

# FMD Project

Rebecca

2025-04-21

```
#load the neccessary libraries
library(readxl)

## Warning: package 'readxl' was built under R version 4.4.1

library(dplyr)

##
## Attaching package: 'dplyr'

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

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

library(tidyverse)

## -- Attaching core tidyverse packages ----- tidyverse 2.0.0 --
## vforcats    1.0.0    vreadr      2.1.5
## vggplot2    3.5.1    vstringr   1.5.1
## vlubridate  1.9.3    vtibble     3.2.1
## vpurrr      1.0.2    vtidyrm   1.3.1

## -- 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

library(stats)
library(ggplot2)
library(caret)

## Warning: package 'caret' was built under R version 4.4.3
```

```

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

library(gridExtra)

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

library(rpart.plot)

## Warning: package 'rpart.plot' was built under R version 4.4.3

## Loading required package: rpart

library(e1071)

## Warning: package 'e1071' was built under R version 4.4.3

library(mice)

##
## Attaching package: 'mice'
##
## The following object is masked from 'package:stats':
##     filter
##
## The following objects are masked from 'package:base':
##     cbind, rbind

library(lme4)

## Warning: package 'lme4' was built under R version 4.4.3

## Loading required package: Matrix
##
## Attaching package: 'Matrix'
##
## The following objects are masked from 'package:tidyverse':
##     expand, pack, unpack

```

```

library(DMwR2)

## Warning: package 'DMwR2' was built under R version 4.4.3

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

library(pROC)

## Warning: package 'pROC' was built under R version 4.4.3

## Type 'citation("pROC")' for a citation.
##
## Attaching package: 'pROC'
##
## The following objects are masked from 'package:stats':
##
##     cov, smooth, var

library(ggcorrplot)

## Warning: package 'ggcorrplot' was built under R version 4.4.3

library(randomForest)

## Warning: package 'randomForest' was built under R version 4.4.3

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

#load the dataset
my_data<-read_excel("C:/Users/HP/Desktop/FMD/Raw Data.xlsx")

```

```

## New names:
## * `` -> '...2'
## * `` -> '...3'
## * `` -> '...4'
## * `` -> '...5'
## * `` -> '...6'
## * `` -> '...7'
## * `` -> '...8'
## * `` -> '...9'
## * `` -> '...10'
## * `` -> '...11'
## * `` -> '...12'
## * `` -> '...13'
## * `` -> '...14'
## * `` -> '...15'
## * `` -> '...16'
## * `` -> '...17'
## * `` -> '...18'
## * `` -> '...19'

glimpse(my_data)

## Rows: 267
## Columns: 19
## $ 'Row data collected from field to investigate seroprevalence of bovine foot and mouth disease and a
## $ ...2
## $ ...3
## $ ...4
## $ ...5
## $ ...6
## $ ...7
## $ ...8
## $ ...9
## $ ...10
## $ ...11
## $ ...12
## $ ...13
## $ ...14
## $ ...15
## $ ...16
## $ ...17
## $ ...18
## $ ...19

#assign proper column names for the first row
colnames(my_data)<-as.character(my_data[1,])
my_data<-my_data[-1,]
my_data

## # A tibble: 266 x 19
##   Animal_ID Sample_Code Lab_Code Region Zone Districts Kebeles Species Breed
##   <chr>      <chr>      <chr>    <chr>  <chr>    <chr>    <chr>    <chr>
##   1 1          S-1       1260     Oromia East_W~ Diga      Arjo_G~ Bovine Local

```

```

## 2 2      S-2      1261    Oromia East_W~ Diga      Arjo_G~ Bovine Local
## 3 3      S-3      1262    Oromia East_W~ Diga      Arjo_G~ Bovine Local
## 4 4      S-4      1263    Oromia East_W~ Diga      Arjo_G~ Bovine Local
## 5 5      S-5      1264    Oromia East_W~ Diga      Efa     Bovine Cross
## 6 6      S-6      1265    Oromia East_W~ Diga      Arjo_G~ Bovine Local
## 7 7      S-7      1266    Oromia East_W~ Diga      Arjo_G~ Bovine Cross
## 8 8      S-8      1267    Oromia East_W~ Diga      Arjo_G~ Bovine Local
## 9 9      S-9      1268    Oromia East_W~ Diga      Arjo_G~ Bovine Local
## 10 10   S-10     1269    Oromia East_W~ Diga      Arjo_G~ Bovine Cross
## # i 256 more rows
## # i 10 more variables: Sex <chr>, Age <chr>, Body_Condition <chr>,
## # Physiology <chr>, Sample_Type <chr>, Longitude <chr>, Latitude <chr>,
## # Altitude <chr>, Agro_Climate <chr>, FMD_Status <chr>

#DATA PREPROCESSING
# 1. handling missing values
missing_counts<-my_data%>% summarise (across(everything(), ~sum(is.na(.))))
missing_counts

## # A tibble: 1 x 19
##   Animal_ID Sample_Code Lab_Code Region Zone Districts Kebeles Species Breed
##       <int>        <int>     <int>  <int> <int>     <int>    <int> <int>
## 1          0           0       0     0     0       0       0     0     0
## # i 10 more variables: Sex <int>, Age <int>, Body_Condition <int>,
## # Physiology <int>, Sample_Type <int>, Longitude <int>, Latitude <int>,
## # Altitude <int>, Agro_Climate <int>, FMD_Status <int>

# 2. handling duplicate values
sum(duplicated(my_data))

## [1] 0

#proportion of the FMD_Status output
prop.table(table(my_data$FMD_Status))

## 
## Negative Positive
## 0.7969925 0.2030075

# Load required package
library(ROSE)

## Warning: package 'ROSE' was built under R version 4.4.3

## Loaded ROSE 0.0-4

# Oversample the minority class
balanced_data <- ovun.sample(FMD_Status ~ ., data = my_data, method = "over", N = max(table(my_data$FMD_Status)))

# Check the new class distribution
table(balanced_data$FMD_Status)

```

```

## Negative Positive
##      212      212

# Undersample the majority class
balanced_data <- ovun.sample(FMD_Status ~ ., data = my_data, method = "under", N = min(table(my_data$FMD_Status)))

# Check the new class distribution
table(balanced_data$FMD_Status)

## Negative Positive
##      54      54

# Combine over and under sampling
balanced_data <- ovun.sample(FMD_Status ~ ., data = my_data, method = "both", p = 0.5, seed = 123, N = min(table(my_data$FMD_Status)))

# Check class balance
table(balanced_data$FMD_Status)

## Negative Positive
##      134      132

# Combine over and under sampling
balanced_data <- ovun.sample(FMD_Status ~ ., data = my_data, method = "both", p = 0.5, seed = 123, N = min(table(my_data$FMD_Status)))

# Check class balance
table(balanced_data$FMD_Status)

## Negative Positive
##      134      132

#Balancing the response variable
##checking
str(balanced_data)

## 'data.frame': 266 obs. of 19 variables:
##   $ Animal_ID    : chr  "29" "47" "238" "196" ...
##   $ Sample_Code  : chr  "S-29" "S-47" "S-238" "S-196" ...
##   $ Lab_Code     : chr  "1288" "1306" "1497" "1455" ...
##   $ Region       : chr  "Oromia" "Oromia" "Oromia" "Oromia" ...
##   $ Zone         : chr  "East_Wollega" "East_Wollega" "East_Wollega" "East_Wollega" ...
##   $ Districts    : chr  "Diga" "Diga" "Nekemte" "Guto_Gida" ...
##   $ Kebeles      : chr  "Arjo_Gudatu" "Arjo_Gudatu" "Bake_Jema" "Uke" ...
##   $ Species       : chr  "Bovine" "Bovine" "Bovine" "Bovine" ...
##   $ Breed         : chr  "Local" "Local" "Local" "Local" ...
##   $ Sex           : chr  "Female" "Female" "Female" "Female" ...
##   $ Age           : chr  "Young" "Old" "Old" "Adult" ...
##   $ Body_Condition: chr  "Poor" "Poor" "Poor" "Poor" ...

```

```

## $ Physiology      : chr "Pregnant" "Pregnant" "Pregnant" "Heifer" ...
## $ Sample_Type     : chr "Serum" "Serum" "Serum" "Serum" ...
## $ Longitude       : chr "197468" "197468" "228630" "228243" ...
## $ Latitude        : chr "1000164" "1000164" "1004934" "1036664" ...
## $ Altitude         : chr "1327" "1327" "2076" "1374" ...
## $ Agro_Climate    : chr "Lowland" "Lowland" "Midland" "Lowland" ...
## $ FMD_Status       : chr "Negative" "Negative" "Negative" "Negative" ...

```

```
summary(balanced_data)
```

	Animal_ID	Sample_Code	Lab_Code	Region
## Length:266	Length:266	Length:266	Length:266	Length:266
## Class :character	Class :character	Class :character	Class :character	Class :character
## Mode :character	Mode :character	Mode :character	Mode :character	Mode :character
## Zone	Districts	Kebeles	Species	
## Length:266	Length:266	Length:266	Length:266	Length:266
## Class :character	Class :character	Class :character	Class :character	Class :character
## Mode :character	Mode :character	Mode :character	Mode :character	Mode :character
## Breed	Sex	Age	Body_Condition	
## Length:266	Length:266	Length:266	Length:266	Length:266
## Class :character	Class :character	Class :character	Class :character	Class :character
## Mode :character	Mode :character	Mode :character	Mode :character	Mode :character
## Physiology	Sample_Type	Longitude	Latitude	
## Length:266	Length:266	Length:266	Length:266	Length:266
## Class :character	Class :character	Class :character	Class :character	Class :character
## Mode :character	Mode :character	Mode :character	Mode :character	Mode :character
## Altitude	Agro_Climate	FMD_Status		
## Length:266	Length:266	Length:266		
## Class :character	Class :character	Class :character		
## Mode :character	Mode :character	Mode :character		

```

# Load required packages
library(dplyr)
library(summarytools)

```

```
## Warning: package 'summarytools' was built under R version 4.4.3
```

```

##
## Attaching package: 'summarytools'
##
## The following object is masked from 'package:tibble':
##
##     view

```

```

# Generate a summary table
dfSummary(balanced_data, plain.ascii = FALSE, style = "grid", valid.col = FALSE)

```

```
## text graphs are displayed; set 'tmp.img.dir' parameter to activate png graphs
```

```

## #### Data Frame Summary
## ##### balanced_data

```



```

## | | | 3\. Bake_Jema\ | 7 ( 2.6%)\ | \ |
## | | | 4\. Calalak\ | 24 ( 9.0%)\ | I \
## | | | 5\. Efa\ | 17 ( 6.4%)\ | I \
## | | | 6\. Fayinera\ | 26 ( 9.8%)\ | I \
## | | | 7\. Jirata\ | 25 ( 9.4%)\ | I \
## | | | 8\. Uke | 43 (16.2%) | III |
## +---+-----+-----+-----+-----+
## | 8 | Species\ | 1\. Bovine | 266 (100.0%) | IIIIIIIIIIIIIIIIIIII | 0\ |
## | | [character] | | | | | (0.0%) |
## +---+-----+-----+-----+-----+
## | 9 | Breed\ | 1\. Cross\ | 89 (33.5%)\ | IIFFFFF \ | 0\ |
## | | [character] | 2\. Local | 177 (66.5%) | IIFFFFFFF | (0.0%) |
## +---+-----+-----+-----+-----+
## | 10 | Sex\ | 1\. Female\ | 172 (64.7%)\ | IIFFFFFFF \ | 0\ |
## | | [character] | 2\. Male | 94 (35.3%) | IIFFFFF | (0.0%) |
## +---+-----+-----+-----+-----+
## | 11 | Age\ | 1\. Adult\ | 72 (27.1%)\ | IIFFF \ | 0\ |
## | | [character] | 2\. Old\ | 132 (49.6%)\ | IIFFFFFF \ | (0.0%) |
## | | | 3\. Young | 62 (23.3%) | IIFFF | |
## +---+-----+-----+-----+-----+
## | 12 | Body_Condition\ | 1\. Good\ | 53 (19.9%)\ | III \ | 0\ |
## | | [character] | 2\. Medium\ | 19 ( 7.1%)\ | I \ | (0.0%) |
## | | | 3\. Poor | 194 (72.9%) | IIFFFFFFF | |
## +---+-----+-----+-----+-----+
## | 13 | Physiology\ | 1\. Bull\ | 43 (16.2%)\ | III \ | 0\ |
## | | [character] | 2\. Dry\ | 47 (17.7%)\ | III \ | (0.0%) |
## | | | 3\. Heifer\ | 59 (22.2%)\ | IIII \ |
## | | | 4\. Lactating\ | 45 (16.9%)\ | III \ |
## | | | 5\. Ox\ | 51 (19.2%)\ | III \ |
## | | | 6\. Pregnant | 21 ( 7.9%) | I | |
## +---+-----+-----+-----+-----+
## | 14 | Sample_Type\ | 1\. Serum | 266 (100.0%) | IIIIIIIIIIIIIIIIIIII | 0\ |
## | | [character] | | | | | (0.0%) |
## +---+-----+-----+-----+-----+
## | 15 | Longitude\ | 1\. 197468\ | 88 (33.1%)\ | IIFFFFF \ | 0\ |
## | | [character] | 2\. 220247\ | 17 ( 6.4%)\ | I \ | (0.0%) |
## | | | 3\. 223825\ | 25 ( 9.4%)\ | I \ |
## | | | 4\. 226389\ | 26 ( 9.8%)\ | I \ |
## | | | 5\. 228243\ | 43 (16.2%)\ | III \ |
## | | | 6\. 228630\ | 5 ( 1.9%)\ | \ |
## | | | 7\. 229820\ | 38 (14.3%)\ | II \ |
## | | | 8\. 231500 | 24 ( 9.0%) | I | |
## +---+-----+-----+-----+-----+
## | 16 | Latitude\ | 1\. 1000164\ | 88 (33.1%)\ | IIFFFFF \ | 0\ |
## | | [character] | 2\. 1004455\ | 26 ( 9.8%)\ | I \ | (0.0%) |
## | | | 3\. 1004934\ | 5 ( 1.9%)\ | \ |
## | | | 4\. 1005370\ | 24 ( 9.0%)\ | I \ |
## | | | 5\. 1036664\ | 43 (16.2%)\ | III \ |
## | | | 6\. 997045\ | 38 (14.3%)\ | II \ |
## | | | 7\. 998279\ | 17 ( 6.4%)\ | I \ |
## | | | 8\. 999437 | 25 ( 9.4%) | I | |
## +---+-----+-----+-----+-----+
## | 17 | Altitude\ | 1\. 1327\ | 87 (32.7%)\ | IIFFFFF \ | 0\ |
## | | [character] | 2\. 1374\ | 43 (16.2%)\ | III \ | (0.0%) |

```



```

        values_from = c(Count, Percent),
        names_sep = "_") %>%
mutate(Variable = varname) %>%
rename(Level = !!sym(varname)) %>%
select(Variable, Level,
       Count_Negative, Percent_Negative,
       Count_Positive, Percent_Positive,
       Total)
}

# Apply the function to each variable and bind all together
summary_list <- lapply(cat_vars, summarize_by_FMD)
final_summary <- bind_rows(summary_list)

# Print first few rows
print(final_summary, n = 100)

```

```

## # A tibble: 30 x 7
## # Groups:   Level [30]
##   Variable     Level   Count_Negative Percent_Negative Count_Positive
##   <chr>       <chr>           <int>            <dbl>           <int>
## 1 Breed        Cross            42              47.2            47
## 2 Breed        Local            92              52              85
## 3 Districts    Diga             48              37.5            80
## 4 Districts    Guto_Gida       62              57.9            45
## 5 Districts    Nekemte          24              77.4             7
## 6 Kebeles      Abdeta           19              50              19
## 7 Kebeles      Arjo_Gudatu    23              26.7            63
## 8 Kebeles      Bake_Jema       5               71.4             2
## 9 Kebeles      Calalak          19              79.2             5
## 10 Kebeles     Efa              10              58.8             7
## 11 Kebeles     Fayinera         19              73.1             7
## 12 Kebeles     Jirata            15              60              10
## 13 Kebeles     Uke              24              55.8            19
## 14 Sex          Female           69              40.1            103
## 15 Sex          Male              65              69.1            29
## 16 Age          Adult            35              48.6            37
## 17 Age          Old              49              37.1            83
## 18 Age          Young            50              80.6            12
## 19 Body_Condition Good            45              84.9             8
## 20 Body_Condition Medium          8               42.1            11
## 21 Body_Condition Poor            81              41.8            113
## 22 Physiology    Bull             34              79.1             9
## 23 Physiology    Dry              20              42.6            27
## 24 Physiology    Heifer           29              49.2            30
## 25 Physiology    Lactating        11              24.4            34
## 26 Physiology    Ox              31              60.8            20
## 27 Physiology    Pregnant         9               42.9            12
## 28 Agro_Climate  Lowland          47              36.4            82
## 29 Agro_Climate  midland          0               0              3
## 30 Agro_Climate  Midland          87              64.9            47
## # i 2 more variables: Percent_Positive <dbl>, Total <int>

```

```

library(knitr)

#encoding categorical variables
# Convert all character columns to factors and ensure result is a data frame
convert_all_columns_to_factors <- function(df) {
  df[] <- lapply(df, function(x) if (is.character(x)) as.factor(x) else x)
  return(df)
}

# Apply it and assign the result
balanced_data <- convert_all_columns_to_factors(balanced_data)

#FEATURE ENGINEERING
#exploring data
str(balanced_data)

```

```

## 'data.frame': 266 obs. of 19 variables:
## $ Animal_ID : Factor w/ 148 levels "100","101","105",...: 103 117 84 58 32 13 136 137 96 25 ...
## $ Sample_Code : Factor w/ 148 levels "S-100","S-101",...: 103 117 84 58 32 13 136 137 96 25 ...
## $ Lab_Code   : Factor w/ 148 levels "1262","1264",...: 14 27 134 109 85 5 44 45 145 78 ...
## $ Region    : Factor w/ 1 level "Oromia": 1 1 1 1 1 1 1 1 1 ...
## $ Zone       : Factor w/ 1 level "East_Wollega": 1 1 1 1 1 1 1 1 1 ...
## $ Districts  : Factor w/ 3 levels "Diga","Guto_Gida",...: 1 1 3 2 2 1 1 1 2 2 ...
## $ Kebeles    : Factor w/ 8 levels "Abdeta","Arjo_Gudatu",...: 2 2 3 8 6 2 7 7 8 1 ...
## $ Species    : Factor w/ 1 level "Bovine": 1 1 1 1 1 1 1 1 1 ...
## $ Breed      : Factor w/ 2 levels "Cross","Local": 2 2 2 2 1 2 1 2 1 1 ...
## $ Sex        : Factor w/ 2 levels "Female","Male": 1 1 1 1 1 1 2 2 1 1 ...
## $ Age        : Factor w/ 3 levels "Adult","Old",...: 3 2 2 1 1 3 3 2 2 1 ...
## $ Body_Condition: Factor w/ 3 levels "Good","Medium",...: 3 3 3 3 3 3 3 3 1 1 ...
## $ Physiology : Factor w/ 6 levels "Bull","Dry","Heifer",...: 6 6 6 3 3 4 1 1 3 3 ...
## $ Sample_Type : Factor w/ 1 level "Serum": 1 1 1 1 1 1 1 1 1 ...
## $ Longitude  : Factor w/ 8 levels "197468","220247",...: 1 1 6 5 4 1 3 3 5 7 ...
## $ Latitude   : Factor w/ 8 levels "1000164","1004455",...: 1 1 3 5 2 1 8 8 5 6 ...
## $ Altitude   : Factor w/ 8 levels "1327","1374",...: 1 1 4 2 5 1 8 8 2 3 ...
## $ Agro_Climate: Factor w/ 3 levels "Lowland","midland",...: 1 1 3 1 3 1 3 3 1 3 ...
## $ FMD_Status  : Factor w/ 2 levels "Negative","Positive": 1 1 1 1 1 1 1 1 1 ...

```

```
summary(balanced_data)
```

	Animal_ID	Sample_Code	Lab_Code	Region	Zone
## 125	: 7	S-125	: 7	1384 : 7	Oromia:266 East_Wollega:266
## 206	: 5	S-206	: 5	1301 : 5	
## 223	: 5	S-223	: 5	1465 : 5	
## 225	: 5	S-225	: 5	1482 : 5	
## 42	: 5	S-42	: 5	1484 : 5	
## 112	: 4	S-112	: 4	1296 : 4	
## (Other):235	(Other):235	(Other):235	(Other):235		
	Districts	Kebeles	Species	Breed	Sex
## Diga	:128	Arjo_Gudatu:86	Bovine:266	Cross: 89	Female:172
## Guto_Gida:107	Uke	:43		Local:177	Male : 94
## Nekemte	: 31	Abdeta	:38		

```

##          Fayinera :26
##          Jirata   :25
##          Calalak  :24
##          (Other)  :24
##      Age     Body_Condition Physiology Sample_Type Longitude
##  Adult: 72    Good   : 53     Bull    :43 Serum:266  197468 :88
##  Old   :132   Medium  : 19     Dry     :47           228243 :43
##  Young: 62   Poor    :194    Heifer  :59           229820 :38
##                      Lactating:45           226389 :26
##                      Ox      :51           223825 :25
##                      Pregnant:21           231500 :24
##                      (Other):22
##      Latitude Altitude Agro_Climate FMD_Status
##  1000164:88   1327    :87 Lowland:129 Negative:134
##  1036664:43   1374    :43 midland: 3 Positive:132
##  997045 :38   2027    :38 Midland:134
##  1004455:26   2090    :26
##  999437 :25   2233    :25
##  1005370:24   2122    :24
##  (Other):22   (Other):23

```

```
head(balanced_data)
```

```

##   Animal_ID Sample_Code Lab_Code Region          Zone Districts Kebeles
## 1         29       S-29    1288 Oromia East_Wollega      Diga Arjo_Gudatu
## 2         47       S-47    1306 Oromia East_Wollega      Diga Arjo_Gudatu
## 3        238       S-238   1497 Oromia East_Wollega Nekemte Bake_Jema
## 4        196       S-196   1455 Oromia East_Wollega Guto_Gida     Uke
## 5        148       S-148   1407 Oromia East_Wollega Guto_Gida  Fayinera
## 6         12       S-12    1271 Oromia East_Wollega      Diga Arjo_Gudatu
##   Species Breed Sex Age Body_Condition Physiology Sample_Type Longitude
## 1  Bovine Local Female Young       Poor Pregnant Serum 197468
## 2  Bovine Local Female Old        Poor Pregnant Serum 197468
## 3  Bovine Local Female Old        Poor Pregnant Serum 228630
## 4  Bovine Local Female Adult     Poor Heifer Serum 228243
## 5  Bovine Cross Female Adult     Poor Heifer Serum 226389
## 6  Bovine Local Female Young     Poor Lactating Serum 197468
##   Latitude Altitude Agro_Climate FMD_Status
## 1 1000164     1327 Lowland Negative
## 2 1000164     1327 Lowland Negative
## 3 1004934     2076 Midland Negative
## 4 1036664     1374 Lowland Negative
## 5 1004455     2090 Midland Negative
## 6 1000164     1327 Lowland Negative

```

```
#dropping variables with one level
```

```
balanced_data <- balanced_data[, sapply(balanced_data, function(x) !is.factor(x) || length(unique(x)) > 1)]
#dropping animal id
balanced_data$Animal_ID<-NULL
```

```
#summary of FMD status
```

```
FMD_Summary<-balanced_data %>%
  group_by(FMD_Status) %>%
```

```

summarise(
  count=n(),
  percentage=round((n()/nrow(balanced_data))*100,2)
)

print(FMD_Summary)

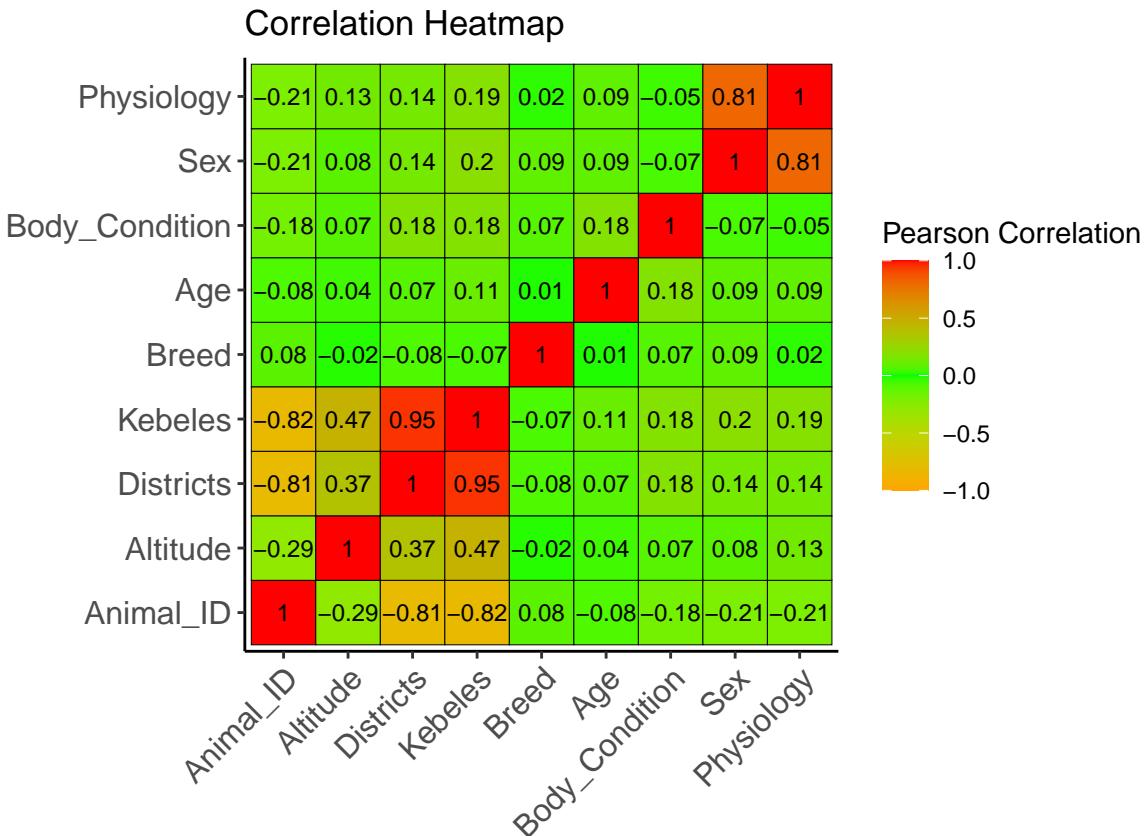
## # A tibble: 2 x 3
##   FMD_Status count percentage
##   <fct>     <int>     <dbl>
## 1 Negative    134      50.4
## 2 Positive    132      49.6

#set a random seed for reproducibility
set.seed(123)

#Importing coded data
library(readxl)
coded_data <- read_excel("C:/Users/HP/Desktop/FINALProject./coded.data.xlsx", sheet = "Coded")
View(coded_data)

#Correlation Analysis
correlat <- cor(coded_data[, setdiff(names(coded_data), 'FMD_Status')])
#Visualize your Results
ggcorrplot(correlat,title = "Correlation Heatmap",lab_col = "black",
           lab = TRUE, legend.title = "Pearson Correlation",
           lab_size = 3, ggtheme = theme_classic(),
           outline.color = "black",
           colors = c("orange", "green", "red"))

```



```

# Create a 70-30 train-test split
# Load the caret package
library(caret)

# Create the partition index
dindex <- createDataPartition(balanced_data$FMD_Status, p = 0.7, list = FALSE)

# Use the index to split into training and test sets
train_data <- balanced_data[dindex, ]
test_data <- balanced_data[-dindex, ]

dindex <- createDataPartition(balanced_data$FMD_Status, p = 0.7, list = FALSE)
train_data<- balanced_data[dindex,]
test_data<- balanced_data[-dindex,]
table(train_data$FMD_Status)

## 
## Negative Positive
##      94       93

# Ensure no perfect multicollinearity:
nzv <- nearZeroVar(train_data, saveMetrics = TRUE)
train_data_clean <- train_data[, !nzv$zeroVar]
test_data <- test_data[, colnames(train_data)]

```

```

#check the dimension
dim(train_data)

## [1] 187 14

dim(test_data)

## [1] 79 14

#Feature selection analysis
library(Boruta)

## Warning: package 'Boruta' was built under R version 4.4.3

library(dplyr)
library(ggplot2)

# Run Boruta (you've done this already)
train_data$FMD_Status <- as.factor(train_data$FMD_Status)
boruta_results <- Boruta(FMD_Status ~ ., data = train_data, doTrace = 2)

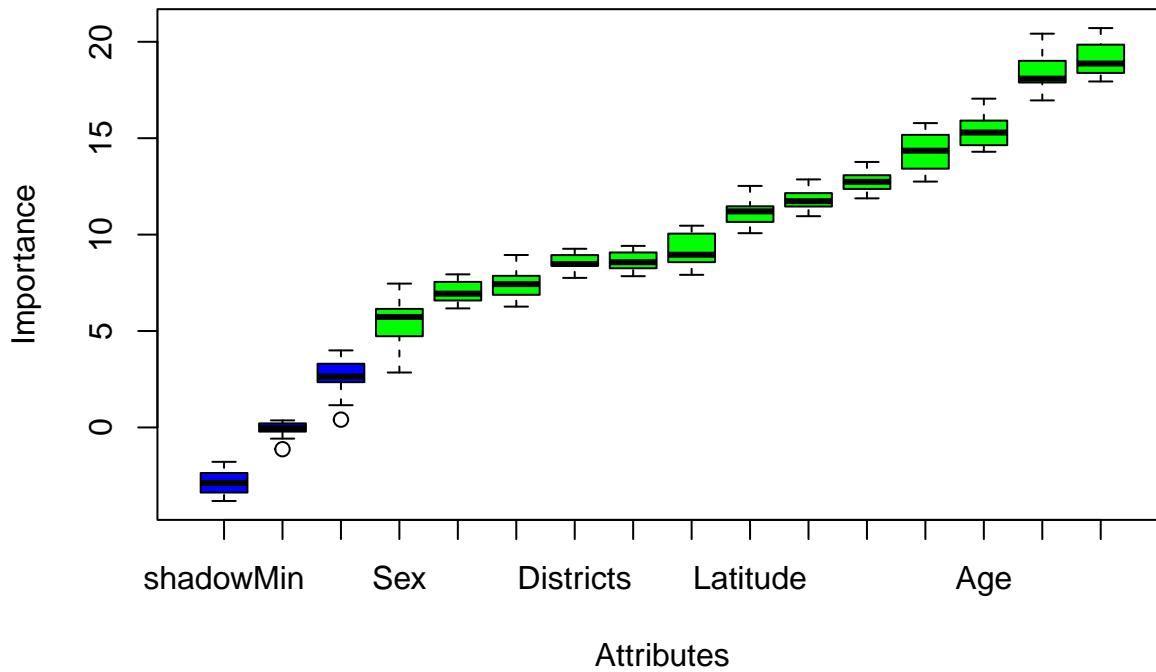
## 1. run of importance source...
## 2. run of importance source...
## 3. run of importance source...
## 4. run of importance source...
## 5. run of importance source...
## 6. run of importance source...
## 7. run of importance source...
## 8. run of importance source...
## 9. run of importance source...
## 10. run of importance source...
## 11. run of importance source...
## After 11 iterations, +1.7 secs:
## confirmed 12 attributes: Age, Agro_Climate, Altitude, Body_Condition, Breed and 7 more;
## still have 1 attribute left.
##
## 12. run of importance source...
## 13. run of importance source...
## 14. run of importance source...
## 15. run of importance source...
## After 15 iterations, +2.2 secs:
## confirmed 1 attribute: Sex;
## no more attributes left.

boruta_results

## Boruta performed 15 iterations in 2.211023 secs.
## 13 attributes confirmed important: Age, Agro_Climate, Altitude,
## Body_Condition, Breed and 8 more;
## No attributes deemed unimportant.

```

```
plot(boruta_results)
```



```
boruta_results <- Boruta(FMD_Status ~ ., data = train_data, doTrace = 2)
```

```
## 1. run of importance source...
## 2. run of importance source...
## 3. run of importance source...
## 4. run of importance source...
## 5. run of importance source...
## 6. run of importance source...
## 7. run of importance source...
## 8. run of importance source...
## 9. run of importance source...
## 10. run of importance source...
## 11. run of importance source...
## After 11 iterations, +1.4 secs:
## confirmed 10 attributes: Age, Agro_Climate, Altitude, Body_Condition, Breed and 5 more;
## still have 3 attributes left.
##
## 12. run of importance source...
## 13. run of importance source...
## 14. run of importance source...
## 15. run of importance source...
## 16. run of importance source...
## 17. run of importance source...
```

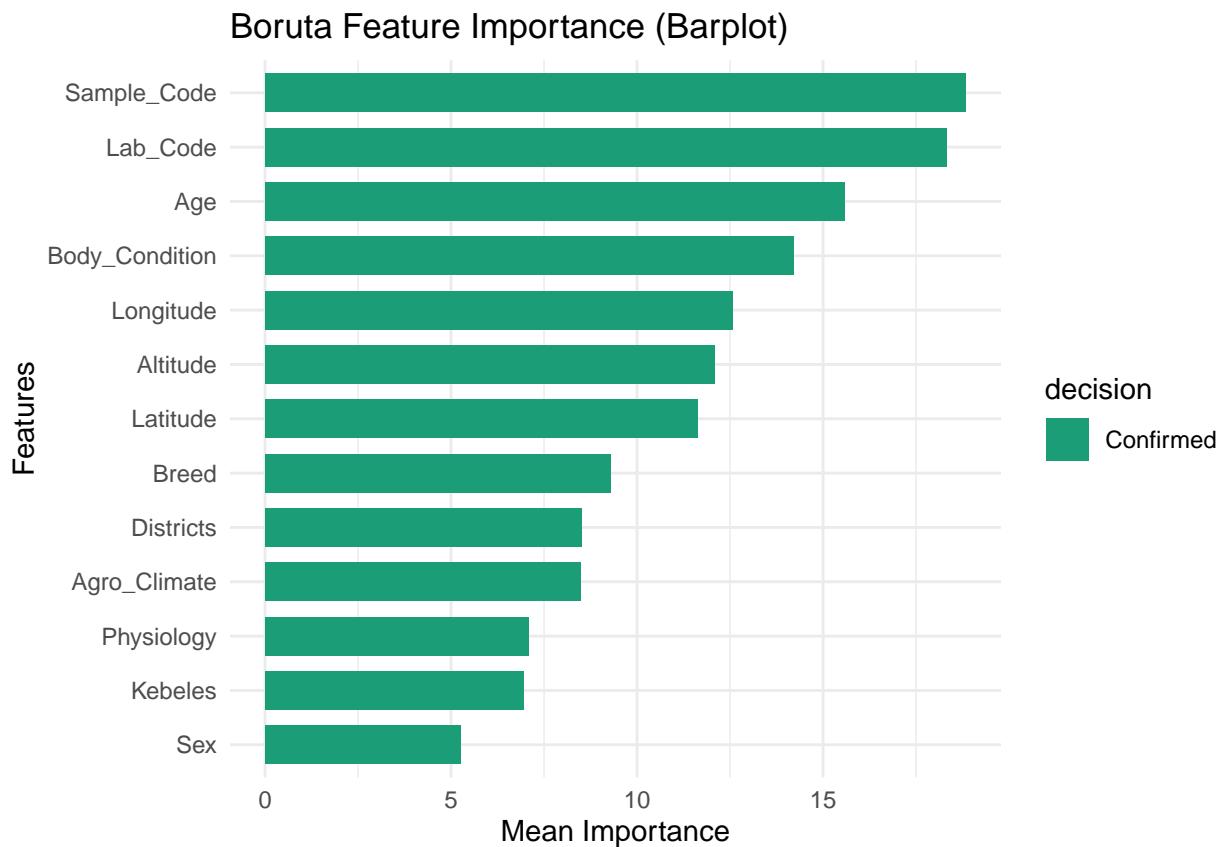
```

## 18. run of importance source...
## After 18 iterations, +2.4 secs:
## confirmed 3 attributes: Kebeles, Physiology, Sex;
## no more attributes left.

# Extract attribute statistics and calculate mean importance
boruta_df <- attStats(boruta_results) %>%
  rownames_to_column(var = "Feature") %>%
  filter(decision != "Rejected") %>%
  arrange(desc(meanImp))

# Plot as a barplot
ggplot(boruta_df, aes(x = reorder(Feature, meanImp), y = meanImp, fill = decision)) +
  geom_bar(stat = "identity", width = 0.7) +
  coord_flip() +
  labs(title = "Boruta Feature Importance (Barplot)",
       x = "Features", y = "Mean Importance") +
  theme_minimal() +
  scale_fill_manual(values = c("Confirmed" = "#1b9e77", "Tentative" = "#d95f02"))

```



```

library(Boruta)
train_data$FMD_Status <- as.factor(train_data$FMD_Status)
boruta_results <- Boruta(FMD_Status ~ ., data = train_data, doTrace = 2)

## 1. run of importance source...

```

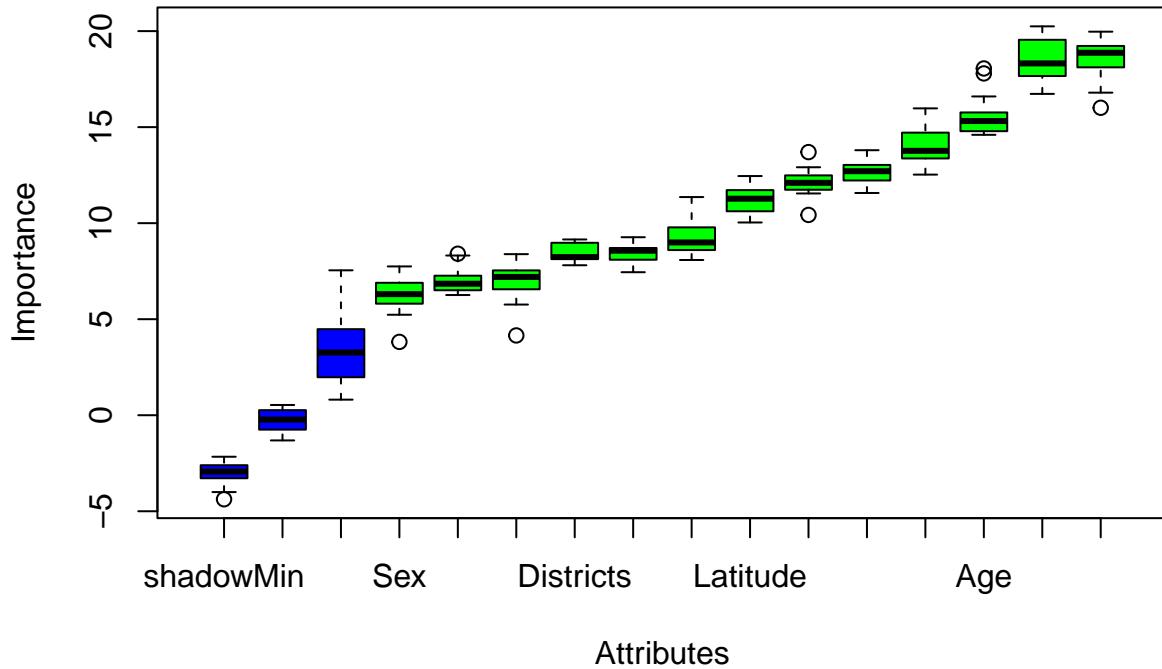
```
## 2. run of importance source...
## 3. run of importance source...
## 4. run of importance source...
## 5. run of importance source...
## 6. run of importance source...
## 7. run of importance source...
## 8. run of importance source...
## 9. run of importance source...
## 10. run of importance source...
## 11. run of importance source...
## After 11 iterations, +1.1 secs:
## confirmed 10 attributes: Age, Agro_Climate, Altitude, Body_Condition, Breed and 5 more;
## still have 3 attributes left.
##
## 12. run of importance source...
## 13. run of importance source...
## 14. run of importance source...
## 15. run of importance source...
## After 15 iterations, +1.5 secs:
## confirmed 3 attributes: Kebeles, Physiology, Sex;
## no more attributes left.
```



```
boruta_results
```

```
## Boruta performed 15 iterations in 1.529004 secs.
## 13 attributes confirmed important: Age, Agro_Climate, Altitude,
## Body_Condition, Breed and 8 more;
## No attributes deemed unimportant.
```

```
plot(boruta_results)
```



```
#Variables falling above the cut off (0.75)
findCorrelation(correlat, cutoff = 0.75)

## [1] 4 3 8

names(train_data)

##  [1] "Sample_Code"      "Lab_Code"        "Districts"       "Kebeles"
##  [5] "Breed"            "Sex"             "Age"             "Body_Condition"
##  [9] "Physiology"        "Longitude"       "Latitude"        "Altitude"
## [13] "Agro_Climate"     "FMD_Status"

# Drop Sample_Code from train and test
library(dplyr)

train_data <- train_data %>% select(-contains("code"))
test_data <- test_data %>% select(-contains("Code"))
#Dropping latitude,longitude and altitude from the train and test set
train_data <- train_data %>% select(-contains("Longitude"), -contains("Latitude"), -contains("Altitude"))

test_data <- test_data %>% select(-contains("Longitude"), -contains("Latitude"), -contains("Altitude"))

library(summarytools)
library(tableone)
```

```
## Warning: package 'tableone' was built under R version 4.4.3
```

```
#create categorical version of FMD_Status
balanced_data$FMD_Status<-factor(balanced_data$FMD_Status)

#specify variable for summary
vars<-c("Districts","Kebeles","Breed","Sex","Age","Body_Condition","Altitude","Physiology","FMD_Status")
#create table
tableII<-CreateTableOne(vars=vars,strata="FMD_Status",data=balanced_data,test=TRUE)
print(tableII,showAllLevels=TRUE,qoute=TRUE,nospaces=TRUE)
```

Stratified by FMD_Status					
	level	Negative	Positive	p	test
## n		134	132		
## Districts (%)	Diga	48 ( 35.8)	80 ( 60.6)	<0.001	
	Guto_Gida	62 ( 46.3)	45 ( 34.1)		
	Nekemte	24 ( 17.9)	7 ( 5.3)		
## Kebeles (%)	Abdetta	19 ( 14.2)	19 ( 14.4)	<0.001	
	Arjo_Gudatu	23 ( 17.2)	63 ( 47.7)		
	Bake_Jema	5 ( 3.7)	2 ( 1.5)		
	Calalak	19 ( 14.2)	5 ( 3.8)		
	Efa	10 ( 7.5)	7 ( 5.3)		
	Fayinera	19 ( 14.2)	7 ( 5.3)		
	Jirata	15 ( 11.2)	10 ( 7.6)		
	Uke	24 ( 17.9)	19 ( 14.4)		
## Breed (%)	Cross	42 ( 31.3)	47 ( 35.6)	0.544	
	Local	92 ( 68.7)	85 ( 64.4)		
## Sex (%)	Female	69 ( 51.5)	103 ( 78.0)	<0.001	
	Male	65 ( 48.5)	29 ( 22.0)		
## Age (%)	Adult	35 ( 26.1)	37 ( 28.0)	<0.001	
	Old	49 ( 36.6)	83 ( 62.9)		
	Young	50 ( 37.3)	12 ( 9.1)		
## Body_Condition (%)	Good	45 ( 33.6)	8 ( 6.1)	<0.001	
	Medium	8 ( 6.0)	11 ( 8.3)		
	Poor	81 ( 60.4)	113 ( 85.6)		
## Altitude (%)	1327	22 ( 16.4)	65 ( 49.2)	<0.001	
	1374	24 ( 17.9)	19 ( 14.4)		
	2027	19 ( 14.2)	19 ( 14.4)		
	2076	6 ( 4.5)	0 ( 0.0)		
	2090	19 ( 14.2)	7 ( 5.3)		
	2122	19 ( 14.2)	5 ( 3.8)		
	2203	10 ( 7.5)	7 ( 5.3)		
	2233	15 ( 11.2)	10 ( 7.6)		
## Physiology (%)	Bull	34 ( 25.4)	9 ( 6.8)	<0.001	
	Dry	20 ( 14.9)	27 ( 20.5)		
	Heifer	29 ( 21.6)	30 ( 22.7)		
	Lactating	11 ( 8.2)	34 ( 25.8)		
	Ox	31 ( 23.1)	20 ( 15.2)		
	Pregnant	9 ( 6.7)	12 ( 9.1)		
## FMD_Status (%)	Negative	134 (100.0)	0 ( 0.0)	<0.001	
	Positive	0 ( 0.0)	132 (100.0)		

```

#prepare a training scheme
fitcontrol<-trainControl(method="repeatedcv",
                           number=10,
                           repeats =3,
                           classProbs = TRUE,
                           summaryFunction = twoClassSummary)

# Remove linearly dependent variables
model_temp <- glm(FMD_Status ~ ., data = train_data, family = "binomial")
alias_info <- alias(model_temp)$Complete
if (!is.null(alias_info)) {
  vars_to_remove <- rownames(alias_info)
  vars_to_remove <- vars_to_remove[vars_to_remove %in% names(train_data)]

  train_data <- train_data[, !(names(train_data) %in% vars_to_remove)]
  test_data <- test_data[, !(names(test_data) %in% vars_to_remove)]
}

# Only keep variables that exist in both train and test data
vars_to_remove <- vars_to_remove[vars_to_remove %in% colnames(train_data)]

# Remove them safely ***
train_data_clean <- train_data[, !(names(train_data) %in% vars_to_remove)]
test_data_clean <- test_data[, !(names(test_data) %in% vars_to_remove)]

# Ensure all categorical variables are factors
train_data_clean[] <- lapply(train_data_clean, function(x) if(is.character(x)) as.factor(x) else x)
test_data_clean[] <- lapply(test_data_clean, function(x) if(is.character(x)) as.factor(x) else x)

# And make sure factor levels are consistent between train and test
for (col in names(train_data_clean)) {
  if (is.factor(train_data_clean[[col]])) {
    common_levels <- intersect(levels(train_data_clean[[col]]), levels(test_data_clean[[col]]))

    if (length(common_levels) > 0) {
      train_data_clean[[col]] <- factor(train_data_clean[[col]], levels = common_levels)
      test_data_clean[[col]] <- factor(test_data_clean[[col]], levels = common_levels)
    } else {
      message(paste("No common levels in column:", col, "- skipping."))
    }
  }
}

#logistic model
logit_model<-glm(FMD_Status~.,data = train_data,family = 'binomial')
# Inspect model coefficients
summary(logit_model)

## 
## Call:

```

```

## glm(formula = FMD_Status ~ ., family = "binomial", data = train_data)
##
## Coefficients: (4 not defined because of singularities)
##                               Estimate Std. Error z value Pr(>|z|)
## (Intercept)           -2.10367   1.28312 -1.640 0.101109
## DistrictsGuto_Gida    0.18071   0.99167  0.182 0.855400
## DistrictsNekemte     -1.15050   1.18701 -0.969 0.332425
## KebelesArjo_Gudatu    1.25405   0.92903  1.350 0.177068
## KebelesBake_Jema      -0.53061   1.60196 -0.331 0.740475
## KebelesCalalak          NA        NA       NA      NA
## KebelesEfa             0.41492   1.06334  0.390 0.696388
## KebelesFayinera      -1.53655   0.93478 -1.644 0.100226
## KebelesJirata          NA        NA       NA      NA
## KebelesUke             0.24608   0.74584  0.330 0.741447
## BreedLocal            -1.42440   0.52683 -2.704 0.006857 **
## SexMale                -2.00851   0.85601 -2.346 0.018957 *
## AgeOld                 0.40990   0.48624  0.843 0.399232
## AgeYoung               -2.48416   0.68646 -3.619 0.000296 ***
## Body_ConditionMedium   2.80516   1.29804  2.161 0.030690 *
## Body_ConditionPoor     3.67073   0.88913  4.128 3.65e-05 ***
## PhysiologyDry          0.74879   0.90949  0.823 0.410334
## PhysiologyHeifer       0.02975   0.77498  0.038 0.969374
## PhysiologyLactating    1.61183   1.01357  1.590 0.111777
## PhysiologyOx            1.15141   0.70934  1.623 0.104543
## PhysiologyPregnant      NA        NA       NA      NA
## Agro_ClimateMidland    19.65449  1385.37822  0.014 0.988681
## Agro_ClimateMidland      NA        NA       NA      NA
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
## Null deviance: 259.23  on 186  degrees of freedom
## Residual deviance: 159.14  on 168  degrees of freedom
## AIC: 197.14
##
## Number of Fisher Scoring iterations: 15

alias_vars <- rownames(alias(logit_model)$Complete)
alias_vars <- alias_vars[alias_vars %in% names(train_data)]

train_data_clean <- train_data[, !(names(train_data) %in% alias_vars)]
test_data_clean <- test_data[, !(names(test_data) %in% alias_vars)]
```

*# Fit model again*

```

logit_model <- glm(FMD_Status ~ ., family = "binomial", data = train_data_clean)

alias_info <- alias(glm(FMD_Status ~ ., data = train_data_clean, family = "binomial"))$Complete
alias_info
```

```

##                               (Intercept) DistrictsGuto_Gida DistrictsNekemte
## KebelesCalalak          0            0            1
## KebelesJirata           1           -1           -1
## PhysiologyPregnant     1            0            0
```

```

## Agro_ClimateMidland 1 0 0
## KebelesArjo_Gudatu KebelesBake_Jema KebelesEfa
## KebelesCalalak 0 -1 0
## KebelesJirata -1 0 -1
## PhysiologyPregnant 0 0 0
## Agro_ClimateMidland -1 0 0
## KebelesFayinera KebelesUke BreedLocal SexMale AgeOld
## KebelesCalalak 0 0 0 0 0
## KebelesJirata 0 0 0 0 0
## PhysiologyPregnant 0 0 0 -1 0
## Agro_ClimateMidland 0 -1 0 0 0
## AgeYoung Body_ConditionMedium Body_ConditionPoor
## KebelesCalalak 0 0 0
## KebelesJirata 0 0 0
## PhysiologyPregnant 0 0 0
## Agro_ClimateMidland 0 0 0
## PhysiologyDry PhysiologyHeifer PhysiologyLactating
## KebelesCalalak 0 0 0
## KebelesJirata 0 0 0
## PhysiologyPregnant -1 -1 -1
## Agro_ClimateMidland 0 0 0
## PhysiologyOx Agro_ClimateMidland
## KebelesCalalak 0 0
## KebelesJirata 0 0
## PhysiologyPregnant 0 0
## Agro_ClimateMidland 0 -1

```

```

logit_prediction <- predict(
  logit_model,
  newdata = test_data_clean %>% select(-FMD_Status),
  type = "response"
)

#confusion matrix for log model
logit_classes<-ifelse(logit_prediction>=0.5,"Positive", "Negative")
actual<-test_data$FMD_Status

logit_classes <- factor(logit_classes, levels = c("Negative", "Positive"))
actual <- factor(actual, levels = c("Negative", "Positive"))

logit_matrix<-confusionMatrix(table(actual,logit_classes))
logit_matrix

```

```

## Confusion Matrix and Statistics
##
##          logit_classes
## actual      Negative Positive
##   Negative      29      11
##   Positive      11      28
##
##          Accuracy : 0.7215
##                95% CI : (0.6093, 0.8165)
## No Information Rate : 0.5063

```

```

##      P-Value [Acc > NIR] : 8.053e-05
##
##          Kappa : 0.4429
##
##  Mcnemar's Test P-Value : 1
##
##          Sensitivity : 0.7250
##          Specificity : 0.7179
##          Pos Pred Value : 0.7250
##          Neg Pred Value : 0.7179
##          Prevalence : 0.5063
##          Detection Rate : 0.3671
##          Detection Prevalence : 0.5063
##          Balanced Accuracy : 0.7215
##
##          'Positive' Class : Negative
##


#RANDOM FOREST
#Train Random Forest
library(randomForest)
library(caret)
library(pROC)

rf_model <- randomForest(FMD_Status ~ ., data = train_data, ntree = 100)
rf_preds <- predict(rf_model, newdata = test_data)
rf_cm <- confusionMatrix(rf_preds, test_data$FMD_Status)
pred_prob_rf <- predict(rf_model, newdata = test_data, type = "prob")
print("Random Forest Performance:")

## [1] "Random Forest Performance:"

print(rf_cm)

## Confusion Matrix and Statistics
##
##          Reference
## Prediction Negative Positive
##   Negative       33        5
##   Positive        7       34
##
##          Accuracy : 0.8481
##          95% CI : (0.7497, 0.919)
##          No Information Rate : 0.5063
##          P-Value [Acc > NIR] : 2.053e-10
##
##          Kappa : 0.6963
##
##  Mcnemar's Test P-Value : 0.7728
##
##          Sensitivity : 0.8250
##          Specificity : 0.8718
##          Pos Pred Value : 0.8684

```

```

##           Neg Pred Value : 0.8293
##           Prevalence : 0.5063
##           Detection Rate : 0.4177
## Detection Prevalence : 0.4810
##           Balanced Accuracy : 0.8484
##
##           'Positive' Class : Negative
##

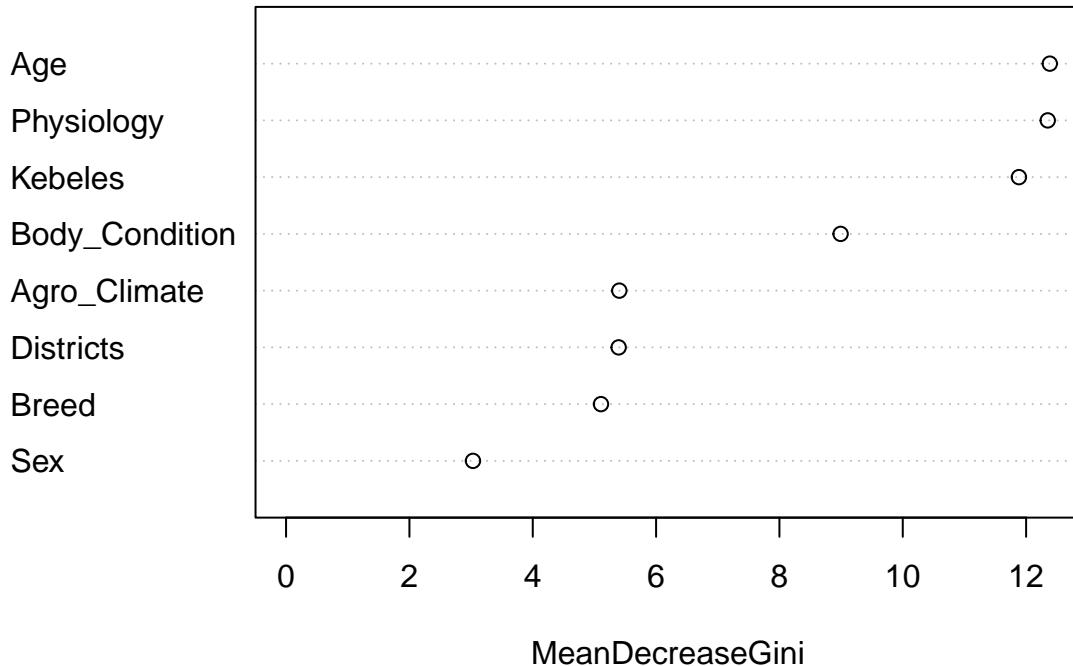
importance(rf_model)

##           MeanDecreaseGini
## Districts          5.393043
## Kebeles           11.883560
## Breed              5.106169
## Sex                3.030743
## Age               12.384340
## Body_Condition    8.991962
## Physiology        12.350770
## Agro_Climate      5.403551

varImpPlot(rf_model, main = "Random Forest Variable Importance")

```

## Random Forest Variable Importance



```

#SUPPORT VECTOR MACHINE (SVM)
#Train Support Vector Machine
svm_model <- svm(FMD_Status ~ ., data = train_data, kernel = "radial", probability = TRUE)
svm_preds <- predict(svm_model, newdata = test_data, probability = TRUE)
svm_cm <- confusionMatrix(svm_preds, test_data$FMD_Status)
print("SVM Performance:")

## [1] "SVM Performance:"


print(svm_cm)

## Confusion Matrix and Statistics
##
##             Reference
## Prediction Negative Positive
##   Negative      31       11
##   Positive       9       28
##
##           Accuracy : 0.7468
##           95% CI : (0.6364, 0.838)
##   No Information Rate : 0.5063
##   P-Value [Acc > NIR] : 1.068e-05
##
##           Kappa : 0.4933
##
##   Mcnemar's Test P-Value : 0.8231
##
##           Sensitivity : 0.7750
##           Specificity  : 0.7179
##   Pos Pred Value : 0.7381
##   Neg Pred Value : 0.7568
##           Prevalence : 0.5063
##           Detection Rate : 0.3924
##   Detection Prevalence : 0.5316
##           Balanced Accuracy : 0.7465
##
##   'Positive' Class : Negative
##

pred_prob_svm <- attr(svm_preds, "probabilities")

# Plot ROC Curve for Logistic Regression
log_prediction_prob <- predict(logit_model, test_data, type="response")
roc_glm <- roc(test_data$FMD_Status, log_prediction_prob)

## Setting levels: control = Negative, case = Positive
## Setting direction: controls < cases

plot(roc_glm, col = "red", main = "ROC Curve Comparison", lwd = 2)

# Plot ROC Curve for Random Forest
roc_rf <- roc(test_data$FMD_Status, pred_prob_rf[, "Positive"])

```

```

## Setting levels: control = Negative, case = Positive
## Setting direction: controls < cases

plot(roc_rf, col = "blue", add = TRUE, lwd = 2)

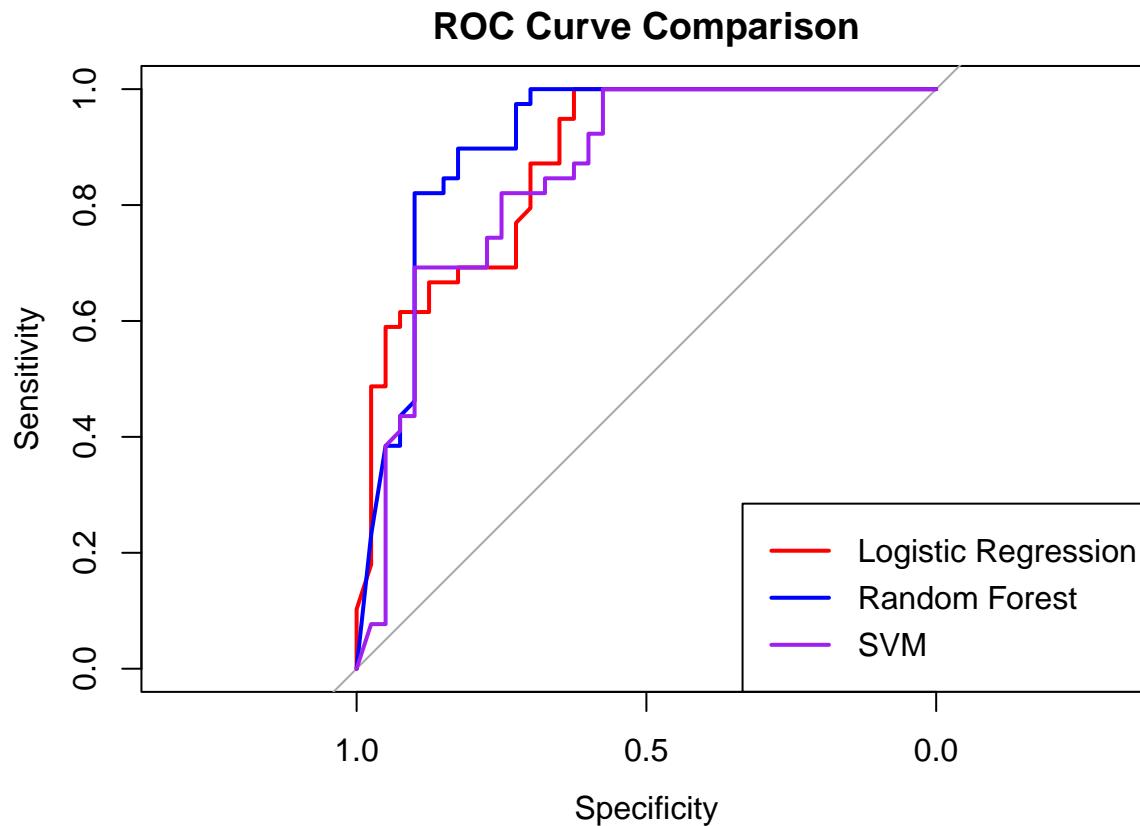
# Plot ROC Curve for SVM
roc_svm <- roc(test_data$FMD_Status, pred_prob_svm[, "Positive"])

## Setting levels: control = Negative, case = Positive
## Setting direction: controls < cases

plot(roc_svm, col = "purple", add = TRUE, lwd = 2)

# Add a legend
legend("bottomright", legend = c("Logistic Regression", "Random Forest", "SVM"),
       col = c("red", "blue", "purple"), lwd = 2)

```



```

#OBJECTIVE TWO

## Logistic Regression
logit_pred <- predict(logit_model, newdata = test_data_clean, type = "response")
logit_pred_class <- ifelse(logit_pred > 0.5, "Positive", "Negative")
#Convert FMD_Status in test data to factor with levels
test_data_clean$FMD_Status <- factor(test_data_clean$FMD_Status,

```

```

levels = c("Negative", "Positive"))
#Convert predicted class to factor with same levels
logit_pred_class <- factor(ifelse(logit_pred > 0.5, "Positive", "Negative"),
                            levels = c("Negative", "Positive"))
#Confusion matrix
library(caret)
logit_cm <- confusionMatrix(logit_pred_class, test_data_clean$FMD_Status)
print("Logistic Confusion Matrix")

```

```
## [1] "Logistic Confusion Matrix"
```

```
logit_cm
```

```

## Confusion Matrix and Statistics
##
##             Reference
## Prediction Negative Positive
##   Negative      29       11
##   Positive      11       28
##
##                   Accuracy : 0.7215
##                   95% CI : (0.6093, 0.8165)
##   No Information Rate : 0.5063
##   P-Value [Acc > NIR] : 8.053e-05
##
##                   Kappa : 0.4429
##
##   Mcnemar's Test P-Value : 1
##
##                   Sensitivity : 0.7250
##                   Specificity  : 0.7179
##   Pos Pred Value : 0.7250
##   Neg Pred Value : 0.7179
##                   Prevalence : 0.5063
##   Detection Rate  : 0.3671
##   Detection Prevalence : 0.5063
##   Balanced Accuracy : 0.7215
##
##   'Positive' Class : Negative
##
```

```

##Random Forest
rf_pred <- predict(rf_model, newdata = test_data_clean)
rf_cm <- confusionMatrix(rf_pred, test_data_clean$FMD_Status)
print("Random Forest Confusion Matrix")

```

```
## [1] "Random Forest Confusion Matrix"
```

```
rf_cm
```

```
## Confusion Matrix and Statistics
```

```

##          Reference
## Prediction Negative Positive
##   Negative      33      5
##   Positive       7     34
##
##                  Accuracy : 0.8481
##                  95% CI : (0.7497, 0.919)
## No Information Rate : 0.5063
## P-Value [Acc > NIR] : 2.053e-10
##
##                  Kappa : 0.6963
##
## Mcnemar's Test P-Value : 0.7728
##
##                  Sensitivity : 0.8250
##                  Specificity : 0.8718
## Pos Pred Value : 0.8684
## Neg Pred Value : 0.8293
## Prevalence : 0.5063
## Detection Rate : 0.4177
## Detection Prevalence : 0.4810
## Balanced Accuracy : 0.8484
##
## 'Positive' Class : Negative
## 

##Support Vector Machine
svm_pred <- predict(svm_model, newdata = test_data_clean)
svm_cm <- confusionMatrix(svm_pred, test_data_clean$FMD_Status)
print("SVM Confusion Matrix")

```

```
## [1] "SVM Confusion Matrix"
```

```
svm_cm
```

```

## Confusion Matrix and Statistics
##
##          Reference
## Prediction Negative Positive
##   Negative      32      12
##   Positive       8     27
##
##                  Accuracy : 0.7468
##                  95% CI : (0.6364, 0.838)
## No Information Rate : 0.5063
## P-Value [Acc > NIR] : 1.068e-05
##
##                  Kappa : 0.4929
##
## Mcnemar's Test P-Value : 0.5023
##
##                  Sensitivity : 0.8000

```

```

##          Specificity : 0.6923
##      Pos Pred Value : 0.7273
##      Neg Pred Value : 0.7714
##          Prevalence : 0.5063
##      Detection Rate : 0.4051
## Detection Prevalence : 0.5570
##      Balanced Accuracy : 0.7462
##
##      'Positive' Class : Negative
##

#ACCURACY
logit_accuracy <- logit_cm$overall["Accuracy"]
rf_accuracy <- rf_cm$overall["Accuracy"]
svm_accuracy <- svm_cm$overall["Accuracy"]

#PRECISION
logit_precision <- logit_cm$byClass["Precision"]
rf_precision <- rf_cm$byClass["Precision"]
svm_precision <- svm_cm$byClass["Precision"]

#RECALL
logit_recall <- logit_cm$byClass["Recall"]
rf_recall <- rf_cm$byClass["Recall"]
svm_recall <- svm_cm$byClass["Recall"]

#F1SCORE
logit_f1 <- logit_cm$byClass["F1"]
rf_f1 <- rf_cm$byClass["F1"]
svm_f1 <- svm_cm$byClass["F1"]

#SUMMARY OF PERFORMANCE METRICS
metrics <- data.frame(
  Model = c( "Random Forest", "SVM", "Logistic Regression"),
  Accuracy = c( rf_accuracy, svm_accuracy, logit_accuracy),
  Precision = c( rf_precision, svm_precision, logit_precision),
  Recall = c( rf_recall, svm_recall, logit_recall),
  F1_Score = c( rf_f1, svm_f1, logit_f1)
)
print(metrics)

##           Model Accuracy Precision Recall  F1_Score
## 1      Random Forest 0.8481013 0.8684211  0.825 0.8461538
## 2                  SVM 0.7468354 0.7272727  0.800 0.7619048
## 3 Logistic Regression 0.7215190 0.7250000  0.725 0.7250000

```