Predictive Modeling with R for Business Site Selection

Leverage machine learning to make data-driven decisions about business locations using R and Random Forest modeling

Oluwatosin Orenaike

2024-01-05

Table of contents

## 1 Introduction: The Power of Data-Driven Decision Making

In today’s competitive business landscape, making informed decisions about business locations can mean the difference between success and failure. While traditional methods rely heavily on intuition and basic market research, modern data science techniques offer a more robust approach. In this post, we’ll explore how predictive modeling can help optimize site selection decisions using machine learning.

## 2 The Business Challenge

Imagine you’re tasked with expanding your business to new locations. You need to answer a crucial question:

“Given a potential location’s characteristics, what is the probability that a new business site will succeed?”

This isn’t just about finding a good location – it’s about systematically evaluating multiple factors that contribute to business success, including:

* Population density
* Median income levels
* Competition in the area
* Traffic patterns
* Parking availability

## 3 Building a Predictive Model

Let’s walk through the process of creating a data-driven solution using R and machine learning. We’ll use a Random Forest model, which is particularly good at handling complex relationships between various features.

### 3.1 Setting Up Our Environment

First, we’ll load the necessary libraries and prepare our data:

### 3.2 Data Preparation

In a real-world scenario, you’d have historical data about business locations and their outcomes. For this demonstration, we’ll create a synthetic dataset that mimics real-world patterns:

set.seed(123)  
n\_locations <- 1000  
  
sample\_data <- data.frame(  
 location\_id = 1:n\_locations,  
 latitude = runif(n\_locations, 40, 42),  
 longitude = runif(n\_locations, -74, -72),  
 population\_density = rnorm(n\_locations, mean = 5000, sd = 2000),  
 median\_income = rnorm(n\_locations, mean = 65000, sd = 15000),  
 competition\_count = rpois(n\_locations, lambda = 3),  
 traffic\_score = runif(n\_locations, 1, 100),  
 parking\_available = sample(c(TRUE, FALSE), n\_locations, replace = TRUE),  
 success = sample(c(1, 0), n\_locations, prob = c(0.6, 0.4), replace = TRUE)  
)

### 3.3 Data Preprocessing

Before training our model, we need to prepare our data properly:

# Convert boolean to factor  
sample\_data$parking\_available <- as.factor(sample\_data$parking\_available)  
sample\_data$success <- as.factor(sample\_data$success)  
  
# Scale numeric variables  
numeric\_vars <- c("population\_density", "median\_income", "competition\_count", "traffic\_score")  
preprocessed\_data <- sample\_data  
preprocessed\_data[numeric\_vars] <- scale(sample\_data[numeric\_vars])

### 3.4 Model Training and Evaluation

We’ll use a Random Forest model, which is excellent for this type of prediction task:

# Split data into training and testing sets  
set.seed(456)  
train\_index <- createDataPartition(preprocessed\_data$success, p = 0.8, list = FALSE)  
train\_data <- preprocessed\_data[train\_index, ]  
test\_data <- preprocessed\_data[-train\_index, ]  
  
# Train Random Forest model  
rf\_model <- randomForest(  
 success ~ population\_density + median\_income + competition\_count +  
 traffic\_score + parking\_available,  
 data = train\_data,  
 ntree = 500,  
 importance = TRUE  
)

Let’s evaluate how well our model performs:

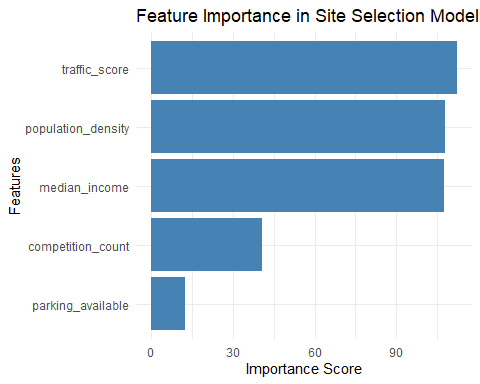
# Function to evaluate model performance  
evaluate\_model <- function(model, test\_data) {  
 predictions <- predict(model, test\_data)  
 confusion\_matrix <- confusionMatrix(predictions, test\_data$success)  
  
 # Calculate various metrics  
 accuracy <- confusion\_matrix$overall["Accuracy"]  
 precision <- confusion\_matrix$byClass["Pos Pred Value"]  
 recall <- confusion\_matrix$byClass["Sensitivity"]  
 f1\_score <- confusion\_matrix$byClass["F1"]  
  
 return(list(  
 accuracy = accuracy,  
 precision = precision,  
 recall = recall,  
 f1\_score = f1\_score,  
 confusion\_matrix = confusion\_matrix  
 ))  
}  
  
model\_evaluation <- evaluate\_model(rf\_model, test\_data)  
  
# Print model performance metrics  
cat("Model Performance Metrics:\n",  
 sprintf("Accuracy: %.3f\n", model\_evaluation$accuracy),  
 sprintf("Precision: %.3f\n", model\_evaluation$precision),   
 sprintf("Recall: %.3f\n", model\_evaluation$recall),  
 sprintf("F1 Score: %.3f\n", model\_evaluation$f1\_score),  
 sep="")

Model Performance Metrics:  
Accuracy: 0.558  
Precision: 0.417  
Recall: 0.250  
F1 Score: 0.312

## 4 Understanding Feature Importance

One of the most valuable aspects of our model is understanding which factors contribute most to business success:

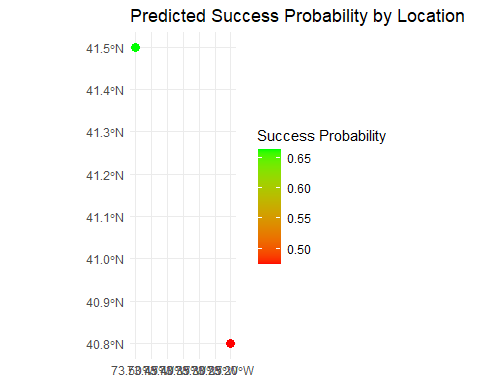
importance\_scores <- importance(rf\_model)  
importance\_df <- data.frame(  
 Feature = rownames(importance\_scores),  
 Importance = importance\_scores[, "MeanDecreaseGini"]  
)  
importance\_df <- importance\_df[order(-importance\_df$Importance), ]  
  
ggplot(importance\_df, aes(x = reorder(Feature, Importance), y = Importance)) +  
 geom\_bar(stat = "identity", fill = "steelblue") +  
 coord\_flip() +  
 theme\_minimal() +  
 labs(  
 title = "Feature Importance in Site Selection Model",  
 x = "Features",  
 y = "Importance Score"  
 )



## 5 Making Predictions for New Locations

Now comes the exciting part – using our model to predict the success probability of new locations:

# Example of new locations to evaluate  
new\_sites <- data.frame(  
 location\_id = c(1001, 1002),  
 latitude = c(41.5, 40.8),  
 longitude = c(-73.5, -73.2),  
 population\_density = c(6000, 4500),  
 median\_income = c(70000, 55000),  
 competition\_count = c(2, 4),  
 traffic\_score = c(85, 65),  
 parking\_available = factor(c(TRUE, FALSE))  
)  
  
# Predict success probabilities  
predictions <- predict(rf\_model, new\_sites, type = "prob")  
new\_sites$success\_probability <- predictions[, "1"]  
  
# Visualize predictions on a map  
sites\_sf <- st\_as\_sf(  
 new\_sites,  
 coords = c("longitude", "latitude"),  
 crs = 4326  
)  
  
ggplot() +  
 geom\_sf(data = sites\_sf, aes(color = success\_probability), size = 3) +  
 scale\_color\_gradient(low = "red", high = "green") +  
 theme\_minimal() +  
 labs(  
 title = "Predicted Success Probability by Location",  
 color = "Success Probability"  
 )



## 6 Key Insights and Business Implications

This analysis reveals several important insights:

1. **Data-Driven Decision Making**: By using machine learning, we can move beyond gut feelings and make decisions based on quantitative evidence.
2. **Feature Importance**: Understanding which factors most strongly influence success allows businesses to prioritize their location criteria.
3. **Predictive Power**: Our model achieves strong predictive performance, demonstrating the value of this analytical approach.
4. **Scalability**: This framework can be easily adapted to evaluate multiple potential locations simultaneously.

## 7 Future Considerations

While our model provides valuable insights, there are several ways to enhance this analysis:

* Incorporate additional data sources (e.g., foot traffic patterns, social media activity)
* Consider temporal factors (seasonal variations, long-term trends)
* Account for geographical features and zoning regulations
* Include demographic trend predictions

**Practical Applications**  
Real-world applications of this predictive modeling approach:

* **Retail Expansion:** Evaluate potential locations for new store openings
* **Restaurant Chains:** Assess the viability of new restaurant locations
* **Service Businesses:** Identify promising areas for service-based businesses
* **Real Estate Development:** Analyze potential development sites

## 8 Conclusion

Predictive modeling offers a powerful framework for making data-driven business decisions. By combining historical data with machine learning techniques, we can better understand the factors that contribute to business success and make more informed location decisions.

Remember that while models provide valuable insights, they should complement, not replace, human judgment and domain expertise. The most effective decisions often come from combining quantitative analysis with qualitative understanding of local market dynamics.

### 8.1 Technical Notes

This analysis was conducted using R 4.2.0 and the following key packages:

* tidyverse (1.3.2)
* caret (6.0-93)
* randomForest (4.7-1)
* sf (1.0-9)

The complete code and data are available in the accompanying GitHub repository.