*This is the main submission document**. Save and rename this document filename with your registered full name as Prefix before submission.** Submit both this word document and PDF format.*

Class	Seminar 5
Full Name	Stephen Michael Lee
University Email	SLEE155@E.NTU.EDU.SG

** : Delete and replace as appropriate.*

Declaration of Academic Integrity

By submitting this assignment for assessment, I declare that this submission is my own work, unless otherwise quoted, cited, referenced or credited. I have read and understood the Instructions to CBA.PDF provided and the Academic Integrity Policy.

I am aware that failure to act in accordance with the University's Academic Integrity Policy may lead to the imposition of penalties which may include the requirement to revise and resubmit an assignment, receiving a lower grade, or receiving an F grade for the assignment; suspension from the University or termination of my candidature.

I consent to the University copying and distributing any or all of my work in any form and using third parties to verify whether my work contains plagiarised material, and for quality assurance purposes.

Please insert an "X" within the square bracket below to indicate your selection.

[X] I have read and accept the above.

Table of Contents

Declaration on Use of GenAI (Generative Artificial Intelligence)[1]

I _____Stephen Michael Lee_____ (student name),

___SLEE155_____@e.ntu.edu.sg (NTU email) honestly and sincerely make the following declaration in relation to the following course submission:

1. Name of course: Analytics I: Visual and Predictive Analytics
2. Course Code: BC2406
3. Title of Assignment/Project Submission: CBA

In relation to the foregoing, I hereby declare that fully and properly in accordance with the Assignment/Project Instructions, I have **(insert an "X" in the relevant square bracket)**:

i. Used GenAI as permitted to assist in generating key ideas. **[ ]**

ii. Used GenAI as permitted to assist in generating a first text. **[ ]**

And/Or
iii. Used GenAI to refine syntax and grammar for correct language submission. **[ ]**

Or

iv. Did not use GenAI in any way. **[X]**

I also declare that I have:

a. Fully and honestly submitted the digital paper trail required under the assignment/project instructions in the appendix of this document; and that
b. Wherever GenAI assistance has been employed in the submission in word or paraphrase or inclusion of a significant idea or fact suggested by the GenAI, I have acknowledged this by a footnote or in-text reference; and that,
c. Apart from the foregoing notices, the submission is wholly my own work.

Stephen Michael Lee — 25/10/25

…… ……..

Student Name and Signature — Date

[1] GenAI cannot be used to produce the entire submission. Even with GenAI outputs, internet research or discussions, the student must refine the work on his/her own so that the submission is substantively the student's work. The list of all the prompts given to the GenAI and the GenAI outputs must be listed in the appendix of this document.

Answer to Q1:

Part 1: Data Exploration

I began by installing the relevant packages and importing the data. I used the na.strings parameter to potentially catch any missing values/NA in the dataset. This ensures that data quality issues are detected early.

```
> # Packages ------------------------------------------------------------------
> library(data.table)
> library(caTools)
> library(rpart)
> library(rpart.plot)
> library(ggplot2)
> library(corrplot)
> library(hms)
>
> # Import Data ---------------------------------------------------------------
> setwd("~/Desktop/CBA Question Paper")
> misinformation.dt <- fread("misinformation2.csv", header = T, na.strings = c("NA", "na", "N/A", "",
+                                                                               ".", "m", "M"))
```

I did an initial exploratory data analysis (EDA) to better understand the structure, data types and summary statistics of all variables. No obvious data quality anomalies were found.

```
> # EDA Before Preprocessing --------------------------------------------------
> ## General Info
> class(misinformation.dt)
[1] "data.table" "data.frame"
> summary(misinformation.dt)
       id          platform           timestamp                      date
 Min.   :  1.0   Length:500         Min.   :2024-01-01 22:35:00   Length:500
 1st Qu.:125.8   Class :character   1st Qu.:2024-05-14 07:02:15   Class :character
 Median :250.5   Mode  :character   Median :2024-10-05 08:25:30   Mode  :character
 Mean   :250.5                      Mean   :2024-10-18 05:07:12
 3rd Qu.:375.2                      3rd Qu.:2025-03-29 12:24:15
 Max.   :500.0                      Max.   :2025-08-20 20:02:00
     time              month              weekday            country             city
 Length:500         Length:500         Length:500         Length:500         Length:500
 Class :character   Class :character   Class :character   Class :character   Class :character
 Mode  :character   Mode  :character   Mode  :character   Mode  :character   Mode  :character

  timezone          author_id         author_followers author_verified  text_length
Length:500         Length:500         Min.   :   146   Min.   :0.000   Min.   : 20.0
Class :character   Class :character   1st Qu.:256249   1st Qu.:0.000   1st Qu.: 88.0
Mode  :character   Mode  :character   Median :493295   Median :0.000   Median :155.0
                                      Mean   :505746   Mean   :0.498   Mean   :151.6
                                      3rd Qu.:768602   3rd Qu.:1.000   3rd Qu.:215.2
                                      Max.   :998936   Max.   :1.000   Max.   :280.0
 token_count    readability_score    num_urls     num_mentions    num_hashtags
Min.   : 3.00   Min.   :30.00     Min.   :0.00   Min.   :0.000   Min.   :0.000
1st Qu.:18.00   1st Qu.:41.00     1st Qu.:0.00   1st Qu.:1.000   1st Qu.:1.000
Median :32.00   Median :55.56     Median :2.00   Median :3.000   Median :3.000
Mean   :35.47   Mean   :54.69     Mean   :1.49   Mean   :2.538   Mean   :2.572
3rd Qu.:47.25   3rd Qu.:68.16     3rd Qu.:2.00   3rd Qu.:4.000   3rd Qu.:4.000
Max.   :92.00   Max.   :79.92     Max.   :3.00   Max.   :5.000   Max.   :5.000
sentiment_score     toxicity_score   detected_synthetic_score external_factchecks_count
Min.   :-1.000000   Min.   :0.0010   Min.   :0.0010           Min.   :0.000
1st Qu.:-0.509250   1st Qu.:0.2557   1st Qu.:0.2285           1st Qu.:1.000
Median : 0.034000   Median :0.5045   Median :0.4780           Median :2.000
Mean   : 0.003634   Mean   :0.4959   Mean   :0.4854           Mean   :2.006
3rd Qu.: 0.508500   3rd Qu.:0.7465   3rd Qu.:0.7308           3rd Qu.:3.000
Max.   : 0.999000   Max.   :0.9970   Max.   :0.9940           Max.   :5.000
source_domain_reliability   engagement   is_misinformation
Min.   :0.026             Min.   :   4   Min.   :0.000
1st Qu.:3.232             1st Qu.:3049   1st Qu.:0.000
Median :5.577             Median :5686   Median :1.000
Mean   :5.312             Mean   :5397   Mean   :0.536
3rd Qu.:7.151             3rd Qu.:7893   3rd Qu.:1.000
Max.   :9.996             Max.   :9977   Max.   :1.000
```

The dataset contains 500 observations and 26 variables. Noticed some variables in improper formats (date, time) and categorical variables are not factors (platform, is_misinformation, etc.). These are to be converted later.

```
> dim(misinformation.dt) # 500 rows & 26 columns
[1] 500  26
> str(misinformation.dt)
Classes ‘data.table’ and 'data.frame':  500 obs. of  26 variables:
 $ id                        : int  1 2 3 4 5 6 7 8 9 10 ...
 $ platform                  : chr  "Reddit" "Reddit" "Telegram" "Twitter" ...
 $ timestamp                 : POSIXct, format: "2024-03-06 10:01:00" "2025-08-07 18:30:00" "2024-12-13 03:15:00" ...
 $ date                      : chr  "6 03 2024" "7 08 2025" "13 12 2024" "5 04 2024" ...
 $ time                      : chr  "10:01:00" "18:30:00" "3:15:00" "7:10:00" ...
 $ month                     : chr  "March" "August" "December" "April" ...
 $ weekday                   : chr  "Wednesday" "Thursday" "Friday" "Friday" ...
 $ country                   : chr  "USA" "Germany" "USA" "USA" ...
 $ city                      : chr  "New York" "Berlin" "New York" "Chicago" ...
 $ timezone                  : chr  "EST" "CET" "EST" "EST" ...
 $ author_id                 : chr  "A1117" "A7669" "A9786" "A8886" ...
 $ author_followers          : int  74491 199709 470455 224180 63968 439639 651382 977067 869684 610855 ...
 $ author_verified           : int  0 0 0 1 1 1 0 0 1 0 ...
 $ text_length               : int  137 144 118 228 131 146 57 174 46 96 ...
 $ token_count               : int  34 24 29 38 43 48 19 58 15 32 ...
 $ readability_score         : num  44 68.6 68.6 46.5 74.4 ...
 $ num_urls                  : int  1 0 3 0 0 2 0 1 0 3 ...
 $ num_mentions              : int  0 5 1 1 0 1 4 2 0 1 ...
 $ num_hashtags              : int  0 3 2 3 4 5 1 0 2 0 ...
 $ sentiment_score           : num  -0.223 -0.718 -0.989 -0.283 -0.378 -0.761 0.045 -0.371 -0.542 0.267 ...
 $ toxicity_score            : num  0.271 0.802 0.815 0.116 0.325 0.713 0.428 0.509 0.077 0.871 ...
 $ detected_synthetic_score : num  0.829 0.075 0.707 0.863 0.73 0.761 0.025 0.908 0.29 0.804 ...
 $ external_factchecks_count: int  0 5 2 1 1 0 3 1 2 2 ...
 $ source_domain_reliability: num  1.323 5.51 5.474 0.909 7.623 ...
 $ engagement                : int  3899 7651 7260 7454 8320 4351 5718 4616 7239 5862 ...
 $ is_misinformation         : int  1 0 0 1 0 1 0 0 1 0 ...
 - attr(*, ".internal.selfref")=<externalptr>
```

I have checked for duplicated rows and NA values, but there were none.

```
> ## Check duplicated rows
> sum(duplicated(misinformation.dt)) # no duplicated rows
[1] 0
>
> ## Check NA values
> sum(is.na(misinformation.dt)) # no NA
[1] 0
```

Checked for some not obvious categorical variables. There are 4 platforms, 5 countries, 15 cities and 5 time zones.

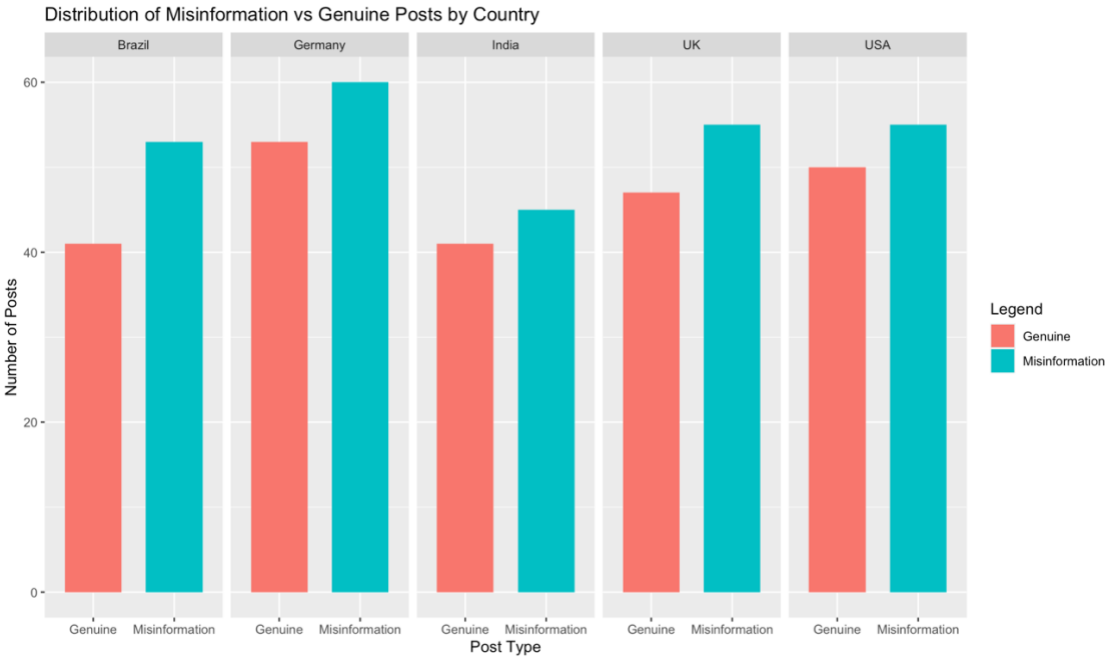
```
> unique(misinformation.dt$platform)
[1] "Reddit"   "Telegram" "Twitter"  "Facebook"
> unique(misinformation.dt$country)
[1] "USA"     "Germany" "India"   "UK"      "Brazil"
> unique(misinformation.dt$city)
 [1] "New York"       "Berlin"         "Chicago"        "Hamburg"        "Delhi"
 [6] "Bangalore"      "Mumbai"         "London"         "Sao Paulo"      "Manchester"
[11] "Birmingham"     "Brasilia"       "Los Angeles"    "Munich"         "Rio de Janeiro"
> unique(misinformation.dt$timezone)
[1] "EST" "CET" "IST" "GMT" "BRT"
```

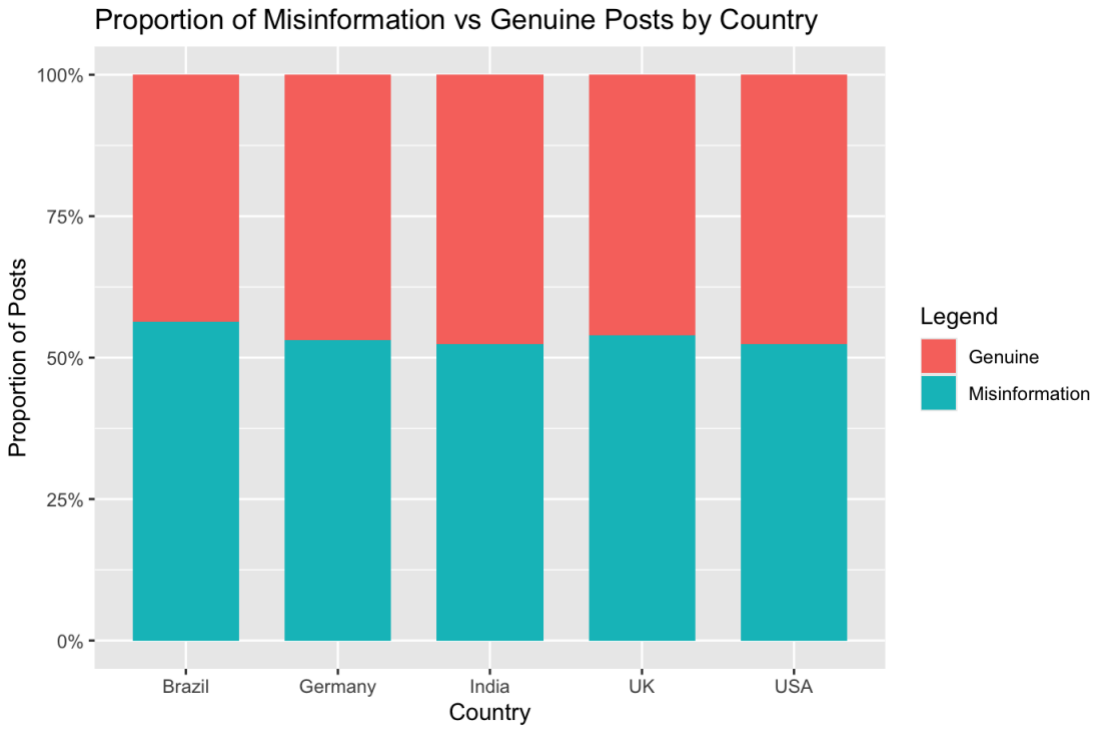
Posts span from January 2024 to August 2025, providing roughly 20 months’ worth of data.

```
> min(misinformation.dt$timestamp)
[1] "2024-01-01 22:35:00 UTC"
> max(misinformation.dt$timestamp)
[1] "2025-08-20 20:02:00 UTC"
```

Part 2: General Analysis

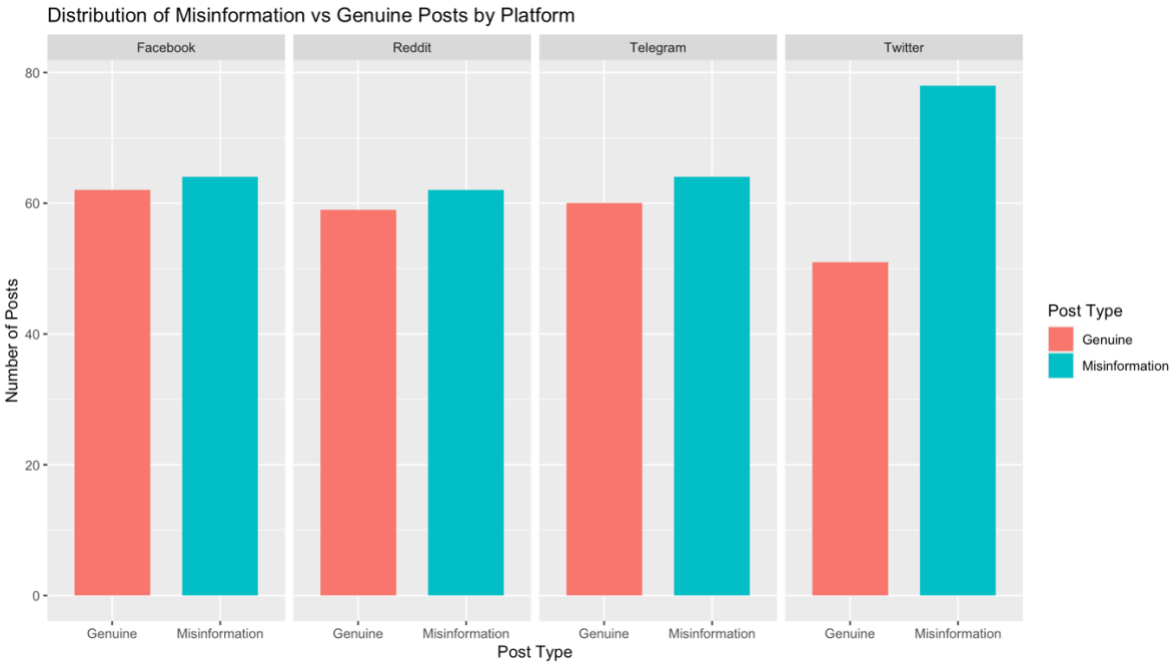
To further examine how misinformation varies across regions, I analysed the distribution of posts by country. Across all five countries, misinformation consistently accounts for more than 50% of the total posts. This pattern shows that misinformation is a widespread global issue and not confined to any single region or cultural context. Drivers of misinformation transcend geographical boundaries and are likely to be influenced by other factors.

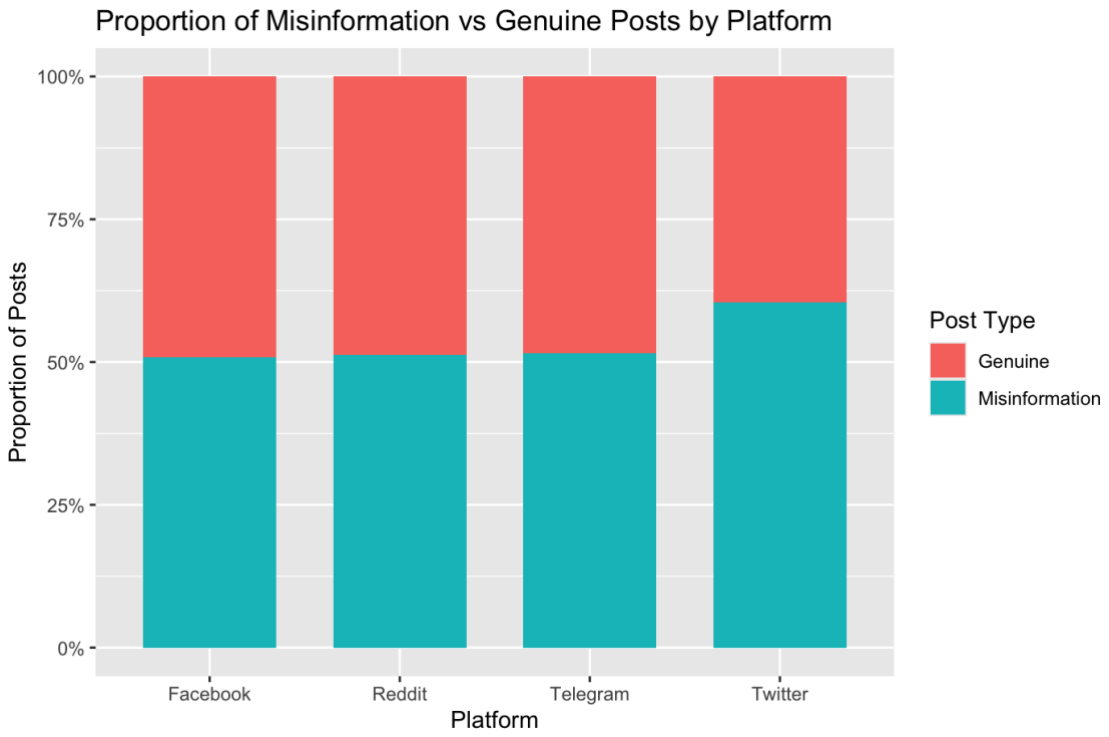




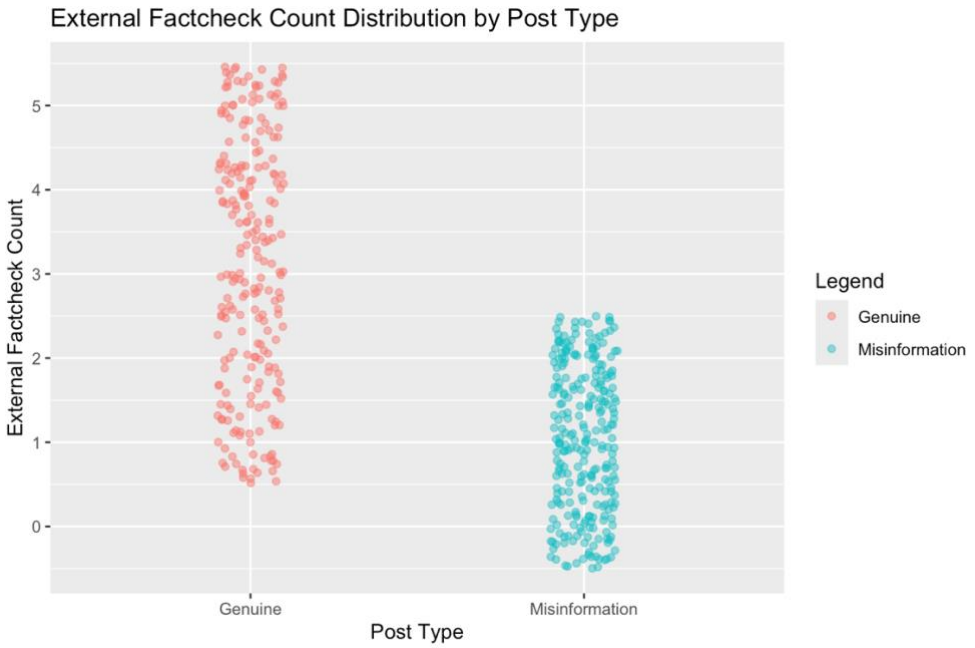
Part 3: Notable Findings

1. When comparing across social media platforms, Twitter showed the highest proportion of misinformation posts. Other platforms like Facebook, Reddit and Telegram have more balanced distributions between genuine and misinformation posts. This suggests that misinformation may spread more rapidly or gain greater visibility on Twitter, possibly due to its public posting system, retweet feature and trending algorithms that amplify viral content. In contrast, platforms with more private or moderated spaces (e.g., Reddit or Telegram) might inherently restrict the uncontrolled spread of such content.





2. The jitter plot revealed that misinformation posts were associated with significantly fewer external verifications. In other words, false or misleading information tends to escape detection or correction, remaining unverified compared to genuine content. This finding underscores a potential gap in fact-checking coverage. Posts that require verification most urgently may be the least scrutinised. Genuine posts, on the other hand, are more likely to be linked with credible sources that have been verified multiple times.



3. The boxplot clearly showed that misinformation posts mainly came from low-reliability domains, while genuine posts clustered around higher reliability scores. The median reliability score for misinformation sources was substantially lower, indicating a strong negative relationship between source credibility and misinformation likelihood. This pattern suggests that source credibility plays a central role in determining content trustworthiness.

Answer to Q2:

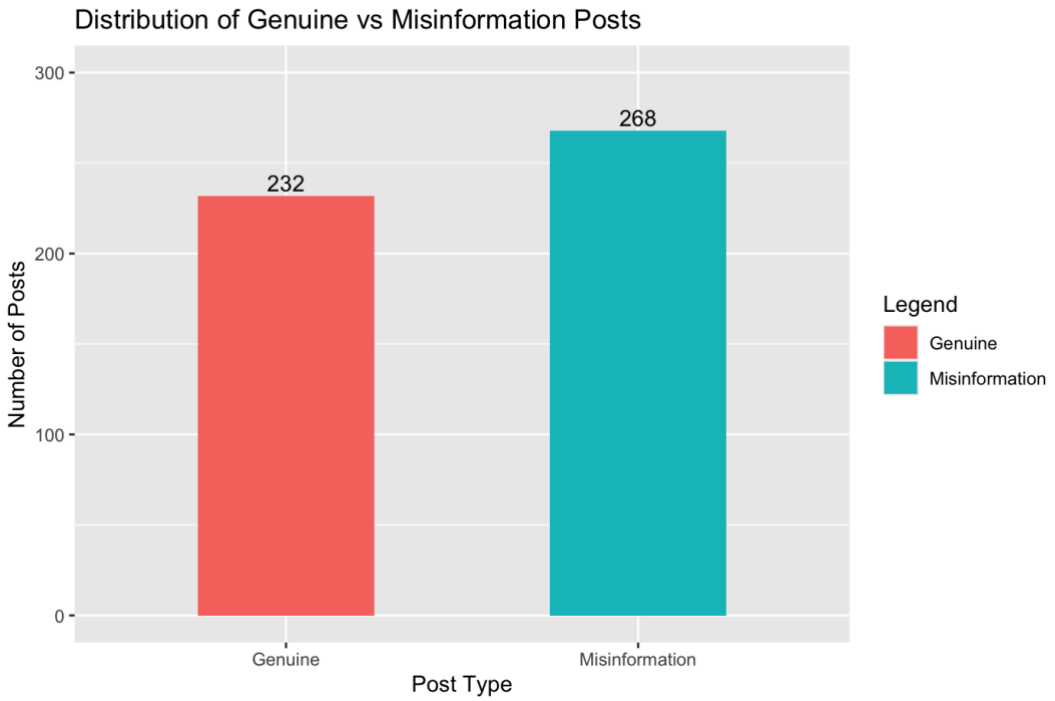
In the original dataset, some variables such as date and time were stored as characters, so I reformatted them into proper date/time formats. In addition, categorical variables were converted into factor variables to facilitate modelling and improve interpretability. I also converted binary variables like author_verified and is_misinformation into factors and ordered nominal variables such as month and weekday based on their natural flow.

```
> # Clean Date & Time ------------------------------------------------------------
> misinformation.dt[, cleaned_date := as.Date(date, format = "%d %m %Y")]
> misinformation.dt[, cleaned_time := as_hms(time)]
> class(misinformation.dt$cleaned_date)
[1] "Date"
> class(misinformation.dt$cleaned_time)
[1] "hms"      "difftime"
> # Factorize --------------------------------------------------------------------
> ## Binary 0/1
> misinformation.dt[, author_verified := factor(author_verified, levels = c(0, 1), labels = c("Not Verified", "Verified"))]
> misinformation.dt[, is_misinformation := factor(is_misinformation, levels = c(0, 1), labels = c("Genuine", "Misinformation"))]
>
> ## Text / Nominal
> factor_cols <- c("platform", "month", "weekday", "country", "city", "timezone")
> misinformation.dt[, (factor_cols) := lapply(.SD, factor), .SDcols = factor_cols]
>
> ## Order factors for readability
> misinformation.dt[, month := factor(month, levels = month.name, ordered = T)]
> misinformation.dt[, weekday := factor(weekday, levels = c("Monday","Tuesday","Wednesday","Thursday","Friday","Saturday","Sunday"),
ordered = T)]
```

After preprocessing, the original dataset expanded to 28 columns, as 2 new columns (cleaned_date and cleaned_time) were added to retain both original and reformatted date-time data.

```
> dim(misinformation.dt) # 500 rows and 28 columns
[1] 500  28
```

Before using models, I also wanted to analyse the distribution of post types to check if the dataset is balanced. I found that there were 232 genuine posts (46.4%) and 268 misinformation posts (53.6%), which is quite even.



Next, I dropped the unnecessary variables as there would be a multi-collinearity effect, also they bring no extra value to my models later. Especially unique identifier columns like id and author_id. Removed city because it is too granular and timezone as its determined by country already. Removed time-date columns as they only indicate when a post is made, doesn't classify whether is misinformation.

```
> drop_cols <- c("id", "author_id", "timestamp", "date", "time", "cleaned_date", "cleaned_time", "time
zone", "city", "month", "weekday")
> model.dt <- misinformation.dt[, !drop_cols, with = F]
```

For my final dataset, I have 500 rows and 17 columns. This will be the dataset I use to train my models.

```
> dim(model.dt)
[1] 500  17
```

These are the variables in my model.dt.

```
> summary(model.dt)
     platform      country    author_followers     author_verified  text_length     token_count    readability_score
 Facebook:126   Brazil : 94   Min.   :   146   Not Verified:251    Min.   : 20.0   Min.   : 3.00   Min.   :30.00
 Reddit  :121   Germany:113   1st Qu.:256249   Verified    :249    1st Qu.: 88.0   1st Qu.:18.00   1st Qu.:41.00
 Telegram:124   India  : 86   Median :493295                       Median :155.0   Median :32.00   Median :55.56
 Twitter :129   UK     :102   Mean   :505746                       Mean   :151.6   Mean   :35.47   Mean   :54.69
                USA    :105   3rd Qu.:768602                       3rd Qu.:215.2   3rd Qu.:47.25   3rd Qu.:68.16
                               Max.   :998936                       Max.   :280.0   Max.   :92.00   Max.   :79.92
    num_urls     num_mentions    num_hashtags   sentiment_score     toxicity_score   detected_synthetic_score
 Min.   :0.00   Min.   :0.000   Min.   :0.000   Min.   :-1.000000   Min.   :0.0010   Min.   :0.0010
 1st Qu.:0.00   1st Qu.:1.000   1st Qu.:1.000   1st Qu.:-0.509250   1st Qu.:0.2557   1st Qu.:0.2285
 Median :2.00   Median :3.000   Median :3.000   Median : 0.034000   Median :0.5045   Median :0.4780
 Mean   :1.49   Mean   :2.538   Mean   :2.572   Mean   : 0.003634   Mean   :0.4959   Mean   :0.4854
 3rd Qu.:2.00   3rd Qu.:4.000   3rd Qu.:4.000   3rd Qu.: 0.508500   3rd Qu.:0.7465   3rd Qu.:0.7308
 Max.   :3.00   Max.   :5.000   Max.   :5.000   Max.   : 0.999000   Max.   :0.9970   Max.   :0.9940
 external_factchecks_count source_domain_reliability   engagement       is_misinformation
 Min.   :0.000             Min.   :0.026             Min.   :   4   Genuine       :232
 1st Qu.:1.000             1st Qu.:3.232             1st Qu.:3049   Misinformation:268
 Median :2.000             Median :5.577             Median :5686
 Mean   :2.006             Mean   :5.312             Mean   :5397
 3rd Qu.:3.000             3rd Qu.:7.151             3rd Qu.:7893
 Max.   :5.000             Max.   :9.996             Max.   :9977
```

This procedure retains the master dataset, misinformation.dt and creates a separate modelling dataset, model.dt. This ensures traceability and flexibility for further exploration or future feature engineering.

Answer to Q3:

I set seed as 123 for reproducibility and created a train-test split.

```
> # Train-test Split ------------------------------------------------------------
> set.seed(123)
> train <- sample.split(model.dt$is_misinformation, SplitRatio = 0.7)
> trainset <- model.dt[train == T]
> testset <- model.dt[train == F]
```

There doesn't seem to be any overwhelming amount of each. The trainset is relatively balanced, so I proceeded without further action.

```
> summary(trainset$is_misinformation)
       Genuine Misinformation
           162            188
```

Did the first regression with all variables.

```
> ## Model 1 (Full)
> m1 <- glm(is_misinformation ~ ., data = trainset, family = binomial)
> summary(m1)

Coefficients:
                            Estimate Std. Error z value Pr(>|z|)
(Intercept)                1.067e+01  2.365e+00   4.511 6.44e-06 ***
platformReddit            -6.346e-02  8.712e-01  -0.073    0.942
platformTelegram           1.208e-01  7.702e-01   0.157    0.875
platformTwitter            1.134e+00  8.083e-01   1.403    0.161
countryGermany             5.261e-01  8.451e-01   0.623    0.534
countryIndia               5.991e-01  8.666e-01   0.691    0.489
countryUK                  1.116e+00  8.630e-01   1.293    0.196
countryUSA                 5.840e-01  7.966e-01   0.733    0.463
author_followers          -3.979e-07  9.489e-07  -0.419    0.675
author_verifiedVerified    4.160e-01  5.688e-01   0.731    0.465
text_length               -5.994e-03  6.947e-03  -0.863    0.388
token_count                2.228e-02  2.456e-02   0.907    0.364
readability_score          9.639e-03  1.749e-02   0.551    0.581
num_urls                   1.304e-02  2.312e-01   0.056    0.955
num_mentions               1.304e-01  1.614e-01   0.808    0.419
num_hashtags               4.581e-02  1.631e-01   0.281    0.779
sentiment_score            4.812e-02  4.573e-01   0.105    0.916
toxicity_score             4.536e-01  9.497e-01   0.478    0.633
detected_synthetic_score  -9.324e-01  9.519e-01  -0.980    0.327
external_factchecks_count -1.896e+00  3.435e-01  -5.521 3.37e-08 ***
source_domain_reliability -1.514e+00  2.540e-01  -5.960 2.52e-09 ***
engagement                 4.391e-05  9.777e-05   0.449    0.653
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 483.27  on 349  degrees of freedom
Residual deviance: 108.75  on 328  degrees of freedom
AIC: 152.75

Number of Fisher Scoring iterations: 8
```

Only external_factchecks_count and source_domain_reliabilty were significant.

Next, I tried another regression model by only keeping the 2 significant factors.

```
> ## Model 2 (Simplified, keep significant - manual)
> m2 <- glm(is_misinformation ~ external_factchecks_count + source_domain_reliability, data = trainse
t, family = binomial)
> summary(m2)

Call:
glm(formula = is_misinformation ~ external_factchecks_count +
    source_domain_reliability, family = binomial, data = trainset)

Coefficients:
                          Estimate Std. Error z value Pr(>|z|)
(Intercept)                11.3562     1.4870   7.637 2.23e-14 ***
external_factchecks_count  -1.6912     0.2822  -5.993 2.06e-09 ***
source_domain_reliability  -1.3930     0.2125  -6.555 5.57e-11 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 483.27  on 349  degrees of freedom
Residual deviance: 118.18  on 347  degrees of freedom
AIC: 124.18

Number of Fisher Scoring iterations: 7
```

To reinforce this, I tried another model using the step function.

```
> ## Model confirmation using step function
> m.test <- glm(is_misinformation ~ ., data = trainset, family = binomial)
> m.step <- step(m.test, direction = "both", trace = T)

> summary(m.step)

Call:
glm(formula = is_misinformation ~ external_factchecks_count +
    source_domain_reliability, family = binomial, data = trainset)

Coefficients:
                          Estimate Std. Error z value Pr(>|z|)
(Intercept)                11.3562     1.4870   7.637 2.23e-14 ***
external_factchecks_count  -1.6912     0.2822  -5.993 2.06e-09 ***
source_domain_reliability  -1.3930     0.2125  -6.555 5.57e-11 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 483.27  on 349  degrees of freedom
Residual deviance: 118.18  on 347  degrees of freedom
AIC: 124.18

Number of Fisher Scoring iterations: 7
```

It returned the same result as my second regression model.

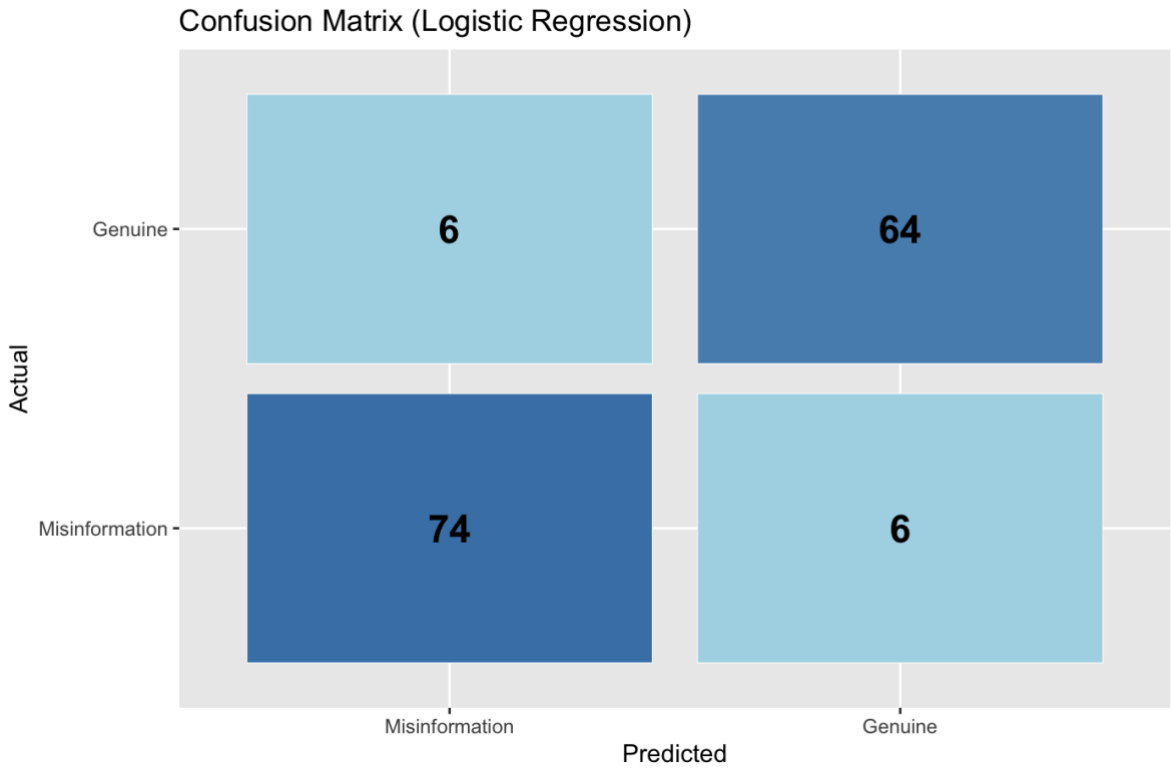
I then used the AIC function to check the AIC for all 3 to find the best logistic regression.

```
> ## AIC
> AIC(m1, m2, m.step)
        df      AIC
m1      22 152.7530
m2       3 124.1758
m.step   3 124.1758
```

From here, we can see that model 2 has the lowest AIC, which means it's the best fit.

Next, I created the confusion matrix on the testset.

```
> ## Confusion Matrix on testset using m2
> prob.test <- predict(m2, newdata = testset, type = "response")
> threshold1 <- 0.5
> m2.predict.test <- ifelse(prob.test > threshold1, "Misinformation", "Genuine")
> conf.mtx.lr <- table(Actual = relevel(testset$is_misinformation, ref = "Misinformation"),
+                      Predicted = relevel(as.factor(m2.predict.test), ref = "Misinformation"), depars
e.level = 2)
> conf.mtx.lr
                Predicted
Actual           Misinformation Genuine
  Misinformation             74       6
  Genuine                     6      64
```



Now, we will be moving on to building the CART model.

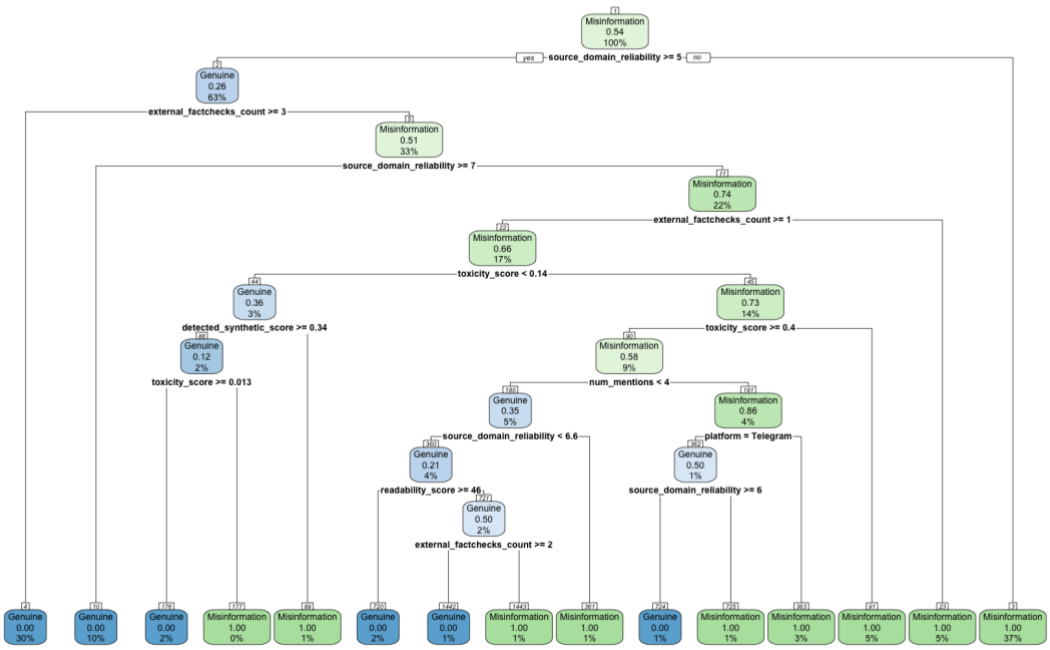
```
> m3 <- rpart(is_misinformation ~ ., data = trainset,
+             method = "class", control = rpart.control(minsplit = 2, cp = 0))
```

I used class method as we are predicting for a binary outcome.

I first grew the tree to a maximal.

```
> printcp(m3)

Classification tree:
rpart(formula = is_misinformation ~ ., data = trainset, method = "class",
    control = rpart.control(minsplit = 2, cp = 0))

Variables actually used in tree construction:
[1] detected_synthetic_score  external_factchecks_count num_mentions             platform
[5] readability_score         source_domain_reliability toxicity_score

Root node error: 162/350 = 0.46286

n= 350

        CP nsplit rel error  xerror     xstd
1 0.6419753      0  1.000000 1.00000 0.057582
2 0.1172840      1  0.358025 0.41975 0.045691
3 0.0123457      3  0.123457 0.13580 0.028028
4 0.0092593      9  0.037037 0.21605 0.034645
5 0.0061728     11  0.018519 0.22222 0.035081
6 0.0000000     14  0.000000 0.23457 0.035927
```

Maximal Tree

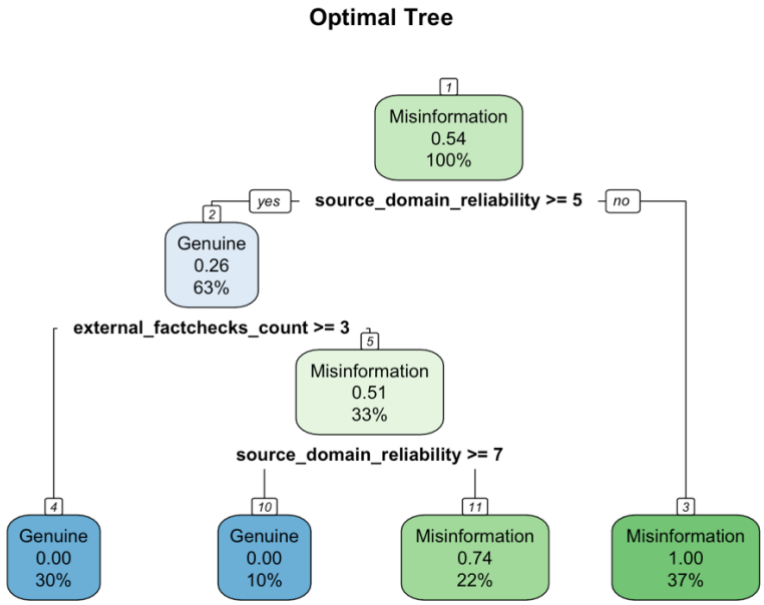


Without pruning, this would be too complex as there is way too many splits. We need to derive the optimal CP before pruning.

```
> ## Compute CP
> CVerror.cap <- m3$cptable[which.min(m3$cptable[, "xerror"]), "xerror"] + m3$cptable[which.min(m3$cptabl
e[, "xerror"]), "xstd"]
>
> i <- 1
> while (m3$cptable[i, 4] > CVerror.cap) {
+   i <- i + 1
+ }
>
> cp1 <- ifelse(i > 1, sqrt(m3$cptable[i, 1] * m3$cptable[i - 1, 1]), 1)
```

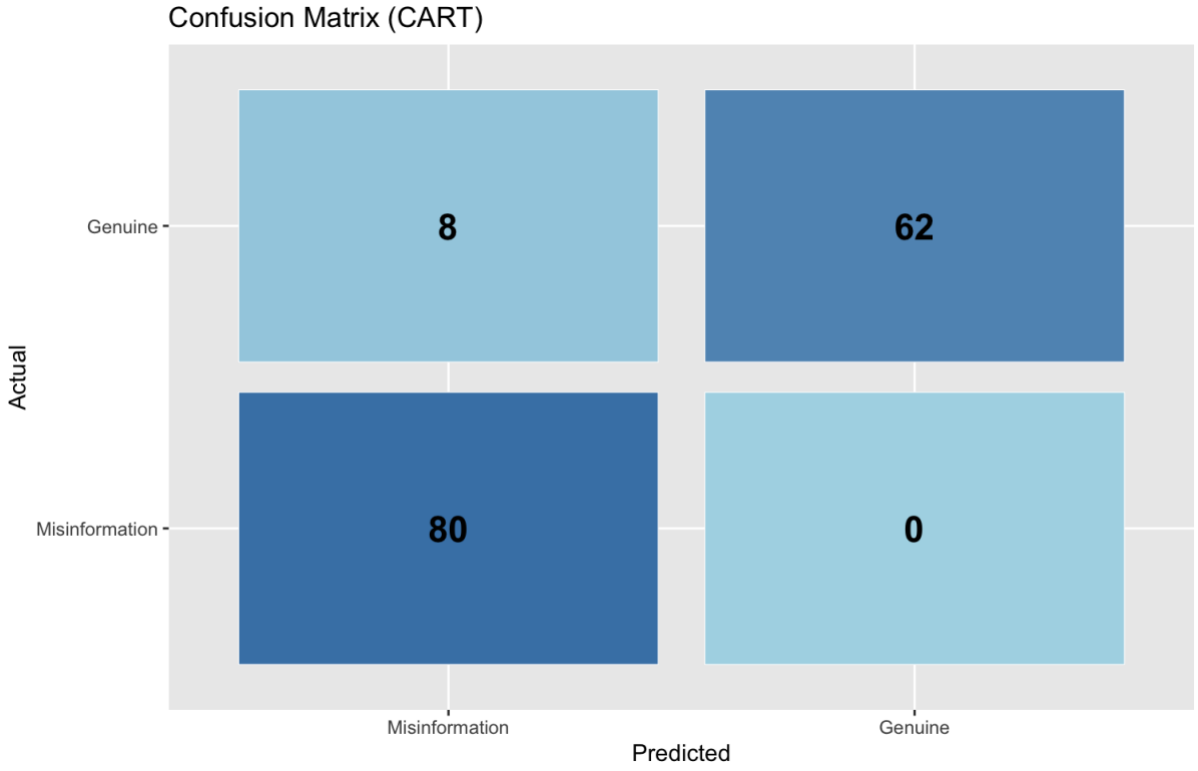
Afterwards, we can derive the optimal tree.

```
> ## Prune Tree
> m3.pruned <- prune(m3, cp = cp1)
```



Now, I'll create the confusion matrix on the test set.

```
> ## Confusion Matrix on testset using m3.pruned
> cart.pred.prob <- predict(m3.pruned, newdata = testset, type = "class")
>
> conf.mtx.cart <- table(Actual = relevel(testset$is_misinformation, ref = "Misinformation"),
+                        Predicted = relevel(cart.pred.prob, ref = "Misinformation"), deparse.level = 2)
> conf.mtx.cart
                Predicted
Actual           Misinformation Genuine
  Misinformation             80       0
  Genuine                     8      62
```



I then computed and tabulated the results from the 2 models into a table.

```
## Calculations
tp.lr <- as.numeric(conf.mtx.lr[1, 1])
fp.lr <- as.numeric(conf.mtx.lr[2, 1])
tn.lr <- as.numeric(conf.mtx.lr[2, 2])
fn.lr <- as.numeric(conf.mtx.lr[1, 2])

accuracy.lr <- (tp.lr + tn.lr) / sum(conf.mtx.lr)
fpr.lr <- fp.lr / (fp.lr + tn.lr)
fnr.lr <- fn.lr / (fn.lr + tp.lr)
overall_error.lr <- 1 - accuracy.lr


## Calculations
tp.cart <- as.numeric(conf.mtx.cart[1, 1])
fp.cart <- as.numeric(conf.mtx.cart[2, 1])
tn.cart <- as.numeric(conf.mtx.cart[2, 2])
fn.cart <- as.numeric(conf.mtx.cart[1, 2])

accuracy.cart <- (tp.cart + tn.cart) / sum(conf.mtx.cart)
fpr.cart <- fp.cart / (fp.cart + tn.cart)
fnr.cart <- fn.cart / (fn.cart + tp.cart)
overall_error.cart <- 1 - accuracy.cart
```

```
> # Tabulation ------------------------------------------------------------------
> ## Logistic Regression metrics
> lr.data <- data.frame(Model = "Logistic Regression",
+                       Model_Complexity = "2 X-variables",
+                       False_Positive_Rate = round(fpr.lr, 3),
+                       False_Negative_Rate = round(fnr.lr, 3),
+                       Overall_Error = round(overall_error.lr, 3))
>
> ## CART metrics
> cart.data <- data.frame(Model = "CART",
+                         Model_Complexity = "4 terminal nodes",
+                         False_Positive_Rate = round(fpr.cart, 3),
+                         False_Negative_Rate = round(fnr.cart, 3),
+                         Overall_Error = round(overall_error.cart, 3))
>
> ## Combine into 1 table
> results.table <- rbind(lr.data, cart.data)
> results.table
                Model Model_Complexity False_Positive_Rate False_Negative_Rate Overall_Error
1 Logistic Regression    2 X-variables              0.086              0.075         0.080
2                CART 4 terminal nodes              0.114              0.000         0.053
```

Below is the table of results for both models.

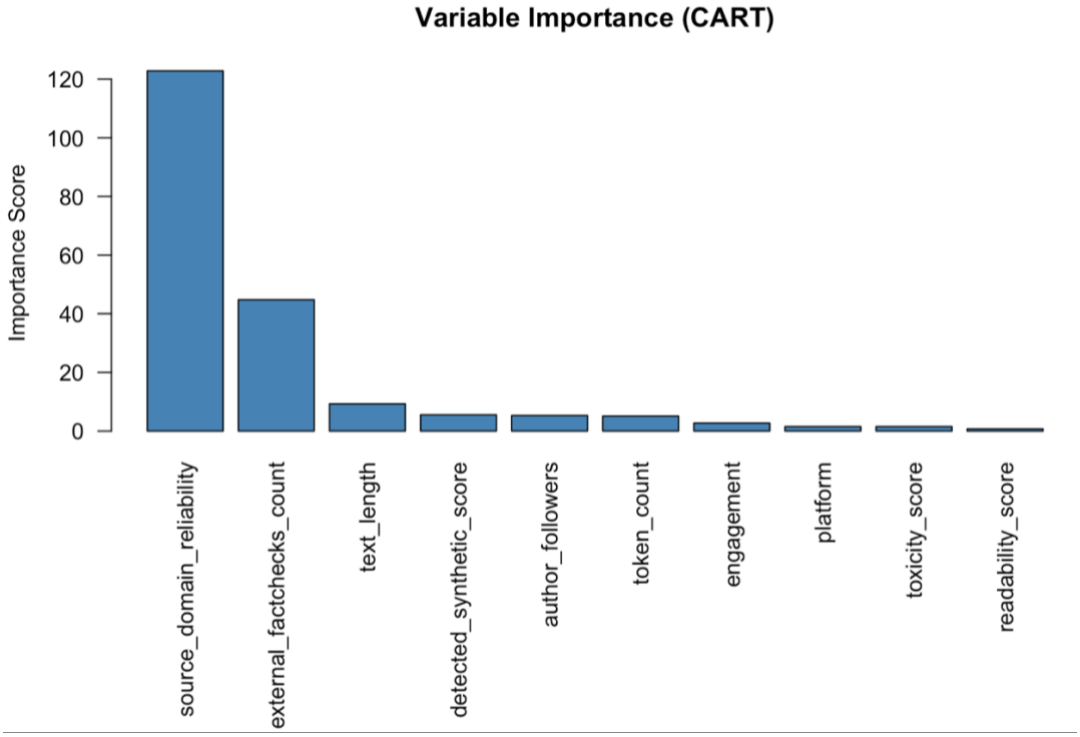
Model	**Model Complexity**	**False Positive Rate**	**False Negative Rate**	**Overall Error**
Logistic Regression	2 X variables	8.6%	7.5%	8%
CART	4 terminal nodes	11.4%	0%	5.3%

Both models performed relatively well, but CART had a slightly lower overall error and managed to achieve a 0% false negative rate despite a higher false positive rate. Though it might have flagged more genuine posts as misinformation, this trade-off may be acceptable when the priority is to minimise missed misinformation.

Answer to Q4:

I first analysed the variable importance of the predictors in the CART model.

```
> # Variable Importance (CART) ------------------------------------------------
> m3.pruned$variable.importance
source_domain_reliability external_factchecks_count              text_length  detected_synthetic_score
                122.8297709                44.7956218                9.2887399                5.5323873
          author_followers               token_count               engagement                 platform
                  5.3170817                 5.1189866                 2.7266494                1.5132704
           toxicity_score         readability_score
                  1.5132704                 0.7566352
```



1. The CART model identified source_domain_reliability (122.8) and external_factchecks_count (44.8) as the 2 most influential predictors for detecting misinformation. These variables contributed the most to reducing classification error, indicating that posts from less reliable domains and those with fewer external fact checks were strong indicators of misinformation.

2. The other variables had minor contributions, suggesting limited predictive power.

Next, I analysed the odds ratio and odds ratio confidence interval for the logistic regression model.

```
> # OR CI ---------------------------------------------------------------------
> OR.m2 <- exp(coef(m2))
> OR.m2
             (Intercept) external_factchecks_count source_domain_reliability
            8.549188e+04              1.842901e-01              2.483268e-01
> OR.CI.m2 <- exp(confint(m2))

> OR.CI.m2
                                2.5 %       97.5 %
(Intercept)                6.487378e+03 2.330380e+06
external_factchecks_count  9.844263e-02 3.008663e-01
source_domain_reliability  1.544507e-01 3.584555e-01
```

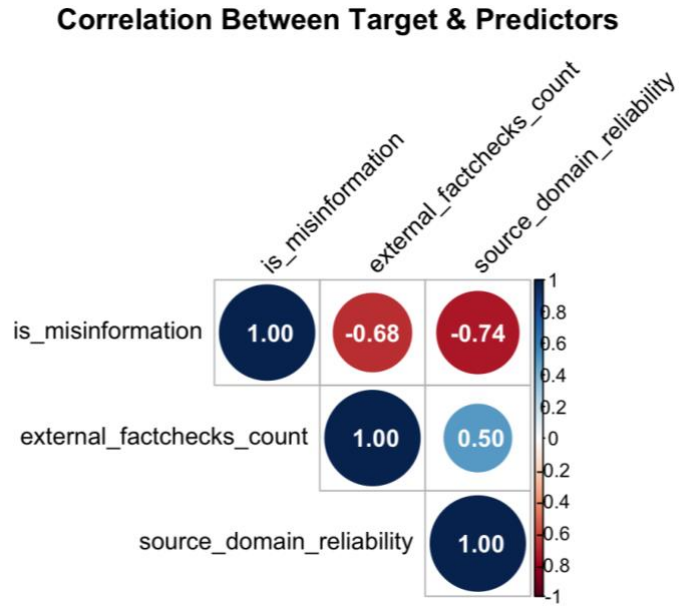
1. We can see that both odds ratios are less than 1, which suggests a negative association. This means that as either fact check count or source reliability increases, the likelihood of a post being misinformation drops.

2. The odds ratio confidence intervals for both do not include 1 in their 95% confidence interval range. This indicates that both predictors are statistically significant contributors to the model.

3. These findings reinforce that source reliability and external fact check counts are decisive factors in detecting misinformation.

Lastly, I did a correlation analysis on the full dataset to better understand the overall relationships between the 2 predictors and the target variable.

```
> # Corr Plot ------------------------------------------------------------------
> ## Convert factor target to numeric (1 = Misinformation, 0 = Genuine)
> misinfo_numeric <- ifelse(misinformation.dt$is_misinformation == "Misinformation", 1, 0)
>
> ## Create a data frame for correlation
> corr.df <- data.frame(is_misinformation = misinfo_numeric,
+                       external_factchecks_count = misinformation.dt$external_factchecks_count,
+                       source_domain_reliability = misinformation.dt$source_domain_reliability)
>
> ## Compute & plot correlation
> corr_matrix <- cor(corr.df, use = "complete.obs")
> corr_matrix
                        is_misinformation external_factchecks_count source_domain_reliability
is_misinformation               1.0000000               -0.6759750               -0.7433622
external_factchecks_count      -0.6759750                1.0000000                0.5010074
source_domain_reliability      -0.7433622                0.5010074                1.0000000
```



1. The results show that is_misinformation is strongly negatively correlated with both predictors, source_domain_reliability (r = - 0.74) and external_factchecks_count (r = - 0.68). This supports the idea that posts from more reliable domains or with more external fact checks are less likely to be misinformation.

2. The 2 predictors show a moderately positive correlation with each other (r = + 0.50). This means that reliable sources may also tend to have more external fact checks.

3. These findings are consistent with both the logistic regression and CART models, reinforcing that source reliability and external fact checking are key factors for misinformation detection.

Answer to Q5:

This study explored the use of machine learning models to detect misinformation using a dataset of 500 social media posts. The goal was to identify key predictors of misinformation and evaluate model performance using logistic regression and CART.

Initial Exploratory Data Analysis confirmed that there were no missing values or duplicate rows. Categorical variables were factorised, and irrelevant ones were dropped after pre-processing. The target variable (is_misinformation) was evenly distributed between genuine and misinformation posts.

Both models demonstrated good predictive ability, with the CART model achieving a lower overall error rate (5.3%) than logistic regression (8%). CART also recorded a 0% false-negative rate, successfully identifying all misinformation posts, though at a slightly higher false-positive rate (11.4%).

The variable importance results showed that source_domain_reliability (122.8) and external_factchecks_count (44.8) were the two most influential features. Other variables such as text_length, detected_synthetic_score, etc. contributed marginally. Odds-ratio analysis supported these findings, with both predictors showing OR < 1, indicating that higher domain reliability or more fact-checks significantly reduce misinformation likelihood.

Correlation analysis revealed strong negative correlations between misinformation and both predictors. r = –0.74 for reliability and r = –0.68 for fact check count, confirming consistent patterns across all methods.

Overall, source_domain_reliability and external_factchecks_count emerged as the strongest indicators of detecting misinformation.

In conclusion, though both models are effective, CART was advantageous in terms of overall accuracy and fewer false negatives (0% false negative rate). This makes it more suitable for practical applications in detecting misinformation even if it flags genuine posts by mistake.

Appendix: List of GenAI Prompts and Outputs

[For each question in the question paper, state the complete list of GenAI prompts used (if any), in which GenAI tool (ChatGPT, Gemini, etc…) and entire GenAI Outputs.]