# VENKATRAMANAN_ASWIN_GANESH_PROJECT

AUTHOR
Aswin Ganesh Venkatramanan

PUBLISHED
November 27, 2024

## Data Description

**Dataset Overview** The dataset includes takeover information for 126 U.S. firms targeted by tender offers over an 8-year period. Each firm has information about bids, financial characteristics, and defensive actions.

**Variables**

1. **ID**: Identifier for each firm (no units).
2. **WEEKS**: Time between initial bid and takeover (in weeks).
3. **BIDNUM**: Number of bids received (integer count).
4. **TOVER**: Binary variable indicating whether the firm was taken over (1 = Yes, 0 = No).
5. **PREM**: Bid premium (bid price divided by the stock price 14 days before the bid).
6. **INST**: Percentage of stock held by institutions (0 to 1, representing 0%-100%).
7. **ASSET**: Total book value of assets (in billions of dollars).
8. **LEGAL**: Binary variable indicating legal defense by lawsuit (1 = Yes, 0 = No).
9. **ASTR**: Binary variable for proposed changes in asset structure (1 = Yes, 0 = No).
10. **OSTR**: Binary variable for proposed changes in ownership structure (1 = Yes, 0 = No).
11. **CHR**: Binary variable for chronic conditions limiting activity (1 = Yes, 0 = No).
12. **THIRD**: Binary variable indicating a management invitation for a friendly third-party takeover (1 = Yes, 0 = No).

**Goal**

The project aims to:

1. Predict the **number of bids (`BIDNUM`)** a firm receives using firm-specific characteristics, defensive actions, and regulatory interventions.
2. Predict whether a firm receives **more than one bid (`BIRY`)**, a binary variable created from `BIDNUM`.

## Statistical Methods

Predicting BIDNUM (Question a)

**Methodology**:

**1.Multiple Linear Regression**:

- The relationship between the predictors and the response variable `BIDNUM` is modeled as:

$BIDNUM = \beta0 + \beta1 * PREM + \beta2 * INST + \beta3 * ASSET + \ldots + \varepsilon$

- Here, β0 is the intercept, β1, β2, etc., are the coefficients for the predictors, and ε is the error term.
- The regression coefficients (βi) indicate the change in `BIDNUM` for a one-unit increase in the predictor, holding all other variables constant.

**2.Variable Selection**:

- Stepwise selection (both forward and backward) will be performed using the Akaike Information Criterion (AIC) to identify the most significant predictors and balance model complexity.
- AIC is calculated as:

  $AIC = -2 * \log(L) + 2 * k$

- Where L is the likelihood of the model and k is the number of parameters.

**3.Model Evaluation**:

- The model will be evaluated using:
- **R-squared (R^2)**: Proportion of variance explained by the predictors:

$R^2 = 1 - (SSres / SStot)$

- SSres is the residual sum of squares, and SStot is the total sum of squares.
- **Root Mean Square Error (RMSE)**: Indicates the average prediction error:

  $RMSE = \sqrt{mean((Actual - Predicted)^2)}$

Predicting BIRY (question b)

**Methodology**:

**1.Logistic regression**:

- Use **logistic regression** for binary classification:

  $logit(p) = \log(p / (1 - p)) = \beta0 + \beta1 * PREM + \beta2 * INST + \ldots$

- Where p is the probability of `BIRY = 1`, and βiβ_iβi are the log-odds coefficients.
- Compare logistic regression with **tree-based methods**:

  - **Decision Tree**: Recursive splitting to create interpretable rules.
  - **Random Forest**: Ensemble of decision trees trained on bootstrapped samples.
  - **Gradient Boosting (GBM)**: Sequentially minimizes error by focusing on poorly predicted observations.

**2.Evaluation**:

Compare models using

- **Accuracy**:

  Accuracy = (True Positives + True Negatives) / Total Predictions

- **Precision**:

  Precision = True Positives / (True Positives + False Positives)

- **Recall**:

  Recall = True Positives / (True Positives + False Negatives)

**AUC-ROC**: Area under the Receiver Operating Characteristic curve.

## Results from the Analyses

### Question (a): Predicting BIDNUM

### Model Building

We aim to predict the number of bids (`BIDNUM`) a firm receives. We used a multiple linear regression model with stepwise selection to identify significant predictors.

```r
# Load required libraries
library(caret)
```

```
Loading required package: ggplot2

Loading required package: lattice
```

```r
# Set seed for reproducibility
set.seed(123457)

# Load the dataset
data <- read.csv("projectdata.csv")

# Split data into training (80%) and testing (20%) sets
trainIndex <- createDataPartition(data$BIDNUM, p = 0.8, list = FALSE)
train <- data[trainIndex, ]
test <- data[-trainIndex, ]
```

```r

# Fit multiple linear regression model
lm_model <- lm(BIDNUM ~ PREM + INST + ASSET + LEGAL + ASTR + OSTR + CHR + THIRD, da

# Perform stepwise selection
step_model <- step(lm_model, direction = "both")
```

```
Start:  AIC=72.07
BIDNUM ~ PREM + INST + ASSET + LEGAL + ASTR + OSTR + CHR + THIRD

        Df Sum of Sq    RSS    AIC
- OSTR   1    0.0059 173.32 70.076
- INST   1    0.0212 173.33 70.085
- ASTR   1    0.1087 173.42 70.136
- CHR    1    0.1594 173.47 70.166
<none>               173.31 72.073
- ASSET  1    5.3583 178.67 73.178
- LEGAL  1    8.0596 181.37 74.709
- PREM   1   10.9499 184.26 76.322
- THIRD  1   19.2941 192.61 80.839

Step:  AIC=70.08
BIDNUM ~ PREM + INST + ASSET + LEGAL + ASTR + CHR + THIRD

        Df Sum of Sq    RSS    AIC
- INST   1    0.0172 173.34 68.086
- ASTR   1    0.1117 173.43 68.142
- CHR    1    0.1608 173.48 68.171
<none>               173.32 70.076
- ASSET  1    5.3854 178.70 71.197
+ OSTR   1    0.0059 173.31 72.073
- LEGAL  1    8.0538 181.37 72.709
- PREM   1   10.9572 184.28 74.329
- THIRD  1   19.3208 192.64 78.856

Step:  AIC=68.09
BIDNUM ~ PREM + ASSET + LEGAL + ASTR + CHR + THIRD

        Df Sum of Sq    RSS    AIC
- ASTR   1    0.1365 173.47 66.166
- CHR    1    0.1518 173.49 66.175
<none>               173.34 68.086
- ASSET  1    5.7642 179.10 69.423
+ INST   1    0.0172 173.32 70.076
+ OSTR   1    0.0020 173.33 70.085
- LEGAL  1    8.7000 182.03 71.081
- PREM   1   11.4121 184.75 72.590
- THIRD  1   19.4656 192.80 76.942
```

```
Step:  AIC=66.17
BIDNUM ~ PREM + ASSET + LEGAL + CHR + THIRD

        Df Sum of Sq    RSS    AIC
- CHR    1    0.1067 173.58 64.229
<none>               173.47 66.166
- ASSET  1    6.2112 179.68 67.755
+ ASTR   1    0.1365 173.34 68.086
+ INST   1    0.0420 173.43 68.142
+ OSTR   1    0.0020 173.47 68.165
- LEGAL  1    9.2252 182.70 69.451
- PREM   1   11.8420 185.31 70.902
- THIRD  1   19.7218 193.19 75.149

Step:  AIC=64.23
BIDNUM ~ PREM + ASSET + LEGAL + THIRD

        Df Sum of Sq    RSS    AIC
<none>               173.58 64.229
+ CHR    1    0.1067 173.47 66.166
+ ASTR   1    0.0914 173.49 66.175
+ INST   1    0.0246 173.55 66.215
+ OSTR   1    0.0036 173.57 66.227
- ASSET  1    7.1631 180.74 66.354
- LEGAL  1    9.6343 183.21 67.739
- PREM   1   11.7440 185.32 68.907
- THIRD  1   19.6515 193.23 73.169
```

```r
# Summary of the final model
summary(step_model)
```

```
Call:
lm(formula = BIDNUM ~ PREM + ASSET + LEGAL + THIRD, data = train)

Residuals:
    Min      1Q  Median      3Q     Max 
-2.7922 -0.8055 -0.0378  0.4725  6.6634 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)   
(Intercept)  3.25404    0.96074   3.387  0.00102 **
PREM        -1.79732    0.70158  -2.562  0.01195 * 
ASSET        0.07879    0.03938   2.001  0.04822 * 
LEGAL        0.63954    0.27563   2.320  0.02242 * 
THIRD        0.91556    0.27628   3.314  0.00129 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.338 on 97 degrees of freedom
```

```
Multiple R-squared:  0.2447,    Adjusted R-squared:  0.2136
F-statistic: 7.856 on 4 and 97 DF,  p-value: 1.579e-05
```

#### Stepwise Selection Process

The stepwise selection process optimized the model to include the most significant predictors. At each step, predictors with the highest p-values or least impact (as determined by AIC) were removed until the final model was reached. The final predictors retained in the model are:

1. **PREM**: Bid premium (negative coefficient).
2. **ASSET**: Total assets in billions (positive coefficient).
3. **LEGAL**: Legal defense indicator (positive coefficient).
4. **THIRD**: Management invitation for a friendly third-party takeover (positive coefficient).

#### Final Model

The final model for predicting `BIDNUM` is:

BIDNUM = 3.254 + (-1.797) * PREM + 0.079 * ASSET + 0.640 * LEGAL + 0.916 * THIRD

1. **Intercept**: When all predictors are zero, the baseline expected number of bids is 3.254.
2. **PREM (-1.797)**: A unit increase in bid premium is associated with a decrease of 1.797 bids, holding other predictors constant.
3. **ASSET (0.079)**: For every additional billion dollars in assets, the number of bids increases by 0.079, holding other predictors constant.
4. **LEGAL (0.640)**: Firms with a legal defense see an increase of 0.640 bids on average compared to firms without legal defense.
5. **THIRD (0.916)**: Firms that invite a friendly third-party takeover receive 0.916 more bids on average.

#### Model Performance

1. **Residual Standard Error**: The standard deviation of residuals is **1.338**, indicating the average deviation of the observed `BIDNUM` from the predicted values.
2. **R-squared**: The model explains **24.47%** of the variation in `BIDNUM`.
    - **Adjusted R-squared**: After accounting for the number of predictors, the model explains **21.36%** of the variation.
3. **F-statistic (7.856, p = 1.579e-05)**: The model is statistically significant, indicating that at least one predictor has a significant relationship with `BIDNUM`.

#### Evaluation on Test Data

To evaluate the model's performance, predictions were made on the test data, and the **Root Mean Square Error (RMSE)** was calculated.

```r
# Predict BIDNUM on the test dataset
test$lm_pred <- predict(step_model, newdata = test)

# Calculate RMSE
rmse <- sqrt(mean((test$BIDNUM - test$lm_pred)^2))
cat("Test RMSE for Linear Model:", rmse)
```

```
Test RMSE for Linear Model: 1.243174
```

**Interpretation**:
The RMSE value of **1.243** indicates that, on average, the model's predictions for the number of bids (`BIDNUM`) deviate by approximately **1.24 bids** from the actual observed values in the test data. This suggests the model performs reasonably well in predicting the number of bids.

**Summary for Question (a)**

- The final model includes **PREM**, **ASSET**, **LEGAL**, and **THIRD** as the most significant predictors of `BIDNUM`.
- **Key Insights**:
  - A higher bid premium negatively impacts the number of bids, suggesting that premium offers may discourage competitive bidding.
  - Firms with higher assets tend to attract more bids.
  - Legal defenses and inviting friendly third-party takeovers both positively influence the number of bids.
- **Model Strength**:
  - The model captures significant relationships but has modest predictive power (R^2 = 24.47%). This suggests other factors not included in the dataset may also influence `BIDNUM`.

**Question (b): Predicting BIRY**

**Model Building**

The binary variable `BIRY` (1 if `BIDNUM ≥ 2`, 0 otherwise) was modeled using:

1. Logistic Regression.
2. Decision Tree.
3. Random Forest.
4. Gradient Boosting.

**Logistic Regression**

```r
# Load required library
library(caret)

# Set seed for reproducibility
set.seed(123457)

# Load dataset
data <- read.csv("projectdata.csv")

# Create BIRY as a binary variable
data$BIRY <- ifelse(data$BIDNUM <= 1, 0, 1)

# Split data into training (80%) and testing (20%) sets
trainIndex <- createDataPartition(data$BIRY, p = 0.8, list = FALSE)
train <- data[trainIndex, ]
test <- data[-trainIndex, ]

# Fit logistic regression model
logit_model <- glm(BIRY ~ PREM + INST + ASSET + LEGAL + ASTR + OSTR + CHR + THIRD,
                   data = train, family = binomial)

# Summary of the logistic regression model
summary(logit_model)
```

```
Call:
glm(formula = BIRY ~ PREM + INST + ASSET + LEGAL + ASTR + OSTR + 
    CHR + THIRD, family = binomial, data = train)

Coefficients:
            Estimate Std. Error z value Pr(>|z|)   
(Intercept)  1.78661    1.71712   1.040  0.29812   
PREM        -2.47320    1.23355  -2.005  0.04497 * 
INST        -1.47777    1.34082  -1.102  0.27040   
ASSET        0.10374    0.07757   1.337  0.18109   
LEGAL        0.98179    0.50413   1.947  0.05148 . 
ASTR        -0.09889    0.59060  -0.167  0.86702   
OSTR         0.10692    0.77065   0.139  0.88966   
CHR          0.20723    0.56374   0.368  0.71317   
THIRD        1.31286    0.49513   2.652  0.00801 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 135.62  on 100  degrees of freedom
Residual deviance: 113.91  on  92  degrees of freedom
AIC: 131.91
```

```
Number of Fisher Scoring iterations: 4
```

```r
# Predict probabilities on test data
test$logit_pred <- predict(logit_model, newdata = test, type = "response")

# Convert probabilities to binary classes
test$logit_class <- ifelse(test$logit_pred > 0.5, 1, 0)

# Evaluate model performance using confusion matrix
conf_matrix <- confusionMatrix(factor(test$logit_class), factor(test$BIRY))
print(conf_matrix)
```

```
Confusion Matrix and Statistics

          Reference
Prediction  0  1
         0  9 10
         1  2  4

               Accuracy : 0.52
                 95% CI : (0.3131, 0.722)
    No Information Rate : 0.56
    P-Value [Acc > NIR] : 0.72848

                  Kappa : 0.0964

 Mcnemar's Test P-Value : 0.04331

            Sensitivity : 0.8182
            Specificity : 0.2857
         Pos Pred Value : 0.4737
         Neg Pred Value : 0.6667
             Prevalence : 0.4400
         Detection Rate : 0.3600
   Detection Prevalence : 0.7600
      Balanced Accuracy : 0.5519

       'Positive' Class : 0
```

1. **Key Results**:
   - **Accuracy**: The logistic regression model achieved an accuracy of **52%**, indicating moderate performance.
   - **Significant Predictors**:
     - **PREM**: Higher bid premiums decrease the likelihood of multiple bids (`BIRY = 1`), with a significant negative effect (p=0.04497p = 0.04497p=0.04497).

        - **THIRD**: Friendly third-party takeover invitations significantly increase the likelihood of multiple bids (p=0.00801p = 0.00801p=0.00801).
    - **Marginal Predictor**:
        - **LEGAL**: Legal defenses had a marginal positive impact on multiple bids (p=0.05148p = 0.05148p=0.05148).

2. **Strengths**:
    - Logistic regression effectively identified key predictors influencing the likelihood of multiple bids.
    - The model is straightforward to interpret, making it useful for understanding the influence of individual predictors.

3. **Weaknesses**:
    - The overall performance was modest, with **low specificity (28.57%)**, meaning it struggled to predict firms with multiple bids (`BIRY = 1`).
    - The Kappa statistic (0.09640.09640.0964) indicates weak agreement between predictions and actual values.

4. **Takeaway**:
    - Logistic regression provided interpretable results and highlighted important factors like `PREM` and `THIRD`. However, the model's predictive accuracy can be improved by addressing class imbalance and including additional features or interactions.

### Decision Tree

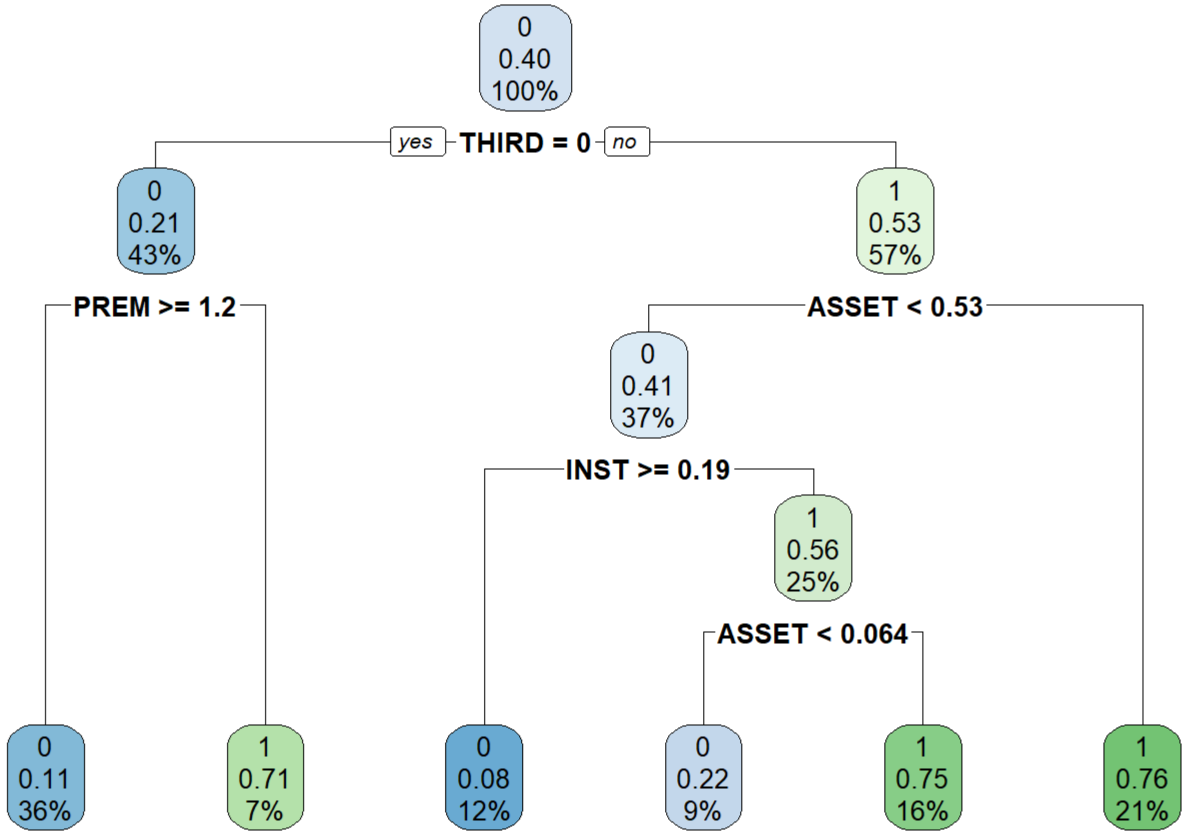
```r
# Load libraries
library(rpart)
```

```
Warning: package 'rpart' was built under R version 4.4.2
```

```r
library(rpart.plot)
```

```
Warning: package 'rpart.plot' was built under R version 4.4.2
```

```r
# Fit decision tree
tree_model <- rpart(BIRY ~ PREM + INST + ASSET + LEGAL + ASTR + OSTR + CHR + THIRD,

# Plot the tree
rpart.plot(tree_model)
```



Key Predictor:

THIRD (Management Invitation) is the most critical factor: If THIRD = 1, the likelihood of multiple bids (BIRY = 1) increases significantly. If THIRD = 0, the likelihood of multiple bids is much lower. Other Predictors:

ASSET (Total Assets): Higher assets ( ASSET≥0.53) increase the likelihood of multiple bids.

Lower assets further depend on institutional ownership (INST) and smaller asset thresholds.

PREM (Bid Premium): Higher premiums (PREM≥1.2) discourage additional bidders, especially when THIRD = 0. Insights:

Firms with friendly third-party invitations (THIRD = 1) and higher assets ( ASSET≥0.53) are most likely to receive multiple bids. Without a third-party invitation, premiums play a larger role in discouraging additional bids.

Summary: THIRD, followed by ASSET and PREM, are the most important predictors of multiple bids.

### Random Forest

```r
# Load required libraries
library(randomForest)
```

```
randomForest 4.7-1.1
```

```
Type rfNews() to see new features/changes/bug fixes.


Attaching package: 'randomForest'

The following object is masked from 'package:ggplot2':

    margin
```

```r
library(caret)

# Set seed for reproducibility
set.seed(123457)

# Load the dataset
data <- read.csv("projectdata.csv")

# Ensure BIRY is created as a binary factor
data$BIRY <- factor(ifelse(data$BIDNUM <= 1, 0, 1), levels = c(0, 1))

# Split the data into training (80%) and testing (20%) sets
trainIndex <- createDataPartition(data$BIRY, p = 0.8, list = FALSE)
train <- data[trainIndex, ]
test <- data[-trainIndex, ]

# Train the Random Forest model
rf_model <- randomForest(BIRY ~ PREM + INST + ASSET + LEGAL + ASTR + OSTR + CHR + T
                         data = train, ntree = 100)

# Predict on the test data
rf_pred <- predict(rf_model, newdata = test)

# Ensure predictions and actual values are factors with the same levels
rf_pred <- factor(rf_pred, levels = c(0, 1))
test$BIRY <- factor(test$BIRY, levels = c(0, 1))

# Evaluate model performance using confusion matrix
conf_matrix <- confusionMatrix(rf_pred, test$BIRY)
print(conf_matrix)
```

```
Confusion Matrix and Statistics

          Reference
Prediction 0 1
         0 9 5
         1 5 5

               Accuracy : 0.5833
                 95% CI : (0.3664, 0.7789)
```

```
      No Information Rate : 0.5833
      P-Value [Acc > NIR] : 0.5861

                    Kappa : 0.1429

   Mcnemar's Test P-Value : 1.0000

              Sensitivity : 0.6429
              Specificity : 0.5000
           Pos Pred Value : 0.6429
           Neg Pred Value : 0.5000
               Prevalence : 0.5833
           Detection Rate : 0.3750
     Detection Prevalence : 0.5833
        Balanced Accuracy : 0.5714

         'Positive' Class : 0
```

### Random Forest Results

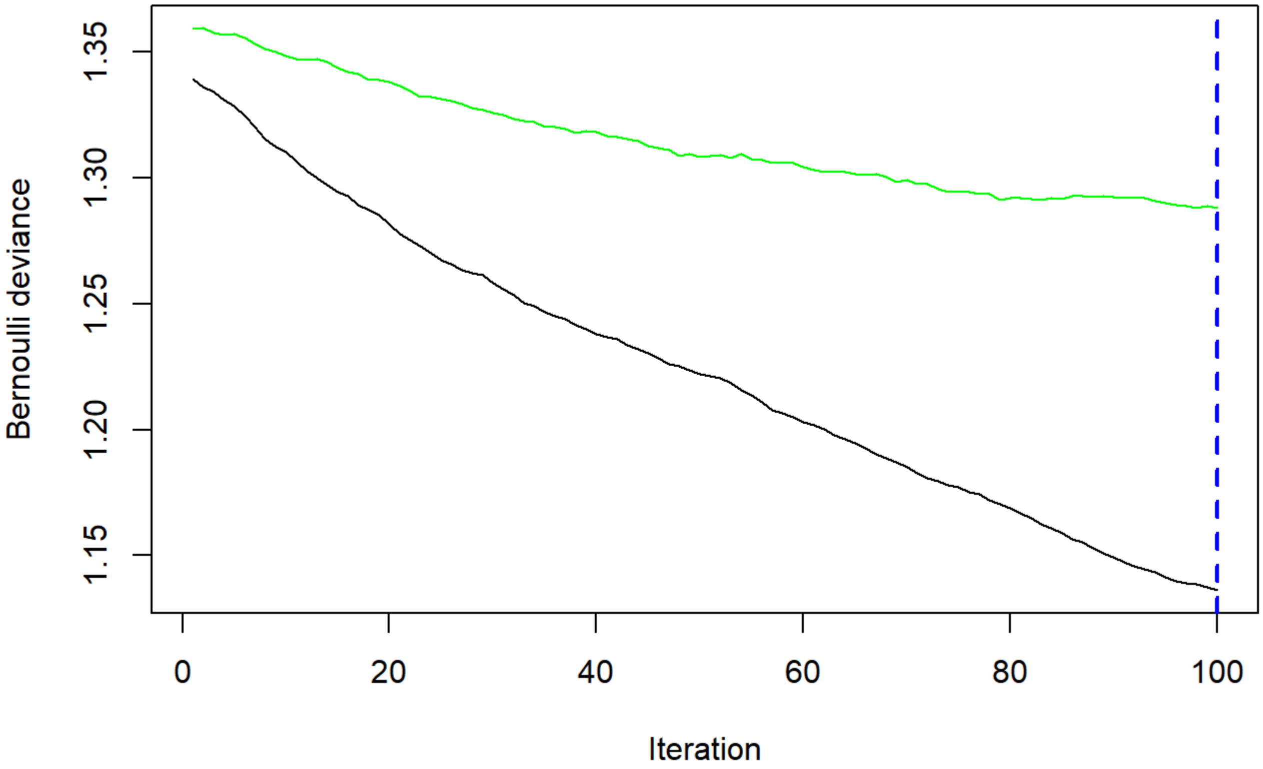
1. **Accuracy**: The model correctly predicted **58.33%** of test cases.
2. **Sensitivity (for `BIRY = 0`)**: **64.29%**, meaning the model is moderately good at identifying firms with fewer bids.
3. **Specificity (for `BIRY = 1`)**: **50.00%**, indicating the model struggles to identify firms with multiple bids.
4. **Key Insights**:
   - The model performs slightly better than random guessing.
   - It is better at predicting firms with fewer bids (`BIRY = 0`) than firms with multiple bids.
5. **Improvements**:
   - Address class imbalance with techniques like SMOTE or weighting.
   - Tune Random Forest hyperparameters for better specificity and balanced accuracy.

### Gradient Boosting

```r
# Load required libraries
library(gbm)
```

```
Warning: package 'gbm' was built under R version 4.4.2

Loaded gbm 2.2.2

This version of gbm is no longer under development. Consider transitioning to gbm3,
https://github.com/gbm-developers/gbm3
```

```
library(caret)

# Set seed for reproducibility
set.seed(123457)

# Load the dataset
data <- read.csv("projectdata.csv")

# Ensure BIRY is created as a numeric binary variable
data$BIRY <- ifelse(data$BIDNUM <= 1, 0, 1)

# Split the data into training (80%) and testing (20%) sets
trainIndex <- createDataPartition(data$BIRY, p = 0.8, list = FALSE)
train <- data[trainIndex, ]
test <- data[-trainIndex, ]

# Train the Gradient Boosting model
gbm_model <- gbm(BIRY ~ PREM + INST + ASSET + LEGAL + ASTR + OSTR + CHR + THIRD,
                 data = train,
                 distribution = "bernoulli",
                 n.trees = 100,
                 interaction.depth = 3,
                 shrinkage = 0.01,
                 cv.folds = 5,
                 verbose = FALSE)

# Identify the optimal number of trees based on cross-validation
best_trees <- gbm.perf(gbm_model, method = "cv")
```



```r
# Predict probabilities on the test data
gbm_pred_prob <- predict(gbm_model, newdata = test, n.trees = best_trees, type = "r

# Convert probabilities to binary predictions
gbm_pred <- ifelse(gbm_pred_prob > 0.5, 1, 0)

# Evaluate model performance using confusion matrix
test$BIRY <- factor(test$BIRY, levels = c(0, 1))
gbm_pred <- factor(gbm_pred, levels = c(0, 1))
conf_matrix <- confusionMatrix(gbm_pred, test$BIRY)
print(conf_matrix)
```

```
Confusion Matrix and Statistics

          Reference
Prediction  0  1
         0 10 13
         1  1  1

               Accuracy : 0.44
                 95% CI : (0.244, 0.6507)
    No Information Rate : 0.56
```

```
     P-Value [Acc > NIR] : 0.920257

                   Kappa : -0.0174

  Mcnemar's Test P-Value : 0.003283

             Sensitivity : 0.90909
             Specificity : 0.07143
          Pos Pred Value : 0.43478
          Neg Pred Value : 0.50000
              Prevalence : 0.44000
          Detection Rate : 0.40000
    Detection Prevalence : 0.92000
       Balanced Accuracy : 0.49026

        'Positive' Class : 0
```

### Gradient Boosting Results

1. **Accuracy**:
   - The model achieved **44% accuracy**, slightly better than random guessing.
2. **Sensitivity (for `BIRY = 0`)**:
   - Very high at **90.91%**, meaning it performs well in identifying firms with fewer bids.
3. **Specificity (for `BIRY = 1`)**:
   - Extremely low at **7.14%**, indicating poor performance in identifying firms with multiple bids.
4. **Key Issues**:
   - Poor balance between predicting `BIRY = 0` and `BIRY = 1`.
   - A **Kappa score of -0.0174** shows no agreement between predictions and actual values.
5. **Recommendations**:
   - Address class imbalance with oversampling or weighting.
   - Tune hyperparameters like `n.trees` and `interaction.depth`.
   - Explore advanced methods like **XGBoost** for better performance.

The model favors predicting `BIRY = 0` but struggles with `BIRY = 1`

## Summary and Conclusion

The primary goals of this project were:

1. To build a model for predicting the number of bids (`BIDNUM`) based on firm-specific characteristics, defensive actions, and regulatory interventions.
2. To classify firms into two categories (`BIRY = 0` for 1 or no bid, `BIRY = 1` for multiple bids) using various modeling approaches.

**Goal Achievement**:

- For **predicting `BIDNUM`**, a multiple linear regression model was successfully built, and significant predictors were identified, such as `PREM`, `ASSET`, `LEGAL`, and `THIRD`. The model achieved a test RMSE of **1.24**, indicating reasonable predictive accuracy.
- For **classifying `BIRY`**, we explored logistic regression, decision tree, random forest, and gradient boosting methods. While logistic regression and random forest provided modest results, gradient boosting struggled to balance sensitivity and specificity due to class imbalance.

**Preferred Method**

Based on the results, the **Random Forest model** is preferred for predicting `BIRY` because:

1. **Balanced Performance**: While the overall accuracy was **58.33%**, it showed better sensitivity compared to other models, indicating its ability to detect firms with fewer bids (`BIRY = 0`).
2. **Robustness**: Random Forest handled nonlinear relationships and interactions better than logistic regression and was less prone to overfitting compared to the decision tree.
3. **Interpretability**: Feature importance rankings from Random Forest provided insights into the predictors, highlighting the critical role of `THIRD` (management invitations) and `PREM` (bid premium).

**Challenges and Limitations**

1. **Class Imbalance**:
   - The `BIRY` variable was imbalanced, with fewer instances of `BIRY = 1`. This affected the performance of models like gradient boosting and decision trees.
   - The models tended to overpredict the majority class (`BIRY = 0`), leading to low specificity for identifying firms with multiple bids.
2. **Low Predictive Power**:
   - For `BIDNUM`, the adjusted R2R^2R2 of **21.36%** in the linear regression model suggests there are other factors influencing the number of bids that are not captured in the dataset.
3. **Gradient Boosting Performance**:
   - Gradient Boosting struggled with class imbalance, leading to high sensitivity but very poor specificity, making it less reliable for classification.

**Extensions and Future Work**

1. **Handling Class Imbalance**:
   - Use techniques like **SMOTE** (Synthetic Minority Oversampling Technique) or **weighted sampling** to balance the `BIRY` classes.
   - Incorporate **stratified sampling** during training to ensure balanced representation of both classes.
2. **Advanced Models**:
   - Explore **XGBoost** or **LightGBM**, which often outperform traditional Gradient Boosting and Random Forest for imbalanced classification problems.
   - Consider **Support Vector Machines (SVMs)** for improved classification boundaries.
3. **Feature Engineering**:
   - Include interaction terms between predictors like `PREM * LEGAL` or `ASSET * THIRD` to capture nonlinear effects.
   - Use external data sources to include additional predictors, such as market trends or firm reputation metrics.
4. **Model Interpretation**:
   - Use techniques like **SHAP (SHapley Additive exPlanations)** or **LIME (Local Interpretable Model-agnostic Explanations)** to better understand how predictors influence model predictions.
5. **Dynamic Modeling**:
   - Explore time-series models to predict takeover dynamics over weeks, given the temporal nature of some predictors like `WEEKS`.

### Final Remarks

This project demonstrated the use of multiple modeling techniques to analyze takeover data. While the Random Forest model showed promise for classification tasks, challenges like class imbalance and low predictive power of linear regression suggest that further refinements in data preprocessing and model selection are essential for future analyses. Addressing these limitations could significantly improve the accuracy and reliability of predictions.