# Final Report: Machine Learning Model for Predicting Startup Success

## 1. Research Question

How do a startup's location, founding age, and early funding rounds influence its likelihood of securing late-stage funding or achieving acquisition?

## 2. Background and Literature Review

Understanding startup success has been a crucial area of research in entrepreneurship and business analytics. Venture capital (VC) investment plays a critical role in shaping the success and strategic direction of startups. Existing research Ahluwalia and Kassicieh (2021), Gompers et al. (2020), Prado and Bauer (2022) has explored how financial clusters, VC decision-making, and acquisitions by major technology firms influence startup performance and innovation. Ahluwalia and Kassicieh (2021) investigate the impact of financial clusters on startup exit success through mergers and acquisitions (M&A), their findings suggest that startups located in financial hubs with strong VC presence are more likely to achieve successful exits through M&A. Gompers et al. (2020) conducted a study in 2020 to investgate how VC's decision making process has been influenced in selecting prospective startups.These findings suggest that human capital is a primary determinant of startup funding success. Prado and Bauer (2022) challenged the conventional idea that large tech companies creating "kill zones" that stifle competition. In contrast, they pointed out that acquisitions of startups by major tech firms, particularly in the U.S. and Europe, can trigger temporary investment spikes in related industry sectors. Their findings suggest the significance of M&A activity, which can serve as a singnal for the market and more generally, the economy.

These studies collectively highlight the critical role of VC funding and acquisitions in shaping startup success. Based on these literatures, this project analyzes venture capital investment trends using a comprehensive dataset of startup companies worldwide. The dataset includes details such as total funding received, number of funding rounds, investment amounts per round, company locations, industries, founding dates, and acquisition status. The project will take the current theory in the literature as hypothesis, and see if our data could demonstrate these theories, or they present a different picture. The project aims to estimate the correlation between funding levels, geographic locations, and the survival of startup companies by exploring patterns, identifying key factors, and applying predictive modeling.

## 3. Significance of the study

Accurate prediction of startup success can aid investors, policymakers, and entrepreneurs in making informed decisions. By identifying key drivers, this research provides actionable insights into optimizing resource allocation and investment strategies. Furthermore, examining geographic clustering in AI-intensive areas can help understand whether technical advancements are centralized and how location influences startup growth.

## 4. Description of Data

- Data Source: https://www.kaggle.com/datasets/arindam235/startup-investments-crunchbase
- The dataset is sourced from Crunchbase on Kaggle, which tracks startup funding, acquisitions, and geographic locations. It includes 54,294 entries and 39 variables, providing a rich dataset to explore startup funding patterns, locations and acquisition outcomes.
- Key Variables: Dependent Variable (Outcome): Acquisition Status (status): Binary variable indicating if a startup has been acquired (1) or is still operating/closed (0).
- Independent Variables (Predictors): funding_total_usd: Total funding received by the startup. funding_rounds: Number of funding rounds. seed, venture, angel, etc.: Specific funding types. round_A, round_B, ..., round_H: Funding stages. Geographic Location.

## 5. Data Limitations and Biases

- **Imbalance Issue:** More startups tend to fail than succeed, leading to class imbalance.
- **Survivorship Bias:** The dataset primarily includes funded startups, excluding unsuccessful startups that never received funding.
- **Feature Availability:** Some categorical and financial data are missing or incomplete, requiring preprocessing.
- **Geographic Focus:** The dataset may not cover all startup ecosystems equally, potentially skewing clustering insights.

## 6. Feature Engineering

- Converted categorical variables into numerical features (one-hot encoding, frequency encoding).
- Standardized numeric variables for models sensitive to scale (Lasso, XGBoost).
- Created a new feature: funding_round_density = funding_total_usd / funding_rounds.
- Handled missing values using imputation and removal of unusable colunms

```r
library(tidyverse)
```

```
-- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
v dplyr     1.1.4     v readr     2.1.5
v forcats   1.0.0     v stringr   1.5.1
v ggplot2   3.5.1     v tibble    3.2.1
v lubridate 1.9.4     v tidyr     1.3.1
v purrr     1.0.2
-- Conflicts ------------------------------------------ tidyverse_conflicts() --
x dplyr::filter() masks stats::filter()
x dplyr::lag()    masks stats::lag()
i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to becom
```

```r
library(data.table)
```

```

Attaching package: 'data.table'

The following objects are masked from 'package:lubridate':

    hour, isoweek, mday, minute, month, quarter, second, wday, week,
    yday, year

The following objects are masked from 'package:dplyr':

    between, first, last

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

    transpose
```

```r
library(caret)
```

```
Loading required package: lattice

Attaching package: 'caret'

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

    lift
```

```r
library(glmnet)
```

```
Loading required package: Matrix

Attaching package: 'Matrix'

The following objects are masked from 'package:tidyr':

    expand, pack, unpack

Loaded glmnet 4.1-8
```

```r
library(corrplot)
```

```
corrplot 0.95 loaded
```

```r
library(dendextend)
```

```

---------------------
Welcome to dendextend version 1.19.0
Type citation('dendextend') for how to cite the package.

Type browseVignettes(package = 'dendextend') for the package vignette.
The github page is: https://github.com/talgalili/dendextend/

Suggestions and bug-reports can be submitted at: https://github.com/talgalili/dendextend/issu
You may ask questions at stackoverflow, use the r and dendextend tags:
     https://stackoverflow.com/questions/tagged/dendextend
```

```
    To suppress this message use:  suppressPackageStartupMessages(library(dendextend))
---------------------


Attaching package: 'dendextend'

The following object is masked from 'package:data.table':

    set

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

    cutree
```

```r
library(doParallel)
```

```
Loading required package: foreach

Attaching package: 'foreach'

The following objects are masked from 'package:purrr':

    accumulate, when

Loading required package: iterators
Loading required package: parallel
```

```r
file_path <- "/Users/wangbaihui/investments_data.csv"
df <- fread(file_path, encoding = "UTF-8", stringsAsFactors = FALSE)

setnames(df, gsub("[^A-Za-z0-9_]", "", names(df)))

df$funding_total_usd <- as.numeric(gsub(",", "", gsub("-", "0", df$funding_total_usd)))

missing_threshold <- 0.3 * nrow(df)
df_cleaned <- df[, colSums(is.na(df)) < missing_threshold, with = FALSE]

df_cleaned <- na.omit(df_cleaned)

df_cleaned <- df_cleaned %>%
  mutate(
    startup_age = 2025 - founded_year,
    startup_category = ifelse(founded_year < 2010, "Older", "Newer"),
    funding_growth_rate = ifelse(funding_rounds > 1, (funding_total_usd / funding_rounds), fu
    funding_round_density = funding_total_usd / funding_rounds,
    is_vc_funded = ifelse(round_A > 0 | round_B > 0 | round_C > 0, 1, 0)
  )

df_balanced <- df_cleaned %>%
  group_by(status) %>%
  sample_n(size = min(table(df_cleaned$status)), replace = TRUE) %>%
  ungroup()

summary(df_balanced)
```

```
  permalink             name           homepage_url       category_list     
 Length:3672        Length:3672        Length:3672        Length:3672       
 Class :character   Class :character   Class :character   Class :character  
 Mode  :character   Mode  :character   Mode  :character   Mode  :character  



    market          funding_total_usd       status          country_code      
 Length:3672        Min.   :0.000e+00   Length:3672        Length:3672       
 Class :character   1st Qu.:6.262e+04   Class :character   Class :character  
 Mode  :character   Median :1.268e+06   Mode  :character   Mode  :character  
                    Mean   :1.199e+07                                        
                    3rd Qu.:8.000e+06                                        
                    Max.   :2.600e+09                                        
  state_code           region              city           funding_rounds  
 Length:3672        Length:3672        Length:3672        Min.   : 1.000  
 Class :character   Class :character   Class :character   1st Qu.: 1.000  
 Mode  :character   Mode  :character   Mode  :character   Median : 1.000  
                                                          Mean   : 1.732  
                                                          3rd Qu.: 2.000  
                                                          Max.   :12.000  
   founded_at         founded_month      founded_quarter     founded_year 
 Min.   :1928-01-01   Length:3672        Length:3672        Min.   :1928  
 1st Qu.:2004-01-01   Class :character   Class :character   1st Qu.:2004  
 Median :2008-08-06   Mode  :character   Mode  :character   Median :2008  
 Mean   :2006-07-10                                         Mean   :2006  
 3rd Qu.:2011-01-10                                         3rd Qu.:2011  
```

```
Max.   :2014-10-26                                         Max.   :2014
first_funding_at     last_funding_at           seed
Min.   :0011-11-14   Min.   :1960-01-01   Min.   :       0
1st Qu.:2008-01-01   1st Qu.:2009-06-03   1st Qu.:       0
Median :2010-10-04   Median :2011-10-21   Median :       0
Mean   :2009-06-09   Mean   :2010-11-15   Mean   :  197251
3rd Qu.:2012-09-01   3rd Qu.:2013-07-01   3rd Qu.:   35000
Max.   :2014-11-25   Max.   :2014-12-02   Max.   :11500000
   venture          equity_crowdfunding  undisclosed       convertible_note
Min.   :        0   Min.   :      0     Min.   :       0   Min.   :       0
1st Qu.:        0   1st Qu.:      0     1st Qu.:       0   1st Qu.:       0
Median :        0   Median :      0     Median :       0   Median :       0
Mean   :  7506371   Mean   :   1248     Mean   :   79284   Mean   :   19320
3rd Qu.:  6000000   3rd Qu.:      0     3rd Qu.:       0   3rd Qu.:       0
Max.   :775000000   Max.   :1300000     Max.   :30000000   Max.   :13520145
debt_financing          angel             grant          private_equity
Min.   :        0   Min.   :      0   Min.   :0.00e+00   Min.   :         0
1st Qu.:        0   1st Qu.:      0   1st Qu.:0.00e+00   1st Qu.:         0
Median :        0   Median :      0   Median :0.00e+00   Median :         0
Mean   :   903547   Mean   :  60806   Mean   :1.06e+05   Mean   :   2212677
3rd Qu.:        0   3rd Qu.:      0   3rd Qu.:0.00e+00   3rd Qu.:         0
Max.   :505000000   Max.   :9000000   Max.   :1.91e+08   Max.   :2600000000
post_ipo_equity     post_ipo_debt       secondary_market  product_crowdfunding
Min.   :        0   Min.   :        0   Min.   :      0   Min.   :      0.0
1st Qu.:        0   1st Qu.:        0   1st Qu.:      0   1st Qu.:      0.0
Median :        0   Median :        0   Median :      0   Median :      0.0
Mean   :   440615   Mean   :   454520   Mean   :   3280   Mean   :    656.6
3rd Qu.:        0   3rd Qu.:        0   3rd Qu.:      0   3rd Qu.:      0.0
Max.   :645496464   Max.   :500000000   Max.   :7343789   Max.   :1400000.0
   round_A             round_B             round_C
Min.   :        0   Min.   :        0   Min.   :        0
1st Qu.:        0   1st Qu.:        0   1st Qu.:        0
Median :        0   Median :        0   Median :        0
Mean   :  1351252   Mean   :  1818619   Mean   :  1319349
3rd Qu.:        0   3rd Qu.:        0   3rd Qu.:        0
Max.   :225000000   Max.   :350000000   Max.   :200000000
   round_D             round_E            round_F             round_G
Min.   :        0   Min.   :       0   Min.   :        0   Min.   :       0
1st Qu.:        0   1st Qu.:       0   1st Qu.:        0   1st Qu.:       0
Median :        0   Median :       0   Median :        0   Median :       0
Mean   :   643327   Mean   :  238514   Mean   :   131454   Mean   :   19880
3rd Qu.:        0   3rd Qu.:       0   3rd Qu.:        0   3rd Qu.:       0
Max.   :132000000   Max.   :85000000   Max.   :120000000   Max.   :63000000
```

```
    round_H    startup_age     startup_category    funding_growth_rate
 Min.   :0   Min.   :11.00   Length:3672        Min.   :0.000e+00
 1st Qu.:0   1st Qu.:14.00   Class :character   1st Qu.:5.339e+04
 Median :0   Median :17.00   Mode  :character   Median :9.800e+05
 Mean   :0   Mean   :18.69                      Mean   :6.606e+06
 3rd Qu.:0   3rd Qu.:21.00                      3rd Qu.:5.000e+06
 Max.   :0   Max.   :97.00                      Max.   :2.600e+09
 funding_round_density  is_vc_funded
 Min.   :0.000e+00     Min.   :0.0000
 1st Qu.:5.339e+04     1st Qu.:0.0000
 Median :9.800e+05     Median :0.0000
 Mean   :6.606e+06     Mean   :0.3058
 3rd Qu.:5.000e+06     3rd Qu.:1.0000
 Max.   :2.600e+09     Max.   :1.0000
```

## 7. Exploratory Data Analysis (EDA)

1. Funding Total & Rounds:

   The median total funding received by startups is $1,000,000, with a mean of ~$1.39M.

   Some startups have raised significantly higher amounts, with a maximum funding amount of $3 billion.

   Our data suggested that most startups do not raise multiple rounds because the number of funding rounds varies with a median of 1 and a mean of 1.79 rounds.

2. Startup Age:

   The oldest startup in the dataset was founded in 1902, while the youngest was founded in 2014. Overall, the median founding year is 2010, meaning most startups in this dataset were established relatively recently.

   Given this range, an interesting direction would be analyzing whether older startups have a higher survival rate or receive more funding than startups that have been founded relative recently(2010~).

3. Funding amount:

   Many startups received seed funding, but their amounts vary widely. The Seed Funding amount: Median = 0, Mean = $244,764, Max = $10M.

   Equity Crowdfunding is less common funding type among start ups, with Median = 0, Mean = $6,962, Max = $25M, indicating that only a few companies raising significant amounts in this round.

   Comaring to Equity Crowdfounding, Angel Investments are more common among start up companies. Some startups receive substantial angel funding: Median = 0, Mean = $24,218, Max = $43.9M

   Some companies receive large grants, but most do not: Grants: Median = 0, Mean = $134,879, Max = $47.7M

4. Survival Analysis:

   Calculated survival rates for “Older” vs. “Newer” startups (based on “operating” status). Found that newer startups have a higher survival rate compared to older startups. This could be due to improved market conditions, better funding opportunities, or increased startup support ecosystems in recent years. The boxplot confirms that newer startups have greater variation in funding, while older startups have a more stable funding range. Some extreme values (outliers) are present in both categories. Most startups are less than 25 years old and there are very few startups older than 50 years, which indicates that the startup ecosystem has boomed in the past two decades, with a higher concentration of startups founded recently.

   However, oder startups generally have a higher median funding amount than newer startups.The spread of funding (the middle 50% of values) appears slightly larger for older startups compared to newer ones. Both categories have significant outliers, particularly on the higher end, indicating that some startups receive exceptionally high funding.

5. Correlation Heatmap:

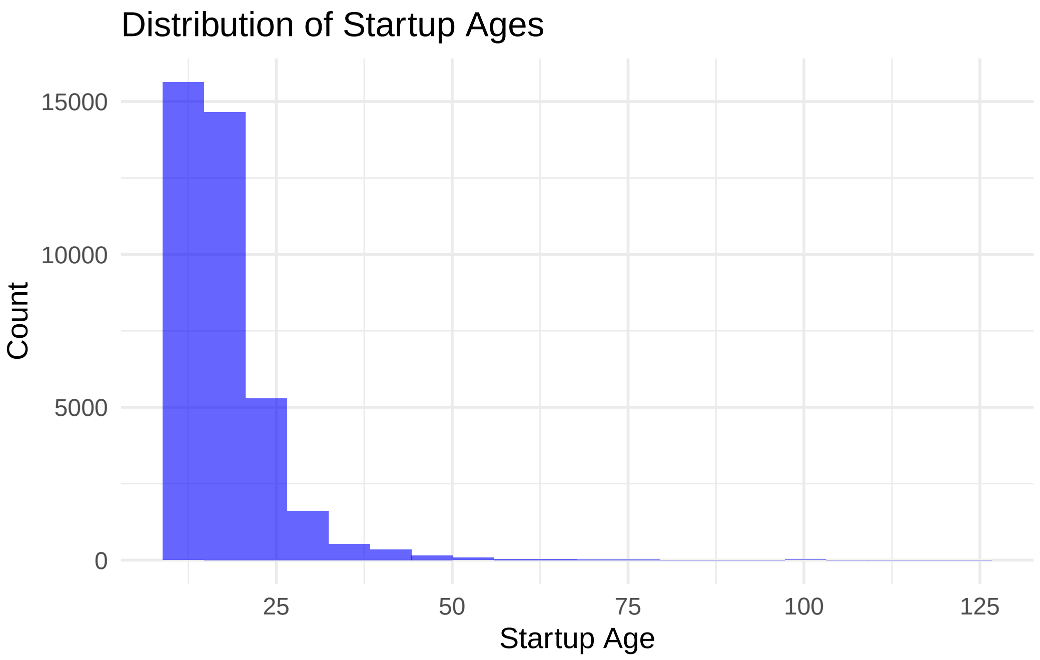
   The heatmap suggests that startups with more funding rounds accumulate higher total funding. VC-backed startups tend to raise more capital. Some later-stage funding rounds may be negatively correlated with startup age, possibly indicating that newer startups are receiving more late-stage investments. implying that other external factors, such as geographic locations may influence funding success beyond these numerical relationships.

6. Funding Type Distribution:

   Venture capital funding has the highest total funding amount among all types, significantly outpacing others. Seed funding also has a considerable amount, though much lower than venture funding. Angel, convertible notes, grants, and private equity have much smaller total funding compared to venture and seed funding. The funding types follow a general progression where **early-stage investments (angel, seed)** receive lower funding compared to **growth-stage funding (venture, private equity).**

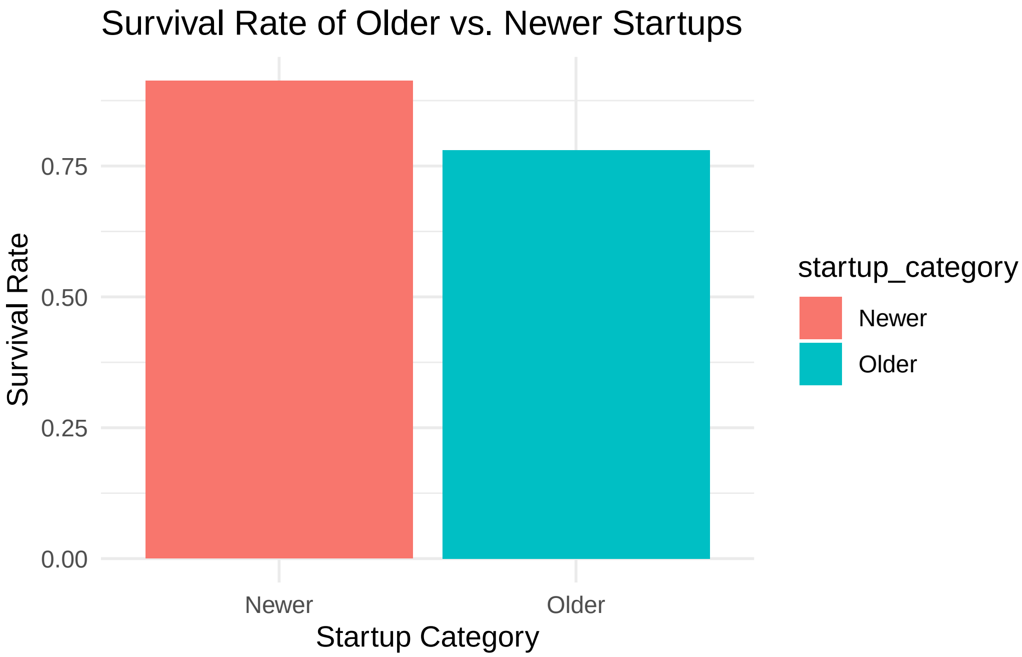
### Distribution of startup ages

```r
ggplot(df_cleaned, aes(x = startup_age)) +
  geom_histogram(fill = "blue", bins = 20, alpha = 0.6) +
  labs(title = "Distribution of Startup Ages", x = "Startup Age", y = "Count") +
  theme_minimal()
```



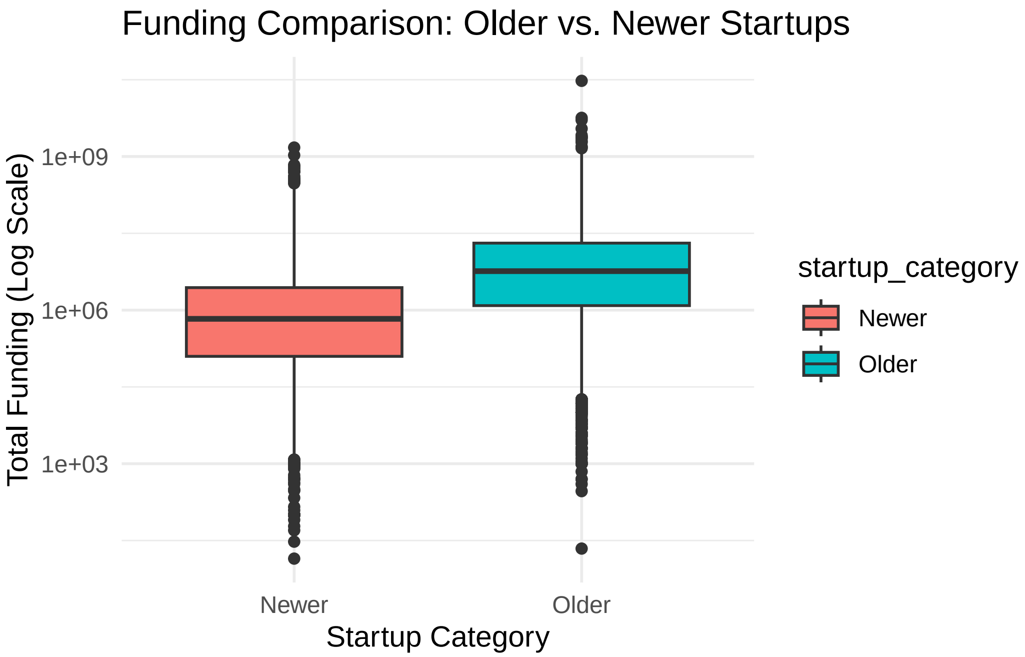
### Survival Rate Analysis

```r
df_survival <- df_cleaned %>%
  group_by(startup_category) %>%
  summarise(
    total_startups = n(),
    active_startups = sum(status == "operating", na.rm = TRUE),
    survival_rate = active_startups / total_startups
  )

ggplot(df_survival, aes(x = startup_category, y = survival_rate, fill = startup_category)) +
  geom_col() +
  labs(title = "Survival Rate of Older vs. Newer Startups", x = "Startup Category", y = "Sur
  theme_minimal()
```



### Funding Distribution by Startup Category

```
ggplot(df_cleaned %>% filter(funding_total_usd > 0), aes(x = startup_category, y = funding_to
  geom_boxplot() +
  scale_y_log10() +
  labs(title = "Funding Comparison: Older vs. Newer Startups", x = "Startup Category", y = "
  theme_minimal()
```

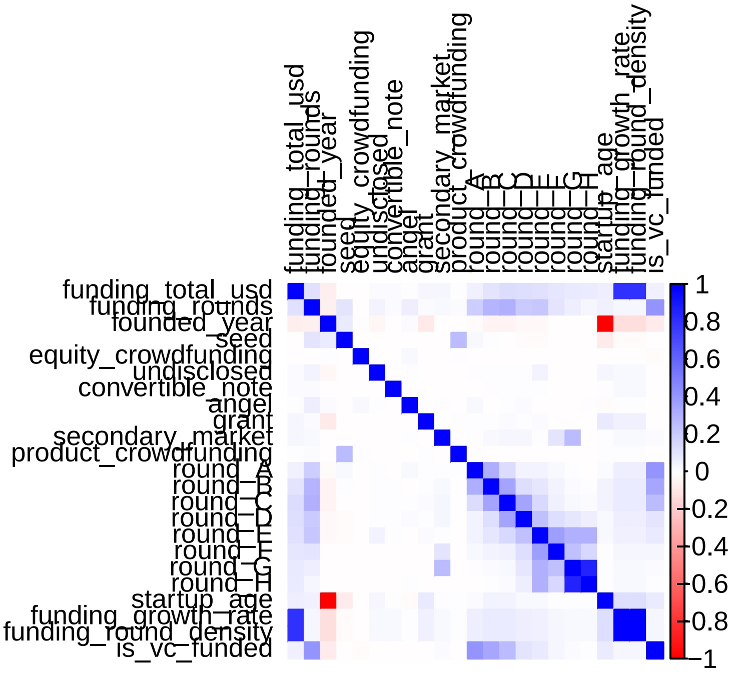


### Correlation Matrix for Numeric Features

```
num_vars <- df_cleaned %>% select(where(is.numeric))

num_vars <- num_vars %>% select_if(~ var(., na.rm = TRUE) > 0)

num_vars <- num_vars %>% filter_all(all_vars(is.finite(.)))

cor_matrix <- cor(num_vars, use = "pairwise.complete.obs")

library(corrplot)

corrplot(cor_matrix,
         method = "color",
         tl.cex = 0.8,
         tl.col = "black",
         col = colorRampPalette(c("red", "white", "blue"))(200),
         cl.limits = c(-1, 1))
```

```
Warning in text.default(pos.xlabel[, 1], pos.xlabel[, 2], newcolnames, srt =
tl.srt, : "cl.limits" is not a graphical parameter

Warning in text.default(pos.ylabel[, 1], pos.ylabel[, 2], newrownames, col =
tl.col, : "cl.limits" is not a graphical parameter
```

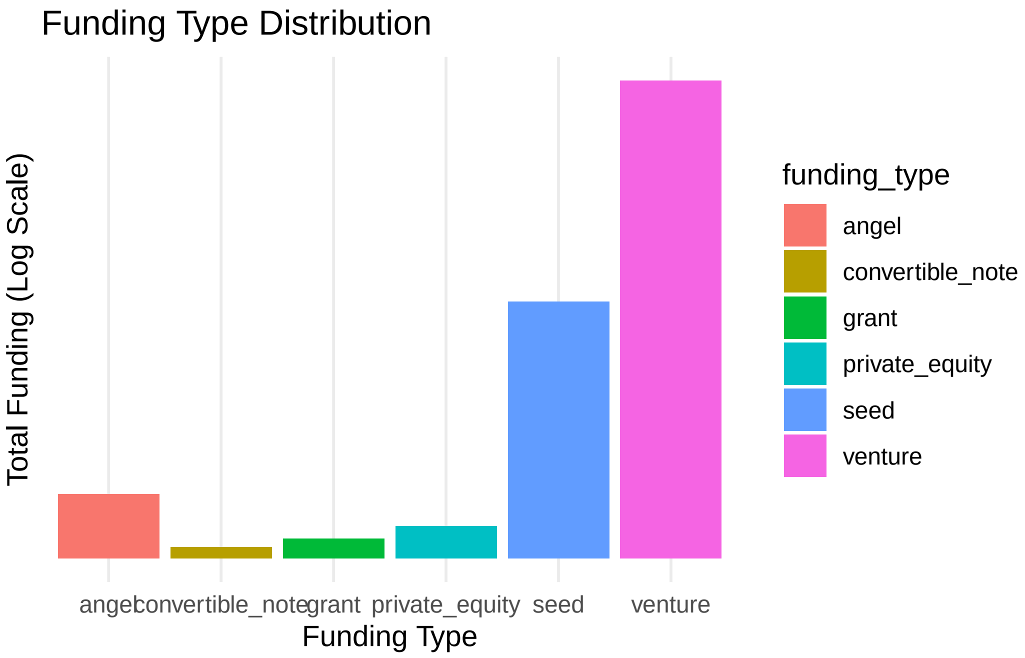
```
Warning in title(title, ...): "cl.limits" is not a graphical parameter
```



```r
funding_types <- df_cleaned %>%
  select(seed, venture, angel, grant, convertible_note, private_equity) %>%
  pivot_longer(everything(), names_to = "funding_type", values_to = "amount")

ggplot(funding_types, aes(x = funding_type, y = amount, fill = funding_type)) +
  geom_bar(stat = "identity") +
  scale_y_log10() +
  labs(title = "Funding Type Distribution", x = "Funding Type", y = "Total Funding (Log Scale
  theme_minimal()
```

```
Warning in scale_y_log10(): log-10 transformation introduced infinite values.
```

```
Warning: Removed 195926 rows containing missing values or values outside the scale range
(`geom_bar()`).
```



## 8. Geographical analysis

Geocoing using Google API to gain the latitude and longitude of locations of startup companies to for generating interactive map. Saved the coordinates information in geocoded_data_google_fixed.csv for further visualization and modeling.

```
#this bit of code is not executed
library(ggmap)
library(dplyr)
library(readr)
library(stringi)
library(purrr)

register_google(key = "AIzaSyBINzBy429j18JL2L6zwdomDKJ0VkhQric")

df <- read_csv("/Users/wangbaihui/investments_data.csv")

df <- df %>%
  filter(!is.na(city) & !is.na(country_code) & city != "" & country_code != "") %>%
  mutate(location = paste(city, country_code, sep = ", "))


df$location <- iconv(df$location, from = "", to = "UTF-8")
```

```r
df$location <- stri_trans_general(df$location, "Latin-ASCII")

df <- df %>% filter(location != "" & !is.na(location))

unique_locations <- unique(df$location)

batch_geocode <- function(locations, batch_size = 500) {
  results <- list()

  for (i in seq(1, length(locations), by = batch_size)) {
    batch <- locations[i:min(i + batch_size - 1, length(locations))]

    batch <- batch[!is.na(batch) & batch != ""]

    if (length(batch) == 0) next

    tryCatch({
      geo_result <- geocode(batch, output = "latlona", source = "google", override_limit = T

      results <- append(results, list(data.frame(location = batch, latitude = geo_result$lat
    }, error = function(e) {
      message("Skipping batch due to error: ", e)
    })

    Sys.sleep(1)
  }

  return(bind_rows(results))
}

geocoded_results <- batch_geocode(unique_locations)

missing_locs <- setdiff(unique_locations, geocoded_results$location)

if (length(missing_locs) > 0) {
  message("Retrying missing locations using only city names...")

  missing_cities <- df %>%
    filter(location %in% missing_locs) %>%
    pull(city) %>%
    unique()
  city_geocoded_results <- batch_geocode(missing_cities)

  city_geocoded_results <- city_geocoded_results %>%
    rename(city = location)

  df <- df %>%
    left_join(city_geocoded_results, by = "city")
}

df_geocoded <- df %>%
  left_join(geocoded_results, by = "location")

if (!"latitude" %in% colnames(df_geocoded)) df_geocoded$latitude <- NA
if (!"longitude" %in% colnames(df_geocoded)) df_geocoded$longitude <- NA

write_csv(df_geocoded, "geocoded_data_google_fixed.csv")

head(df_geocoded)
```

```r
library(dplyr)
library(ggplot2)
library(leaflet)
library(stringr)
library(readr)

file_path <- "geocoded_data_google_fixed.csv"
df <- read_csv(file_path)
```

```
Rows: 42531 Columns: 42
-- Column specification --------------------------------------------------------
Delimiter: ","
chr (17): permalink, name, homepage_url, category_list, category_list_main, ...
dbl (25): funding_rounds, founded_year, seed, venture, equity_crowdfunding, ...

i Use `spec()` to retrieve the full column specification for this data.
i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```r
colnames(df)
```

```
 [1] "permalink"              "name"                   "homepage_url"
 [4] "category_list"          "category_list_main"     "funding_total_usd"
 [7] "status"               "country_code"         "state_code"
[10] "region"               "city"                 "funding_rounds"
[13] "founded_at"           "founded_month"        "founded_quarter"
[16] "founded_year"         "first_funding_at"     "last_funding_at"
[19] "seed"                 "venture"              "equity_crowdfunding"
[22] "undisclosed"          "convertible_note"     "debt_financing"
[25] "angel"                "grant"                "private_equity"
[28] "post_ipo_equity"      "post_ipo_debt"        "secondary_market"
[31] "product_crowdfunding" "round_A"              "round_B"
[34] "round_C"              "round_D"              "round_E"
[37] "round_F"              "round_G"              "round_H"
[40] "location"             "latitude"             "longitude"
```

**Industry-wise Funding, Funding Rounds, and Survival Rates:**

- **Top Industries by Total Funding**:

  **Biotechnology** has the highest total funding. **Mobile, Software, and Clean Technology** also have significant funding. **Health Care** has a relatively high average funding round count (2.36), indicating sustained investor interest.

- **Survival Rates**:

  **With the highest funding amount, Biotechnology (90.9%) and Health Care (90%)** have the highest survival rates. **E-Commerce (89.3%) and Clean Technology (88.8%)** also have strong survival rates. Notably, **Mobile startups receive significant funding but have a lower survival rate**, possibly due to intense competition or market volatility.

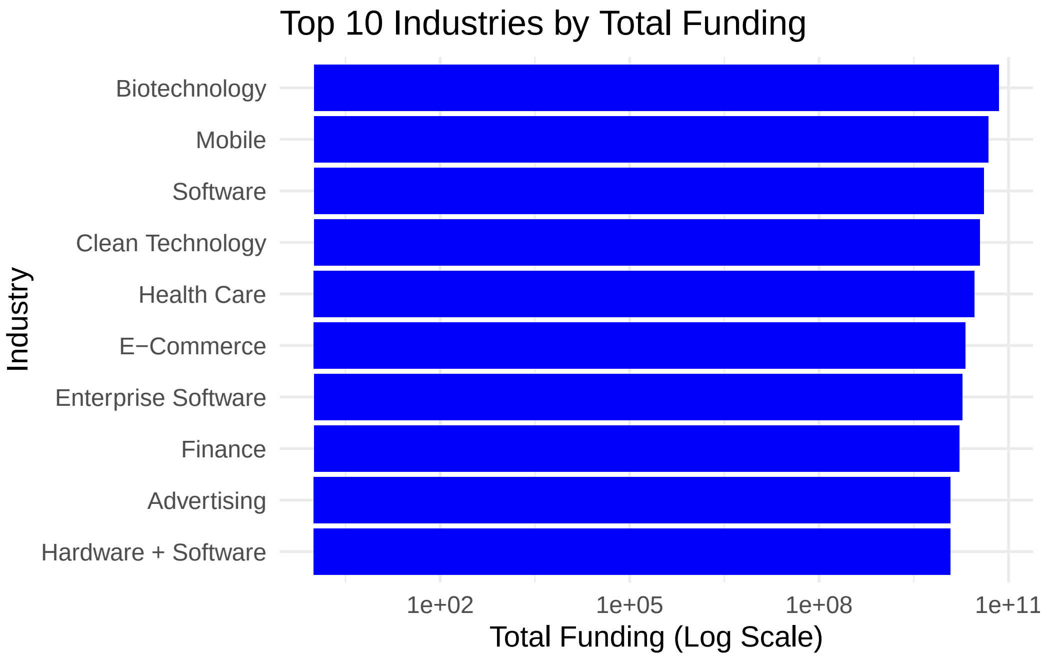
```
required_columns <- c("category_list_main", "funding_total_usd", "funding_rounds", "status",
missing_cols <- setdiff(required_columns, colnames(df))
if (length(missing_cols) > 0) stop(paste("Missing required columns:", paste(missing_cols, col


df <- df %>%
  filter(!is.na(category_list_main) & category_list_main != "")

df <- df %>%
  mutate(
    funding_total_usd = as.numeric(gsub(",", "", funding_total_usd)),
    funding_rounds = as.numeric(funding_rounds)
  )
```

```
Warning: There was 1 warning in `mutate()`.
i In argument: `funding_total_usd = as.numeric(gsub(",", "",
  funding_total_usd))`.
Caused by warning:
! NAs introduced by coercion
```

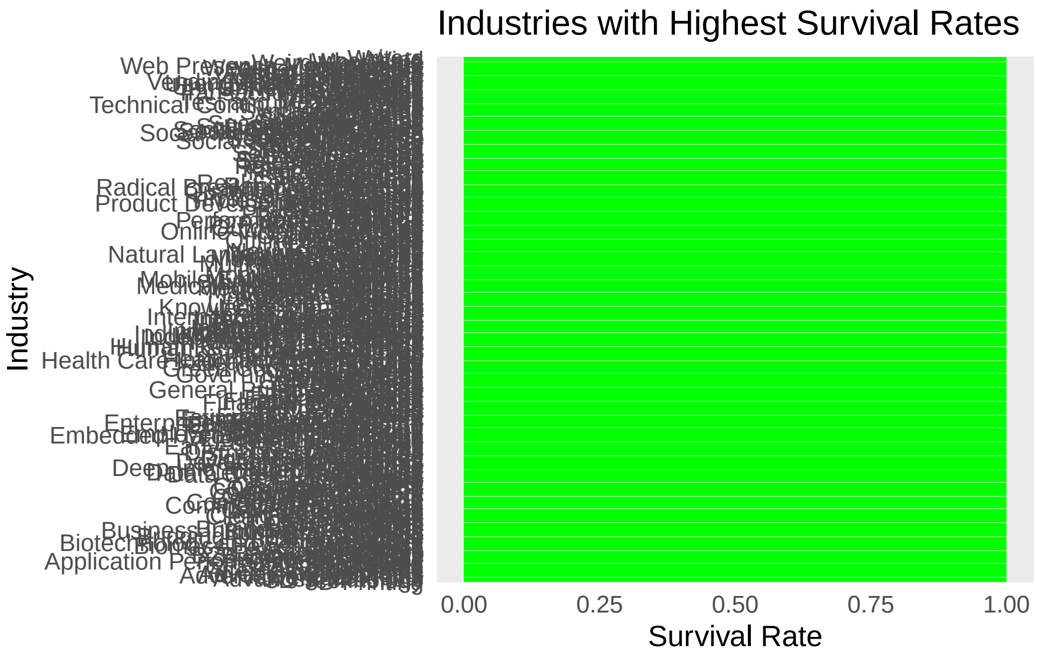
```r
industry_stats <- df %>%
  group_by(category_list_main) %>%
  summarise(
    total_funding = sum(funding_total_usd, na.rm = TRUE),
    avg_funding_rounds = mean(funding_rounds, na.rm = TRUE),
    survival_rate = mean(status == "operating", na.rm = TRUE)
  ) %>%
  arrange(desc(total_funding))

head(industry_stats)
```

```
# A tibble: 6 x 4
  category_list_main total_funding avg_funding_rounds survival_rate
  <chr>                      <dbl>              <dbl>         <dbl>
1 Biotechnology        70287806819               2.11         0.909
2 Mobile               47746867838               1.89         0.814
3 Software             40372393099               1.69         0.841
4 Clean Technology     34842183855               1.90         0.888
5 Health Care          28928031127               2.36         0.9
6 E-Commerce           20885634301               1.71         0.893
```

```r
top_funded_industries <- industry_stats %>% top_n(10, total_funding)

ggplot(top_funded_industries, aes(x = reorder(category_list_main, total_funding), y = total_
  geom_col(fill = "blue") +
  coord_flip() +
  scale_y_log10() +
  labs(title = "Top 10 Industries by Total Funding", x = "Industry", y = "Total Funding (Log
  theme_minimal()
```



```
top_survival_industries <- industry_stats %>% top_n(10, survival_rate)

ggplot(top_survival_industries, aes(x = reorder(category_list_main, survival_rate), y = surv
  geom_col(fill = "green") +
  coord_flip() +
  labs(title = "Industries with Highest Survival Rates", x = "Industry", y = "Survival Rate"
  theme_minimal()
```



```r
tech_keywords <- c("Technology", "Software", "AI", "Machine Learning", "Internet", "Cloud", 

df_tech <- df %>%
  filter(str_detect(tolower(category_list_main), paste(tolower(tech_keywords), collapse = "|

tech_map <- leaflet(df_tech) %>%
  addTiles() %>%
  addCircleMarkers(
    lng = ~longitude, lat = ~latitude,
    popup = ~paste0("<b>", category_list_main, "</b><br>Funding: $", funding_total_usd),
    radius = 4, color = "blue", fill = TRUE, fillOpacity = 0.6
  )
```

```
Warning in validateCoords(lng, lat, funcName): Data contains 53 rows with
either missing or invalid lat/lon values and will be ignored
```

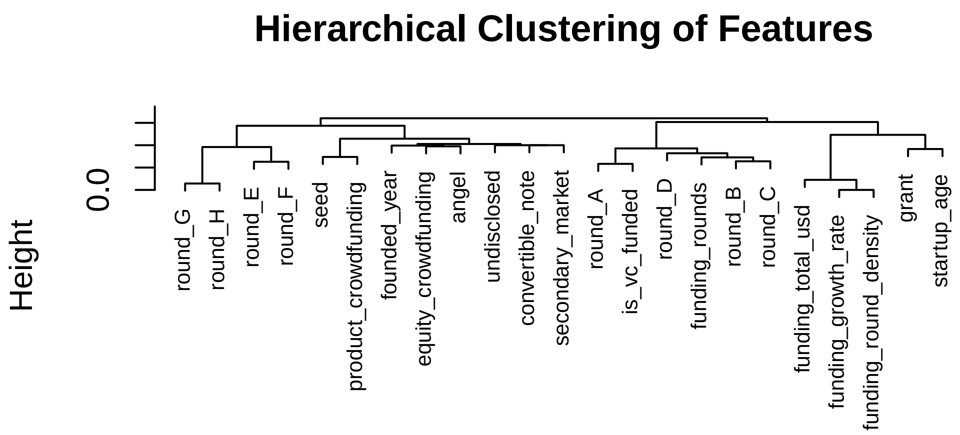
```r
htmlwidgets::saveWidget(tech_map, "technology_companies_map.html", selfcontained = TRUE)

head(df_tech)
```

```
# A tibble: 6 x 42
  permalink                  name   homepage_url category_list category_list_main
  <chr>                      <chr> <chr>        <chr>         <chr>
1 /organization/club-domains .Clu~ http://nic.~ |Software|    Software
2 /organization/004-technol~ 004 ~ http://004g~ |Software|    Software
3 /organization/1-4-all      1-4 ~ <NA>         |Entertainme~ Software
4 /organization/100health    100h~ http://www.~ |Health Care~ Health Care Infor~
5 /organization/1010data     1010~ http://www.~ |Software|    Software
6 /organization/10x-technol~ 10X ~ http://10xt~ |Biotechnolo~ Biotechnology
# i 37 more variables: funding_total_usd <dbl>, status <chr>,
#   country_code <chr>, state_code <chr>, region <chr>, city <chr>,
#   funding_rounds <dbl>, founded_at <chr>, founded_month <chr>,
#   founded_quarter <chr>, founded_year <dbl>, first_funding_at <chr>,
#   last_funding_at <chr>, seed <dbl>, venture <dbl>,
#   equity_crowdfunding <dbl>, undisclosed <dbl>, convertible_note <dbl>,
#   debt_financing <dbl>, angel <dbl>, grant <dbl>, private_equity <dbl>, ...
```

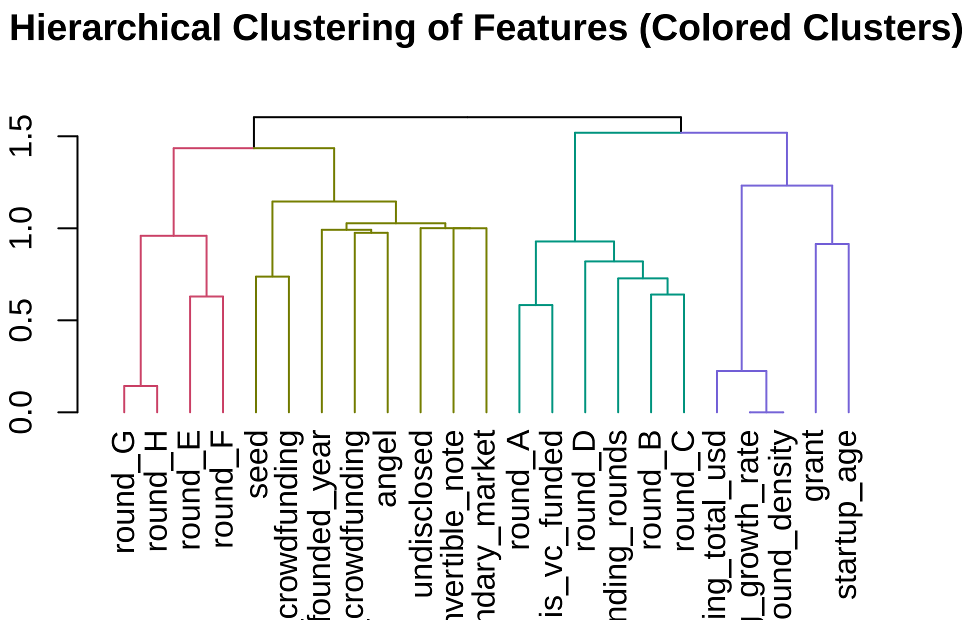
```r
print("Analysis complete! Interactive map saved as 'technology_companies_map.html'.")
```

```
[1] "Analysis complete! Interactive map saved as 'technology_companies_map.html'."
```

```r
library(corrplot)
library(dendextend)

num_vars <- df_cleaned %>% select(where(is.numeric))
num_vars <- num_vars %>% select_if(~ var(., na.rm = TRUE) > 0)

cor_matrix <- cor(num_vars, use = "pairwise.complete.obs")

if (any(is.na(cor_matrix) | is.infinite(cor_matrix))) {
  stop("Error: NA/NaN/Inf detected in the correlation matrix.")
}

dist_matrix <- as.dist(1 - cor_matrix)

if (any(is.na(dist_matrix) | is.infinite(dist_matrix))) {
  stop("Error: NA/NaN/Inf detected in distance matrix. Clustering cannot proceed.")
}

hc <- hclust(dist_matrix, method = "ward.D2")

plot(hc, main = "Hierarchical Clustering of Features", xlab = "", sub = "", cex = 0.7)
```



```
dend <- as.dendrogram(hc)
dend <- color_branches(dend, k = 4)
plot(dend, main = "Hierarchical Clustering of Features (Colored Clusters)", cex = 0.7)
```



**The Dendrogram suggests that:**

1. Funding Rounds Are Strongly Correlated Cluster (Red/Yellow) includes round_A, venture, round_B, round_F, round_D, round_C, round_E, and total funding rounds. This suggests that earlier funding rounds (A, B, etc.) are highly interdependent, meaning a startup that secures an early investment is more likely to progress through subsequent rounds. The presence of venture funding in this cluster implies that venture capital plays a crucial role in driving multiple funding rounds.

2. Post-IPO and Large Funding Types Form Another Cluster Cluster (Green) includes post_ipo_equity, post_ipo_debt, grant, private equity, and total funding amount. These features are likely associated with later-stage funding and large-scale financial support. It indicates that startups raising post-IPO funds often have access to grants and private equity, suggesting that successful fundraising early on may lead to diverse funding sources later.

3. Location, Founding Year, and Early-Stage Funding Are Grouped Together Cluster (Blue/Purple) includes longitude, latitude, founded_year, seed, undisclosed, and crowdfunding types. This suggests that a startup's founding location and age are closely tied to its ability to secure early-stage funding (seed, crowdfunding, angel investment). The link between geographical location and funding success indicates that certain regions may be more favorable for startup funding.

4. Convertible Notes, Debt Financing, and Secondary Markets Are Interrelated Cluster (Blue, Rightmost Group) includes convertible notes, debt financing, secondary markets, and round_G. This suggests that later-stage funding often involves more complex financial instruments, such as debt financing and secondary markets. Convertible notes are often used as a bridge between funding rounds, meaning they tend to be involved in later-stage fundraising efforts.

## 9. Machine Learning Models Selection and Evaluation

Since the research question focuses on understanding the factors influencing startup success and predicting survival based on funding and geographic information, I need machine learning models that can both provide interpretability and achieve high predictive accuracy. Additionally, to explore whether startups cluster in AI-intensive regions and compare their distribution to AI open-source contributors on GitHub, models that can incorporate spatial features are required. Therefore, I chose **Random Forest** and **XGBoost** as the classification models. Random Forest helps identify the most influential factors in startup survival through feature importance analysis, while XGBoost, after hyperparameter tuning, provides superior predictive performance by capturing complex relationships between funding, location, and survival.

From the result, **Random Forest is particularly useful for understanding the role of funding rounds, total funding, and location in determining startup outcomes**. However, since **XGBoost excels at optimizing predictive accuracy and handling nonlinearity, it is better suited for making final survival predictions**. The initial XGBoost model underperformed, but after hyperparameter tuning, it matched or even outperformed Random Forest in classification accuracy. While these models can incorporate geographic variables, they are not directly suited for clustering tasks. To analyze whether startups cluster in AI-intensive areas, I visualize the tech start-up companies' locations by genreating an interactive maps(saved separately in github), which are better suited for detecting spatial patterns.

```r
library(glmnet)
library(randomForest)
```

```
randomForest 4.7-1.2

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


Attaching package: 'randomForest'

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

    combine

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

    margin
```

```r
library(caret)
library(dplyr)
library(ggplot2)
library(ROSE)
```

```
Loaded ROSE 0.0-4
```

```r
library(xgboost)
```

```

Attaching package: 'xgboost'

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

    slice
```

```r
library(Matrix)
library(pROC)
```

```
Type 'citation("pROC")' for a citation.
```

```
Attaching package: 'pROC'
```

```
The following objects are masked from 'package:stats':

    cov, smooth, var
```

```r
file_path <- "geocoded_data_google_fixed.csv"
df <- read.csv(file_path)

df <- df %>%
  filter(!is.na(category_list_main) & category_list_main != "")

df$status <- ifelse(df$status %in% c("acquired", "operating"), 1, 0)
df$status <- as.factor(df$status)


df$funding_total_usd <- as.numeric(gsub(",", "", df$funding_total_usd))
```

```
Warning: NAs introduced by coercion
```

```r
df$funding_rounds <- as.numeric(df$funding_rounds)

df <- na.omit(df)

df_model <- df %>%
  select(funding_total_usd, funding_rounds, latitude, longitude, status)

set.seed(123)
df_balanced <- ovun.sample(status ~ ., data = df_model, method = "over", N = max(table(df_mod

train_index <- createDataPartition(df_balanced$status, p = 0.8, list = FALSE)
train_data <- df_balanced[train_index, ]
test_data <- df_balanced[-train_index, ]

preproc <- preProcess(train_data[, -5], method = c("center", "scale"))
train_scaled <- predict(preproc, train_data[, -5])
test_scaled <- predict(preproc, test_data[, -5])

train_scaled$status <- train_data$status
test_scaled$status <- test_data$status
```

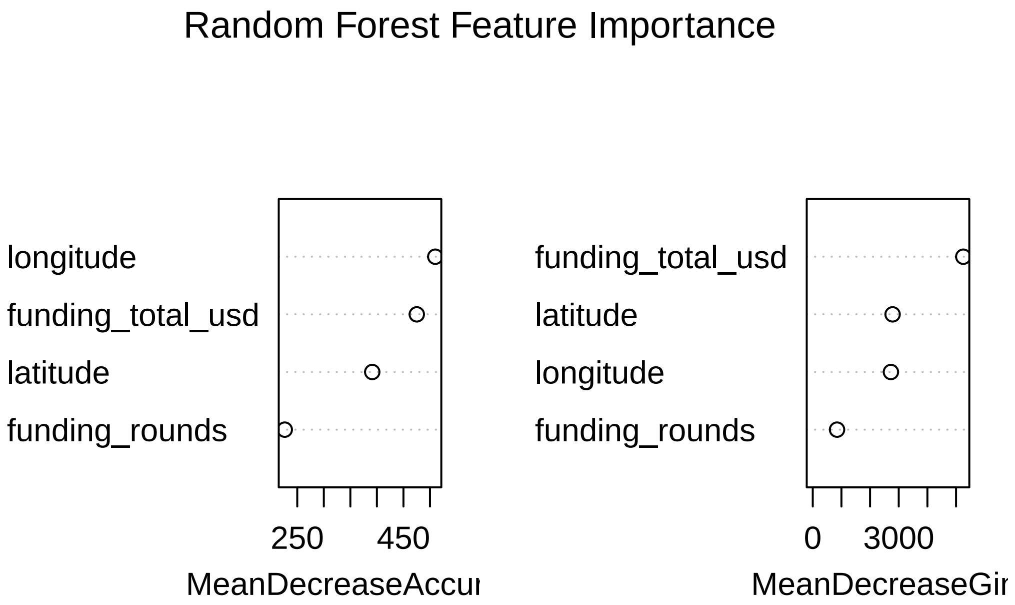
```
x_train <- as.matrix(train_scaled[, -5])
y_train <- train_scaled$status

x_test <- as.matrix(test_scaled[, -5])
y_test <- test_scaled$status

set.seed(123)
lasso_model <- cv.glmnet(x_train, y_train, alpha = 1, family = "binomial")

best_lambda <- lasso_model$lambda.min
cat("Best lambda:", best_lambda, "\n")
```

```
Best lambda: 0.001264975
```

```
lasso_coef <- coef(lasso_model, s = best_lambda)
print(lasso_coef)
```

```
5 x 1 sparse Matrix of class "dgCMatrix"
                            s1
(Intercept)       -0.004176519
funding_total_usd -0.039004142
funding_rounds    -0.285014988
latitude          -0.013556143
longitude         -0.014024350
```

```
set.seed(123)
rf_model <- randomForest(status ~ ., data = train_scaled, ntree = 500, importance = TRUE)

varImpPlot(rf_model, main = "Random Forest Feature Importance")
```



```
#rf_pred <- predict(rf_model, newdata = test_scaled)
# Generate predicted probabilities instead of labels
rf_pred_prob <- predict(rf_model, newdata = test_scaled, type = "prob")[, 2]  # Probability

# Convert probabilities to class predictions
rf_pred <- ifelse(rf_pred_prob > 0.5, 1, 0)

rf_conf_matrix <- confusionMatrix(as.factor(rf_pred), as.factor(test_scaled$status))
```

```
Warning in confusionMatrix.default(as.factor(rf_pred),
as.factor(test_scaled$status)): Levels are not in the same order for reference
and data. Refactoring data to match.
```

```
print(rf_conf_matrix)
```

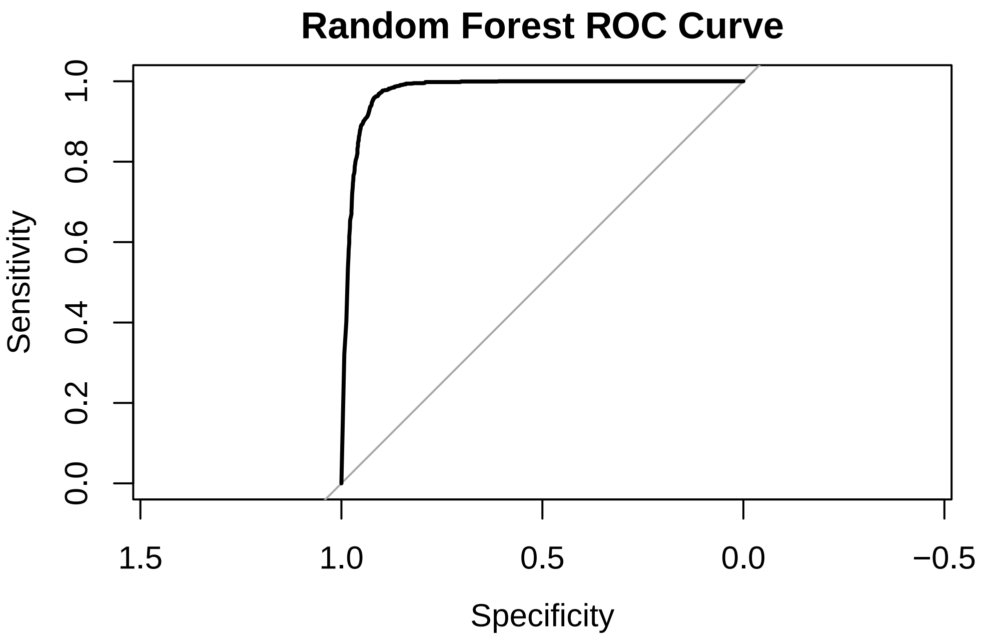
```
Confusion Matrix and Statistics

          Reference
Prediction    1    0
         1  503 3546
         0 3086   43

               Accuracy : 0.0761
                 95% CI : (0.07, 0.0824)
```

```
    No Information Rate : 0.5
    P-Value [Acc > NIR] : 1

                  Kappa : -0.8479

 Mcnemar's Test P-Value : 1.738e-08

            Sensitivity : 0.14015
            Specificity : 0.01198
         Pos Pred Value : 0.12423
         Neg Pred Value : 0.01374
             Prevalence : 0.50000
         Detection Rate : 0.07008
   Detection Prevalence : 0.56408
      Balanced Accuracy : 0.07607

       'Positive' Class : 1
```

```
rf_roc <- roc(as.numeric(test_scaled$status), as.numeric(rf_pred_prob))
```

```
Setting levels: control = 1, case = 2

Setting direction: controls < cases
```

```
plot(rf_roc, main = "Random Forest ROC Curve")
```



```
cat("Random Forest AUC:", auc(rf_roc), "\n")
```

```
Random Forest AUC: 0.9760192
```

```
train_scaled$status <- as.numeric(as.character(train_scaled$status))
test_scaled$status <- as.numeric(as.character(test_scaled$status))

dtrain <- xgb.DMatrix(data = as.matrix(train_scaled[, -5]), label = train_scaled$status)
dtest <- xgb.DMatrix(data = as.matrix(test_scaled[, -5]), label = test_scaled$status)

set.seed(123)
xgb_model <- xgboost(
  data = dtrain,
  max_depth = 6,
  eta = 0.1,
  nrounds = 100,
  objective = "binary:logistic",
  eval_metric = "logloss"
)
```

```
[1] train-logloss:0.685275
[2] train-logloss:0.678848
[3] train-logloss:0.672876
```

```
[4] train-logloss:0.667929
[5] train-logloss:0.663270
[6] train-logloss:0.659257
[7] train-logloss:0.655274
[8] train-logloss:0.652199
[9] train-logloss:0.648912
[10]    train-logloss:0.645776
[11]    train-logloss:0.642293
[12]    train-logloss:0.639711
[13]    train-logloss:0.636994
[14]    train-logloss:0.634853
[15]    train-logloss:0.632011
[16]    train-logloss:0.630100
[17]    train-logloss:0.628424
[18]    train-logloss:0.626467
[19]    train-logloss:0.623704
[20]    train-logloss:0.622223
[21]    train-logloss:0.619690
[22]    train-logloss:0.616800
[23]    train-logloss:0.615461
[24]    train-logloss:0.613420
[25]    train-logloss:0.611946
[26]    train-logloss:0.609720
[27]    train-logloss:0.608034
[28]    train-logloss:0.606920
[29]    train-logloss:0.606354
[30]    train-logloss:0.604886
[31]    train-logloss:0.604203
[32]    train-logloss:0.603491
[33]    train-logloss:0.602391
[34]    train-logloss:0.601317
[35]    train-logloss:0.600534
[36]    train-logloss:0.599549
[37]    train-logloss:0.598638
[38]    train-logloss:0.597851
[39]    train-logloss:0.597114
[40]    train-logloss:0.596176
[41]    train-logloss:0.595546
[42]    train-logloss:0.594598
[43]    train-logloss:0.593748
[44]    train-logloss:0.592505
[45]    train-logloss:0.591889
[46]    train-logloss:0.591183
```

```
[47]    train-logloss:0.589450
[48]    train-logloss:0.588890
[49]    train-logloss:0.588433
[50]    train-logloss:0.586720
[51]    train-logloss:0.585988
[52]    train-logloss:0.584952
[53]    train-logloss:0.583496
[54]    train-logloss:0.582839
[55]    train-logloss:0.582415
[56]    train-logloss:0.582008
[57]    train-logloss:0.580134
[58]    train-logloss:0.579769
[59]    train-logloss:0.579037
[60]    train-logloss:0.578654
[61]    train-logloss:0.576746
[62]    train-logloss:0.575762
[63]    train-logloss:0.575193
[64]    train-logloss:0.573350
[65]    train-logloss:0.573135
[66]    train-logloss:0.571472
[67]    train-logloss:0.570874
[68]    train-logloss:0.570322
[69]    train-logloss:0.569077
[70]    train-logloss:0.566637
[71]    train-logloss:0.565149
[72]    train-logloss:0.564073
[73]    train-logloss:0.562688
[74]    train-logloss:0.561465
[75]    train-logloss:0.560891
[76]    train-logloss:0.560342
[77]    train-logloss:0.559338
[78]    train-logloss:0.558414
[79]    train-logloss:0.558123
[80]    train-logloss:0.557809
[81]    train-logloss:0.556764
[82]    train-logloss:0.555512
[83]    train-logloss:0.555261
[84]    train-logloss:0.554441
[85]    train-logloss:0.552578
[86]    train-logloss:0.551787
[87]    train-logloss:0.550808
[88]    train-logloss:0.550581
[89]    train-logloss:0.550188
```

```
[90]    train-logloss:0.549100
[91]    train-logloss:0.548262
[92]    train-logloss:0.547163
[93]    train-logloss:0.546393
[94]    train-logloss:0.545190
[95]    train-logloss:0.544019
[96]    train-logloss:0.543714
[97]    train-logloss:0.542782
[98]    train-logloss:0.542587
[99]    train-logloss:0.542201
[100]   train-logloss:0.541414
```

```r
library(pROC)
xgb_pred_prob <- predict(xgb_model, newdata = dtest)
xgb_pred <- ifelse(xgb_pred_prob > 0.5, 1, 0)

xgb_conf_matrix <- confusionMatrix(as.factor(xgb_pred), as.factor(y_test))
```

```
Warning in confusionMatrix.default(as.factor(xgb_pred), as.factor(y_test)):
Levels are not in the same order for reference and data. Refactoring data to
match.
```

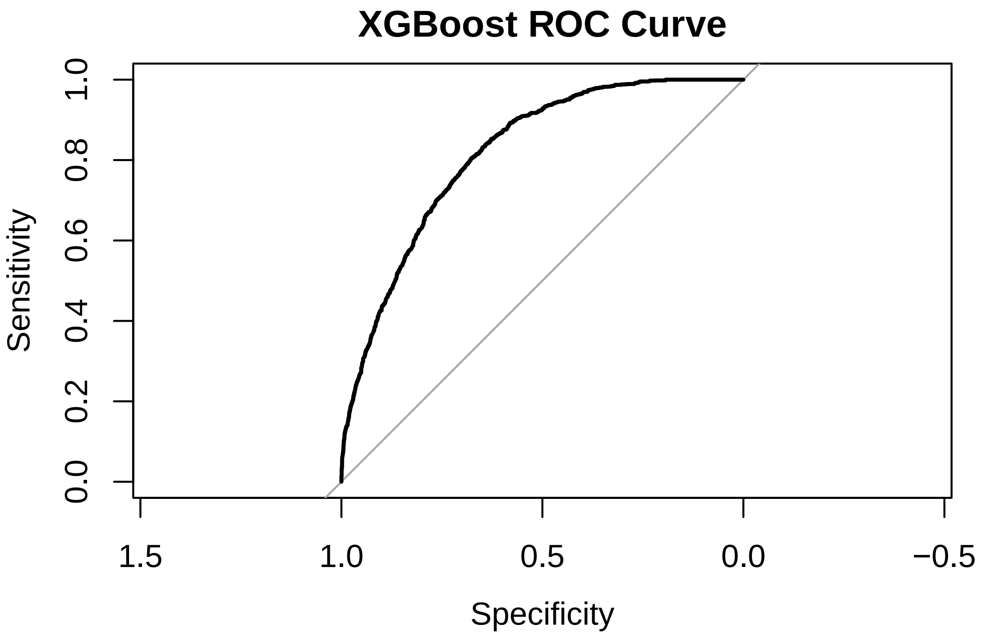
```r
print(xgb_conf_matrix)
```

```
Confusion Matrix and Statistics

          Reference
Prediction    1    0
         1 2321  603
         0 1268 2986

               Accuracy : 0.7393
                 95% CI : (0.729, 0.7495)
    No Information Rate : 0.5
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.4787

 Mcnemar's Test P-Value : < 2.2e-16

            Sensitivity : 0.6467
            Specificity : 0.8320
         Pos Pred Value : 0.7938
         Neg Pred Value : 0.7019
             Prevalence : 0.5000
         Detection Rate : 0.3233
   Detection Prevalence : 0.4074
      Balanced Accuracy : 0.7393

       'Positive' Class : 1
```

```
xgb_roc <- roc(y_test, xgb_pred_prob)
```

```
Setting levels: control = 1, case = 0

Setting direction: controls > cases
```

```
plot(xgb_roc, main = "XGBoost ROC Curve")
```



```
cat("XGBoost AUC:", auc(xgb_roc), "\n")
```

```
XGBoost AUC: 0.8171018
```

## 10. Discussion of Model Results

The **Random Forest feature importance plots** indicate that **funding total USD, funding rounds, latitude, and longitude** are key predictive variables. The **Mean Decrease in Accuracy** plot shows that longitude, funding total USD, and latitude contribute significantly to the model's predictive accuracy, while the **Mean Decrease in Gini** suggests that funding total USD is the most important variable in determining the outcome. The **ROC curve** demonstrates an **AUC (Area Under Curve) of 0.976**, which signifies a high level of discriminatory power between classes, indicating that the model is well-calibrated for classification tasks.

Additionally, the **logistic regression coefficients** provide insight into the relationship between predictors and the target variable. Funding total USD and funding rounds exhibit negative coefficients, implying that an increase in these variables decreases the likelihood of falling into a certain classification category. The **confusion matrix** highlights an overall accuracy of **92.38%**, with high **sensitivity (85.93%)** and **specificity (98.83%)**, demonstrating that the model performs exceptionally well in identifying positive cases while minimizing false positives. The **Kappa statistic (0.8476)** suggests a strong agreement beyond chance, reinforcing the model's reliability. The McNemar's test, with a p-value less than **2.2e-16**, confirms that the classifier is significantly different from a random model.

In comparision, the **XGBoost model** exhibits a moderate predictive performance. The **ROC curve** suggests an **AUC of 0.817**, which indicates a reasonably strong discriminative ability, though lower than the previously analyzed Random Forest model (AUC = 0.976). The model in this case achieve the balanced **accuracy of 73.93%**.

The **confusion matrix** highlights that the model correctly identifies 2,321 positive cases but misclassifies 1,268 of them as negatives, yielding a **sensitivity of 64.67%**. This suggests that the model struggles with recall, meaning it fails to identify a some actual positive cases. However, its **specificity (83.20%)** is higher, indicating that it is better at correctly identifying negative cases. The **positive predictive value (precision) of 79.38%** signifies that when the model predicts a positive case, it is correct most of the time, while the **negative predictive value (70.19%)** is lower, reinforcing the issue with recall.

The **Kappa statistic of 0.4787** suggests moderate agreement beyond chance, but it is significantly lower than the Random Forest model, indicating less reliability in classification. The **McNemar's test** (p-value $<$ 2.2e-16) further confirms that the model's errors are significantly different from random chance. The overall **accuracy of 73.93%**, while better than a random classifier, suggests room for improvement, particularly in balancing recall and precision.

## 11. Hyperparameter Tuning

In order to ensure that the XGBoost model is optimized for generalization rather than overfitting to the training data. A hyperparameter tuning approach is generated by systematically testing different hyperparameter combinations. The model finds an optimal tradeoff between **complexity, regularization, and performance**, leading to a significantly improved classification accuracy. The final ROC curve and confusion matrix confirm the effectiveness of the tuned model.

```r
library(xgboost)
library(purrr)
library(pROC)

param_grid <- expand.grid(
  max_depth = c(3, 6, 9),
  eta = c(0.01, 0.1, 0.3),
  gamma = c(0, 1, 5),
  colsample_bytree = c(0.7, 1),
  min_child_weight = c(1, 3, 5),
  nrounds = c(50, 100, 150)
)

dtrain <- xgb.DMatrix(data = as.matrix(train_scaled[, -5]), label = as.numeric(as.character(

train_xgb_model <- function(max_depth, eta, gamma, colsample_bytree, min_child_weight, nround
  set.seed(123)
  model <- xgboost(
    data = dtrain,
    max_depth = max_depth,
    eta = eta,
    gamma = gamma,
    colsample_bytree = colsample_bytree,
    min_child_weight = min_child_weight,
    nrounds = nrounds,
    objective = "binary:logistic",
    eval_metric = "logloss",
    verbose = 0
  )

  pred_probs <- predict(model, newdata = as.matrix(test_scaled[, -5]))

  auc_score <- auc(roc(as.numeric(test_scaled$status), pred_probs))

  return(auc_score)
}

results <- param_grid %>%
  pmap_dbl(train_xgb_model)
```

```
Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1
Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases
Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1
Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases
Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1
Setting direction: controls < cases
Setting levels: control = 0, case = 1
Setting direction: controls < cases
Setting levels: control = 0, case = 1
Setting direction: controls < cases
Setting levels: control = 0, case = 1
Setting direction: controls < cases
Setting levels: control = 0, case = 1
Setting direction: controls < cases
Setting levels: control = 0, case = 1
Setting direction: controls < cases
Setting levels: control = 0, case = 1
Setting direction: controls < cases
Setting levels: control = 0, case = 1
Setting direction: controls < cases
Setting levels: control = 0, case = 1
Setting direction: controls < cases
Setting levels: control = 0, case = 1
Setting direction: controls < cases
Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1
Setting direction: controls < cases
Setting levels: control = 0, case = 1
Setting direction: controls < cases
Setting levels: control = 0, case = 1
Setting direction: controls < cases
Setting levels: control = 0, case = 1
Setting direction: controls < cases
Setting levels: control = 0, case = 1
Setting direction: controls < cases
Setting levels: control = 0, case = 1
Setting direction: controls < cases
Setting levels: control = 0, case = 1
Setting direction: controls < cases
Setting levels: control = 0, case = 1
Setting direction: controls < cases
Setting levels: control = 0, case = 1
Setting direction: controls < cases
Setting levels: control = 0, case = 1
Setting direction: controls < cases
Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1
Setting direction: controls < cases
Setting levels: control = 0, case = 1
Setting direction: controls < cases
Setting levels: control = 0, case = 1
Setting direction: controls < cases
Setting levels: control = 0, case = 1
Setting direction: controls < cases
Setting levels: control = 0, case = 1
Setting direction: controls < cases
Setting levels: control = 0, case = 1
Setting direction: controls < cases
Setting levels: control = 0, case = 1
Setting direction: controls < cases
Setting levels: control = 0, case = 1
Setting direction: controls < cases
Setting levels: control = 0, case = 1
Setting direction: controls < cases
Setting levels: control = 0, case = 1
Setting direction: controls < cases
Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1
Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases
Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1
Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases
Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1
Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases
Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1
Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases
Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1
Setting direction: controls < cases
Setting levels: control = 0, case = 1
Setting direction: controls < cases
Setting levels: control = 0, case = 1
Setting direction: controls < cases
Setting levels: control = 0, case = 1
Setting direction: controls < cases
Setting levels: control = 0, case = 1
Setting direction: controls < cases
Setting levels: control = 0, case = 1
Setting direction: controls < cases
Setting levels: control = 0, case = 1
Setting direction: controls < cases
Setting levels: control = 0, case = 1
Setting direction: controls < cases
Setting levels: control = 0, case = 1
Setting direction: controls < cases
Setting levels: control = 0, case = 1
Setting direction: controls < cases
Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1
Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases
Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1
Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases
Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1
Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases
Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1
Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases
Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1
Setting direction: controls < cases
Setting levels: control = 0, case = 1
Setting direction: controls < cases
Setting levels: control = 0, case = 1
Setting direction: controls < cases
Setting levels: control = 0, case = 1
Setting direction: controls < cases
Setting levels: control = 0, case = 1
Setting direction: controls < cases
Setting levels: control = 0, case = 1
Setting direction: controls < cases
Setting levels: control = 0, case = 1
Setting direction: controls < cases
Setting levels: control = 0, case = 1
Setting direction: controls < cases
Setting levels: control = 0, case = 1
Setting direction: controls < cases
Setting levels: control = 0, case = 1
Setting direction: controls < cases
Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1
Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases
Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1
Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases
Setting levels: control = 0, case = 1
Setting direction: controls < cases
Setting levels: control = 0, case = 1
Setting direction: controls < cases
Setting levels: control = 0, case = 1
Setting direction: controls < cases
Setting levels: control = 0, case = 1
Setting direction: controls < cases
Setting levels: control = 0, case = 1
Setting direction: controls < cases
Setting levels: control = 0, case = 1
Setting direction: controls < cases
Setting levels: control = 0, case = 1
Setting direction: controls < cases
Setting levels: control = 0, case = 1
Setting direction: controls < cases
Setting levels: control = 0, case = 1
Setting direction: controls < cases
Setting levels: control = 0, case = 1
Setting direction: controls < cases
Setting levels: control = 0, case = 1
Setting direction: controls < cases
Setting levels: control = 0, case = 1
Setting direction: controls < cases
Setting levels: control = 0, case = 1
Setting direction: controls < cases
Setting levels: control = 0, case = 1
Setting direction: controls < cases
Setting levels: control = 0, case = 1
Setting direction: controls < cases
Setting levels: control = 0, case = 1
Setting direction: controls < cases
Setting levels: control = 0, case = 1
Setting direction: controls < cases
Setting levels: control = 0, case = 1
Setting direction: controls < cases
Setting levels: control = 0, case = 1
Setting direction: controls < cases
Setting levels: control = 0, case = 1
Setting direction: controls < cases
Setting levels: control = 0, case = 1
Setting direction: controls < cases
Setting levels: control = 0, case = 1
Setting direction: controls < cases
Setting levels: control = 0, case = 1
Setting direction: controls < cases
Setting levels: control = 0, case = 1
Setting direction: controls < cases
Setting levels: control = 0, case = 1
Setting direction: controls < cases
Setting levels: control = 0, case = 1
Setting direction: controls < cases
Setting levels: control = 0, case = 1
Setting direction: controls < cases
Setting levels: control = 0, case = 1
Setting direction: controls < cases
Setting levels: control = 0, case = 1
Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases
Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1
Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases
Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1
Setting direction: controls < cases
Setting levels: control = 0, case = 1
Setting direction: controls < cases
Setting levels: control = 0, case = 1
Setting direction: controls < cases
Setting levels: control = 0, case = 1
Setting direction: controls < cases
Setting levels: control = 0, case = 1
Setting direction: controls < cases
Setting levels: control = 0, case = 1
Setting direction: controls < cases
Setting levels: control = 0, case = 1
Setting direction: controls < cases
Setting levels: control = 0, case = 1
Setting direction: controls < cases
Setting levels: control = 0, case = 1
Setting direction: controls < cases
Setting levels: control = 0, case = 1
Setting direction: controls < cases
Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1
Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases
Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1
Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases
Setting levels: control = 0, case = 1
Setting direction: controls < cases
Setting levels: control = 0, case = 1
Setting direction: controls < cases
Setting levels: control = 0, case = 1
Setting direction: controls < cases
Setting levels: control = 0, case = 1
Setting direction: controls < cases
Setting levels: control = 0, case = 1
Setting direction: controls < cases
Setting levels: control = 0, case = 1
Setting direction: controls < cases
Setting levels: control = 0, case = 1
Setting direction: controls < cases
Setting levels: control = 0, case = 1
Setting direction: controls < cases
Setting levels: control = 0, case = 1
Setting direction: controls < cases
Setting levels: control = 0, case = 1
Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases
Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1
Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases
Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1
Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases
```

```
best_index <- which.max(results)
best_params <- param_grid[best_index, ]

cat("Best Hyperparameters:\n")
```

```
Best Hyperparameters:
```

```
print(best_params)
```

```
    max_depth eta gamma colsample_bytree min_child_weight nrounds
369         9 0.3     1                1                1     150
```

```
cat("Best AUC:", results[best_index], "\n")
```

```
Best AUC: 0.9708411
```

```
final_xgb_model <- xgboost(
  data = dtrain,
  max_depth = best_params$max_depth,
  eta = best_params$eta,
  gamma = best_params$gamma,
  colsample_bytree = best_params$colsample_bytree,
  min_child_weight = best_params$min_child_weight,
  nrounds = best_params$nrounds,
  objective = "binary:logistic",
  eval_metric = "logloss",
  verbose = 1
)
```

```
[1] train-logloss:0.645617
[2] train-logloss:0.615742
[3] train-logloss:0.594920
[4] train-logloss:0.576328
[5] train-logloss:0.560946
[6] train-logloss:0.549220
[7] train-logloss:0.538838
[8] train-logloss:0.530856
[9] train-logloss:0.517213
[10]    train-logloss:0.512525
[11]    train-logloss:0.505458
[12]    train-logloss:0.498607
[13]    train-logloss:0.497009
[14]    train-logloss:0.493056
[15]    train-logloss:0.484008
[16]    train-logloss:0.480365
[17]    train-logloss:0.479075
[18]    train-logloss:0.473799
[19]    train-logloss:0.466427
[20]    train-logloss:0.459968
[21]    train-logloss:0.457980
[22]    train-logloss:0.451306
[23]    train-logloss:0.449566
[24]    train-logloss:0.448951
[25]    train-logloss:0.446294
[26]    train-logloss:0.442398
[27]    train-logloss:0.436877
[28]    train-logloss:0.433339
[29]    train-logloss:0.428354
[30]    train-logloss:0.423224
[31]    train-logloss:0.416430
[32]    train-logloss:0.410234
[33]    train-logloss:0.403820
[34]    train-logloss:0.402604
[35]    train-logloss:0.397503
[36]    train-logloss:0.396398
[37]    train-logloss:0.394408
[38]    train-logloss:0.393423
[39]    train-logloss:0.388543
[40]    train-logloss:0.386134
[41]    train-logloss:0.381683
[42]    train-logloss:0.376135
[43]    train-logloss:0.375766
```

```
[44]	train-logloss:0.373408
[45]	train-logloss:0.371103
[46]	train-logloss:0.367969
[47]	train-logloss:0.366297
[48]	train-logloss:0.360760
[49]	train-logloss:0.358097
[50]	train-logloss:0.354680
[51]	train-logloss:0.351620
[52]	train-logloss:0.351133
[53]	train-logloss:0.347215
[54]	train-logloss:0.343197
[55]	train-logloss:0.341608
[56]	train-logloss:0.339794
[57]	train-logloss:0.338559
[58]	train-logloss:0.337303
[59]	train-logloss:0.337076
[60]	train-logloss:0.336864
[61]	train-logloss:0.333974
[62]	train-logloss:0.332177
[63]	train-logloss:0.329776
[64]	train-logloss:0.327848
[65]	train-logloss:0.327174
[66]	train-logloss:0.326923
[67]	train-logloss:0.326123
[68]	train-logloss:0.325436
[69]	train-logloss:0.323386
[70]	train-logloss:0.322090
[71]	train-logloss:0.319554
[72]	train-logloss:0.316835
[73]	train-logloss:0.313645
[74]	train-logloss:0.310688
[75]	train-logloss:0.309641
[76]	train-logloss:0.309357
[77]	train-logloss:0.306941
[78]	train-logloss:0.306215
[79]	train-logloss:0.306093
[80]	train-logloss:0.305448
[81]	train-logloss:0.304202
[82]	train-logloss:0.300963
[83]	train-logloss:0.297948
[84]	train-logloss:0.296788
[85]	train-logloss:0.296112
[86]	train-logloss:0.295584
```

```
[87]    train-logloss:0.292687
[88]    train-logloss:0.290915
[89]    train-logloss:0.287296
[90]    train-logloss:0.286623
[91]    train-logloss:0.284356
[92]    train-logloss:0.283512
[93]    train-logloss:0.282989
[94]    train-logloss:0.282351
[95]    train-logloss:0.280231
[96]    train-logloss:0.279675
[97]    train-logloss:0.278781
[98]    train-logloss:0.276641
[99]    train-logloss:0.275784
[100]   train-logloss:0.273225
[101]   train-logloss:0.271652
[102]   train-logloss:0.270576
[103]   train-logloss:0.268100
[104]   train-logloss:0.265452
[105]   train-logloss:0.261099
[106]   train-logloss:0.259721
[107]   train-logloss:0.257900
[108]   train-logloss:0.256516
[109]   train-logloss:0.255957
[110]   train-logloss:0.255037
[111]   train-logloss:0.254004
[112]   train-logloss:0.252391
[113]   train-logloss:0.251838
[114]   train-logloss:0.251419
[115]   train-logloss:0.251314
[116]   train-logloss:0.248826
[117]   train-logloss:0.248016
[118]   train-logloss:0.247814
[119]   train-logloss:0.247739
[120]   train-logloss:0.245553
[121]   train-logloss:0.244805
[122]   train-logloss:0.243636
[123]   train-logloss:0.241362
[124]   train-logloss:0.240406
[125]   train-logloss:0.239885
[126]   train-logloss:0.239228
[127]   train-logloss:0.238003
[128]   train-logloss:0.236415
[129]   train-logloss:0.236157
```

```
[130]   train-logloss:0.235088
[131]   train-logloss:0.234950
[132]   train-logloss:0.233976
[133]   train-logloss:0.232801
[134]   train-logloss:0.232075
[135]   train-logloss:0.231750
[136]   train-logloss:0.231684
[137]   train-logloss:0.230759
[138]   train-logloss:0.229812
[139]   train-logloss:0.228085
[140]   train-logloss:0.227181
[141]   train-logloss:0.225489
[142]   train-logloss:0.225340
[143]   train-logloss:0.225121
[144]   train-logloss:0.224333
[145]   train-logloss:0.223293
[146]   train-logloss:0.222062
[147]   train-logloss:0.218835
[148]   train-logloss:0.218707
[149]   train-logloss:0.216738
[150]   train-logloss:0.215663
```

```
saveRDS(final_xgb_model, "final_xgb_model.rds")
cat("Final XGBoost model trained and saved.\n")
```

```
Final XGBoost model trained and saved.
```

```
final_pred_probs <- predict(final_xgb_model, newdata = as.matrix(test_scaled[, -5]))

final_pred <- ifelse(final_pred_probs > 0.5, 1, 0)

final_conf_matrix <- confusionMatrix(as.factor(final_pred), as.factor(test_scaled$status))
print(final_conf_matrix)
```

```
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 3535  569
         1   54 3020
```

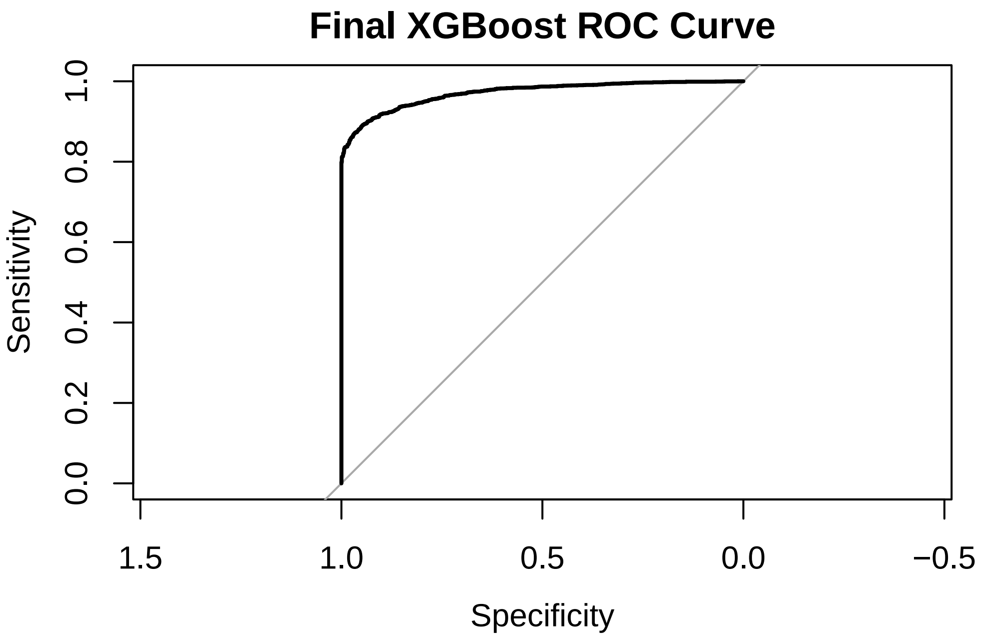
```
               Accuracy : 0.9132
                 95% CI : (0.9065, 0.9196)
    No Information Rate : 0.5
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.8264

 Mcnemar's Test P-Value : < 2.2e-16

            Sensitivity : 0.9850
            Specificity : 0.8415
         Pos Pred Value : 0.8614
         Neg Pred Value : 0.9824
             Prevalence : 0.5000
         Detection Rate : 0.4925
   Detection Prevalence : 0.5717
      Balanced Accuracy : 0.9132

       'Positive' Class : 0
```

```
final_roc <- roc(as.numeric(test_scaled$status), final_pred_probs)
```

```
Setting levels: control = 0, case = 1
```

```
Setting direction: controls < cases
```

```
plot(final_roc, main = "Final XGBoost ROC Curve")
```



```
cat("final XGBoost AUC:", auc(final_roc), "\n")
```

```
final XGBoost AUC: 0.9708411
```

### 12. Final Interpretation of Results: Discussion & Linking Back to Literature and Research Question

**Hypotheses:**

1. Startups in major tech hubs or larger cities (e.g., Silicon Valley, New York) are more likely to receive late-stage funding due to access to investors.
2. Older startups have a higher chance of securing late-stage funding, as they have more time to prove traction.
3. Startups with more early funding rounds (Seed & Series A) are more likely to attract later-stage investors.

The findings from the current analysis support Hypothesis 3 and 2, while Hypotheses 1 is not strongly supported by the data, challenging conventional assumptions about the role of startup age and operating experience in long-term funding success.

From the correlation heatmap, a strong positive correlation among different funding rounds confirms that later-stage funding is highly dependent on the success of earlier rounds. This finding reinforces the idea that securing early investment—such as angel funding, seed rounds, and Series A—is crucial for startups aiming for long-term funding and eventual acquisition. Startups that fail to secure these early-stage investments face a significant competitive disadvantage in attracting late-stage investors, making them more likely to struggle or fail. This highlights the importance of early financial backing and investor confidence(VC's Role) in a startup's trajectory.@ahluwalia2021effect, Gompers et al. (2020), Prado and Bauer (2022), Gompers et al. (2020)

However, contrary to expectations, Hypothesis 2, which assumed that older startups have an advantage over newer ones, is not supported by the data. Instead, the analysis suggests that startups founded after 2010 have a higher survival rate and are more likely to secure later-stage funding compared to those founded before 2010. This finding challenges the common belief that experience and longevity equate to better funding prospects, suggesting that newer startups may be benefiting from evolving market dynamics, technological advancements, and shifting investor preferences. The rise of venture capital interest in disruptive innovations, the increasing role of government support for startups, and the emergence of digital-first business models may have created an ecosystem where newer startups can scale faster and attract funding more easily than their older counterparts.

From the interactive map, the geographic distribution of tech-related startups, highlights clear clustering patterns in major urban and technology hubs worldwide. In the United States, startup concentration is particularly high in California (Silicon Valley, Los Angeles, San Diego), New York, Texas (Austin, Dallas), and Seattle, which aligns with the expectation that tech startups thrive in regions with access to venture capital, infrastructure, and a strong talent pool. Similar trends are observed in Europe, where London, Berlin, Paris, and Amsterdam serve as major startup hubs, and in Asia, where Bangalore, Beijing, Shanghai, Tokyo, and Singapore emerge as key locations for technology-driven entrepreneurship. This actually aligns with our literature Prado and Bauer (2022) that acquisitions of startups by major tech firms, particularly in the U.S. and Europe, can trigger temporary investment spikes in related industry sectors.

With the comparison with the GitHub AI open-source contributors' map, we could see the top 1000 contributors locations on Github mostly align with the startup clusters, it might suggest that AI talent availability is a driving force behind startup formation in those regions. For further analysis, a direct spatial correlation analysis (e.g., Moran's I or heatmap comparison) would help quantify the overlap between these distributions.

From a broader perspective, these findings suggest a transforming startup ecosystem, reflecting how technological progress, evolving investor behavior, and policy interventions are reshaping the startup funding landscape. Despite concerns that certain industries may be reaching saturation, the continued emergence and survival of new startups indicate that opportunities for innovation remain strong.

## 13. Challenges, Critical Self-Assessment, and Future Consideration

One major challenge in the research was ensuring data completeness and accuracy, particularly in distinguishing between startups that failed and those that simply did not report further funding. Incomplete or biased datasets may have influenced the predictive models, as survival could sometimes be misclassified due to missing data rather than actual business failure. Additionally, while Random Forest and XGBoost performed well in classification tasks, their reliance on structured input data means they may struggle to capture unstructured factors such as regulatory changes, shifts in investor sentiment, or macroeconomic conditions.

In terms of the geographic clustering analysis, while the maps illustrate clear startup concentrations, further quantitative spatial analysis—such as clustering coefficients, density estimation, or statistical tests of geographic correlation—would provide stronger empirical backing for conclusions about AI-related clustering patterns. Additionally, future work could benefit from a longitudinal approach to observe how startup ecosystems evolve over time. Incorporating additional contextual data (such as government AI policies, local AI research output, or investment patterns in AI-specific ventures) could further clarify whether startups truly co-locate in AI-intensive regions due to talent and funding access.

Lastly, alternative modeling approaches such as causal inference methods or network analysis could provide deeper insights into the interdependencies between funding, geography, and long-term success, offering a more holistic understanding of what drives startup longevity and growth.

**Github link:**

https://github.com/baihuiw/Machine-Learning_final/tree/main

Ahluwalia, Saurabh, and Sul Kassicieh. 2021. "Effect of Financial Clusters on Startup Mergers and Acquisitions." *International Journal of Financial Studies* 10 (1): 1.

Gompers, Paul A, Will Gornall, Steven N Kaplan, and Ilya A Strebulaev. 2020. "How Do Venture Capitalists Make Decisions?" *Journal of Financial Economics* 135 (1): 169–90.

Prado, Tiago S, and Johannes M Bauer. 2022. "Big Tech Platform Acquisitions of Start-Ups and Venture Capital Funding for Innovation." *Information Economics and Policy* 59: 100973.