What Makes the Honey So Sweet?

Project Overview

The goal of this project is to develop a machine learning model that predicts honey purity, classifying it into categories such as “good,” “normal,” and “bad” based on various features such as water content, color score, electrical conductivity, density, pH, price, etc.

```
# loading packages
library(ggplot2)
library(dplyr)
library(caret)
library(randomForest)
library(GGally)
library(readr)
```

Loading the Dataset

```
# loading the dataset
honey_data <- read_csv("honey_purity_dataset.csv")

head(honey_data)
```

```
## # A tibble: 6 × 11
##      CS Density    WC    pH    EC     F     G Pollen_analysis Viscosity Purity
##   <dbl>   <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <chr>               <dbl>  <dbl>
## 1  2.81    1.75  23.0  6.29  0.76  39.0  33.6 Blueberry           4844.   0.68
## 2  9.47    1.82  17.5  7.2   0.71  38.2  34.4 Alfalfa             6689.   0.89
## 3  4.61    1.84  23.7  7.31  0.8   27.5  34.4 Chestnut            6884.   0.66
## 4  1.77    1.4   16.6  4.01  0.78  31.5  28.2 Blueberry           7168.   1
## 5  6.11    1.25  19.6  4.82  0.9   29.6  42.5 Alfalfa             5125.   1
## 6  2.17    1.35  20.7  4.11  0.75  27.2  43.5 Borage              3967.   0.8
## # ℹ 1 more variable: Price <dbl>
```

Data Preprocessing

```
# check for missing values
sum(is.na(honey_data))
```

```
## [1] 0
```

```
# assigning categorical values to quality column
honey_data$cat_purity <- ifelse(honey_data$Purity < 0.75, "bad",
  ifelse(honey_data$Purity >= 0.75 & honey_data$Purity < 0.97, "normal", "good"))

honey_data$cat_purity <- as.factor(honey_data$cat_purity)
```

The honey is categorized based on purity levels: “bad,” “normal,” and “good.” These labels serve as the response variable for classification.

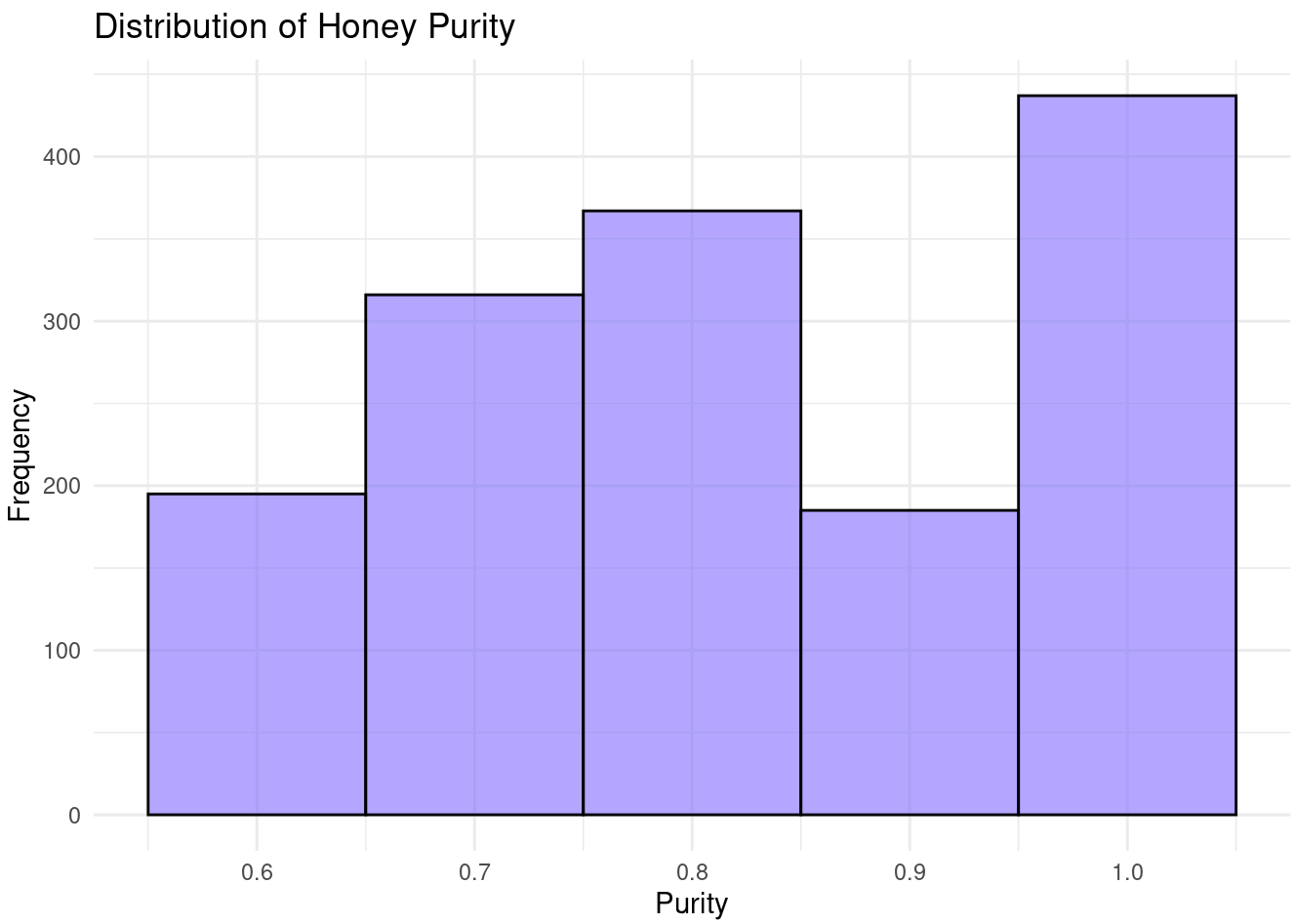
```
# making dataset size smaller to avoid long computation times
honey_data <- honey_data %>%
  group_by(cat_purity) %>%
  sample_frac(size = 1500 / nrow(.)) %>%
  ungroup()
```

```
# how many values are associated with each quality?
table(honey_data$cat_purity)
```

```
##
##    bad   good normal
##    511    434    555
```

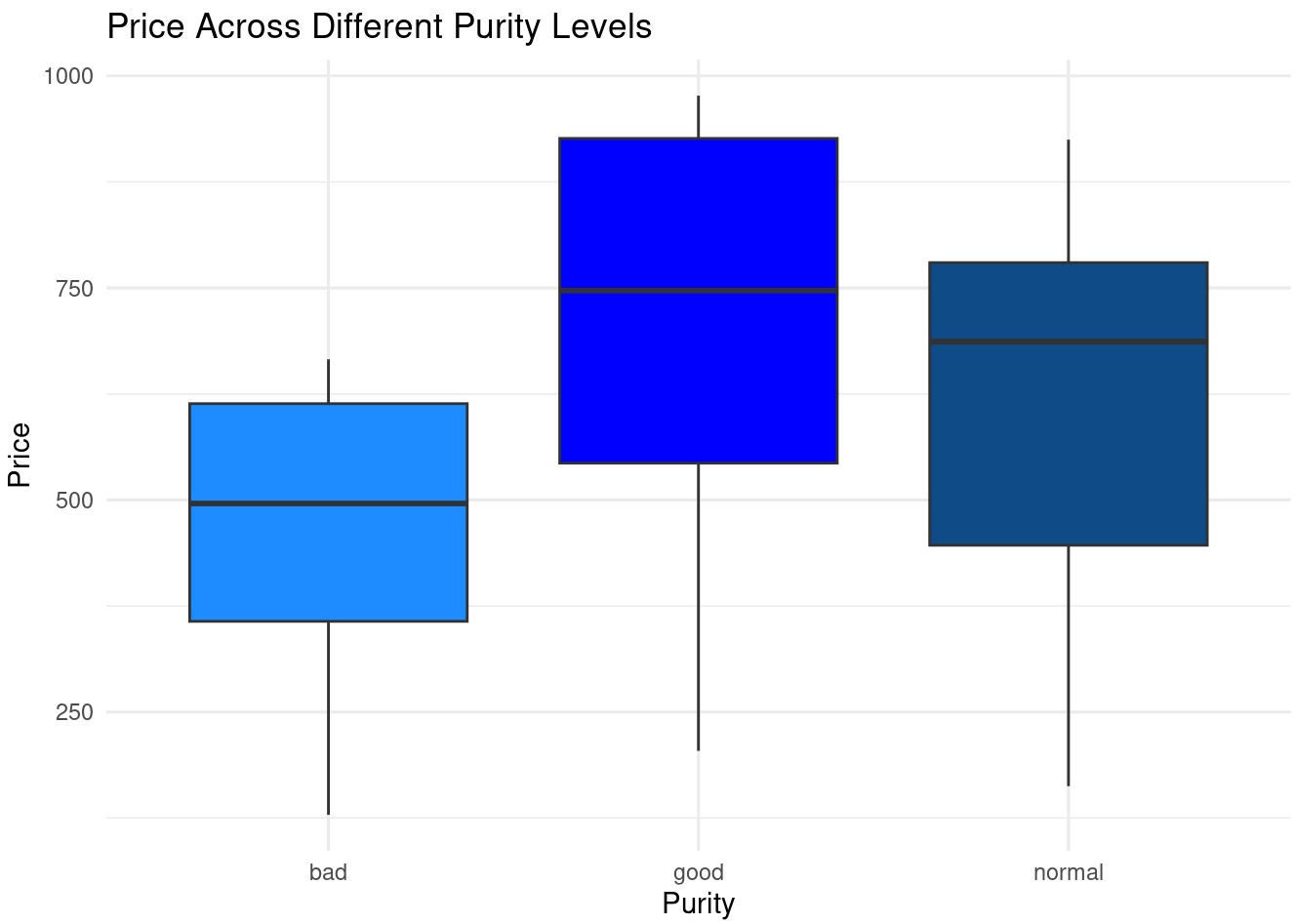
Exploratory Data Analysis (EDA)

```
# histogram -> frequency of honey purity values
ggplot(honey_data, aes(x = Purity)) +
  geom_histogram(binwidth = 0.1, fill = "lightslateblue", color = "black", alpha = 0.6) +
  labs(title = "Distribution of Honey Purity",
       x = "Purity",
       y = "Frequency") +
  theme_minimal()
```



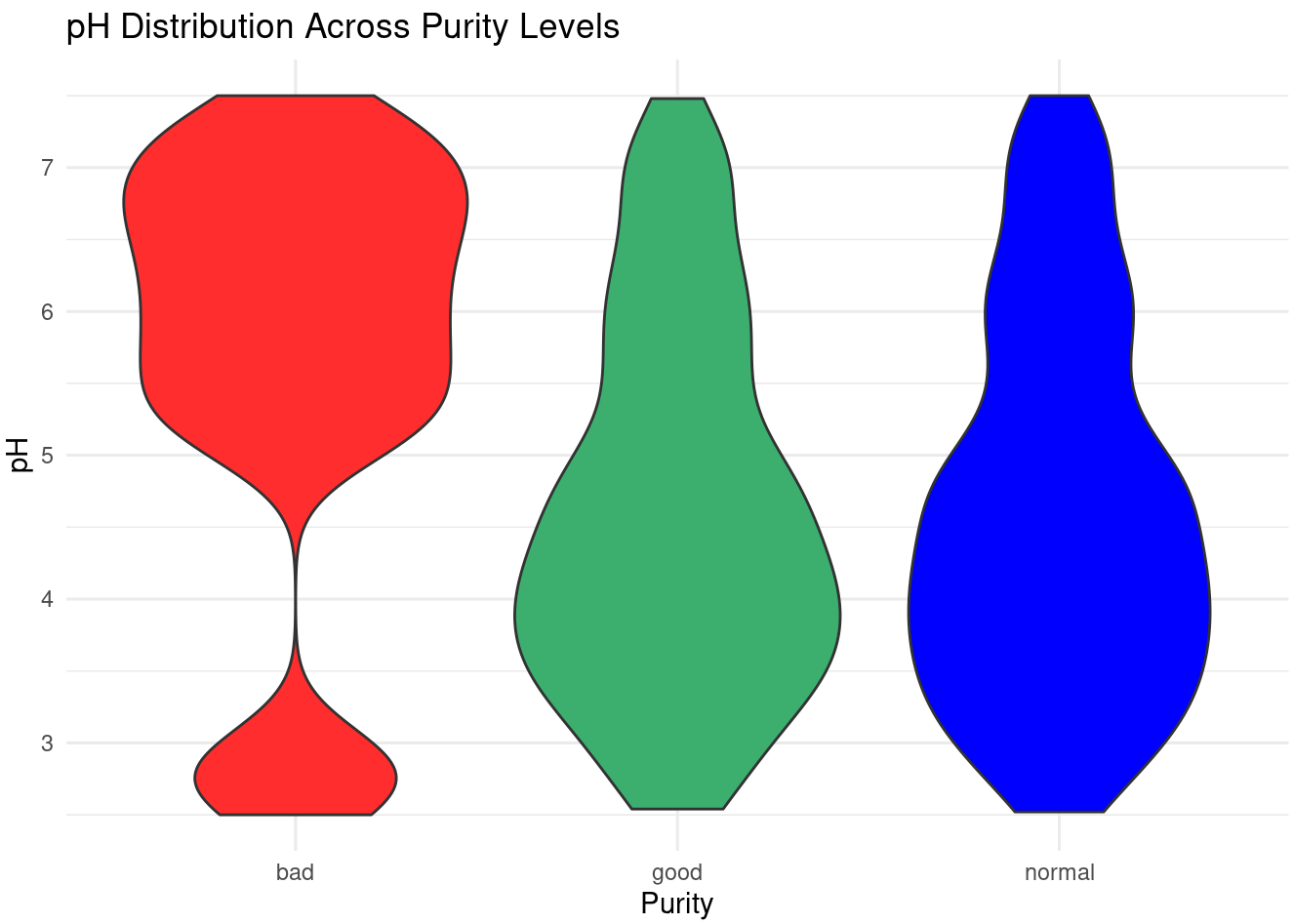
The distribution of the Purity values is visualized to understand the spread of the data and identify any potential outliers.

```
# boxplot => distribution of price across honey purity levels
ggplot(honey_data, aes(x = cat_purity, y = Price, fill = cat_purity)) +
  geom_boxplot() +
  labs(title = "Price Across Different Purity Levels",
       x = "Purity",
       y = "Price") + scale_fill_manual(values = c("bad" = "dodgerblue", "normal" = "dodgerblue
4", "good" = "blue")) +
  theme_minimal() + guides(fill = "none")
```



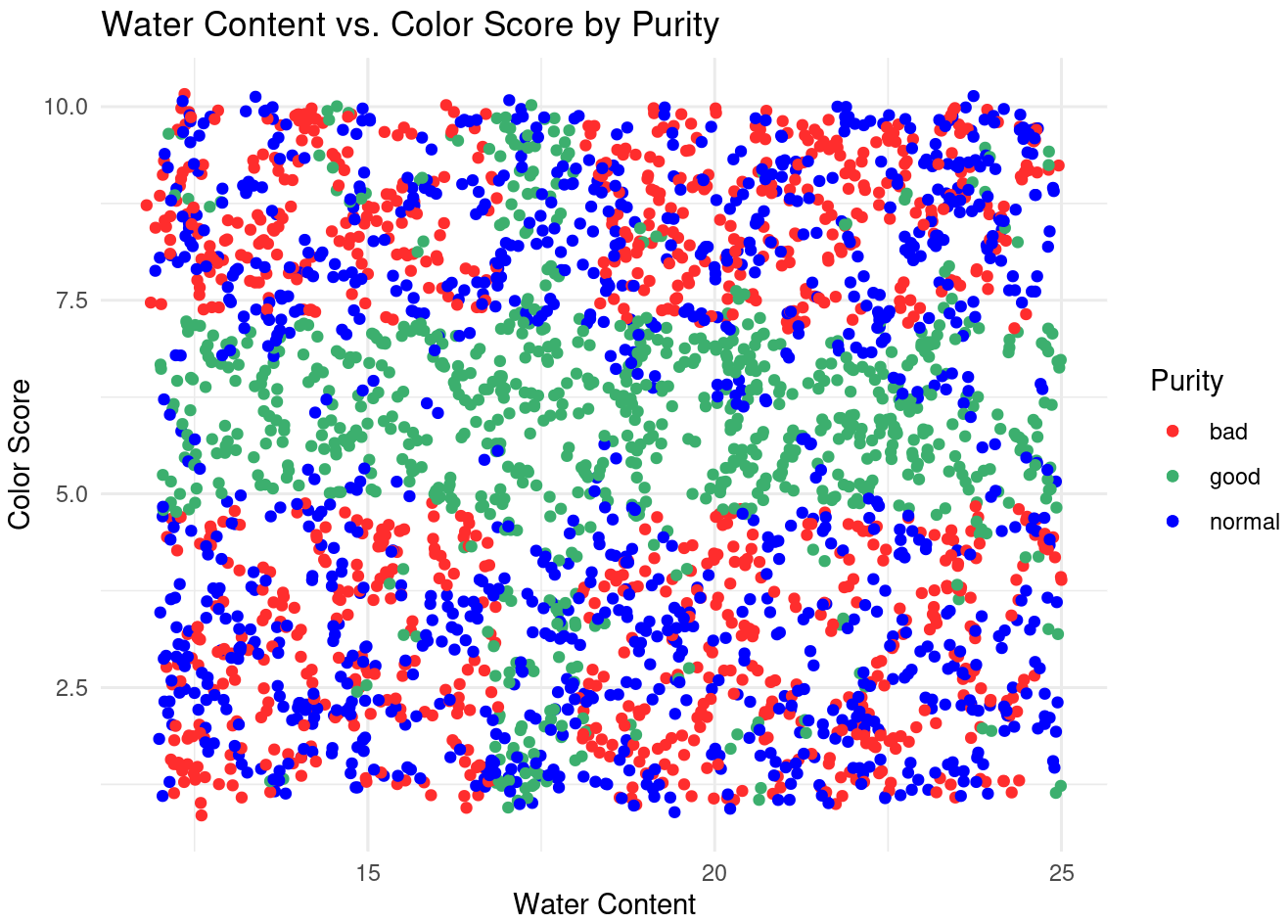
This boxplot examines how honey prices vary across different purity levels, providing insight into the economic factors linked to honey quality. As seen above, the price appears that it will be a strong predictor for purity as the price distributions vary by a fair amount across all purities.

```
# violin plot => distribution of pH across purity levels
ggplot(honey_data, aes(x = cat_purity, y = pH, fill = cat_purity)) +
  geom_violin() +
  labs(title = "pH Distribution Across Purity Levels",
       x = "Purity",
       y = "pH") + scale_fill_manual(values = c("bad" = "firebrick1", "normal" = "blue", "good"
= "mediumseagreen")) +
  theme_minimal() + guides(fill = "none")
```



The violin plot represents the distribution of pH levels for different honey purity categories, helping us assess the potential impact of acidity on honey quality. As seen in the plot above, every purity level seems to have different curves at which most of the pH values are condensed. This indicates that pH could also be a strong predictor for honey purity.

```
# scatterplot => how does the Water Content vs. Color Score of honey correlate across
# different purity levels?
ggplot(honey_data, aes(x = WC, y = CS, color = cat_purity)) +
  geom_point() +
  geom_jitter(width = 0.2, height = 0.2) +
  labs(title = "Water Content vs. Color Score by Purity",
       x = "Water Content",
       y = "Color Score") +
  scale_color_manual(name = "Purity",
                     values = c("bad" = "firebrick1",
                                "normal" = "blue",
                                "good" = "mediumseagreen")) +
  theme_minimal()
```



The scatterplot depicts the patterns and/or clusters of how Water Content (WC) vs. Color Score (CS) interact across different honey purity levels. As seen in the scatterplot above, the "good" purity level seems to have a majority of points clusetered at the average WC mapped to an average CS. This could serve as a potential strong indicator for model to predict if honey is at the "good" purity level.

Feature Engineering

```
# one hot encoding for Pollen_analysis variable
dummy <- dummyVars(" ~ . - cat_purity", data=honey_data)
encoded_features <- data.frame(predict(dummy, newdata = honey_data))
final_honey_data <- cbind(encoded_features, cat_purity = honey_data$cat_purity)
```

Training and Evaluating the Model

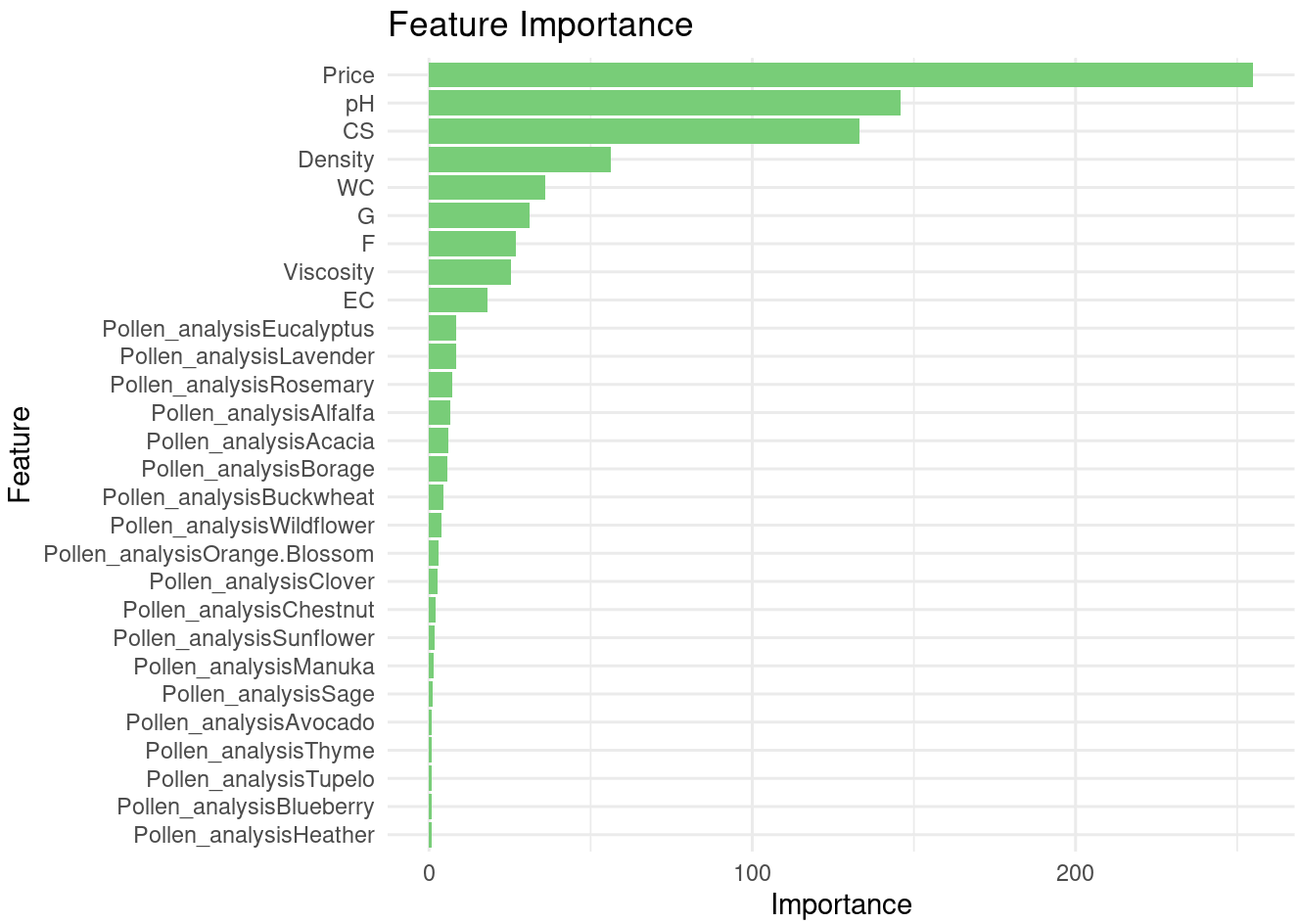
```
# Use a fixed seed for random number generation so the results can be reproduced exactly
set.seed(400)

# splitting data into 80% test and 20% training
samp <- sample(nrow(final_honey_data), 0.8 * nrow(final_honey_data))
train <- final_honey_data[samp, ]
test <- final_honey_data[-samp, ]
```

```
# fitting random forest model on training data
rf_model <- randomForest(cat_purity ~ . -Purity, data = train, ntree = 450, mtry = 10)

# predicting purity on test data
honey_purity_hat <- predict(rf_model, newdata = test)

# calculate accuracy of random forest model
acc_score <- sum(honey_purity_hat==test$cat_purity) / nrow(test)
print(paste("Accuracy Score:", round(acc_score, 4)))
```

```
## [1] "Accuracy Score: 0.94"
```

Feature Importance

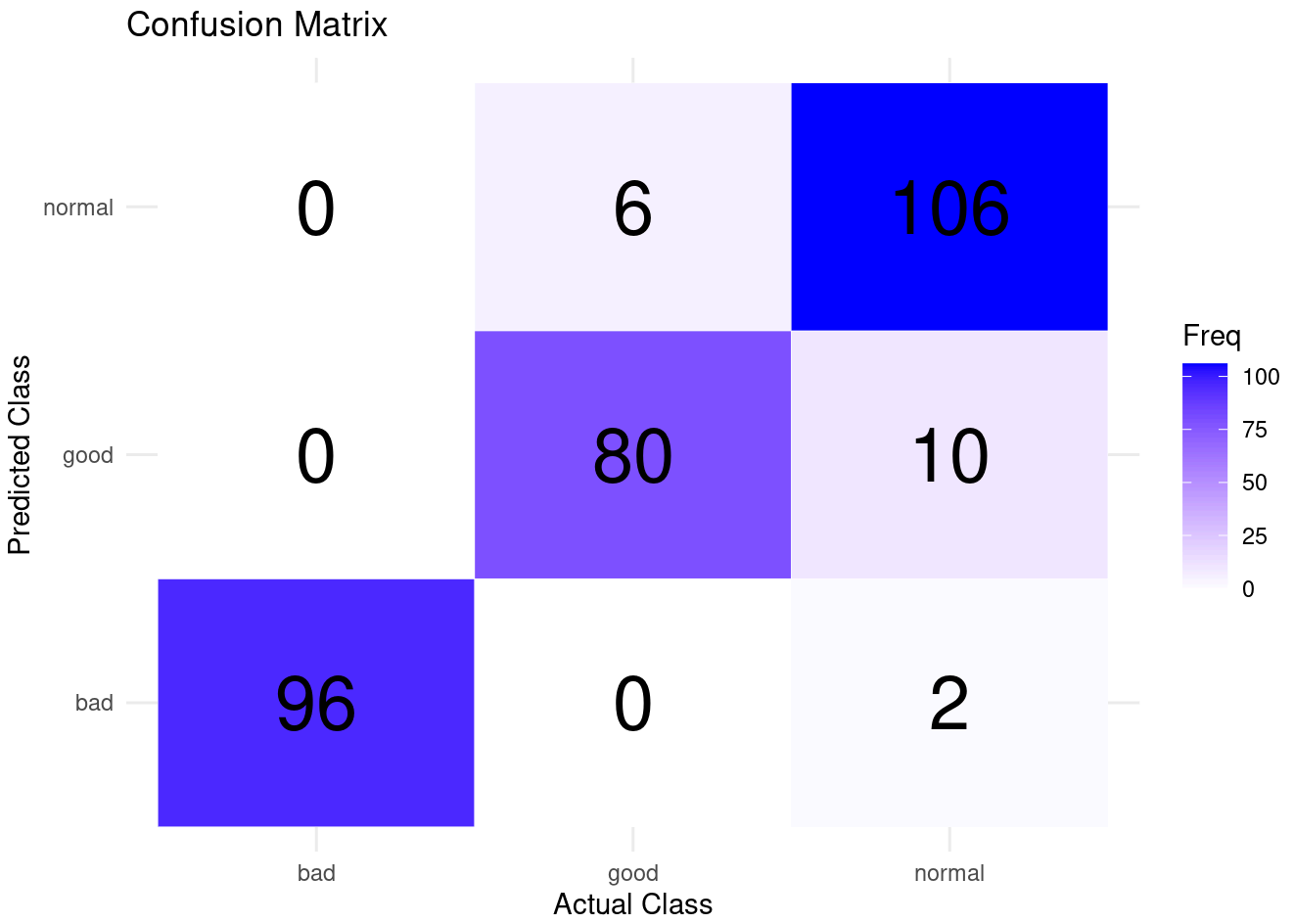
```
# visualizing feature importance of random forest model
importance_values <- importance(rf_model)
importance_df <- data.frame(Feature = rownames(importance_values),
  Importance = importance_values[, 1])
importance_df <- importance_df[order(importance_df$Importance, decreasing = TRUE), ]

ggplot(importance_df, aes(x = reorder(Feature, Importance), y = Importance)) +
  geom_bar(stat = "identity", fill = "palegreen3") +
  coord_flip() +
  labs(title = "Feature Importance", x = "Feature", y = "Importance") +
  theme_minimal()
```



The feature importance helps us to visualize what contributed to the model making its classification decision. As predicted during the exploratory data analysis, price, WC, CS, and pH seem to be one of the most important features. On the other hand, the type of pollen associated with the honey had little to no significance on the model's outcome.

Confusion Matrix

```
# creating a confusion matrix
conf_matrix <- table(Predicted = honey_purity_hat, Actual = test$cat_purity)
conf_matrix_df <- as.data.frame(as.table(conf_matrix))

ggplot(data = conf_matrix_df, aes(x = Actual, y = Predicted, fill = Freq)) +
  geom_tile(color = "white") +  # Tiles with white borders
  scale_fill_gradient(low = "white", high = "blue") +  # Gradient fill
  labs(title = "Confusion Matrix", x = "Actual Class", y = "Predicted Class") +
  theme_minimal() +
  geom_text(aes(label = Freq), color = "black", size = 10)
```



The Confusion Matrix shows how well the model's predictions aligned with the actual purity categories. It also states how many observations in each category were misclassifed as another category => for ex: the # of "good" observations that the model predicted as "normal."