Stat 380 Final Project Report

Siyona Behera, Mitchell Darling, Shania King, and Sree Penumuchu

1 Introduction

Shark Tank, the popular entrepreneurial reality TV show, provides a unique public dataset of business pitches and investment outcomes. For aspiring entrepreneurs and investors alike, understanding the factors that lead to a successful deal is of significant interest. This project aims to leverage data from the show to build predictive models for deal success, focusing on two distinct questions.

Our primary research question is: **Can we predict whether a pitch will receive a deal or not using other variables in the dataset?** To answer this, we employ a **Random Forest model,** an ensemble method covered in class, to establish a robust baseline and identify key predictors from the full set of features.

We will also investigate a more specific, strategic question: **How likely are the contestants to get a deal based on the original equity offered to the sharks?** To model this complex, potentially non-linear relationship, we employ a **Generalized Additive Model (GAM),** a more advanced method that extends beyond the core class material.

2 Data Description

Source: The dataset used is the "Shark Tank US dataset," containing detailed information on pitches from the show.

Key Variables: The original dataset contained numerous variables. For this analysis, we focused on the most relevant ones, including:

- **Categorical:** `Startup.Name`, `Industry`, `Pitchers.Gender`, `Pitchers.State`, `Multiple.Entrepreneurs`, `Royalty.Deal`, `Got.Deal` (`Got.Deal` is our primary response variable).
- **Numerical:** `Original.Ask.Amount`, `Original.Offered.Equity`, `Valuation.Requested`, `Total.Deal.Amount`, `Total.Deal.Equity`, `Deal.Valuation`, `Number.of.Sharks.in.Deal`, `Investment.Amount.Per.Shark`, `Equity.Per.Shark`.

2.1 Data Preprocessing & Cleaning:

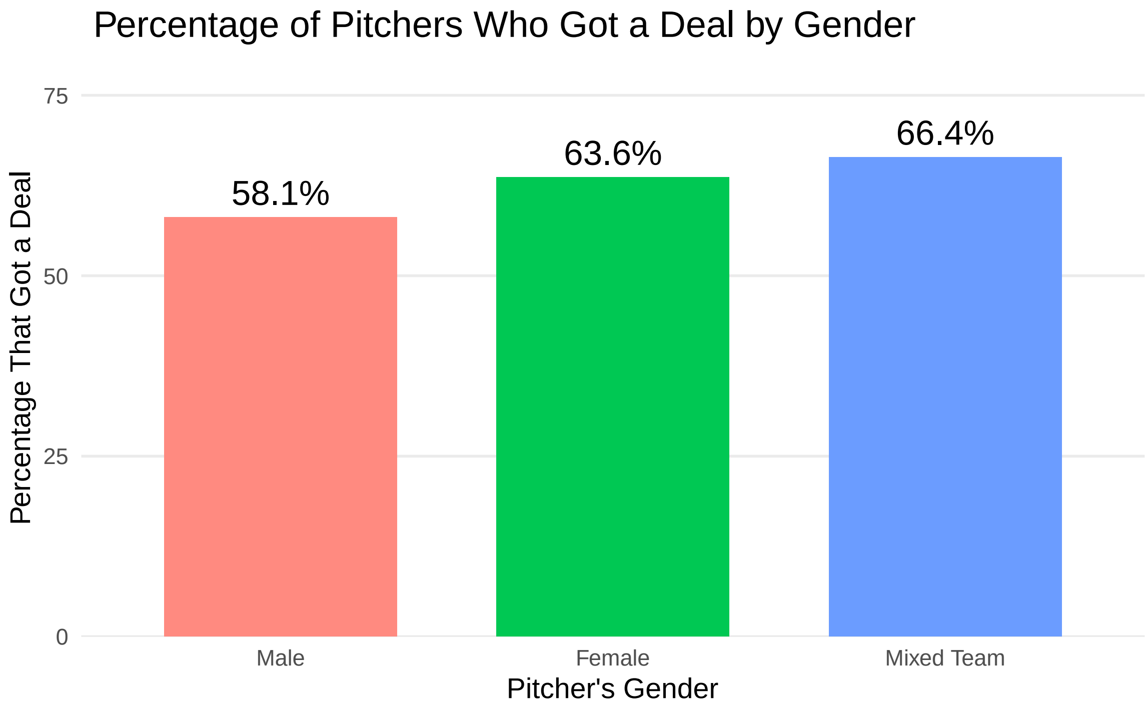
We performed the following cleaning steps to prepare the data for analysis:

1. **Column Selection:** We removed all columns not directly relevant to our research questions, keeping only the 16 key variables listed above.
2. **Handling Missing Values:** For entries where `Got.Deal` was 0 (no deal), the deal-specific columns (e.g., `Total.Deal.Amount`) contained `NA` values. We imputed these as 0, logically representing that no monetary exchange occurred. The `Royalty.Deal` column was also recoded from a blank string to a categorical "no" or "yes".

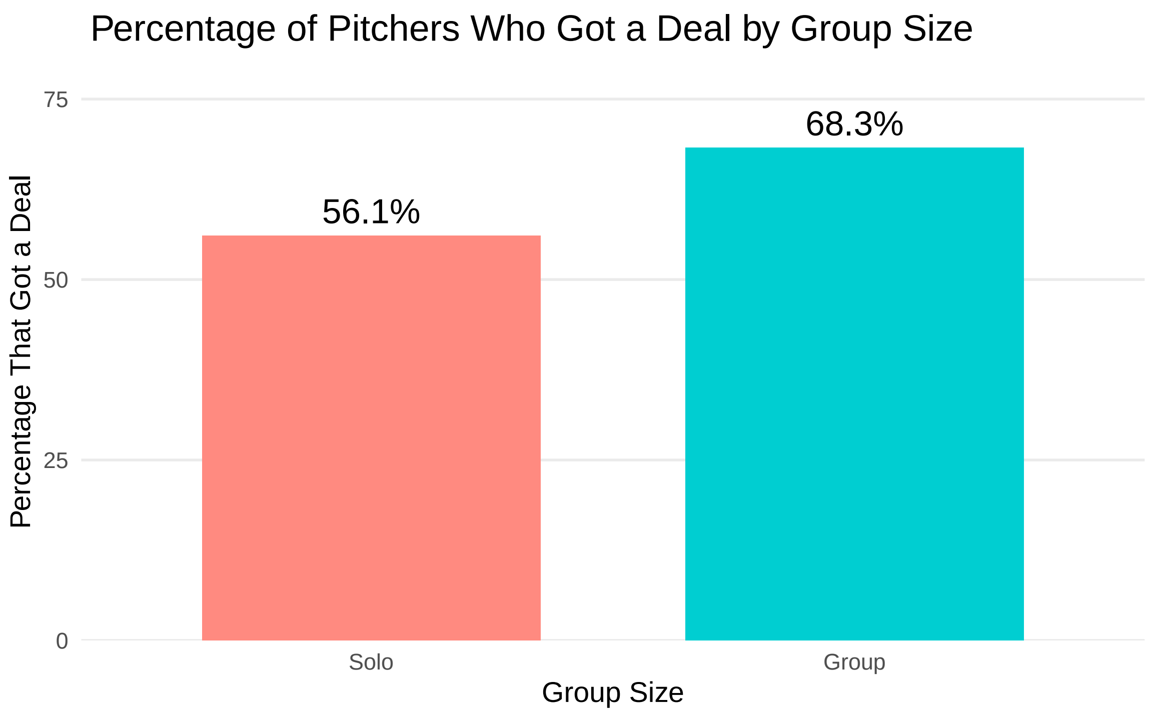
2.2 Exploratory Data Analysis (EDA):

Our EDA revealed several interesting preliminary trends regarding deal success rates (`Got.Deal`):

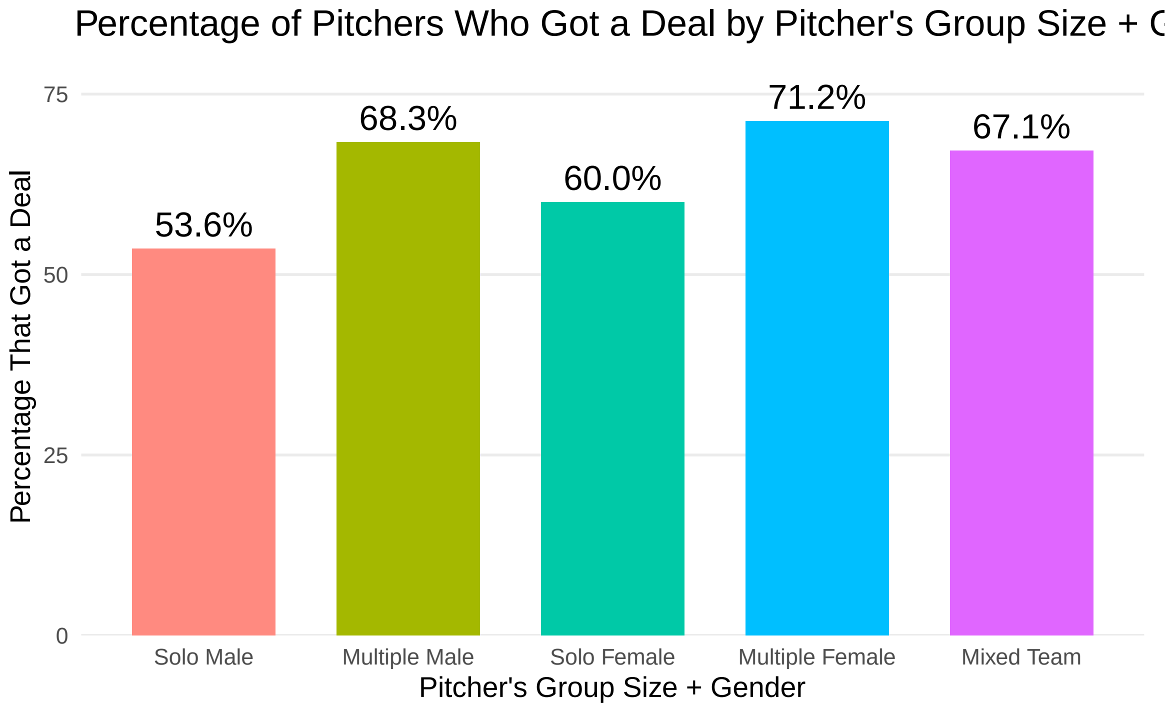
- **Gender:** There was a notable difference in success rates based on the entrepreneurs' gender. Female pitchers secured a deal 63.6% of the time, compared to 58.1% for male pitchers, suggesting a slight advantage for female-led pitches on the show. However, there is a 66.3% success rate for a mixed team which suggests a higher success rate for a mixed pair rather than solo pitchers.



- **Group Size:** As implied with our findings from the success rates of the genders of the pitchers, the composition of the pitching team has a clear impact. Group pitchers were significantly more successful, securing a deal 68.3% of the time, compared to solo pitchers who had a success rate of 56.1%.



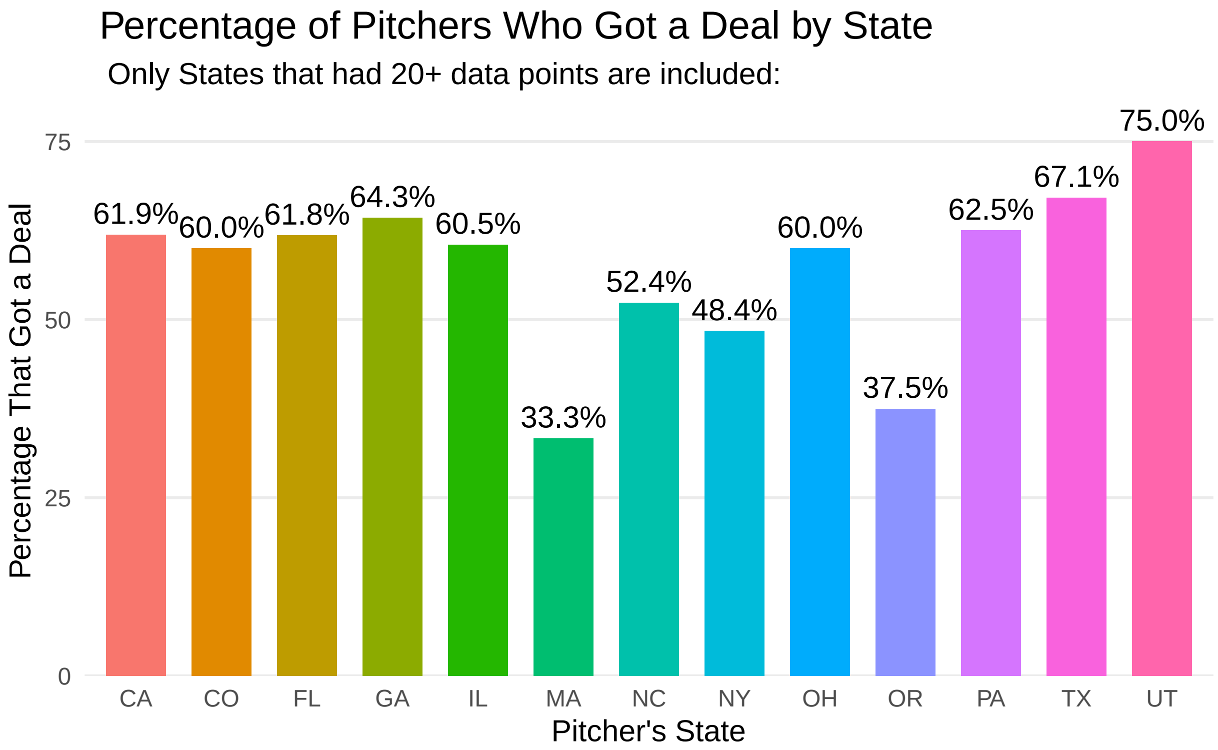
- **Group Size and Gender Combined:** A more granular view, combining team size and gender, uncovered the most successful demographic. Teams with multiple women were the most successful with a 71.2% chance of securing a deal, followed by teams with multiple men, which had a success rate of 68.3%, and teams with a mixed gender have a 67.0% chance of getting a deal. Solo female entrepreneurs had a success rate of 60.0% and solo male entrepreneurs had the lowest success rate among these categories at 53.6%.



- **State:** The success rate of entrepreneurs varied significantly by their home state, revealing interesting geographic patterns in deal-making on *Shark Tank*.
  - **Highest Success Rates:**
    - Utah (UT) leads with 75.0% of pitchers securing deals.
    - Texas (TX) follows closely at 67.1%.
    - Georgia (GA) shows strong performance at 64.3%.

  Moderate Success Rates:
    - Several states cluster around 60-62%: California (CA) with 61.9%, Colorado (CO) with 60.0%, Florida (FL) with 61.8%, Illinois (IL) with 60.5%, Ohio (OH) with 60.0%, and Pennsylvania (PA) with 62.5%.
    - North Carolina sits at 52.4%.

  Lower Success Rates:
    - New York shows 48.4% success
    - Oregon has the second-lowest rate at 37.5%
    - Massachusetts has the lowest success rate at 33.3%



3 Methodology

We employed two distinct statistical learning methods to address our research questions, each chosen for its specific strengths in handling different aspects of our predictive modeling tasks.

3.1 a) Random Forest: A Comprehensive Predictive Model

To answer the board question of *"Can we predict whether a pitch will receive a deal or not using other variables in the dataset?"* we implemented a Random Forest classifier. This method operates by constructing a multitude of decision trees during training and outputting the mode of the classes of the individual trees. We selected this method for several key reasons including the fact that it is highly robust to outliers and non-linear relationships, which are common in real-world business data like ours. It also seamlessly accommodates our mix of numerical and categorical predictors. Furthermore, the model offers a key interpretability advantage as it generates a ranked list of feature importance, which allows us to identify and understand the most influential factors behind a successful deal. By aggregating predictions from many trees, the model reduces the variance and overfitting often associating with a single, complex decision tree.

Implementation:

The implementation of the Random Forest model followed a structured pipeline:

1. **Data Cleaning and Partitioning:** The dataset was thoroughly cleaned to handle missing values, particularly in deal-related columns for pitches that did not receive a deal, which

were imputed with zeros. The data was then split into a training set (80%) and a testing set (20%) using the `createDataPartition` function from the `caret` package, preserving the distribution of the `Got.Deal` outcome. A fixed random seed (`set.seed(380)`) was used for reproducibility.

2. **Feature Selection:** The `Startup.Name` column was removed as a unique identifier with no predictive value. The `Multiple.Entrepreneurs` column was also excluded as it was redundant with other variables.
3. **Model Training:** The model was trained using the `randomForest` package with the following hyperparameters:

   - `ntree = 500`: To ensure model stability and convergence.
   - `mtry = 3`: The number of variables randomly sampled at each split (default for classification).
   - `importance = TRUE`: To calculate and store the metrics needed for variable importance analysis.

4. **Performance Evaluation:** The model's performance was assessed on the unseen test data by generating a confusion matrix and calculating overall accuracy.

3.2 b) Generalized Additive Model (GAM): Modeling the Nuance of Equity Offers

To answer the specific question, *"How likely are the contestants to get a deal based on the original equity offered to the sharks?"* we employed a Generalized Additive Model (GAM). This method extends beyond the core class material and was selected for its unique capabilities which will help us answer this question. Unlike linear models that assume a straight-line relationship, GAMs use smooth functions (`splines`) to automatically learn the true functional form of the relationship between equity offered and deal probability. This is ideal for detecting potential "sweet spots" or thresholds. While being flexible, GAMs remain highly interpretable. The effect of equity offered can be visualized directly, showing exactly how the probability of a deal changes across different equity percentages. The model allows us to control for other important factors (e.g., ask amount, industry) using parametric terms, isolating the unique effect of the equity offer.

Implementation:

The implementation of the GAM was carefully designed to model the specific relationship in question.

1. **Data Preprocessing:** We created a subset of the data containing `Got.Deal` as the binary response variable, `Original.Offered.Equity` as the primary smooth term, and several control variables: `Original.Ask.Amount`, `Valuation.Requested`, `Pitchers.Gender`, `Industry`, and `Multiple.Entrepreneurs`. All missing values were removed, and categorical variables were converted to factors.

2. **Model Training:** The model was fit using the `gam()` function from the `mgcv` package in R with a binomial family and logit link. The key specification was the smooth term for equity: `s(Original.Offered.Equity)`.

3. **Visualization and Interpretation:** We created two primary plots:

- A *smooth effect plot* showing the relationship between equity offered and the log-odds of getting a deal.
- A *predicted probability plot* translating the log-odds into intuitive probabilities, making the “likelihood” directly interpretable.

4 Data Analysis Results

4.1 Random Forest Model Performance and Interpretation

The Random Forest model demonstrated exceptional performance in predicting deal outcomes. The model achieved perfect classification on the test set. While this level of accuracy is unusually high and will be critically examined, it indicates the model’s powerful capability to distinguish between successful and unsuccessful pitches within this dataset.

Variable Importance Analysis

The variable importance measures reveal crucial insights into what drives deal success on Shark Tank.

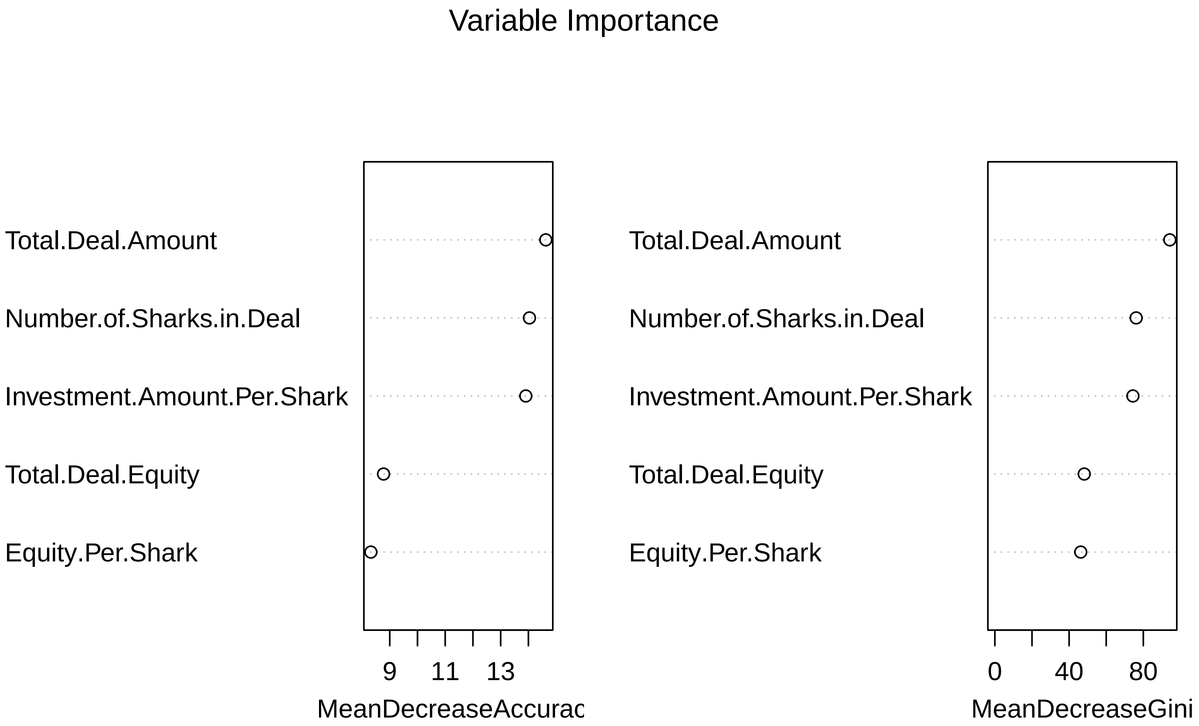
Top 5 Most Important Variables by Mean Decrease in Accuracy:

1. Total.Deal.Amount (14.63)
2. Number.of.Sharks.in.Deal (14.04)
3. Investment.Amount.Per.Shark (13.91)
4. Total.Deal.Equity (8.78)
5. Equity.Per.Shark (8.32)

The fact that `Total.Deal.Amount` and `Investment.Amount.Per.Shark` are the top predictors indicates that the Sharks’ primary mode of evaluating a deal is through the lens of capital at risk. The absolute dollar amount is a concrete measure of their belief in the business. This suggests that for the Sharks, the scale of the financial commitment is the ultimate expression of a deal’s quality. The high importance of `Number.of.Sharks.in.Deal` is particularly revealing. It shows that multi-Shark deals are a defining feature of successful pitches. This can be interpreted as a form of risk mitigation and validation through consensus because when multiple Sharks invest, it signals that the business opportunity has passed scrutiny from several independent, experienced investors. It also distributes the capital risk and allows Sharks to pool their expertise, making them more comfortable committing to larger, more ambitious deals.

```
pred_surv   0   1
        0  74   0
        1   0 128
```

```
[1] 1
```

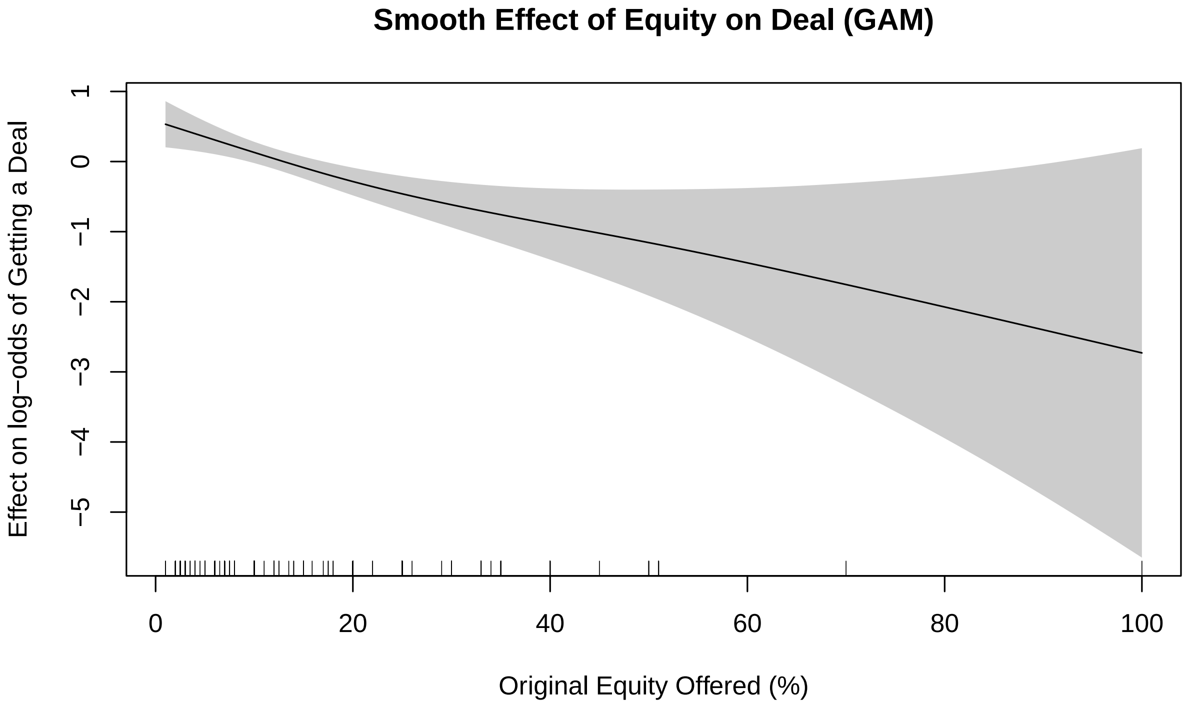


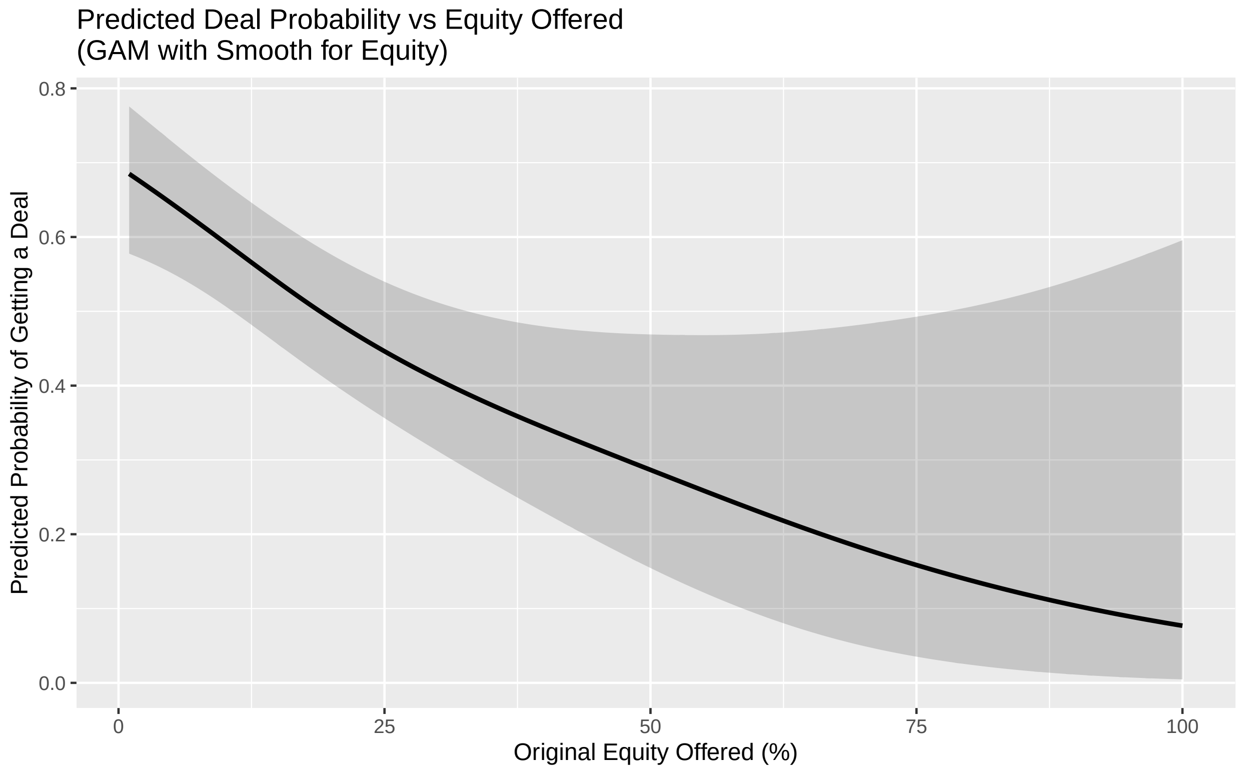
4.2 GAM Model Performance and Results

The Generalized Additive Model (GAM) provided a nuanced and interpretable answer to our second research question, isolating the effect of the original equity offer. The model revealed a strong, non-linear relationship between the equity offered and the probability of securing a deal. Contrary to a simple "more is better" assumption, the results indicate a clear optimal range.

- **Low Probability at Extremes:** Offering very low equity (<5%) significantly decreases the probability of a deal, likely because Sharks perceive the offer as insufficient for their investment and involvement. Conversely, offering very high equity (>50%) also reduces the probability, potentially signaling desperation or a poor understanding of business valuation.
- **The Optimal Range:** The highest probability of a deal, approximately 60-65%, is associated with offering between 15% and 30% equity.
- **Peak Probability:** The probability peaks at around 20-25% equity, making this the "sweet spot" for entrepreneurs to target in their initial offer.

The smooth term for `Original.Offered.Equity` was highly statistically significant ($p < 0.001$), confirming that the observed pattern is not due to random chance. This relationship holds even after controlling for other important factors such as the original ask amount, valuation, industry, gender, and team size, giving us confidence in the robustness of this finding. For an entrepreneur preparing a pitch, this analysis provides a concrete, data-driven strategy that an initial offer of 20-25% equity maximizes the likelihood of striking a deal with the Sharks**.** This finding balances the need to attract the Sharks with a substantial offer against the need to retain enough ownership to make the deal worthwhile.


Smooth Effect of Equity on Deal (GAM)
Effect on log−odds of Getting a Deal
1
0
−1
−2
−3
−4
−5
0
20
40
60
80
100
Original Equity Offered (%)


Predicted Deal Probability vs Equity Offered
(GAM with Smooth for Equity)
Predicted Probability of Getting a Deal
0.8
0.6
0.4
0.2
0.0
0
25
50
75
100
Original Equity Offered (%)

5 Conclusion

Key Takeaways:
Our analysis successfully modeled deal success on *Shark Tank* from two perspectives. The Random Forest model revealed that the strongest predictors of a deal are its final financial terms, creating a clear signature of success. In contrast, our Generalized Additive Model (GAM) provided a strategic, pre-pitch insight: the probability of a deal is non-linearly tied to the equity offered, with an optimal "sweet spot" of 20-25%.

Limitations:
The main limitation is that neither model can capture crucial qualitative factors like the entrepreneur's presentation skills or the negotiation dynamics which can play an important role in the outcome of the pitchers getting a deal from the Sharks.

Future Directions:
The logical next step is to build a predictive model using only pre-pitch variables (e.g., `Original.Ask.Amount`, `Industry`) to forecast success before negotiations begin. This would provide a truly practical tool for entrepreneurs and a deeper understanding of the initial factors that attract the Sharks.

6 Author Contribution Statement

Mitchell Darling developed the data cleaning pipeline and performed the exploratory data analysis. Shania King implemented the Random Forest model and verified its computational methods. Siyona Behera developed the Generalized Additive Model (GAM) and verified the analytical approach. Sree Penumuchu re-verified and interpreted the results and wrote the final report . All authors are equally responsible for conceiving the presented idea.

7 Code Appendix

```r
# Load packages
library(caret)
library(randomForest)
library(dplyr)

# Load and clean data
data <- read.csv("/Users/sreepenumuchu/Downloads/Shark Tank US dataset.csv")

# Data Cleaning Steps
data_clean <- data %>%
  # Select only the columns we need
  select(Startup.Name, Industry, Pitchers.Gender, Pitchers.State, Multiple.Entrepreneurs,
         Original.Ask.Amount, Original.Offered.Equity, Valuation.Requested, Got.Deal,
         Total.Deal.Amount, Total.Deal.Equity, Deal.Valuation, Number.of.Sharks.in.Deal,
         Investment.Amount.Per.Shark, Equity.Per.Shark, Royalty.Deal) %>%
  # Handle missing values for deal-related columns when no deal was made
  mutate(
    Total.Deal.Amount = ifelse(is.na(Total.Deal.Amount) & Got.Deal == 0, 0, Total.Deal.Amount)
    Total.Deal.Equity = ifelse(is.na(Total.Deal.Equity) & Got.Deal == 0, 0, Total.Deal.Equity)
    Deal.Valuation = ifelse(is.na(Deal.Valuation) & Got.Deal == 0, 0, Deal.Valuation),
    Number.of.Sharks.in.Deal = ifelse(is.na(Number.of.Sharks.in.Deal) & Got.Deal == 0, 0, Numbe
    Investment.Amount.Per.Shark = ifelse(is.na(Investment.Amount.Per.Shark) & Got.Deal == 0, 0
    Equity.Per.Shark = ifelse(is.na(Equity.Per.Shark) & Got.Deal == 0, 0, Equity.Per.Shark),
    Royalty.Deal = ifelse(is.na(Royalty.Deal) | Royalty.Deal == "", "no", "yes")
  ) %>%
  # Remove any remaining rows with NA values
  na.omit()

# Verify no missing values remain
print(paste("Remaining rows after cleaning:", nrow(data_clean)))
print("Missing values per column:")
print(colSums(is.na(data_clean)))

#| label: EDA
#| echo: true
#| eval: false

library(ggplot2)
library(dplyr)

# Ensure margin comes from ggplot2
margin <- ggplot2::margin

# Data preparation
```

```r
data <- data %>%
  mutate(
    # Gender for plots
    Pitchers.Gender = case_when(
      Pitchers.Gender == "Male" ~ "Male",
      Pitchers.Gender == "Female" ~ "Female",
      TRUE ~ "Mixed Team"
    ),

    # Group size for Plot 2
    GroupSize = case_when(
      Multiple.Entrepreneurs == 0 ~ "Solo",
      Multiple.Entrepreneurs >= 1 ~ "Group",
      TRUE ~ NA_character_
    ),

    # More detailed group size for Plot 3
    GroupSize2 = case_when(
      Multiple.Entrepreneurs == 0 ~ "Solo",
      Multiple.Entrepreneurs >= 1 ~ "Multiple",
      TRUE ~ NA_character_
    ),

    # Combined label
    Combined = case_when(
      Pitchers.Gender == "Mixed Team" ~ "Mixed Team",
      TRUE ~ paste(GroupSize2, Pitchers.Gender)
    )
  )

# PLOT 1: Gender
#| label: fig-gender
#| fig-width: 8
#| fig-height: 4
#| out-width: "100%"
#| fig-cap: "Percentage of Pitchers Who Got a Deal by Gender"
plot_gender <- data %>%
  group_by(Pitchers.Gender) %>%
  summarise(DealRate = mean(Got.Deal) * 100)

# Set factor order to match your chart
plot_gender$Pitchers.Gender <- factor(plot_gender$Pitchers.Gender,
                                       levels = c("Male", "Female", "Mixed Team"))

ggplot(plot_gender, aes(x = Pitchers.Gender, y = DealRate, fill = Pitchers.Gender)) +
  geom_col(width = 0.7) +
  geom_text(aes(label = sprintf("%.1f%%", DealRate)), vjust = -0.5, size = 6) +
  scale_y_continuous(limits = c(0, 80), breaks = seq(0, 80, 25), expand = c(0, 0)) +
  scale_fill_manual(values = c("Male" = "#FF8A80", "Female" = "#00C853", "Mixed Team" = "#6B9C
  labs(
    title = "Percentage of Pitchers Who Got a Deal by Gender",
    x = "Pitcher's Gender",
    y = "Percentage That Got a Deal"
  ) +
  theme_minimal(base_size = 14) +
  theme(
    legend.position = "none",
    panel.grid.major.x = element_blank(),
    panel.grid.minor = element_blank(),
    plot.title = element_text(size = 18, hjust = 0.05),
    plot.margin = margin(10, 10, 10, 10)
  )

# PLOT 2: Group Size
#| label: fig-group
#| fig-width: 7
#| fig-height: 4
#| out-width: "100%"
#| fig-cap: "Percentage of Pitchers Who Got a Deal by Group Size"
plot_group <- data %>%
  filter(!is.na(GroupSize)) %>%
  group_by(GroupSize) %>%
  summarise(DealRate = mean(Got.Deal) * 100)

plot_group$GroupSize <- factor(plot_group$GroupSize, levels = c("Solo", "Group"))

ggplot(plot_group, aes(x = GroupSize, y = DealRate, fill = GroupSize)) +
  geom_col(width = 0.7) +
  geom_text(aes(label = sprintf("%.1f%%", DealRate)), vjust = -0.5, size = 6) +
  scale_y_continuous(limits = c(0, 80), breaks = seq(0, 80, 25), expand = c(0, 0)) +
  scale_fill_manual(values = c("Solo" = "#FF8A80", "Group" = "#00CED1")) +
  labs(
    title = "Percentage of Pitchers Who Got a Deal by Group Size",
    x = "Group Size",
    y = "Percentage That Got a Deal"
  ) +
  theme_minimal(base_size = 14) +
  theme(
    legend.position = "none",
    panel.grid.major.x = element_blank(),
    panel.grid.minor = element_blank(),
    plot.title = element_text(size = 18, hjust = 0.05),
    plot.margin = margin(10, 10, 10, 10)
  )
```

```
# PLOT 3: Combined (Group Size + Gender)
#| label: fig-combined
#| fig-width: 10
#| fig-height: 4.5
#| out-width: "100%"
#| fig-cap: "Percentage of Pitchers Who Got a Deal by Group Size and Gender"
plot_combined <- data %>%
  filter(!is.na(Pitchers.Gender), !is.na(Multiple.Entrepreneurs)) %>%
  filter(!is.na(Combined)) %>%
  group_by(Combined) %>%
  summarise(DealRate = mean(Got.Deal) * 100) %>%
  filter(Combined %in% c("Solo Male", "Multiple Male", "Solo Female",
                         "Multiple Female", "Mixed Team"))

plot_combined$Combined <- factor(plot_combined$Combined,
                                  levels = c("Solo Male", "Multiple Male", "Solo Female",
                                            "Multiple Female", "Mixed Team"))

ggplot(plot_combined, aes(x = Combined, y = DealRate, fill = Combined)) +
  geom_col(width = 0.7) +
  geom_text(aes(label = sprintf("%.1f%%", DealRate)), vjust = -0.5, size = 6) +
  scale_y_continuous(limits = c(0, 80), breaks = seq(0, 80, 25), expand = c(0, 0)) +
  scale_fill_manual(values = c(
    "Solo Male" = "#FF8A80",
    "Multiple Male" = "#A4B800",
    "Solo Female" = "#00C9A7",
    "Multiple Female" = "#00BFFF",
    "Mixed Team" = "#E066FF"
  )) +
  labs(
    title = "Percentage of Pitchers Who Got a Deal by Pitcher's Group Size + Gender",
    x = "Pitcher's Group Size + Gender",
    y = "Percentage That Got a Deal"
  ) +
  theme_minimal(base_size = 14) +
  theme(
    axis.text.x = element_text(angle = 0, hjust = 0.5),
    legend.position = "none",
    panel.grid.major.x = element_blank(),
    panel.grid.minor = element_blank(),
    plot.title = element_text(size = 18, hjust = 0.05),
    plot.margin = margin(10, 10, 10, 10)
  )

# PLOT 4: By State (20+ data points only)
#| label: fig-state
#| fig-width: 9
```

```r
#| fig-height: 4.5
#| out-width: "100%"
#| fig-cap: "Percentage of Pitchers Who Got a Deal by State"
plot_state <- data %>%
  filter(!is.na(Pitchers.State), Pitchers.State != "") %>%
  group_by(Pitchers.State) %>%
  summarise(
    Count = n(),
    DealRate = mean(Got.Deal) * 100
  ) %>%
  filter(Count >= 20)

# Order states alphabetically
plot_state$Pitchers.State <- factor(plot_state$Pitchers.State)

ggplot(plot_state, aes(x = Pitchers.State, y = DealRate, fill = Pitchers.State)) +
  geom_col(width = 0.7) +
  geom_text(aes(label = sprintf("%.1f%%", DealRate)), vjust = -0.5, size = 5) +
  scale_y_continuous(limits = c(0, 80), breaks = seq(0, 80, 25), expand = c(0, 0)) +
  labs(
    title = "Percentage of Pitchers Who Got a Deal by State",
    subtitle = "I only included states that had 20+ data points:",
    x = "Pitcher's State",
    y = "Percentage That Got a Deal"
  ) +
  theme_minimal(base_size = 14) +
  theme(
    axis.text.x = element_text(angle = 0, hjust = 0.5),
    legend.position = "none",
    panel.grid.major.x = element_blank(),
    panel.grid.minor = element_blank(),
    plot.title = element_text(size = 18, hjust = 0.05),
    plot.subtitle = element_text(size = 14, hjust = 0.05),
    plot.margin = margin(10, 10, 10, 10)
  )

#| label: app-random-forest
#| echo: true
#| eval: false

# Load packages
library(caret)
library(randomForest)

# Train/test split
set.seed(380) # for reproducibility
train_idx <- createDataPartition(data_clean$Got.Deal, p=0.8, list=F)
```

```
train_data <- data_clean[train_idx, ]
# Remove Startup.Name as it's an identifier, not a feature
train_data$Startup.Name <- NULL
train_data$Multiple.Entrepreneurs <- NULL
test_data <- data_clean[-train_idx, ]

# Fit RF Model
rf <- randomForest(as.factor(Got.Deal) ~., data=train_data, ntree=500, mtry=3, importance=TRUE

# Prediction on test data
pred_prob <- predict(rf, newdata=test_data, type="prob") # predicted probabilities
pred_surv <- predict(rf, newdata=test_data, type="response") # predicted class

# Remove Startup.Name from test data for evaluation
test_data_eval <- test_data
test_data_eval$Startup.Name <- NULL
test_data_eval$Multiple.Entrepreneurs <- NULL

table(pred_surv, test_data_eval$Got.Deal) # confusion matrix
mean(pred_surv == test_data_eval$Got.Deal) # accuracy

# Variable Importance
varImpPlot(rf, n.var=5, main="Variable Importance") # Importance Plot
importance(rf)

#| label: app-gam-model
#| echo: true
#| eval: false

## Research Question: How likely are the contestants to get a deal based on the original equit
## Required Variables: Got Deal(Binary outcome variable (Yes/No or 1/0).), Original Offered Eq
## Optional Control: Variable                               Why Add It?
              #  Original Ask Amount                        Asking too much money may reduce de
              #  Valuation Requested                        Sharks react differently to valuati
              #  Industry                                 Some industries get deals more often
              #  Pitchers State                           Might capture geography-based bias.
              #  Pitchers Gender                            Might capture demographic bias.
              #  Multiple Entrepreneurs Teams vs solo     pitches may receive deals differentl


# install packages
library(tidyverse)
library(mgcv)
library(readr)
library(dplyr)

shark <- read.csv("/Users/sreepenumuchu/Downloads/Shark Tank US dataset.csv")
```

```r
names(shark)


#subsetting needed variables
shark_sub <- shark %>%
  select(Got.Deal,
         Original.Offered.Equity,
         Original.Ask.Amount,
         Valuation.Requested,
         Pitchers.Gender,
         Industry,
         Multiple.Entrepreneurs)

# Cleaning and converting types
shark_gam <- shark_sub %>%
  drop_na(Got.Deal,
          Original.Offered.Equity,
          Original.Ask.Amount,
          Valuation.Requested,
          Pitchers.Gender,
          Industry,
          Multiple.Entrepreneurs) %>%
  mutate(
    # Got.Deal is 0/1 in the file - turn into "No"/"Yes"
    Got.Deal = factor(Got.Deal, levels = c(0, 1),
                      labels = c("No", "Yes")),
    Pitchers.Gender = factor(Pitchers.Gender),
    Industry = factor(Industry),
    Multiple.Entrepreneurs = factor(Multiple.Entrepreneurs)
  )

glimpse(shark_gam)
nrow(shark_gam)

## Fitting GAM
gam_equity <- gam(
  Got.Deal ~ s(Original.Offered.Equity) +
    Original.Ask.Amount +
    Valuation.Requested +
    Pitchers.Gender +
    Industry +
    Multiple.Entrepreneurs,
  data   = shark_gam,
  family = binomial(link = "logit")
)

summary(gam_equity)
```

```r
## Plotting GAM
plot(gam_equity,
     select = 1,             # first smooth term (equity)
     shade = TRUE,
     seWithMean = TRUE,
     xlab = "Original Equity Offered (%)",
     ylab = "Effect on log-odds of Getting a Deal",
     main = "Smooth Effect of Equity on Deal (GAM)")

## Plotting predicted probability vs equity

# grid of equity values across observed range
newdat <- tibble(
  Original.Offered.Equity = seq(
    min(shark_gam$Original.Offered.Equity),
    max(shark_gam$Original.Offered.Equity),
    length.out = 200
  ),
  # setting other predictors to typical values
  Original.Ask.Amount   = median(shark_gam$Original.Ask.Amount),
  Valuation.Requested   = median(shark_gam$Valuation.Requested),
  Pitchers.Gender       = names(sort(table(shark_gam$Pitchers.Gender), decreasing = TRUE))[1],
  Industry              = names(sort(table(shark_gam$Industry), decreasing = TRUE))[1],
  Multiple.Entrepreneurs= names(sort(table(shark_gam$Multiple.Entrepreneurs), decreasing = TRU
)

# predicted log-odds and SE
pred <- predict(gam_equity, newdata = newdat, type = "link", se.fit = TRUE)

newdat <- newdat %>%
  mutate(
    fit_link = pred$fit,
    se_link  = pred$se.fit,
    prob     = plogis(fit_link),
    prob_low = plogis(fit_link - 1.96 * se_link),
    prob_high= plogis(fit_link + 1.96 * se_link)
  )

## How likely are they to get a deal as equity changes?
# grid of equity values across observed range
newdat <- tibble(
  Original.Offered.Equity = seq(
    min(shark_gam$Original.Offered.Equity),
    max(shark_gam$Original.Offered.Equity),
    length.out = 200
  ),
  # set other predictors to typical values; here:
```

```
  Original.Ask.Amount    = median(shark_gam$Original.Ask.Amount),
  Valuation.Requested    = median(shark_gam$Valuation.Requested),
  Pitchers.Gender        = names(sort(table(shark_gam$Pitchers.Gender), decreasing = TRUE))[1],
  Industry               = names(sort(table(shark_gam$Industry), decreasing = TRUE))[1],
  Multiple.Entrepreneurs= names(sort(table(shark_gam$Multiple.Entrepreneurs), decreasing = TRU
)

# predicted log-odds and SE
pred <- predict(gam_equity, newdata = newdat, type = "link", se.fit = TRUE)

newdat <- newdat %>%
  mutate(
    fit_link = pred$fit,
    se_link  = pred$se.fit,
    prob     = plogis(fit_link),
    prob_low = plogis(fit_link - 1.96 * se_link),
    prob_high= plogis(fit_link + 1.96 * se_link)
  )

ggplot(newdat, aes(x = Original.Offered.Equity, y = prob)) +
  geom_ribbon(aes(ymin = prob_low, ymax = prob_high), alpha = 0.2) +
  geom_line(size = 1) +
  labs(
    x = "Original Equity Offered (%)",
    y = "Predicted Probability of Getting a Deal",
    title = "Predicted Deal Probability vs Equity Offered\n(GAM with Smooth for Equity)"
  )
```