Isolation Forest

2023-12-10

isolation forest theory and explanation

Isolation Forests represent a novel approach in anomaly detection, focusing on isolating anomalies instead of identifying normal data patterns. The method constructs a ‘forest’ of random binary decision trees, based on the principle that anomalies are easier to isolate from the rest of the sample. In an Isolation Forest, each tree aims to isolate data points by randomly selecting a feature and a split value, with the path length—the number of splits needed to isolate a data point—being the key metric. Anomalies, being few and distinct, typically require fewer splits for isolation, which is reflected in their inverse relationship with the anomaly score. For constructing these trees, given a dataset $X = \{x_1, \ldots, x_n\}$ where each x_i is a point in a d-dimensional space, a subset $X' \subset X$ is used. The trees are built by recursively partitioning X' through random selection of an attribute q and a split value p, resulting in a binary tree where each internal node represents a division based on $q < p$. This process continues until a node either contains a single instance or all instances at the node are identical, effectively leveraging the trees’s capacity to progressively narrow down the space in which data points reside, making it highly effective for identifying anomalies that are rare and distinct.

Code

```
#install.packages("isotree")
library(isotree)
data <- read.csv("/users/Navneet/Documents/GitHub/vignette-anomaly-detection/Data/bc_data_prepared.csv")
```

```
# Split the data set into benign and malignant
benign_data <- subset(data, diagnosis == 'B')
malignant_data <- subset(data, diagnosis == 'M')

# Remove the 'diagnosis' column for modeling
benign_data <- benign_data[,-which(names(benign_data) == "diagnosis")]
malignant_data <- malignant_data[,-which(names(malignant_data) == "diagnosis")]

# Train the model on benign data
set.seed(123)
model <- isolation.forest(benign_data, ntrees=100)

# Apply the model to malignant data]
malignant_scores <- predict(model, malignant_data)

# Determine outliers
outliers <- malignant_scores > 0.5
```

```r
# Calculate the accuracy
true_positive <- sum(outliers)
false_negative <- length(outliers) - true_positive

# Accuracy
accuracy <- true_positive / length(outliers)

# Print the results
print(paste("Number of true positives (malignant identified as outliers):", true_positive))
```

```
## [1] "Number of true positives (malignant identified as outliers): 180"
```

```r
print(paste("Number of false negatives (malignant not identified as outliers):", false_negative))
```

```
## [1] "Number of false negatives (malignant not identified as outliers): 32"
```

```r
print(paste("Accuracy:", accuracy))
```

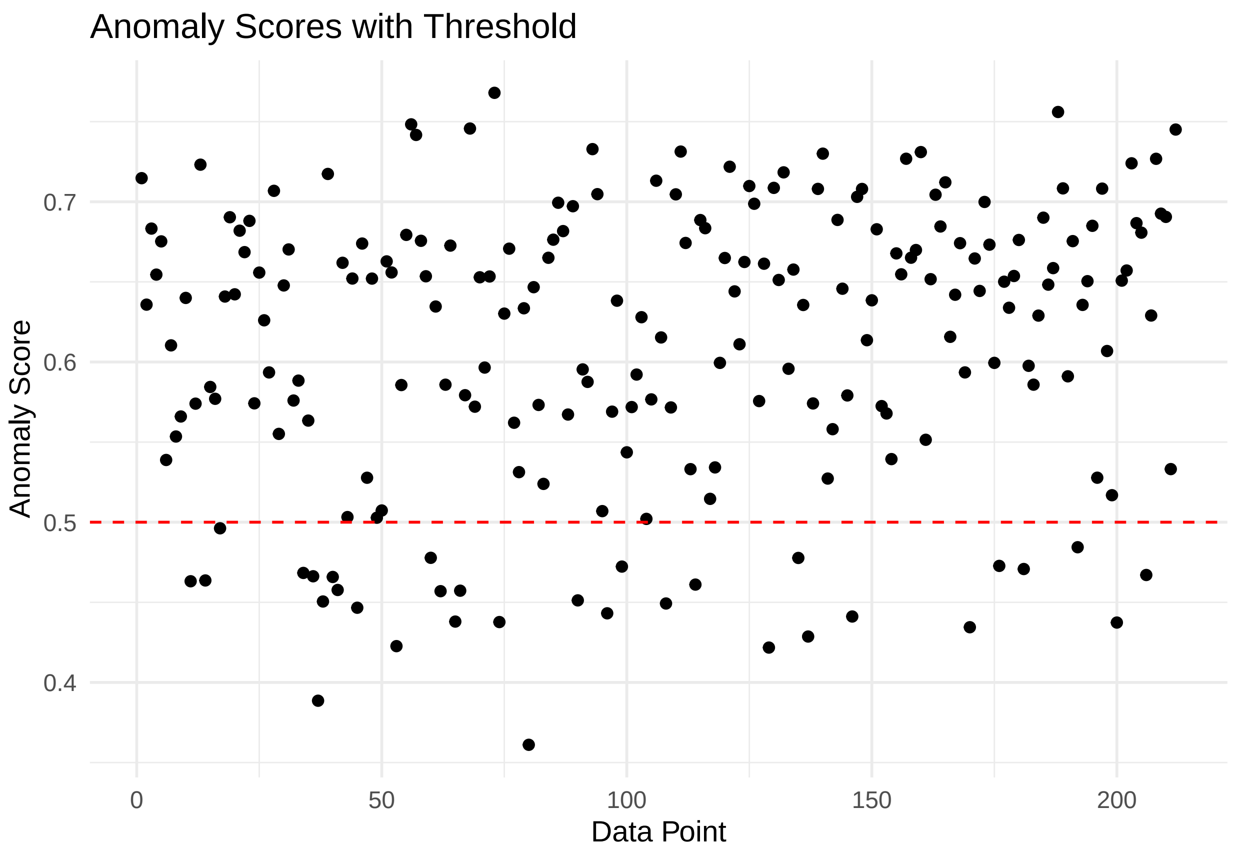
```
## [1] "Accuracy: 0.849056603773585"
```

The output from the isolation forest model indicates that it has correctly identified 180 malignant cases as outliers, which are referred to as true positives. This suggests that the model is effectively distinguishing between benign and malignant cases based on the patterns learned from the benign data it was trained on. However, the model has also yielded 32 false negatives, which are malignant cases that the model failed to flag as outliers. Overall, the model has achieved an accuracy of approximately 84.91%, which quantifies its ability to correctly identify malignant cases as outliers.

```r
library(ggplot2)

# Convert scores to a data frame for ggplot
scores_df <- data.frame(score = malignant_scores)

# Plot
ggplot(scores_df, aes(x = seq_along(score), y = malignant_scores)) +
  geom_point() +
  geom_hline(yintercept = 0.5, linetype = "dashed", color = "red") +
  theme_minimal() +
  labs(title = "Anomaly Scores with Threshold",
       x = "Data Point",
       y = "Anomaly Score")
```

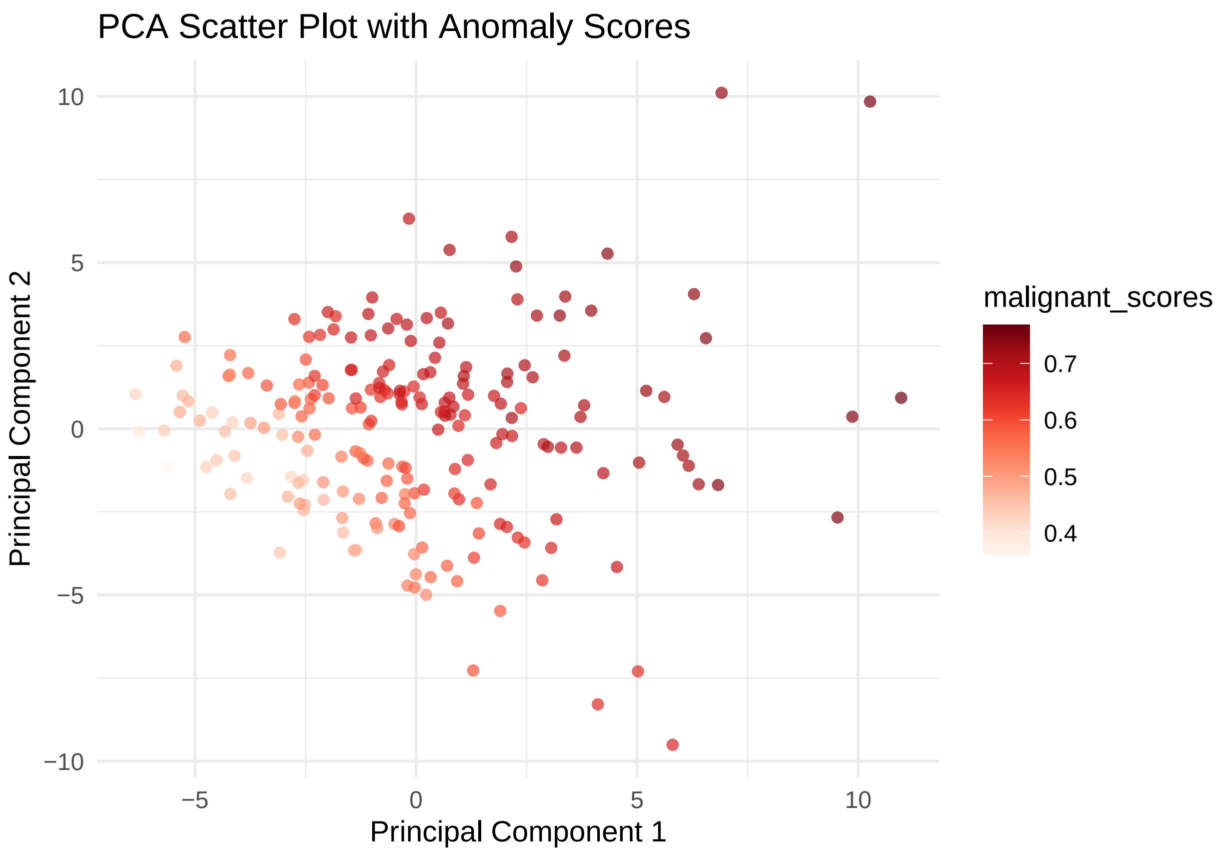


```r
library(ggplot2)
library(RColorBrewer)

# Perform PCA on the data
pca_result <- prcomp(malignant_data[,-c(1, 2)], scale = TRUE)

# Get the first two principal components and add anomaly scores
pca_data <- data.frame(PC1 = pca_result$x[, 1], PC2 = pca_result$x[, 2], Score = malignant_scores)

# Creating the scatter plot
ggplot(pca_data, aes(x = PC1, y = PC2, color = malignant_scores)) +
  geom_point(alpha = 0.7) +
  scale_color_gradientn(colors = brewer.pal(9, "Reds")) +
  theme_minimal() +
  labs(title = "PCA Scatter Plot with Anomaly Scores",
       x = "Principal Component 1",
       y = "Principal Component 2")
```


PCA Scatter Plot with Anomaly Scores
Principal Component 2
Principal Component 1
10
5
0
−5
−10
−5
0
5
10
malignant_scores
0.7
0.6
0.5
0.4