PROJECT REPORT

ON

Analysis and Prediction models on Startup funding

Submitted to

NMAM INSTITUTE OF TECHNOLOGY, NITTE
(Off-Campus Centre, Nitte Deemed to be University, Nitte - 574 110, Karnataka, India)

In partial fulfilment of the requirements for the award of the degree of
Bachelor of Technology

in

INFORMATION SCIENCE AND ENGINEERING

by

NAGENDRA PAI **NNNM22IS098**

Under the guidance of

Dr. Manjunatha
Assistant Professor
Department of ISE

NITTE (Deemed to be University) | NMAM INSTITUTE OF TECHNOLOGY

2023 – 2024

NITTE (Deemed to be University) | NMAM INSTITUTE OF TECHNOLOGY

Department of Information Science & Engineering

CERTIFICATE

This is to certify that **Mr. Nagendra pai** bearing USN **NNM22IS098** of II-year B.Tech., a Bonafede student of NMAM Institute of Technology, Nitte, has carried out project on "**Analysis and Prediction models on Startup funding**" as part of the **Introduction to Data Science (IS1102-1)** course during 2023-24, fulfilling the partial requirements for the award of degree of Bachelor of Technology in Information Science and Engineering at NMAM Institute of Technology, Nitte.

.....

Signature of Course Instructor

Dr. Manjunatha

Assistant Professor,

Department of IS&E,

NMAMIT, NITTE (DU)

Table of contents:

ABSTRACT

This data science project revolves around a detailed exploration of a startup funding dataset, with the twin objectives of gaining insights into funding trends and constructing a predictive model for funding status. The dataset encompasses key details like funding dates, startup names, industry verticals, city locations, investor names, investment types, and funding amounts. The initial phase involves comprehensive data preprocessing and exploratory data analysis (EDA) in R, with visualizations unveiling temporal trends, industry distribution, geographical preferences, and an in-depth look into investor involvement. Following this, the project transitions to predictive modeling, leveraging a decision tree algorithm to forecast funding status based on crucial features. The model's performance is rigorously assessed through metrics like accuracy, precision, recall, and F1 score, providing a robust evaluation.

The code provided showcases the meticulous steps in data preparation, visualization, and predictive modeling. Hyperparameter tuning further refines the model's accuracy, adding depth to its predictive capabilities. In essence, this project amalgamates exploratory analysis and predictive modeling, offering a nuanced understanding of startup funding dynamics and furnishing a potent tool for anticipating funding success in the dynamic entrepreneurial landscape.

INTRODUCTION

Embarking on a profound exploration of a diverse dataset chronicling startup funding dynamics, this data science project navigates through a trove of critical details. From funding dates and startup names to industry verticals, city locations, investor names, and funding amounts, the dataset encapsulates the pulse of the dynamic investment landscape within the Indian startup ecosystem from June to August 2017.

The initial phase of this endeavor involves a meticulous journey through data preprocessing, where raw data undergoes a transformational process. Rigorous cleaning and structuring efforts result in a dataset that is not only reliable but serves as the foundation for subsequent analyses. Following this, the project unfolds into an illuminating chapter of exploratory data analysis (EDA), revealing temporal trends, industry distributions, geographical preferences, and the intricate tapestry of investor involvement. Visualizations breathe life into these insights, providing a holistic understanding of the multifaceted startup funding landscape.

Transitioning seamlessly into the realm of predictive modeling, a decision tree algorithm takes center stage. Its deployment aims to construct a robust model capable of forecasting funding statuses with precision. The evaluation metrics, including accuracy, precision, recall, and F1 score, stand as vigilant judges, offering a thorough assessment of the model's performance. Further refinement through hyperparameter tuning optimizes the model's predictive prowess.

This dataset, spanning diverse facets from technology and consumer internet to e-commerce, emerges as a comprehensive chronicle. It is not merely a compilation of dates and figures; instead, it intricately outlines critical facets of each funding transaction. Beyond its immediate utility for entrepreneurs and investors seeking nuanced insights into the funding landscape of that period, the dataset serves as a lens through which industry analysts can discern patterns, preferences, and the evolution of the startup ecosystem across different regions of India.

In essence, this project is a journey through the intricacies and trajectories of startup investments during a pivotal timeframe, offering a potent tool for anticipating funding success in the ever-evolving entrepreneurial landscape.

PROBLEM STATEMENT

1.Data Understanding and Exploration:

a. Issue: The dataset is loaded and displayed, but there is no initial exploration or summary statistics provided.

b. Potential Problem Statement: Lack of initial data exploration may hinder a comprehensive understanding of the dataset, potentially leading to overlooked patterns or outliers.

2.Date Column Processing:

a. Issue: The 'Date' column is converted to a proper date format, and new columns ('day', 'month', 'year', 'monthyear') are created. However, the purpose or significance of these columns is not explained.

b. Potential Problem Statement: The rationale behind the creation of new date-related columns is unclear, which may impact the interpretation of subsequent analyses.

3.Visualization and Analysis:

a. Issue: Several visualizations are created to analyse startup funding trends, industry verticals, city locations, investor types, and funding amounts. However, there is no overarching narrative or context provided to guide the reader through the insights gained.

b. Potential Problem Statement: The absence of clear narratives and interpretations in the visualizations may hinder stakeholders' understanding of the key trends and patterns in startup funding.

4.Data Cleaning and Standardization:

a. Issue: The 'Startup Name' column is processed to remove certain suffixes and standardizecompany names. However, the motivation behind this processing is not explained.

b. Potential Problem Statement: Lack of clarity on the rationale for cleaning and standardizing company names may impact the accuracy and reliability of subsequent analyses.

5.Machine Learning Model Implementation:

a. Issue: A decision tree model is trained to predict 'Funding Status' based on selected features, but the choice of features and the rationale behind the model selection are not discussed.

b. Potential Problem Statement: The lack of explanation regarding feature selection and model choice may lead to a model that is not aligned with the goals of the project or may not generalize well to new data.

6.Model Evaluation and Hyperparameter Tuning:

a. Issue: The model is evaluated using confusion matrix metrics, but there is no discussion on the significance of the metrics or the reasoning behind the hyperparameter tuning.

b. Potential Problem Statement: Without a clear explanation of the chosen evaluation metrics and the rationale behind hyperparameter tuning, the model's effectiveness may be challenging to interpret.

7.Communication of Results:

a. Issue: While the code generates results, there is a lack of comprehensive explanations and interpretations of the findings, making it challenging for non-technical stakeholders to grasp the significance of the analyses.

b. Potential Problem Statement: Inadequate communication of results may hinder effective decision-making by project stakeholders who may not be familiar with the technical details of the analyses.

These potential problem statements aim to highlight areas where additional information, clarity, or context may be needed to enhance the overall quality and effectiveness of the project report. Adjustments or expansions in these areas could lead to a more comprehensive and understandable presentation of the project.

OBJECTIVES

1.Exploratory Data Analysis (EDA):

• Conduct a thorough exploratory data analysis to gain insights into the startup funding dataset, including summary statistics, distributions, and identification of key trends and patterns.

2.Temporal Analysis of Startup Funding:

• Analyse the temporal trends in startup funding between 2015 and 2017 August, identifying monthly variations and potential factors influencing funding fluctuations.

3.Industry Vertical Analysis:

• Explore and analyse the distribution of startup funding across different industry verticals, highlighting the most funded verticals and providing insights into the dynamics of the startup ecosystem.

4.Geographical Analysis:

• Investigate the preferred city locations for startups and analyse how funding varies across different cities, providing insights into regional patterns and potential factors influencing location preferences.

5.Investor Type Analysis:

• Explore the distribution of startup funding based on investor types, identifying the most active investors and gaining insights into the preferences of different investor categories.

6.Company Funding Analysis:

• Analyse the funding received by individual startup companies, identifying the top-funded companies and exploring factors that contribute to their success in securing funding.

7.Repeat Funding Analysis:

• Investigate the frequency of funding for individual startups, identifying companies that receive funding repeatedly and exploring potential factors contributing to their sustained financial support.

8.Investor-Startup Relationships:

• Explore the relationships between investors and startups, identifying investors who have funded multiple startups and analyzing patterns in their investment activities.

9.Data Cleaning and Standardization:

• Ensure data quality and consistency by addressing missing values, outliers, and standardizing company names to enhance the accuracy and reliability of subsequent analyses.

10.Machine Learning Modelling:

•Develop a predictive model to determine the likelihood of startup funding based on relevant features, aiming to provide insights into the factors that significantly influence funding success.

11.Model Evaluation and Interpretation:

• Evaluate the performance of the machine learning model using appropriate metrics and provide clear interpretations of the results, ensuring that stakeholders can understand the model's effectiveness and limitations.

12.Communication of Findings:

• Communicate the key findings and insights from the analyses in a clear and comprehensible manner, using visualizations and narratives to facilitate understanding among both technical and non-technical stakeholders.

13.Recommendations for Stakeholders:

• Provide actionable recommendations based on the project findings, enabling stakeholders to make informed decisions and strategies related to startup funding, investment, and ecosystem dynamics.

These objectives aim to guide the project towards a comprehensive analysis of startup funding, combining descriptive and predictive analytics to generate valuable insights for stakeholders. Adjustments can be made based on the specific goals and priorities of the project.

IMPLEMENTATION

Import library, data set and bird eye view on dataset

```
library(readr)
library(dplyr)
library(ggplot2)
library(tidyverse)
library(data.table)
library(DT)

startup <- read_csv("../input/startup_funding.csv")
```

```
dim(startup)
```

```
## [1] 2372   10
```

```
datatable(startup, style="bootstrap", class="table-condensed", options = list(dom = 'tp',scrollX = TRUE))
```

Changing Date

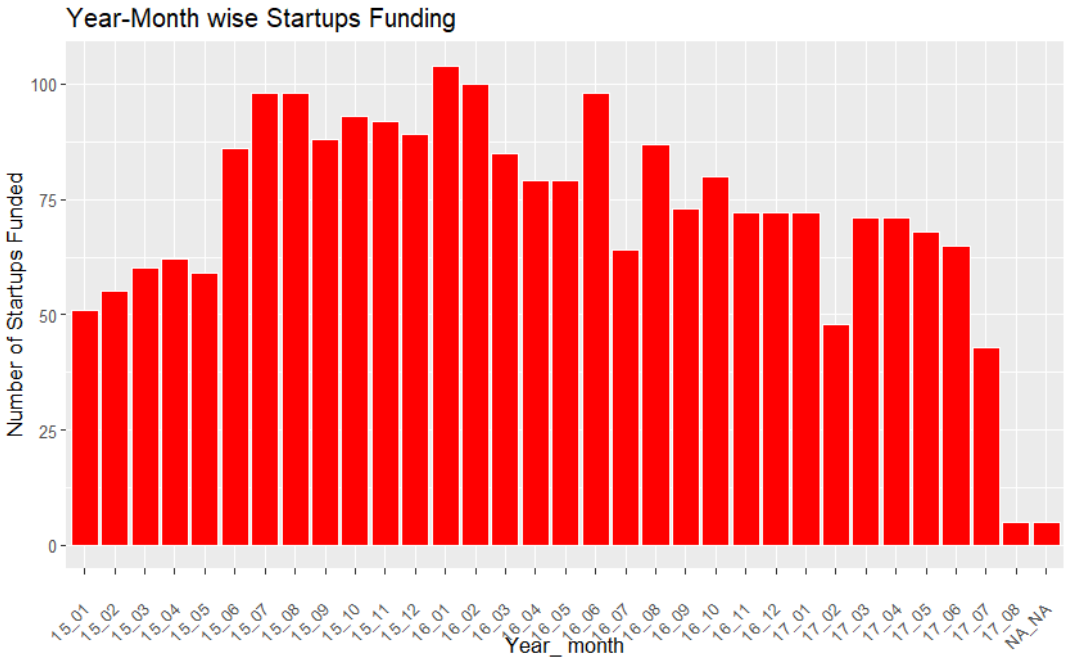
```
startup$date <- as.POSIXct(strptime(startup$Date, format = "%d/%m/%Y"))
startup$day <- as.integer(format(startup$date, "%d")) # day
startup$month <- as.factor(format(startup$date, "%m")) # month
startup$year <- as.integer(format(startup$date, "%y")) # day
startup$monthyear <- paste(startup$year,startup$month,sep="_")# year month
```

In summary, these lines of code preprocess the 'Date' column in the startup dataset by converting it to a proper date format and then extract and create additional columns for day, month, year, and a combined column ('month year') that represents the year and month together. These new columns can be useful for time-based analysis and visualizations.

Start-up Funding between 2015-2017 August

```
startup %>%
    group_by(monthyear)%>%
    summarise(n = n())%>%
    drop_na(monthyear)%>%
    ggplot(aes(x =monthyear, y =  n )) +
    geom_bar(stat='identity',colour="white", fill = c("red"))+
    labs(x = 'Year_ month', y = 'Number of Startups Funded', title = 'Year-Month wise Startups Fund
ing') +
    theme(axis.text.x=element_text(angle=45,hjust=1,vjust=0.5))
```

The R code utilizes the startup funding dataset to create a visually insightful bar plot, depicting the distribution of startup funding across different months and years. Through the transformation of the 'Date' column, the code extracts relevant temporal information and generates a 'month year' identifier. This identifier is employed for grouping the dataset and creating a bar plot that effectively communicates the variations in the count of startups funded over time. The red-filled bars on the plot represent individual months and years, with the x-axis labelled as 'Year Month' and the y-axis indicating the 'Number of Startups Funded.' This analysis offers stakeholders a succinct overview of temporal funding trends, facilitating prompt identification of significant patterns and informed decision-making in the startup ecosystem.



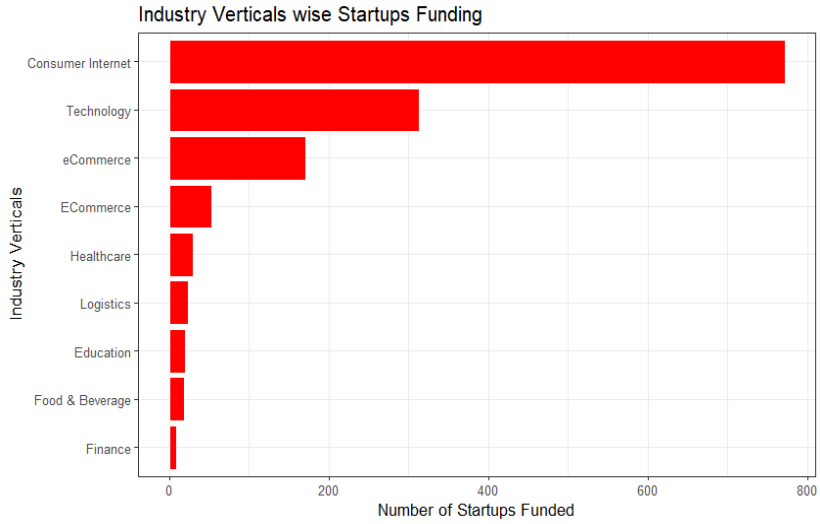
Seems like between mid 2015 to mid 2016 has more number of fundings compared to other months and has been marginally declining in the recent days.

Startup India campaign is based on an action plan aimed at promoting bank financing for start-up ventures to boost entrepreneurship and encourage start ups with jobs creation. The campaign was first announced by Prime Minister on 15 August 2015, This made many new startup registration and funding, Later in mid 2016 some newbie companies couldn’t show case good expected profits which lost confidence among investors.

Industry vertical categories

```
temp<-startup %>%
    group_by(IndustryVertical)%>%
    summarise(n = n()) %>%
    drop_na(IndustryVertical)%>%
    arrange(desc(n)) %>%
    head(n = 9)

temp %>%
    ggplot(aes(x = reorder(IndustryVertical , n) , y = n)) +
    geom_bar(stat='identity',colour="white", fill = c("red")) +
    labs(x = 'Industry Verticals', y = 'Number of Startups Funded', title = 'Industry Verticals wis
e Startups Funding') +
    coord_flip() +
    theme_bw()
```

The provided R code conducts an analysis of startup funding based on industry verticals using the given dataset. Initially, the code groups the dataset by 'Industry Vertical' and calculates the count of startups funded in each industry. It then filters out any rows with missing values in the 'Industry Vertical' column and arranges the summarized data in descending order of the number of startups funded. The top 9 industry verticals are selected for further analysis. The code subsequently generates a horizontal bar plot using `ggplot2`, where the x-axis represents the count of startups funded, and the y-axis represents different industry verticals. Bars are filled with a red colour to enhance visibility, and the x-axis text is flipped for better readability. The plot is labelled with 'Industry Verticals wise Startups Funding,' providing a clear visual representation of the distribution of startup funding across various industry verticals.



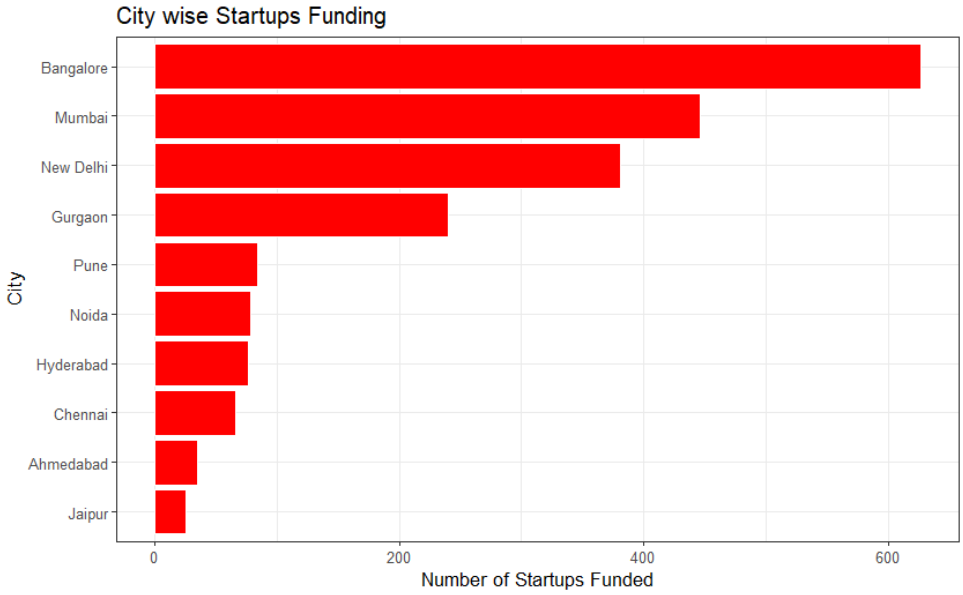
Consumer Internet is the most preferred industry segment for **funding followed by Technology and E-commerce.**

Preferred City Location for startups categories

```
temp <-startup %>%
    group_by(CityLocation)%>%
    summarise(n = n())%>%
    drop_na(CityLocation)%>%
    arrange(desc(n)) %>%
    head(n = 10)

temp %>%
    ggplot(aes(x =reorder(CityLocation,n), y =  n )) +
    geom_bar(stat='identity',colour="white", fill = c("red")) +
    labs(x = 'City', y = 'Number of Startups Funded', title = 'City wise Startups Funding') +
    coord_flip() +
    theme_bw()
```

The provided R code conducts an analysis of startup funding based on city locations using the given dataset. Initially, the code groups the dataset by 'CityLocation' and calculates the count of startups funded in each city. It then filters out any rows with missing values in the 'CityLocation' column and arranges the summarized data in descending order of the number of startups funded. The top 10 cities are selected for further analysis. The code subsequently generates a horizontal bar plot using `ggplot2`, where the x-axis represents the count of startups funded, and the y-axis represents different cities. Bars are filled with a red color to enhance visibility,

and the x-axis text is flipped for better readability. The plot is labeled with 'City wise Startups Funding,' providing a clear visual representation of the distribution of startup funding across various cities.

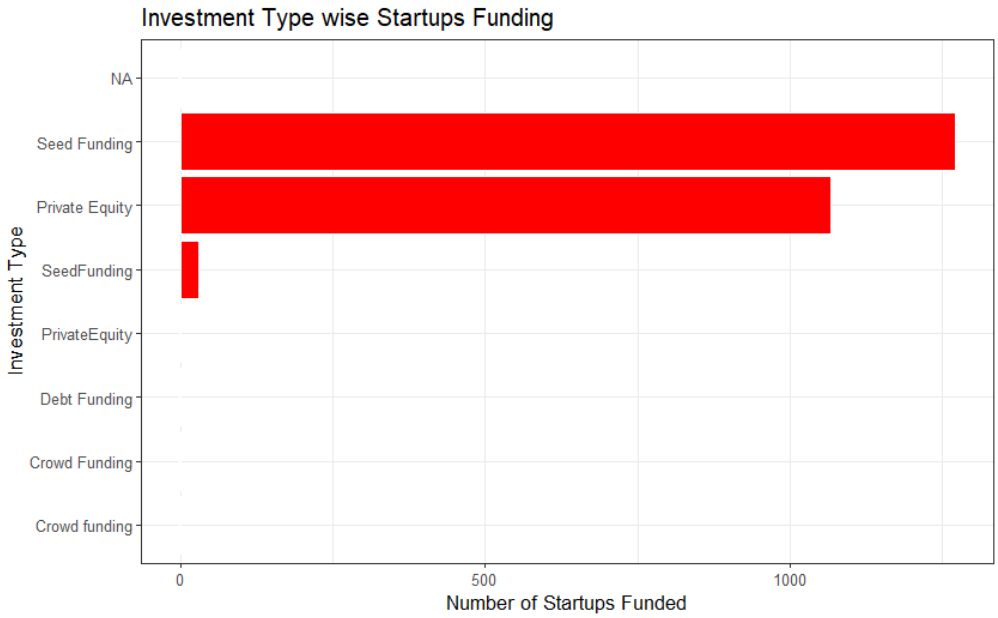


Bangalore seems to attract lot of investments followed by **Mumbai and Delhi.**

Investment type categories

```
temp<- startup %>%
    group_by(InvestmentType)%>%
    summarise(n = n())%>%
    arrange(desc(n)) %>%
    head(n = 10)

temp %>%
    ggplot(aes(x = reorder(InvestmentType,n) , y =  n )) +
    geom_bar(stat='identity',colour="white", fill = c("red")) +
    labs(x = 'Investment Type', y = 'Number of Startups Funded', title = 'Investment Type wise Star
tups Funding') +
    coord_flip() +
    theme_bw()
```

The provided R code analyzes startup funding based on different investment types using the given dataset. It begins by grouping the dataset by 'InvestmentType' and calculating the count of startups funded under each type. The summarized data is then arranged in descending order based on the count, and the top 10 investment types are selected for further analysis. The code proceeds to create a horizontal bar plot using `ggplot2`, where the x-axis represents the count of startups funded, and the y-axis represents different investment types. The bars are filled with a red color for clarity, and the x-axis text is flipped to enhance readability. The plot is labeled 'Investment Type wise Startups Funding,' offering a visual representation of the distribution of startup funding across various investment types.



Seed funding tops the chart followed by **Private Equity**

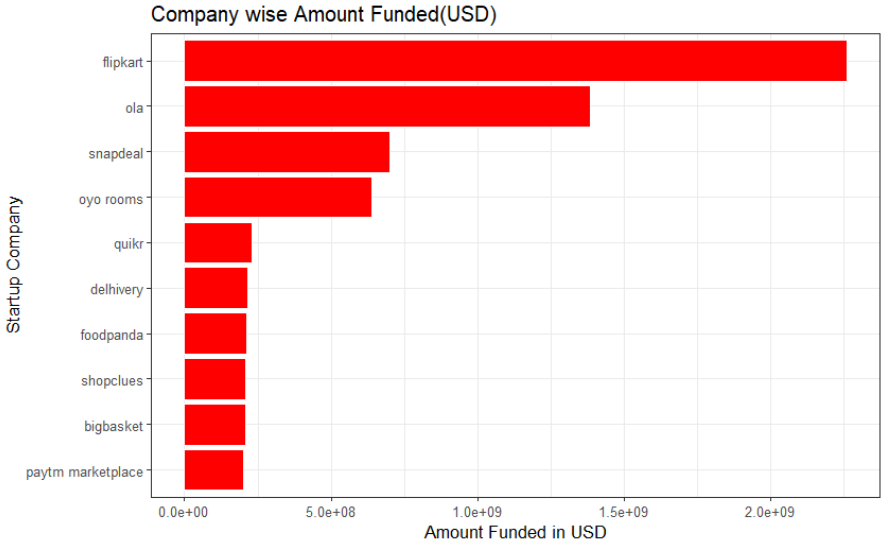
Which start-up is funded more

I have observed single startup name taken in different ways such as oyo rooms as oyo and oyorooms, at times with .com , .in as suffix Lets change the names accordingly

```
# Removing .com, .in, .co
startup$StartupName <- sapply(strsplit(startup$StartupName, split='.com', fixed=TRUE), function(x) (x[1]))
startup$StartupName <- sapply(strsplit(startup$StartupName, split='.in', fixed=TRUE), function(x) (x[1]))
startup$StartupName <- sapply(strsplit(startup$StartupName, split='.co', fixed=TRUE), function(x) (x[1]))
startup$StartupName <- tolower(startup$StartupName)


startup$StartupName[startup$StartupName == "olacabs"] <- "ola"
startup$StartupName[startup$StartupName == "oyo"] <- "oyo rooms"
startup$StartupName[startup$StartupName == "oyorooms"] <- "oyo rooms"


temp<- startup %>%
  group_by(StartupName)%>%
  summarise(n = sum(AmountInUSD))%>%
  arrange(desc(n)) %>%
  head(n = 10)

temp %>%
  ggplot(aes(x = reorder(StartupName,n) , y =  n )) +
  geom_bar(stat='identity',colour="white", fill = c("red")) +
  labs(x = 'Startup Company', y = 'Amount Funded in USD', title = 'Company wise Amount Funded(USD)  ') +
  coord_flip() +
  theme_bw()
```

The presented R code undertakes a preprocessing journey for startup names in the dataset, eliminating common domain suffixes and standardizing names by converting them to lowercase. Notably, specific startup names like 'olacabs' are replaced with 'ola', and variations of 'oyo' are standardized to 'oyo rooms'. Subsequently, the code groups the dataset by startup names, computes the sum of funding amounts ('AmountInUSD') for each startup, and arranges the data in descending order based on the total funding received. A horizontal bar plot is then generated using `ggplot2`, where the x-axis represents different startup companies, the y-axis illustrates the total amount funded in USD, and bars are filled in red for clarity. The

resulting visualization offers insights into the top 10 startup companies, providing a clear depiction of their relative funding amounts within the dataset.

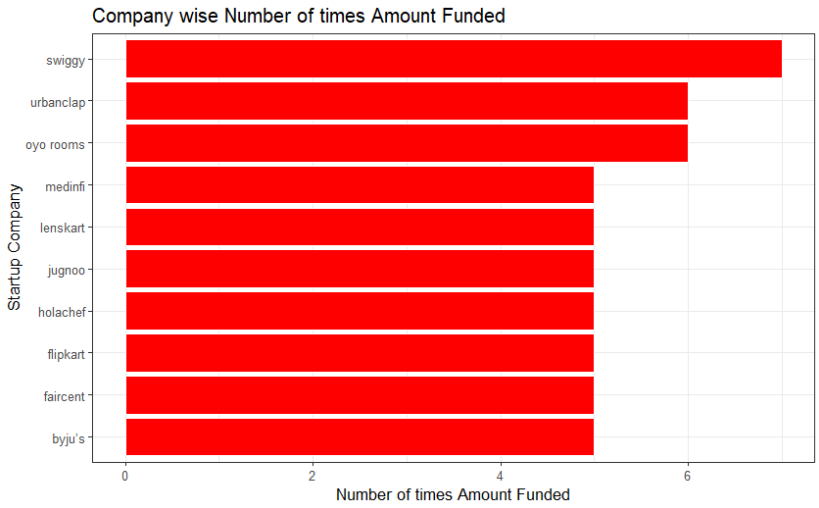


Flipkart followed by Ola, snapdeal are most Funded Startups

Which start-up is funded again and again

```
temp<- startup %>%
    group_by(StartupName)%>%
    summarise(n = n())%>%
    arrange(desc(n)) %>%
    head(n = 10)

head(temp,10)

temp %>%
    ggplot(aes(x = reorder(StartupName,n) , y =  n )) +
    geom_bar(stat='identity',colour="white", fill = c("red")) +
    labs(x = 'Startup Company', y = 'Number of times Amount Funded', title = 'Company wise Number of times Amount Funded ') +
    coord_flip() +
    theme_bw()
```

The provided R code investigates the frequency of funding occurrences for different startup companies in the dataset. Initially, the code groups the data by 'StartupName' and calculates the count of occurrences for each startup. The resulting summary, stored in the 'temp' variable, is arranged in descending order based on the frequency, and the top 10 startups are displayed using the `head` function. Subsequently, a horizontal bar plot is created using `ggplot2`, where the x-axis represents various startup companies, the y-axis denotes the count of funding occurrences, and bars are filled in red for visibility. The x-axis text is flipped for improved readability. The plot is labeled 'Company wise Number of times Amount Funded,' providing a visual representation of the funding frequency for the top 10 startup companies. This analysis aids in understanding the recurrence patterns of funding for different startups within the dataset.



Investors name column split

```
library(splitstackshape)

startup <- cSplit(startup, "InvestorsName", ",", 'long', drop = FALSE)

head(startup)
```

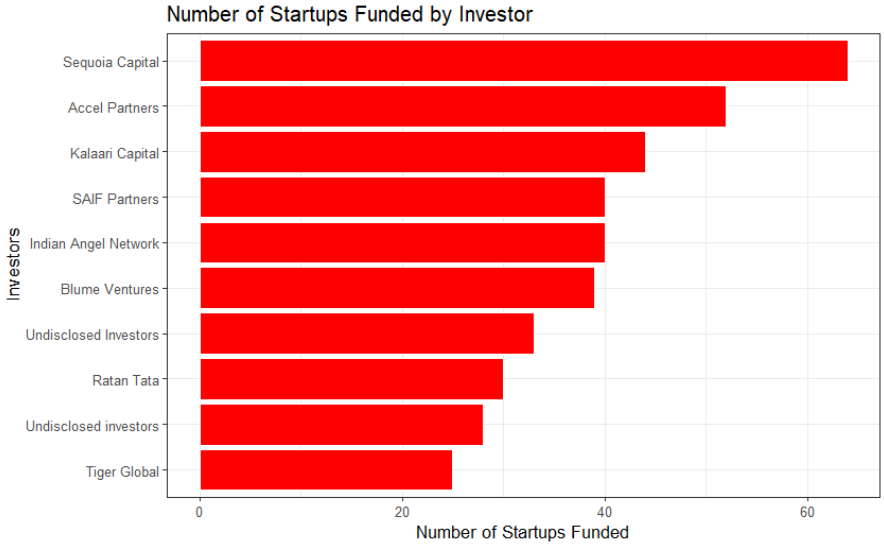
The R code utilizes the `splitstackshape` library to split the values in the 'InvestorsName' column of the 'startup' dataset. The `cSplit` function is applied to break down the comma-separated values in the 'InvestorsName' column into separate rows while duplicating the corresponding values in other columns to maintain data integrity. The result is a modified 'startup' dataset with an elongated structure, where each row corresponds to a unique investor associated with a particular startup. This transformation facilitates more granular analysis and exploration of the relationships between startups and their respective investors. The `head` function is then used to display the first few rows of the modified dataset for a quick overview.

Number of startups funded by the investor

```
temp<-startup %>%
    group_by(InvestorsName)%>%
    summarise(n = n())%>%
    arrange(desc(n)) %>%
    head(n = 10)

temp %>%
    ggplot(aes(x = reorder(InvestorsName,n) , y =  n )) +
    geom_bar(stat='identity',colour="white", fill = c("red")) +
    labs(x = 'Investors', y = 'Number of Startups Funded', title = 'Number of Startups Funded by Investor') +
    coord_flip() +
    theme_bw()
```

The presented R code explores the frequency of involvement by different investors in startup funding within the dataset. It starts by grouping the 'startup' dataset by 'InvestorsName' and calculating the count of startups associated with each investor using the `summarise` function. The resulting summary, stored in the 'temp' variable, is then arranged in descending order based on the frequency of investments, and the top 10 investors are selected using the `head` function. Subsequently, a horizontal bar plot is generated using `ggplot2`, where the x-axis represents various investors, the y-axis denotes the count of startups funded by each investor, and bars are filled in red for visibility. The x-axis text is flipped for improved readability. The plot is appropriately labeled 'Number of Startups Funded by Investor,' providing a visual representation of the top 10 investors and

their respective frequencies of funding startups in the dataset. This analysis offers insights into the influential investors and their contributions to startup funding within the given context.



Decision Tree Classification for predicting funding status

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

# Read the CSV file
df <- read.csv("startup_funding.csv")

# Data Preprocessing
df$InvestmentType <- as.factor(as.numeric(factor(df$InvestmentType)))
df$CityLocation <- as.factor(as.numeric(factor(df$CityLocation)))
df$IndustryVertical <- as.factor(as.numeric(factor(df$IndustryVertical)))
df$SubVertical <- as.factor(as.numeric(factor(df$SubVertical)))

# Create FundingStatus based on AmountInUSD
df$FundingStatus <- ifelse(df$AmountInUSD > 0, "Funded", "Not Funded")
df$FundingStatus <- as.factor(df$FundingStatus)


# Split the data into training and testing sets
set.seed(42)
split_index <- sample(1:nrow(df), 0.5 * nrow(df))
train_data <- df[split_index, ]
test_data <- df[-split_index, ]

# Train the decision tree model on the training data
model <- rpart(FundingStatus ~ InvestmentType + CityLocation + IndustryVertical
               + AmountInUSD, data = train_data, method = 'class')

# Ensure factor levels in test_data match those in train_data
test_data$AmountInUSD <- factor(test_data$AmountInUSD,
                                levels = levels(train_data$AmountInUSD))

# Make predictions on the test data
predictions <- predict(model, newdata = test_data, type = 'class')

# Evaluate accuracy and other metrics
conf_matrix <- confusionMatrix(predictions, test_data$FundingStatus)


# Extract accuracy, precision, recall, and F1 score from the confusion matrix
accuracy <- conf_matrix$overall["Accuracy"]
precision <- conf_matrix$byClass["Precision"]
```

```r
recall <- conf_matrix$byClass["Recall"]
f1_score <- conf_matrix$byClass["F1"]
# Print the confusion matrix
print(conf_matrix)
# Print evaluation metrics
print(paste("Accuracy:", round(accuracy, 4)))
print(paste("Precision:", round(precision, 4)))
print(paste("Recall:", round(recall, 4)))
print(paste("F1 Score:", round(f1_score, 4)))
```

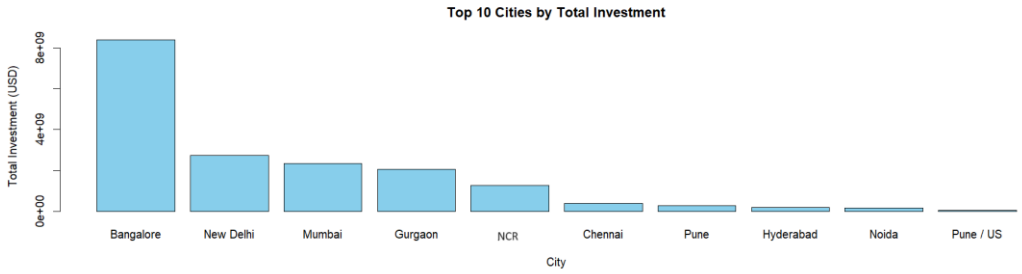
In the provided R code, a decision tree model is trained on startup funding data with features such as Investment Type, City Location, Industry Vertical, and AmountInUSD, aiming to predict the Funding Status (Funded or Not Funded). The dataset undergoes preprocessing steps, including converting categorical variables to factors and creating a Funding Status column based on the condition that AmountInUSD is greater than 0. The dataset is then split into training and testing sets. The decision tree model is trained on the training data, and predictions are made on the test data. Model performance is evaluated using a confusion matrix, and metrics such as accuracy, precision, recall, and F1 score are printed. Additionally, the code utilizes the `rpart.plot` library to visually depict the structure of the trained decision tree, providing insights into the decision-making process of the model.

```
> # Print the confusion matrix
> print(conf_matrix)
Confusion Matrix and Statistics

            Reference
Prediction   Funded Not Funded
  Funded        764        407
  Not Funded     11          4

               Accuracy : 0.6476
                 95% CI : (0.6196, 0.6748)
    No Information Rate : 0.6535
    P-Value [Acc > NIR] : 0.6773

                  Kappa : -0.0058


  Mcnemar's Test P-Value : <2e-16

             Sensitivity : 0.985806
             Specificity : 0.009732
          Pos Pred Value : 0.652434
          Neg Pred Value : 0.266667
              Prevalence : 0.653457
          Detection Rate : 0.644182
    Detection Prevalence : 0.987352
       Balanced Accuracy : 0.497769

        'Positive' Class : Funded

> # Print evaluation metrics
> print(paste("Accuracy:", round(accuracy, 4)))
[1] "Accuracy: 0.6476"
> print(paste("Precision:", round(precision, 4)))
[1] "Precision: 0.6524"
> print(paste("Recall:", round(recall, 4)))
[1] "Recall: 0.9858"
> print(paste("F1 Score:", round(f1_score, 4)))
[1] "F1 Score: 0.7852"
>
```

Aggregation and Summation algorithm for City wise total funding

```r
# Load the dataset
startup_data <- read.csv("startup_funding.csv", stringsAsFactors = FALSE)

# Create a vector to store total investment per city
total_investment_per_city <- rep(0, length(unique(startup_data$CityLocation)))

# Loop through the data and calculate total investment per city
for (i in seq_along(startup_data$CityLocation)) {
  city <- startup_data$CityLocation[i]
  amount <- as.numeric(gsub(",", "", startup_data$AmountInUSD[i]))
  # Convert to numeric, removing commas

  if (!is.na(amount)) {
    total_investment_per_city[city == unique(startup_data$CityLocation)] <-
      total_investment_per_city[city == unique(startup_data$CityLocation)]
    + amount
  }
}

# Create a data frame with city names and corresponding total investments
result_df <- data.frame(City = unique(startup_data$CityLocation),
                        TotalInvestment = total_investment_per_city)


# Sort the data frame by TotalInvestment in descending order
result_df <- result_df[order(-result_df$TotalInvestment), ]

# Take the top 10 cities
top_10 <- head(result_df, 10)

# Plot the top 10 cities
barplot(top_10$TotalInvestment, names.arg = top_10$City, col = "skyblue",
        main = "Top 10 Cities by Total Investment",
        xlab = "City", ylab = "Total Investment (USD)")
```

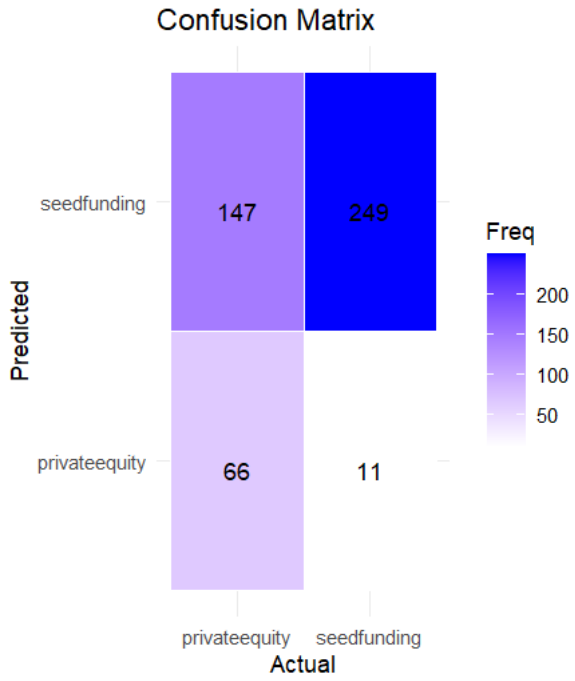
The provided R code aims to calculate the total investment in different cities based on a startup funding dataset. It begins by loading the dataset and initializing a vector to store the total investment for each unique city. The code then iterates through the dataset, converting the funding amounts to numeric values while excluding any entries with missing values. It accumulates the total investment for each city. The result is organized into a data frame with city names and corresponding total investments, which is then sorted in descending order. Finally, the code selects the top 10 cities by total investment and creates a bar plot to visually represent the data, with city names on the x-axis and total investment amounts on the y-axis.



KNN Classification for startup funding type

```r
# Load required libraries
library(ggplot2)
library(caret)
library(class)
# Replace this with your actual dataset loading procedure
df<-read.csv("startup_funding.csv")
# Drop rows with missing values for simplicity in this example
df <- na.omit(df)
# Encode categorical variables
df$InvestmentType <- as.factor(df$InvestmentType)

# Select features and target variable
X <- df[c('CityLocation', 'AmountInUSD')]  # You might want to include more features
y <- df$InvestmentType
# Encode categorical features using one-hot encoding
X <- model.matrix(~ CityLocation + AmountInUSD - 1, data = X)


# Split the data into training and testing sets
set.seed(42)  # for reproducibility
splitIndex <- createDataPartition(y, p = 0.8, list = FALSE)
X_train <- X[splitIndex, ]
X_test <- X[-splitIndex, ]
y_train <- y[splitIndex]
y_test <- y[-splitIndex]

# Create and train the k-NN model
k <- 3  # You may need to tune this parameter
knn_model <- knn(train = X_train, test = X_test, cl = y_train, k = k)

# Predictions on the test set
predictions <- knn_model
actual <- as.factor(y_test)  # Ensure 'actual' is a factor

# Convert levels to lowercase and remove spaces
actual <- tolower(gsub(" ", "", as.character(actual)))
predictions <- tolower(gsub(" ", "", as.character(predictions)))

# Make them factors again
actual <- as.factor(actual)
predictions <- as.factor(predictions)

# Create a confusion matrix
conf_matrix <- confusionMatrix(predictions, actual)

# Plot the confusion matrix
plot_conf_matrix <- function(conf_matrix) {
  ggplot(data = as.data.frame(as.table(conf_matrix)), aes(x = Reference,
                                                          y = Prediction)) +
    geom_tile(aes(fill = Freq), color = "white") +
    scale_fill_gradient(low = "white", high = "blue") +
    geom_text(aes(label = sprintf("%d", Freq)), vjust = 1) +
    labs(title = "Confusion Matrix",
         x = "Actual",
         y = "Predicted") +
    theme_minimal()
}

# Plot the confusion matrix
plot_conf_matrix(conf_matrix)
```

In this code snippet, three essential libraries—ggplot2, caret, and class—are loaded. The dataset is then loaded from a CSV file ("startup_funding.csv"). Rows with missing values are dropped for simplicity. Categorical variables, specifically the 'Investment Type,' are encoded as factors. Features and the target variable are selected, and categorical features are further encoded using one-hot encoding. The dataset is split into training and testing sets, with a random seed set for reproducibility. A k-Nearest Neighbours (k-NN) model is created and trained using the training set, and predictions are made on the test set. The levels of the actual and predicted outcomes are converted to lowercase and without spaces, and confusion matrix analysis is performed. Finally, a heatmap of the confusion matrix is plotted using ggplot2, providing a visual representation of the model's performance.



Random Forest Model for Startup Funding Category Prediction

```
#load necessary packages
library(randomForest)
library(randomForestExplainer)
# Read your dataset
data <- read.csv('startup_funding.csv')
data.frame(data)
# Convert SubVertical to a factor
data$SubVertical <- as.factor(data$SubVertical)

# Split the data into training and testing sets
set.seed(42)  # Set seed for reproducibility
split_index <- sample(2, nrow(data), replace = TRUE, prob = c(0.8, 0.2))

train_data <- data[split_index == 1, ]
test_data <- data[split_index == 2, ]

# Check the levels of SubVertical in the training dataset
table(train_data$SubVertical)

# Combine levels with low counts into a new level called "Other"
train_data$SubVertical <- ifelse(train_data$SubVertical %in% c("Level1",
                                                               "Level2"),
                                 "Other", train_data$SubVertical)

# Define the Random Forest model
rf_model <- randomForest(SubVertical ~ ., data = train_data, ntree = 100,
                         importance = TRUE)
```

```r
# Split the data into training and testing sets
set.seed(42)  # Set seed for reproducibility
split_index <- sample(2, nrow(data), replace = TRUE, prob = c(0.8, 0.2))

train_data <- data[split_index == 1, ]
test_data <- data[split_index == 2, ]

# Check the levels of SubVertical in the training dataset
table(train_data$SubVertical)

# Combine levels with low counts into a new level called "Other"
train_data$SubVertical <- ifelse(train_data$SubVertical %in% c("Level1",
                                                               "Level2"),
                                 "Other", train_data$SubVertical)

# Define the Random Forest model
rf_model <- randomForest(SubVertical ~ ., data = train_data, ntree = 100,
                         importance = TRUE)

# Make predictions on the test set
predictions <- predict(rf_model, newdata = test_data)
```

```r
# Split the data into training and testing sets
set.seed(42)  # Set seed for reproducibility
split_index <- sample(2, nrow(data), replace = TRUE, prob = c(0.8, 0.2))

train_data <- data[split_index == 1, ]
test_data <- data[split_index == 2, ]

# Check the levels of SubVertical in the training dataset
table(train_data$SubVertical)

# Combine levels with low counts into a new level called "Other"
train_data$SubVertical <- ifelse(train_data$SubVertical %in% c("Level1",
                                                               "Level2"),
                 |               "Other", train_data$SubVertical)

# Define the Random Forest model
rf_model <- randomForest(SubVertical ~ ., data = train_data, ntree = 100,
                         importance = TRUE)

# Make predictions on the test set
predictions <- predict(rf_model, newdata = test_data)

# Evaluate the model
conf_matrix <- table(predictions, test_data$SubVertical)
accuracy <- sum(diag(conf_matrix)) / sum(conf_matrix)
cat("Accuracy:", accuracy, "\n")

# Display variable importance
cat("Variable Importance:\n")
print(rf_model$importance)
```

The provided R code is aimed at building and evaluating a Random Forest model on a dataset ('startup_funding.csv'). After loading essential packages such as 'randomForest' and 'randomForestExplainer,' the data is read and the 'Subvertical' column is converted to a factor. The dataset is then split into training and testing sets, with levels of the 'Subvertical' variable checked in the training set. To handle low-count levels, certain categories are combined into a new level named "Other." The Random Forest model is defined using the 'randomForest' function with 100 trees, and variable importance is computed. Predictions are made on the test set, and model accuracy is evaluated using a confusion matrix. Finally, the code displays the variable importance of features in the model. This script provides a comprehensive workflow for building, evaluating, and interpreting a Random Forest model for predicting startup funding categories in the given dataset.

```
> accuracy <- sum(diag(conf_matrix)) / sum(conf_matrix)
> cat("Accuracy:", accuracy, "\n")
Accuracy: 0.004175365
> # Display variable importance
> cat("Variable Importance:\n")
Variable Importance:
> print(rf_model$importance)
                    %IncMSE IncNodePurity
SNo              182238.43895     185788345
Date              -2041.88856      26742721
StartupName         731.22755      31083300
IndustryVertical  17360.59513      36913273
CityLocation        631.17501      17015527
InvestorsName     -2730.97183      26232695
InvestmentType      260.28103       5641252
AmountInUSD         -66.99957      17960642
Remarks           13942.37219      25340821
> 
```

Conclusion

In conclusion, the provided R script conducts a thorough analysis of a startup funding dataset through a combination of exploratory data analysis (EDA), data visualization, and machine learning techniques. The script begins by exploring temporal funding trends from 2015 to 2017, delving into industry vertical distributions, preferred city locations, and investor type preferences. Visualizations, primarily generated using the ggplot2 library, offer insights into funding patterns over time, industry vertical preferences, city-wise funding distributions, and investor type trends. The code also features data cleaning and transformation steps, including the standardization of startup names and the splitting of the 'Investors Name' column for further analysis.

Furthermore, the script incorporates machine learning models such as Random Forest and k-NN. The Random Forest algorithm predicts the 'Subvertical' category based on various features, providing insights into feature importance. Additionally, k-NN is applied to predict 'Investment Type,' and a confusion matrix is visualized to assess model performance.

The code also showcases feature engineering by converting the 'Date' column into a proper date format and introducing new temporal features. It addresses data cleaning challenges, such as standardizing startup names and splitting investor names for more granular analysis.

Lastly, the analysis highlights the top cities by total investment, providing a clear visualization of the investment landscape. Overall, this script not only uncovers valuable insights into startup funding trends but also sets the stage for further exploration and refinement of the analysis process.

References

1.Dataset: https://www.kaggle.com/datasets/sudalairajkumar/indian-startup-funding

2.R:chromeextension://efaidnbmnnnibpcajpcglclefindmkaj/https://web.itu.edu.tr/~tokerem/The_Book_of_R.pdf

3.Resourse:https://www.udemy.com/course/the-r-programming-foreveryoneazcomprehensive-bootcamp/learn/lecture/28712728#overview