, it was observed that airlines A and C have the same mean satisfaction score while airline B and D also share the same mean satisfaction score, which is higher than that of A and C. However, comparing B and D with the same mean score, airline D has a higher rating by customer with 75% of the customers rating them 8 as observed with the box D as compared to box B where 75% of the customer rated them as 7. For airline C, the rating is inconsistent as seen in the size of box C when compared to airline A of the same mean satisfaction score. Although airline C was rated above A looking at 75% of the scores, however, airline A does has a better score when considering the first quartile statistics of the two airlines.

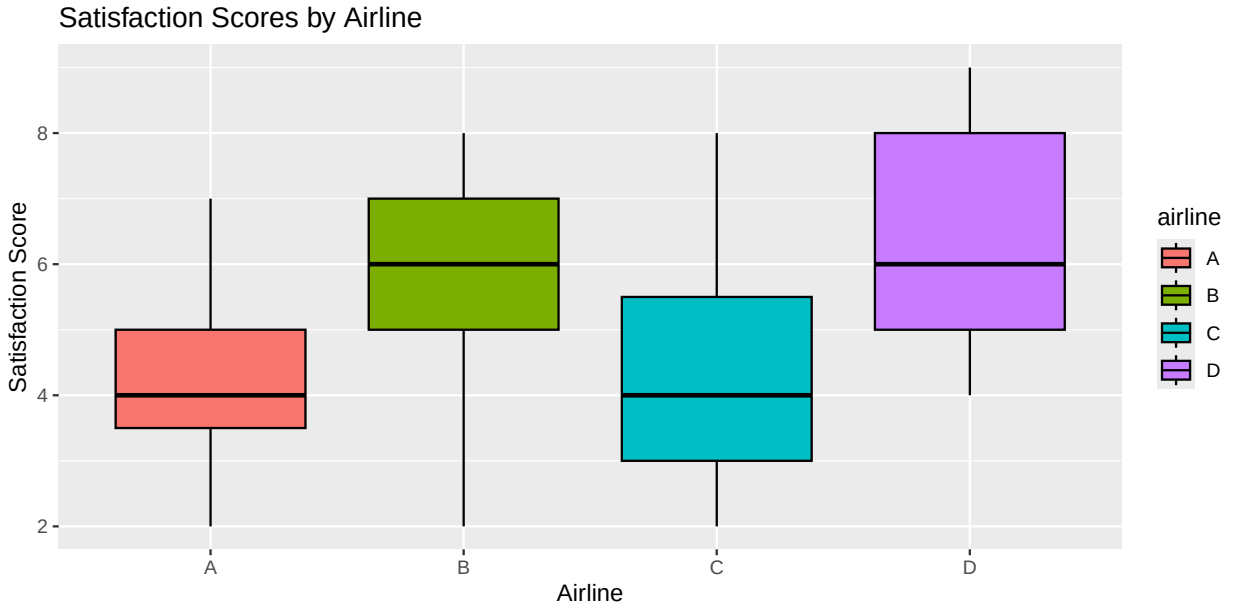

Fig. 12. A boxplot comparing the mean and distribution of satisfaction score among the four airlines.

Interpretation of α_4 for analysis of variance model

An analysis of variance (ANOVA) model was constructed to analyse which airline performed the best by comparing the average values for each group:

$$
\begin{aligned}
y_{ij} \sim N(\mu_{ij}, \sigma^2), \qquad & i = 1, \dots, 4, \quad j = 1, \dots, 15 \\
\mu_{ij} = \mu_1, \qquad & j = 1, \dots, 15. \\
\mu_{2j} = \mu_1 + \alpha_2, \qquad & j = 1, \dots, 15. \\
\mu_{3j} = \mu_1 + \alpha_3, \qquad & j = 1, \dots, 15. \\
\mu_{4j} = \mu_1 + \alpha_4, \qquad & j = 1, \dots, 15.
\end{aligned}
$$

In this context, μ_i denotes the mean satisfaction score across various airlines. The model represents airlines B, C, and D relative to airline A. For instance, the parameter α_4 can be interpreted as the disparity between airline A and D. A positive α_4 implies that, on average, customers exhibit higher satisfaction with airline D compared to airline A.

Fitting model in the frequentist framework

The parameters μ_1, α_2, α_3, and α_4 were estimated through fitting the data using a frequentist approach. The estimated values were found to be: $\mu_1 = 4.33$, $\alpha_2 = 1.33$, $\alpha_3 = 0.13$, and $\alpha_4 = 2.0$. Examining these values, it seems that airline D exhibits significantly higher satisfaction ratings compared to airline A, whereas the distinction between airline B and C in terms of satisfaction score relative to airline A is less clear. Nevertheless, hypothesis tests were carried out at a significance level of 5% to assess whether there were differences in mean satisfaction scores among the four airlines.

The initial null hypothesis states that $\mu_1 = \mu_2 = \mu_3 = \mu_4$ but as the latter three can be expressed in terms of μ_1,the following null and alternative hypotheses can be tested:

$$H_0 : \ \alpha_2 = \alpha_3 = \alpha_4 = 0$$
$$H_1 : \ Not\ H_0$$

The hypothesis tests validated prior assumptions regarding the four airlines. α_2 yielded a non-significant outcome (p = 0.20), suggesting no definitive distinction between airlines A and B. Similarly, α_3 produced an inconsequential disparity (p = 0.09). Conversely, α_4 yielded a significant difference (p = 0.00135). These findings suggest ample evidence to propose that customer satisfaction scores differ between airlines A and D, but not between airlines A and B, or A and C.

Tukey Honest Significant Difference test

Subsequently, a Tukey Honest Significant Difference (HSD) test was employed to discern variations among the four airlines.

```
# Perform Tukey's HSD test
tukey_result <- TukeyHSD(aov(satisfactionscore ~ airline, data = baye_data))
```

The tukey result further support the claim that satisfaction score for airline D is significantly doubled that of airline A (p=0.0072). Also a significant difference was also revealed by the tukey HSD between airline D and C. Customers score airline D 1.87 higher than airline C (p=0.013).

Comparing satisfaction score between airline D with B and C.

An additional hypothesis was carried out to determine if the satisfaction score for airline D is more than 3 points higher than the average satisfaction score for airline B and C?. The hypothesis is stated below:

$$H_0 : \ \mu_3 + 3 < \mu_4 > \mu_2 + 3$$
$$H_1 : \ Not\ H_0$$

The hypothesis was carried out using t-test as shown below:

```r
# calculate mean satisfaction score for airline B and C
mu_B <- mean(baye_data$satisfactionscore[baye_data$airline == "B"])
mu_C <- mean(baye_data$satisfactionscore[baye_data$airline == "C"])

# compare D and B.
t_test_result1 <- t.test(baye_data$satisfactionscore[baye_data$airline == "D"],
                         mu = mu_B + 3, alternative = "greater")

# compare D and C
t_test_result2 <- t.test(baye_data$satisfactionscore[baye_data$airline == "D"],
                         mu = mu_C + 3, alternative = "greater")
```

In both cases, the p value obtained is greater than 0.05 (D vs B: 0.999;D vs A: 0.9857). Therefore, the null hypothesis is rejected and we can conclude that the difference in the average satisfaction score between airline D and airlines B and C is not more than 3 points.

Second Sub-Task: Bayesian Two-ways Analysis of Variance

Performing inference about Bayesian two-ways Analysis of Variance model using jags.

A Bayesian approach was utilized to conduct two-way analysis of variance, investigating potential differences in mean soil carbon levels resulting from the application of five distinct treatments across three different fields. Below are summaries of the posterior probability density functions. It should be noted that in this context, mu_1_1 represents the baseline mean carbon level after applying treatment one to field one. Additionally, alpha[i] $= \alpha_i$ for $i = 1, \ldots, 3$ signifies the change in carbon level between field 1 and field i, beta[j] $= \beta_j$ for $j = 1, \ldots, 5$ denotes the change in carbon between treatment 1 and treatment j, and τ represents the precision.

Table 7: Summary statistics for parameters, using two-way analysis of variance

	Mean	Median	95% Credible Interval
alpha[1]	0.00	0.00	(0, 0)
alpha[2]	105.55	105.46	(13.63, 201.02)
alpha[3]	109.81	109.88	(18.41, 202.36)
beta[1]	0.00	0.00	(0, 0)
beta[2]	101.24	100.82	(-7.28, 209.93)
beta[3]	111.24	112.27	(3.17, 218.2)
beta[4]	102.99	104.11	(-0.21, 207.27)
beta[5]	112.08	110.71	(6.03, 219)
deviance	69.09	71.96	(-55, 179.09)
mu[1,1]	196.06	207.86	(81.24, 224.85)
tau	57.15	0.15	(0, 668.07)

As anticipated from the definition of the corner constraints ($\alpha_1 = 0$ and $\beta_1 = 0$), the average, median, and 95% credible intervals for $alpha[1]$ and $beta[1]$ are zero. The mean precision, denoted by τ, is notably high, signifying precise parameter estimates and minimal variation in our baseline carbon level. Additionally, it is noteworthy that the median closely aligns with the mean across all parameters except τ.

The model will be analyzed using the result of the parameters above. Comparing the treatments, treatment 5 and treatment 3 represented by $beta[5]$ and $beta[3]$ respectively appeared to be treatments that generates the highest level of carbon when compared to treatment 1. The 95% credible interval obtained for treatment

2, *beta*[2] and treatment 4, *beta*[4] suggest that these treatments may not be significantly different from treatment 1. Meanwhile, considering the fields, field 2 represented by *alpha*[2] and field 3, *alpha*[3] appeared to have significantly varied from field 1 as seen in their credible intervals centered around large numbers.

No conclusion can be made yet as the observations are mainly due to the results of credible intervals. However, more explanation will be provided by the traceplot and posterior densities of the parameters.

Traceplots and Posterior densities of α_i and β_j

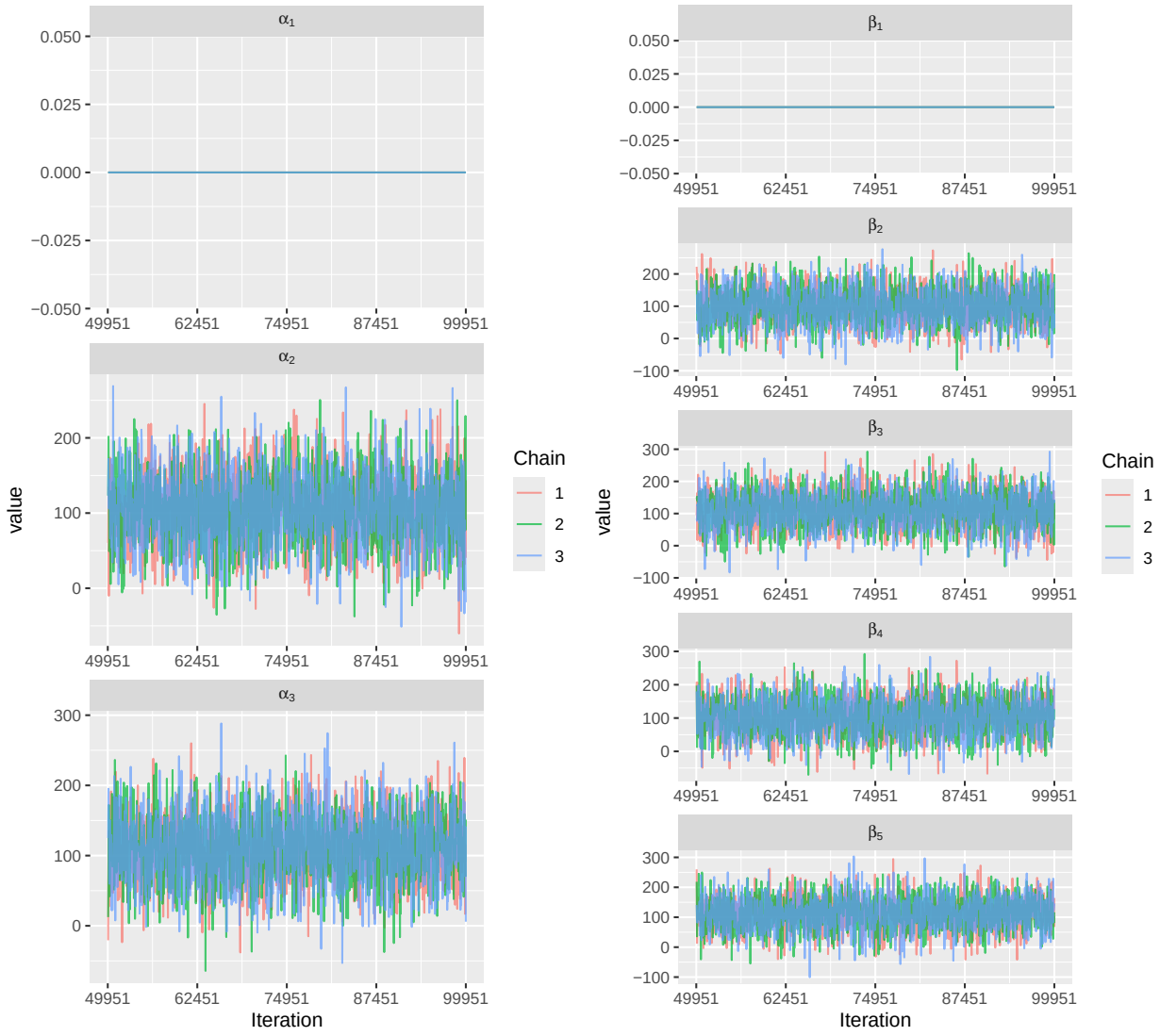

Fig. 13: Traceplots detailing changes in parameters in ergodic phase of sampling.

Figure 13 illustrates the variations in the parameters α_i and β_j as the sampling process progresses into the ergodic phase. It is observed that α_i values exhibit fluctuations regardless of the number of iterations. The distinction in carbon levels between treatments 1 and treatments 2, 3, 4, and 5 is discerned from their respective β_2, β_3, β_4, β_5 trace plots. β_5 and β_3 appeared to be more settled.

Posterior densities

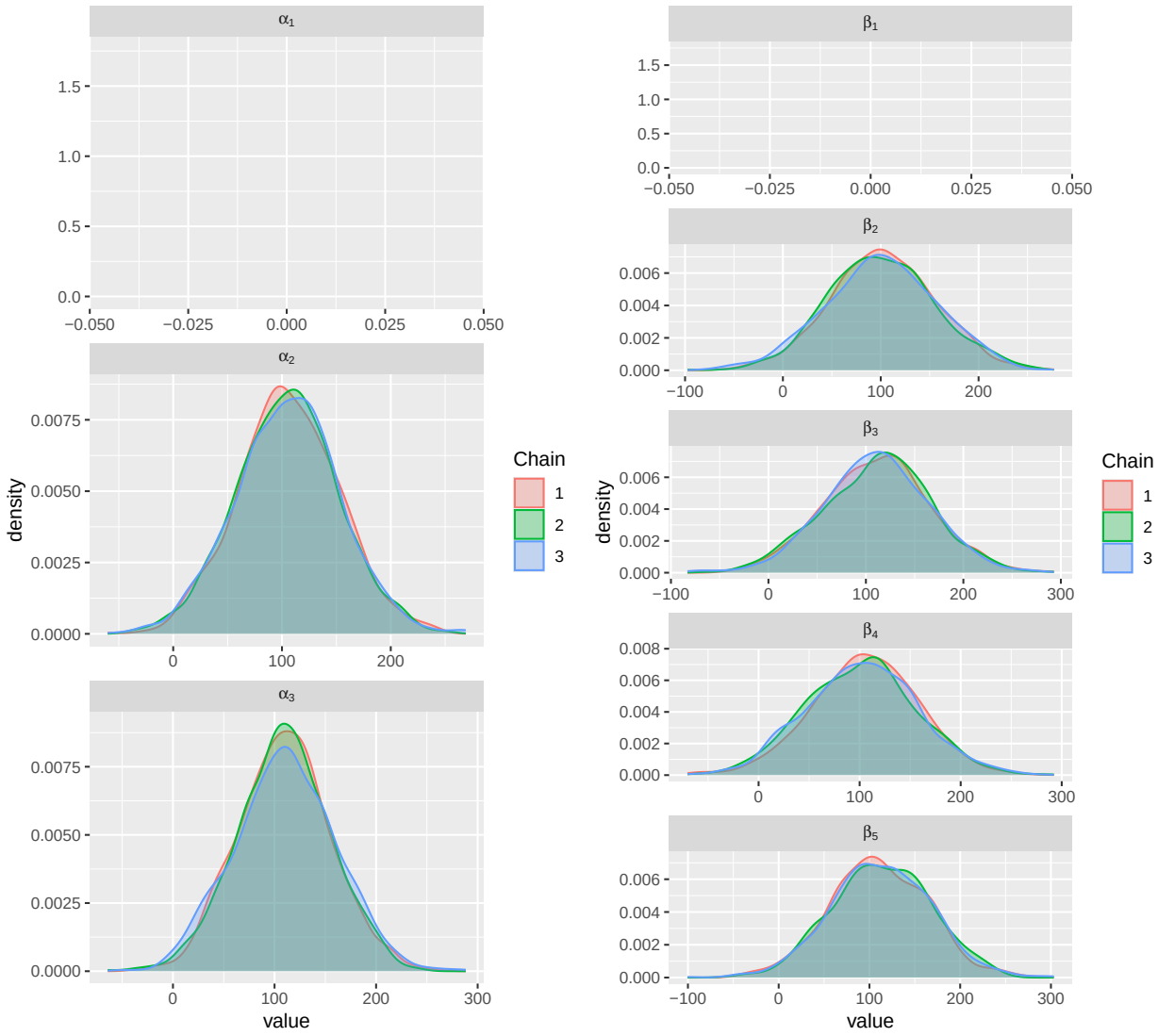

Fig. 14: Posterior density plots highlighting distribution of parameters.

The distributions of the estimated values for variables α_i and β_j are depicted in Figure 14. Previously, it was contended that any treatment applied to fields 1, 2, or 3 would yield similar carbon levels. This assertion finds further support in Figure 14. The plot for β_2 reveals a dense concentration of values skewed towards the left, suggesting that most sampled values of β_2 are below 100. This indicates that the disparity in carbon levels between treatments 1 and 2 is minimal. Additionally, the distribution of β_5 and β_3 values appears notably distinct from other β_j values, providing further supporting evidence. However, not all previous arguments are as unequivocally reinforced. The distribution of β_4, as depicted in Figure 12, does not appear negligible, given the central skewness of its distribution. Moreover, the two α_2 and α_3 representing fields 2 and 3 respectively exhibit a significant difference in their distributions compared to the baseline field 1.

Graphical representation of 95% credible intervals

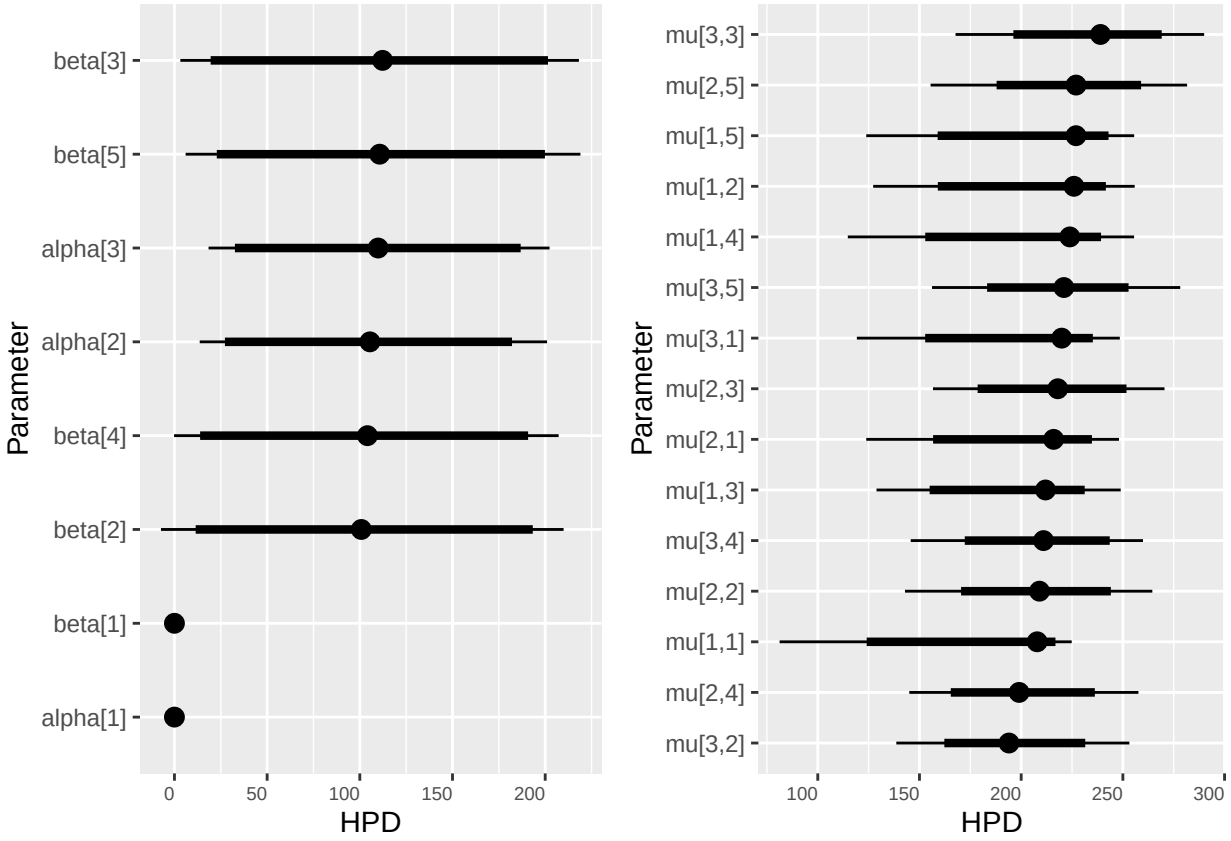

Fig. 15: 95% credible intervals for all parameters, assuming fields and treatments differ in carbon level.

Figure 15 summarizes key observations regarding the varying effectiveness of the five treatments applied across the three fields. Specifically, the left plot illustrates that treatments 5 and 3 induce the most significant changes in carbon levels compared to treatment 1. While treatment 2 appears to have a slightly greater impact than treatment 4, the latter still influences carbon levels, albeit to a lesser extent compared to other treatments.

Examining the right plot in Figure 13 provides a detailed comparison of treatment mixtures relative to the baseline. The baseline treatment and field (mu[1,1]) exhibit the highest variation in carbon levels. However, combinations involving field 2 and treatment 4 (mu[2,4]) and field 3 and treatment 2 demonstrate lower mean carbon values than the baseline, suggesting significant differences. Moreover, mu[3, 3] and mu[2,5] plots affirm the effectiveness of treatments 3 and 5 in carbon sequestration compared to others. Conversely, mu[2, 4] and mu[3, 2] plots at the bottom indicate that treatments 2 and 4 yield negligible differences in carbon levels compared to treatment 1.

In summary, treatments 3 and 5 consistently lead to notable alterations in carbon levels, while the impact of fields varies.

Comparing treatment 4 and remaining treatments

Bayesian inference was conducted to find the differences between β_4 and β_j for $j = 1, 2, 3, 5$ so as to determine whether treatment 4 yield a higher level of carbon sequestration. These are denoted below using the variables $beta\_diff[j]$, to represent $\beta_4 - \beta_j$:

Table 8: Summary statistics for differences in beta, using two-way analysis of variance

	Mean	Median	95% Credible Interval
beta_diff[1]	102.99	104.11	(-0.21, 207.27)
beta_diff[2]	1.75	3.55	(-144.29, 146.37)
beta_diff[3]	-8.25	-7.26	(-152.4, 137.39)
beta_diff[5]	-9.09	-8.71	(-148.35, 136.54)
deviance	69.09	71.96	(-55, 179.09)

Table 8 summarizes the statistics for differences in β_4 representing treatment 4 and others. As observed from the table, the difference between β_4 and β_1 is 102.99 carbon units. β_1 is the baseline and therefore this value does not hold a claim. β_4 is 1.75 higher in carbon level sequestrated than β_2. This shows that treatment 4 outperforms treatment 2. In contrast, treatment 3 and 5 outperform treatment 4 as seen in the negative value of the mean difference. This further supports the evidence we had earlier that treatment 3 and 5 appeared to be the most significant treatments when sequestrating carbon.

More evidence will be provided using the catterpillar plot and posterior densities of the observed parameters.

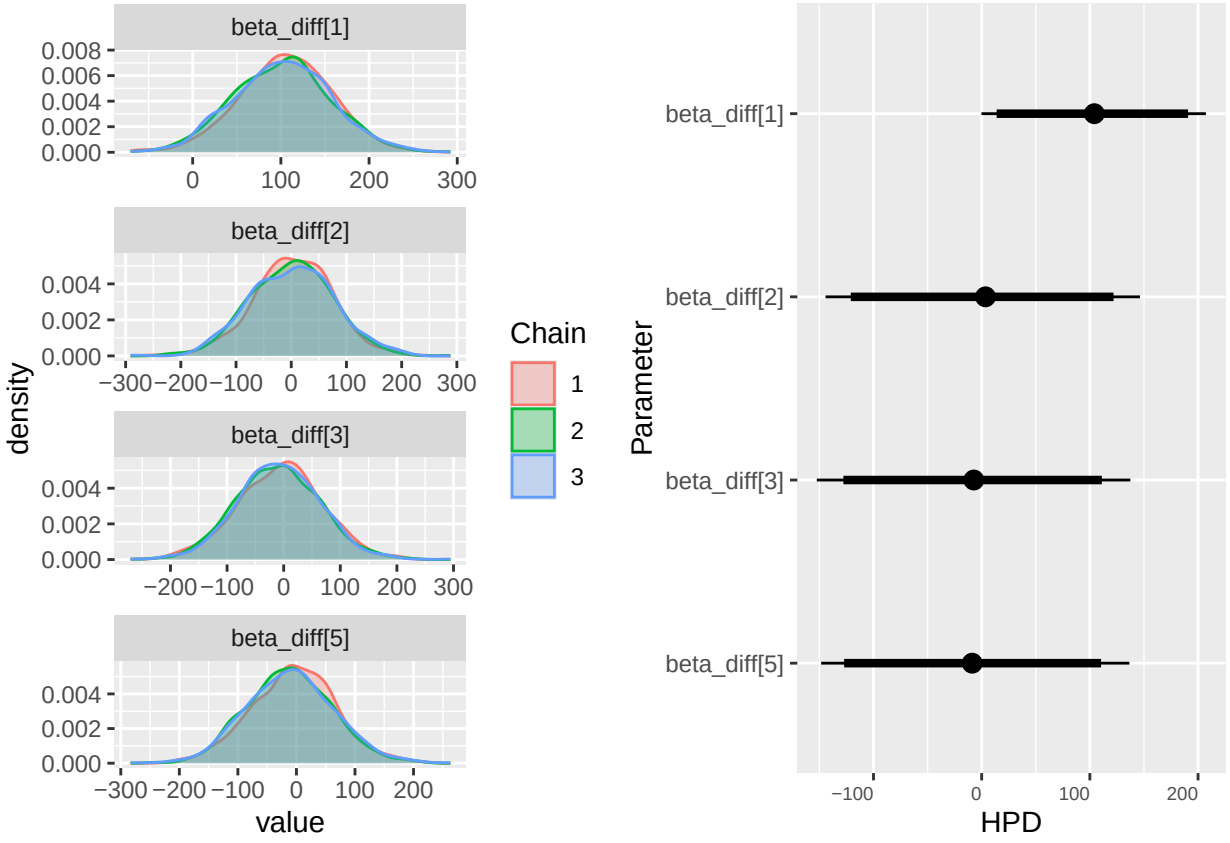

Fig. 16: Caterpillar plot and posterior density plot displaying differences between treatment of interest (j = 4) and remaining treatments. 85% credible intervals for thick lines, 95% for thin lines

The two plots obtained in figure 16 provide similar observations with what we explained earlier using the table. The median value represented by the thick circle in the caterpillar plot, for $beta\_diff[1]$ and $beta\_diff[2]$ are above zero, which shows that treatment 4 is a better option compared to treatments 1 and 2. Meanwhile, the farmer was wrong with treatment 3 and 5 as these treatments are shown to outperform treatment 4.

Third Sub-Task: Simpler Bayesian model

One-way analysis of variance

In a scenario where all fields exhibit identical carbon levels, a one-way analysis of variance was performed within the Bayesian framework. This analysis aimed to ascertain whether there were variations in the average carbon levels when subjecting the fields to five different treatments. Recorded below are the mean values of parameters, denoted as β_j for $j = 1, \ldots, 5$, the baseline carbon level under treatment 1, represented as μ, and the precision, denoted as τ. Additionally, the median and the starting and ending points of the 95% credible intervals are provided.

Table 9: Summary statistics of parameters, using one-way analysis of variance

	Mean	Median	95% Credible Interval
beta[1]	0.00	0.00	(0, 0)
beta[2]	104.66	103.87	(-39.22, 246)
beta[3]	112.00	110.49	(-24.27, 249)
beta[4]	104.94	104.89	(-34.45, 238.72)
beta[5]	112.18	112.38	(-28.21, 251.35)
deviance	117.53	116.64	(111.31, 128.95)
mu[1]	213.49	213.57	(198.14, 227.46)
tau	0.01	0.01	(0, 0.02)

Posterior densities and 95% credible intervals

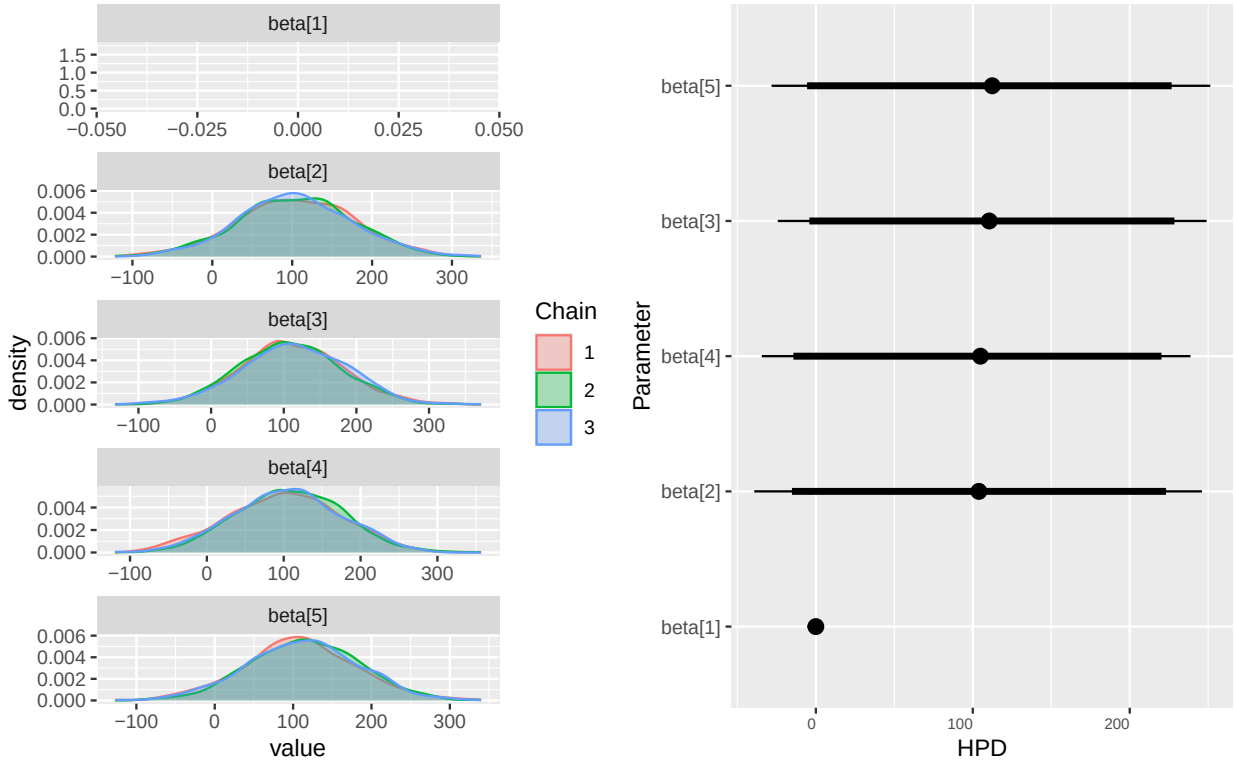

Fig. 17: Caterpillar plot and posterior plot showing the change in differences between the baseline solution (j = 1) and remaining solutions (j = 2,3,4,5), assuming fields have the same carbon level.

The results presented in table 9 and the plots visualized in figure 17 revealed so much similarities between two way ANOVA and the one way ANOVA. Similar to what was discussed earlier, treatment 3 and 5 also appeared to be the most effective in sequestration of carbon. Treatment 2 and treatment 4 hold no significant comparison just as seen in two way ANOVA model.

Comparing one- and two-way analysis of variance Bayesian models

Deviance Information Criterion (DIC) and size of 95% Highest Posterior Density (HPD) credible intervals for parameters allows comparison of Bayesian models.

Table 10: Comparison of summary statistics of parameters present in both one- and two-way analysis of variance models

	Mean (Two way)	Mean (One way)	Median (Two way)	Median (One way)	95% Credible Interval (Two way)	95% Credible Interval (One way)
beta[1]	0.00	0.00	0.00	0.00	(0, 0)	(0, 0)
beta[2]	101.24	104.66	100.82	103.87	(-7.28, 209.93)	(-39.22, 246)
beta[3]	111.24	112.00	112.27	110.49	(3.17, 218.2)	(-24.27, 249)
beta[4]	102.99	104.94	104.11	104.89	(-0.21, 207.27)	(-34.45, 238.72)
beta[5]	112.08	112.18	110.71	112.38	(6.03, 219)	(-28.21, 251.35)
deviance	69.09	117.53	71.96	116.64	(-55, 179.09)	(111.31, 128.95)
mu[1,1]	196.06	NA	207.86	NA	(81.24, 224.85)	NA
mu[1]	NA	213.49	NA	213.57	NA	(198.14, 227.46)
tau	57.15	0.01	0.15	0.01	(0, 668.07)	(0, 0.02)

DIC Values: Two ways = 2479.3; One way = 127.9

Comparing the mean and median of the two models as seen in table 10, we can observed consistently that the mean and median values of the one way ANOVA are larger than that of the two way ANOVA. More appropriately, DIC offers a more acceptable comparison between the two models. A lower Deviance Information Criterion (DIC) generally indicates a superior model, with a DIC of 127.9 being considerably lower than 2479.3. This further implies that the one-way analysis of variance (ANOVA) model is a more appropriate fit. In terms of the precision τ, observed for the two-way ANOVA, it exhibits higher mean and median values. Nevertheless, the 95% credible interval for this precision in the two-way ANOVA model is notably wide, suggesting it may not be the best-fitting model.

Overall, most of the metrics considered favor the one-way ANOVA model. Hence, it is preferable to model the types of treatments used in carbon sequestration solely under the one-way ANOVA Bayesian framework.