Crab Age Prediction EDA

Devin Streeter, Titus Karuri

2025-04-19

## Load Data

CrabTrain <- read.csv("Data/train-1.csv")

#Exploratory Data Analysis (EDA)

#Data Overview

str(CrabTrain)

## 'data.frame': 15000 obs. of 10 variables:  
## $ id : int 0 1 2 3 4 5 6 7 8 9 ...  
## $ Sex : chr "F" "M" "M" "I" ...  
## $ Length : num 1.375 1.238 1.613 1.15 0.938 ...  
## $ Diameter : num 1.05 0.95 1.262 0.863 0.688 ...  
## $ Height : num 0.287 0.312 0.4 0.287 0.212 ...  
## $ Weight : num 20.9 16.8 38 10.4 6.8 ...  
## $ Shucked.Weight: num 9.82 7.54 15.97 4.85 2.95 ...  
## $ Viscera.Weight: num 4.82 3.71 7.98 2.37 1.35 ...  
## $ Shell.Weight : num 5.1 4.39 13.61 3.54 1.98 ...  
## $ Age : num 9 10 18 8 6 11 10 9 9 8 ...

summary(CrabTrain)

## id Sex Length Diameter   
## Min. : 0 Length:15000 Min. :0.200 Min. :0.0000   
## 1st Qu.: 3750 Class :character 1st Qu.:1.150 1st Qu.:0.8875   
## Median : 7500 Mode :character Median :1.375 Median :1.0750   
## Mean : 7500 Mean :1.315 Mean :1.0221   
## 3rd Qu.:11249 3rd Qu.:1.538 3rd Qu.:1.2000   
## Max. :14999 Max. :2.038 Max. :1.5750   
## Height Weight Shucked.Weight Viscera.Weight   
## Min. :0.0000 Min. : 0.2268 Min. : 0.09922 Min. : 0.01418   
## 1st Qu.:0.2875 1st Qu.:13.1967 1st Qu.: 5.66990 1st Qu.: 2.80660   
## Median :0.3625 Median :23.5584 Median : 9.86563 Median : 4.89029   
## Mean :0.3467 Mean :23.2434 Mean :10.04032 Mean : 5.01447   
## 3rd Qu.:0.4125 3rd Qu.:32.1625 3rd Qu.:13.99048 3rd Qu.: 6.98815   
## Max. :0.6000 Max. :75.3246 Max. :42.18406 Max. :20.12814   
## Shell.Weight Age   
## Min. : 0.09922 Min. : 1.000   
## 1st Qu.: 3.82718 1st Qu.: 8.000   
## Median : 6.80388 Median :10.000   
## Mean : 6.67329 Mean : 9.985   
## 3rd Qu.: 9.07184 3rd Qu.:11.000   
## Max. :29.10076 Max. :29.000

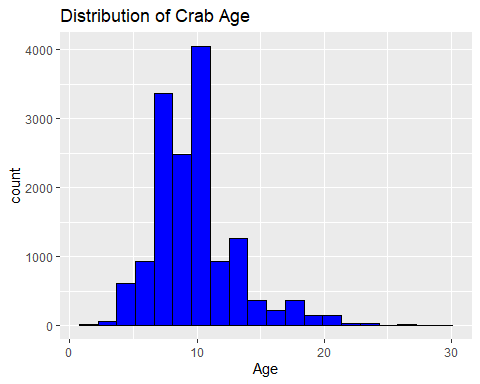
#Correlation with Age

features <- CrabTrain %>%   
 select(where(is.numeric), -id) %>%  
 colnames()  
  
cor\_results <- data.frame(Feature = character(), Correlation = numeric())  
  
for (feature in features) {  
 cor\_val <- cor(CrabTrain[[feature]], CrabTrain$Age)  
 cor\_results <- rbind(cor\_results, data.frame(Feature = feature, Correlation = cor\_val))  
}  
  
cor\_results <- cor\_results %>% arrange(desc(abs(Correlation)))  
print(cor\_results)

## Feature Correlation  
## 1 Age 1.0000000  
## 2 Shell.Weight 0.6856386  
## 3 Height 0.6649325  
## 4 Diameter 0.6292600  
## 5 Length 0.6214662  
## 6 Weight 0.6140155  
## 7 Viscera.Weight 0.5829943  
## 8 Shucked.Weight 0.5039951

#Distribution of Age

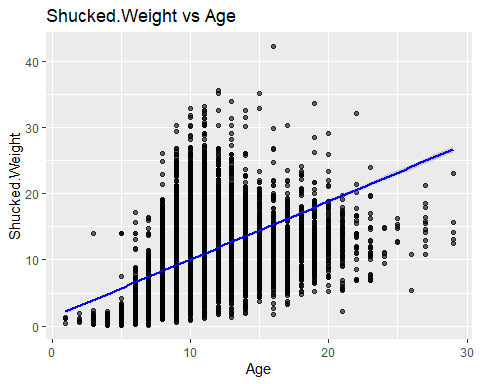
ggplot(CrabTrain, aes(x = Age)) +  
 geom\_histogram(bins = 20, fill = "blue", color = "black") +  
 labs(title = "Distribution of Crab Age")



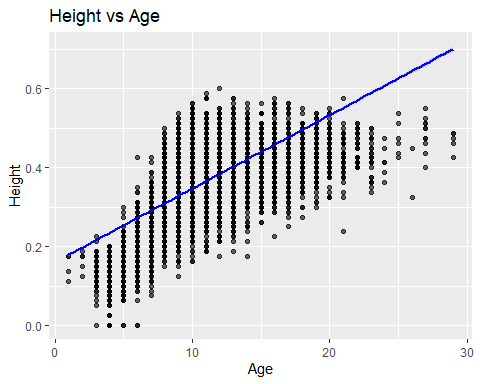
#Scatterplots of Features vs. Age

plot\_list <- c("Shucked.Weight", "Height", "Diameter", "Length", "Weight", "Shell.Weight", "Viscera.Weight")  
  
for (var in plot\_list) {  
 print(  
ggplot(CrabTrain, aes(x = Age, y = !!sym(var))) +  
 geom\_point(alpha = 0.6) +  
 geom\_smooth(method = "lm", se = TRUE, color = "blue") +  
 labs(title = paste(var, "vs Age"))  
 )  
}

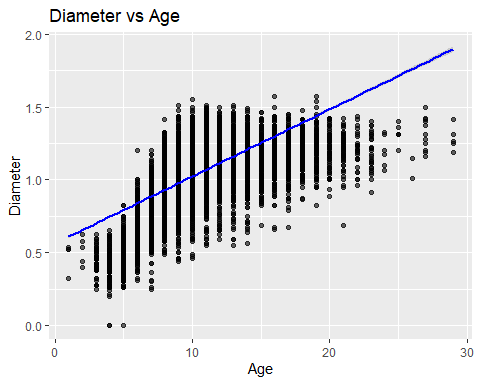
## `geom\_smooth()` using formula = 'y ~ x'



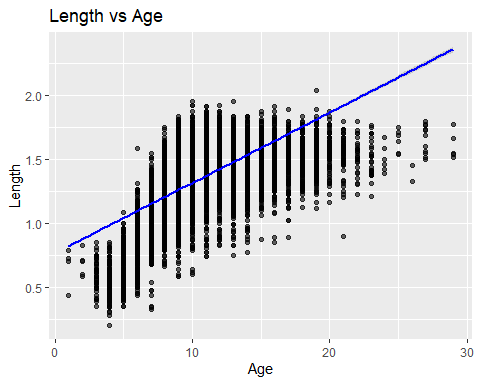
## `geom\_smooth()` using formula = 'y ~ x'



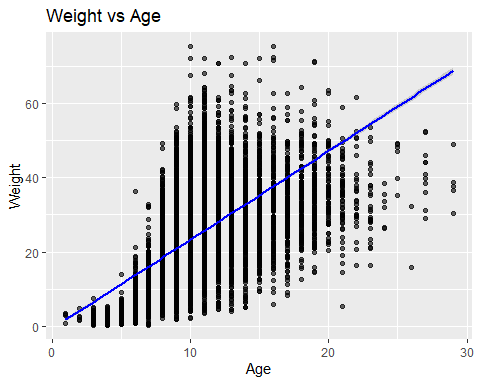
## `geom\_smooth()` using formula = 'y ~ x'



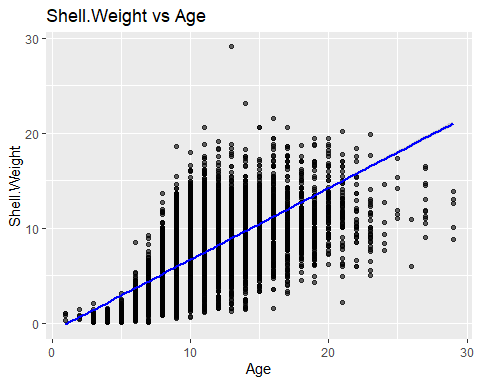
## `geom\_smooth()` using formula = 'y ~ x'



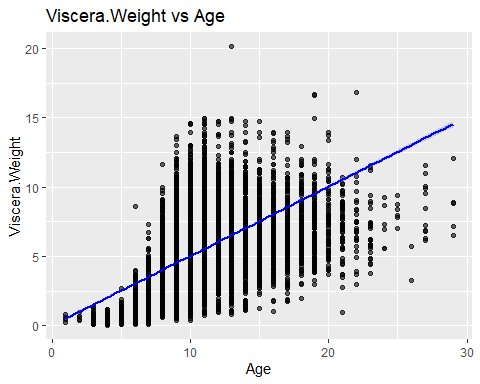
## `geom\_smooth()` using formula = 'y ~ x'



## `geom\_smooth()` using formula = 'y ~ x'



## `geom\_smooth()` using formula = 'y ~ x'



#Feature Selection and Standardization

feature\_df <- CrabTrain %>%  
 select(Shucked.Weight, Height, Diameter, Length, Weight, Age)  
  
feature\_df\_scaled <- feature\_df %>% mutate(across(where(is.numeric), scale))

#Train-Test Split

set.seed(7)  
trainIndex <- createDataPartition(feature\_df\_scaled$Age, p = 0.7, list = FALSE)  
train\_data <- feature\_df\_scaled[trainIndex, ]  
test\_data <- feature\_df\_scaled[-trainIndex, ]

#Linear Regression Model

lm\_model <- lm(Age ~ Shucked.Weight + Height + Diameter + Length + Weight, data = train\_data)  
summary(lm\_model)

##   
## Call:  
## lm(formula = Age ~ Shucked.Weight + Height + Diameter + Length +   
## Weight, data = train\_data)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -3.6659 -0.3883 -0.1131 0.2281 5.2982   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 0.003578 0.006563 0.545 0.586   
## Shucked.Weight -1.412333 0.027370 -51.602 < 2e-16 \*\*\*  
## Height 0.413615 0.020467 20.209 < 2e-16 \*\*\*  
## Diameter 0.234460 0.054005 4.341 1.43e-05 \*\*\*  
## Length -0.050020 0.052679 -0.950 0.342   
## Weight 1.437055 0.033708 42.632 < 2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.6726 on 10497 degrees of freedom  
## Multiple R-squared: 0.5543, Adjusted R-squared: 0.5541   
## F-statistic: 2611 on 5 and 10497 DF, p-value: < 2.2e-16

#Model Evaluation

lm\_pred <- predict(lm\_model, newdata = test\_data)  
mae\_score <- mae(test\_data$Age, lm\_pred)  
cat("Mean Absolute Error (MAE):",mae\_score)

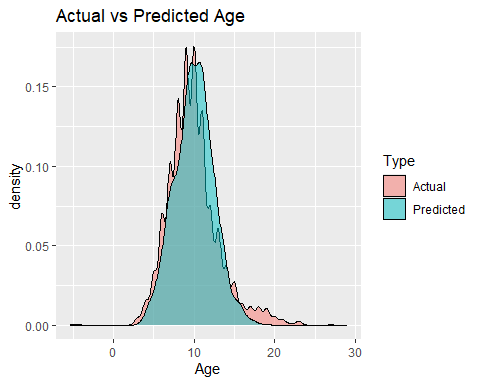
## Mean Absolute Error (MAE): 0.4583566

#Backtransformation for Visualisation

actual\_age <- CrabTrain[-trainIndex, "Age"]  
  
#Original mean and sd  
age\_mean <- mean(CrabTrain[trainIndex, "Age"])  
age\_sd <- sd(CrabTrain[trainIndex, "Age"])  
  
#z-scores to real values  
predicted\_age <- lm\_pred \* age\_sd + age\_mean  
  
#Combine  
df\_compare <- data.frame(Actual = actual\_age,Predicted = predicted\_age) %>%  
 pivot\_longer(cols = everything(), names\_to = "Type", values\_to = "Age")

#Results Density Plot

ggplot(df\_compare, aes(x = Age, fill = Type)) +  
 geom\_density(alpha = 0.5) +  
 labs(title = "Actual vs Predicted Age")



#Residual Plot

residuals\_unscaled <- actual\_age - predicted\_age  
  
ggplot(data.frame(Predicted = predicted\_age, Residuals = residuals\_unscaled),  
 aes(x = Predicted, y = Residuals)) +  
 geom\_point(color = "black", alpha = 0.6) +  
 geom\_hline(yintercept = 0, linetype = "dashed", color = "red") +  
 labs(title = "Residuals vs Predicted Age",x = "Predicted Age", y = "Residuals")

