INSTAGRAM FAKE SPAMMER GENUINE ACCOUNTS

Submitted by – Ishita

Intern ID - UMIP277158

Internship Domain - Data Science Intern

INTRODUCTION

With the growing influence of social media platforms like Instagram, the presence of fake and spammer accounts has become a significant concern. These accounts often exhibit different behavioral patterns compared to genuine users, such as unusual follower counts, posting frequency, bio characteristics, or profile completeness.

This project aims to analyze and classify Instagram accounts into categories like fake, spammer, and genuine based on a variety of features, including:

- Number of posts, followers, and followings
- Presence of profile pictures and external links
- Length and patterns in usernames, full names, and bios
- Account privacy settings (public vs. private)

Using machine learning models such as Logistic Regression and Support Vector Machines (SVM), we evaluate these features to identify accounts that show suspicious or automated behavior.

My goal is to develop a system that not only helps in automatically detecting fake or spam accounts, but also enhances the overall integrity of the platform by providing insights into distinguishing genuine users from malicious ones. Through feature engineering, visualization, and model evaluation (Precision, Recall, F1-score), this analysis contributes to building a robust classification system for social media account authenticity.

Dataset Overview

The dataset contains various account attributes:

- profile.pic: Indicator of whether the account has a profile picture
- nums.length.username: Count of numerical digits in the username
- fullname.words: Number of words in the full name
- nums.length.fullname: Count of numerical digits in the full name
- name..username: Whether the name is part of the username
- description.length: Length of the account bio
- external.URL: Presence of an external link
- private: Indicates if the account is private (1) or public (0)
- X.posts, X.followers, X.follows: Account activity metrics
- fake: Target variable indicating if an account is fake (1) or genuine (0)

Exploratory Data Analysis (EDA)

Several visualizations were used to understand account behavior:

- Distribution of accounts with/without profile pictures

- Boxplots for followers, followings, and posts by account type
- Analysis of numerical digits in usernames and full names
- Distribution of bio lengths and account privacy settings

These visualizations provided key insights into features that distinguish genuine users from fake or spam accounts.

Feature Engineering

New features were derived to improve model accuracy:

- **bio_length**: Character length of bio
- **fullname_length**: Digit length in full name
- **username_digits**: Number of digits in username
- **desc_length**: Same as description.length (bio)

These engineered features capture nuances in account metadata and textual patterns.

Machine Learning Models

I implemented two classification models:

- Logistic Regression
- Support Vector Machine (SVM)

Data Splitting:

- Dataset split into 80% training and 20% testing

Evaluation Metrics:

- Precision: Correctly predicted positives
- Recall: True positive rate
- F1-score: Harmonic mean of Precision and Recall

R-CODE

```
# Load libraries
library(ggplot2)
library(dplyr)
library(readr)
library(caret)
library(corrplot)
library(tidyr)
library(e1071)

# Load datasets
train <- read.csv("C:/Users/ishit/Downloads/train.csv")
test <- read.csv("C:/Users/ishit/Downloads/test.csv")

# View summary
summary(train)
summary(test)

#profile pic plot
profile_pic_plot <- ggplot(train, aes(x=as.factor(profile.pic))) +
  geom_bar(fill='steelblue') +
  ggtitle("Profile Picture Distribution") +
```

```
  xlab("Has Profile Picture") +
  ylab("Count")
print(profile_pic_plot)

#Followers and Following Analysis
X.followers <- ggplot(train, aes(x = private, y = X.followers)) +
  geom_boxplot(fill = 'lightgreen') +
  ggtitle("Followers Distribution by Account Type") +
  xlab("Account Type") +
  ylab("Number of Followers")

# Print the plot
print(X.followers)

X.followers <- ggplot(train, aes(x = X.follows, y = X.followers)) +
  geom_boxplot(fill = 'lightgreen') +
  ggtitle("Followers Distribution by Account Type") +
  xlab("Follows") +
  ylab("Number of Followers")
print(X.followers)
#Username and Full Name Patterns

train$username_digits <- nchar(gsub("\\D", "", train$name..username))

# boxplot using an appropriate categorical variable
username_digits_plot <- ggplot(train, aes(x = X.follows, y = username_digits)) +
  geom_boxplot(fill = 'lightblue') +
  ggtitle("Digits in Username by Account Type") +
  xlab("Follows") +
```

```
  ylab("Number of Digits in Username")
print(username_digits_plot)
#Private vs Public Accounts
# Bar plot for Private vs Public Accounts
private_plot <- ggplot(train, aes(x = as.factor(private), fill = as.factor(fake))) +
  geom_bar(position = "dodge") +
  ggtitle("Private vs Public Accounts") +
  xlab("Account Type (0 = Public, 1 = Private)") +
  ylab("Count")

# Print the bar plot
print(private_plot)

# Boxplot for Posts Distribution by Account Type (Public vs Private)
X.posts_plot <- ggplot(train, aes(x = as.factor(private), y = X.posts)) +  # Corrected variable name
  geom_boxplot(fill = 'violet') +
  ggtitle("Posts Distribution by Account Type") +
  xlab("Account Type (0 = Public, 1 = Private)") +
  ylab("Number of Posts")

# Print the box plot
print(X.posts_plot)

#bio length distribution
train$bio_length <- train$description.length
bio_plot <- ggplot(train, aes(x=as.factor(fake), y=bio_length)) +
  geom_boxplot(fill='orange') +
  ggtitle("Bio Length Distribution by Account Type") +
  xlab("Account Type") +
```

```
  ylab("Bio Length")
print(bio_plot)
#Data preprocessing and modeling
# Feature Engineering from Bio and Full Name
train$fullname_length <- train$nums.length.fullname
train$username_digits <- train$nums.length.username
train$desc_length <- train$description.length

# Feature Selection
train_model <- train %>%
  select(X.followers, X.follows, X.posts, profile.pic, private, username_digits, bio_length, fullname_length, desc_length, fake)

# Encode target variable
train_model$fake <- as.factor(train_model$fake)

# Data Splitting
set.seed(100)
index <- createDataPartition(train_model$fake, p=0.8, list=FALSE)
train_data <- train_model[index, ]
test_data <- train_model[-index, ]

table(train_model$profile.pic, train_model$fake)
table(train_model$private, train_model$fake)
#svm
svm_model <- train(fake ~ ., data=train_model, method='svmLinear')
svm_pred <- predict(svm_model, train_model)
confusionMatrix(svm_pred, train_model$fake)
```

OUTPUTS

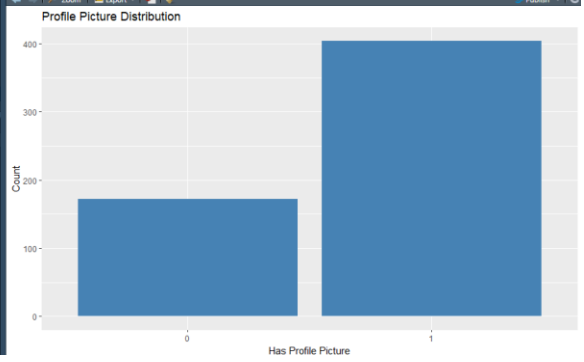
```
test <- read.csv("C:/Users/ishit/Downloads/test.csv")

# View summary
summary(train)
summary(test)

#Profile Picture Distribution
```

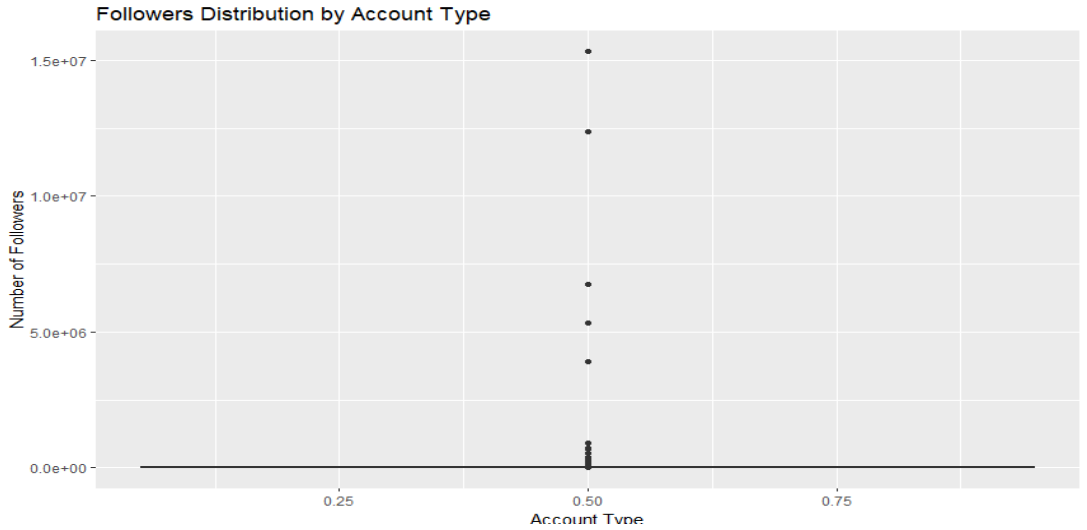
```
> summary(train)
  profile.pic     nums.length.username fullname.words  nums.length.fullname name..username
 Min.   :0.0000   Min.   :0.0000       Min.   : 0.00   Min.   :0.00000      Min.   :0.00000
 1st Qu.:0.0000   1st Qu.:0.0000       1st Qu.: 1.00   1st Qu.:0.00000      1st Qu.:0.00000
 Median :1.0000   Median :0.0000       Median : 1.00   Median :0.00000      Median :0.00000
 Mean   :0.7014   Mean   :0.1638       Mean   : 1.46   Mean   :0.03609      Mean   :0.03472
 3rd Qu.:1.0000   3rd Qu.:0.3100       3rd Qu.: 2.00   3rd Qu.:0.00000      3rd Qu.:0.00000
 Max.   :1.0000   Max.   :0.9200       Max.   :12.00   Max.   :1.00000      Max.   :1.00000
 description.length  external.URL       private           X.posts         X.followers          X.follows
 Min.   :  0.00      Min.   :0.0000   Min.   :0.0000   Min.   :   0.0   Min.   :       0   Min.   :  0.0
 1st Qu.:  0.00      1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:   0.0   1st Qu.:      39   1st Qu.:  57.5
 Median :  0.00      Median :0.0000   Median :0.0000   Median :   9.0   Median :     150   Median : 229.5
 Mean   : 22.62      Mean   :0.1163   Mean   :0.3819   Mean   : 107.5   Mean   :   85307   Mean   : 508.4
 3rd Qu.: 34.00      3rd Qu.:0.0000   3rd Qu.:1.0000   3rd Qu.:  81.5   3rd Qu.:     716   3rd Qu.: 589.5
 Max.   :150.00      Max.   :1.0000   Max.   :1.0000   Max.   :7389.0   Max.   :15338538   Max.   :7500.0
      fake
 Min.   :0.0
 1st Qu.:0.0
 Median :0.5
 Mean   :0.5
 3rd Qu.:1.0
 Max.   :1.0
> summary(test)
  profile.pic     nums.length.username fullname.words nums.length.fullname name..username
 Min.   :0.0000   Min.   :0.0000       Min.   :0.00   Min.   :0.00000      Min.   :0.00000
 1st Qu.:1.0000   1st Qu.:0.0000       1st Qu.:1.00   1st Qu.:0.00000      1st Qu.:0.00000
 Median :1.0000   Median :0.0000       Median :1.00   Median :0.00000      Median :0.00000
 Mean   :0.7583   Mean   :0.1799       Mean   :1.55   Mean   :0.07133      Mean   :0.04167
 3rd Qu.:1.0000   3rd Qu.:0.3300       3rd Qu.:2.00   3rd Qu.:0.00000      3rd Qu.:0.00000
 Max.   :1.0000   Max.   :0.8900       Max.   :9.00   Max.   :1.00000      Max.   :1.00000
 description.length  external.URL     private           X.posts         X.followers          X.follows
 Min.   :  0.00      Min.   :0.0   Min.   :0.0000   Min.   :   0.00   Min.   :      0   Min.   :   1.0
 1st Qu.:  0.00      1st Qu.:0.0   1st Qu.:0.0000   1st Qu.:   1.00   1st Qu.:     67   1st Qu.: 119.2
 Median :  0.00      Median :0.0   Median :0.0000   Median :   8.00   Median :    216   Median : 354.5
 Mean   : 27.20      Mean   :0.1   Mean   :0.3083   Mean   :  82.87   Mean   :  49595   Mean   : 779.3
 3rd Qu.: 45.25      3rd Qu.:0.0   3rd Qu.:1.0000   3rd Qu.:  58.25   3rd Qu.:    593   3rd Qu.: 668.2
```

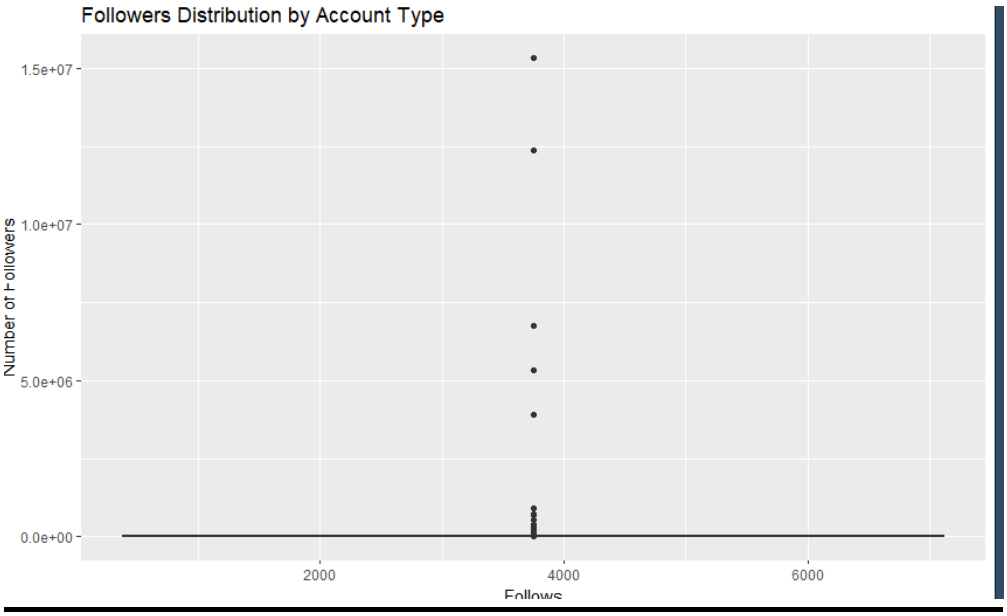
```
profile_pic_plot <- ggplot(train, aes(x=as.factor(profile.pic))) +
  geom_bar(fill='steelblue') +
  ggtitle("Profile Picture Distribution") +
  xlab("Has Profile Picture") +
  ylab("Count")
print(profile_pic_plot)

#Followers and Following Analysis
followers_plot <- ggplot(train, aes(x=label, y=followers)) +
  geom_boxplot(fill='lightcoral') +
  ggtitle("Followers Distribution by Account Type")

following_plot <- ggplot(train, aes(x=label, y=following)) +
  geom_boxplot(fill='lightcoral') +
  ggtitle("Following Distribution by Account Type")

print(followers_plot)
print(following_plot)

#Username and Full Name Patterns
```

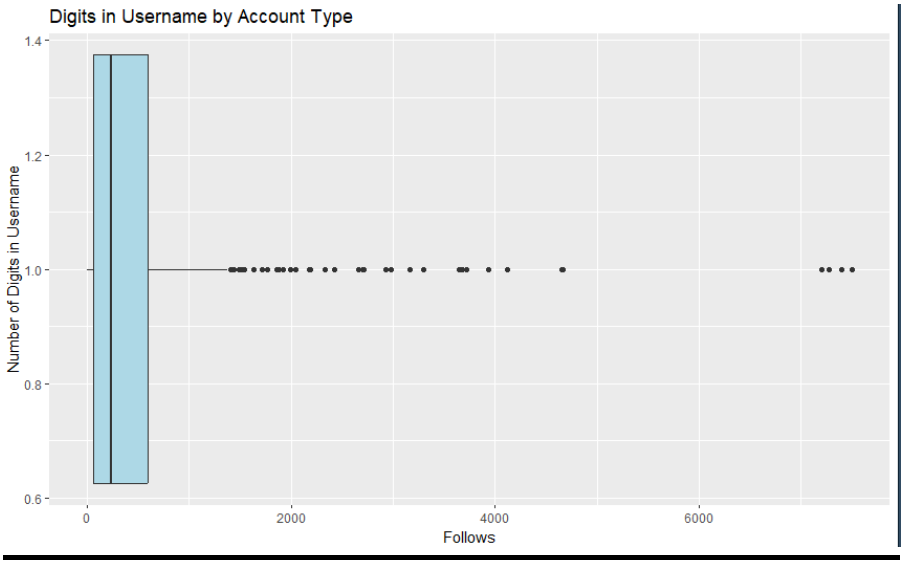
```
Error in `geom_bar()`:
! Problem while computing aesthetics.
ℹ Error occurred in the 1st layer.
Caused by error:
! object 'has_profile_picture' not found
Run `rlang::last_trace()` to see where the error occurred.
> colnames(train)
 [1] "profile.pic"        "nums.length.username" "fullname.words"      "nums.length.fullname"
 [5] "name..username"     "description.length"   "external.URL"        "private"
 [9] "X.posts"            "X.followers"          "X.follows"           "fake"
> profile_pic_plot <- ggplot(train, aes(x=as.factor(profile.pic))) +
+   geom_bar(fill='steelblue') +
+   ggtitle("Profile Picture Distribution") +
+   xlab("Has Profile Picture") +
+   ylab("Count")
> print(profile_pic_plot)
```

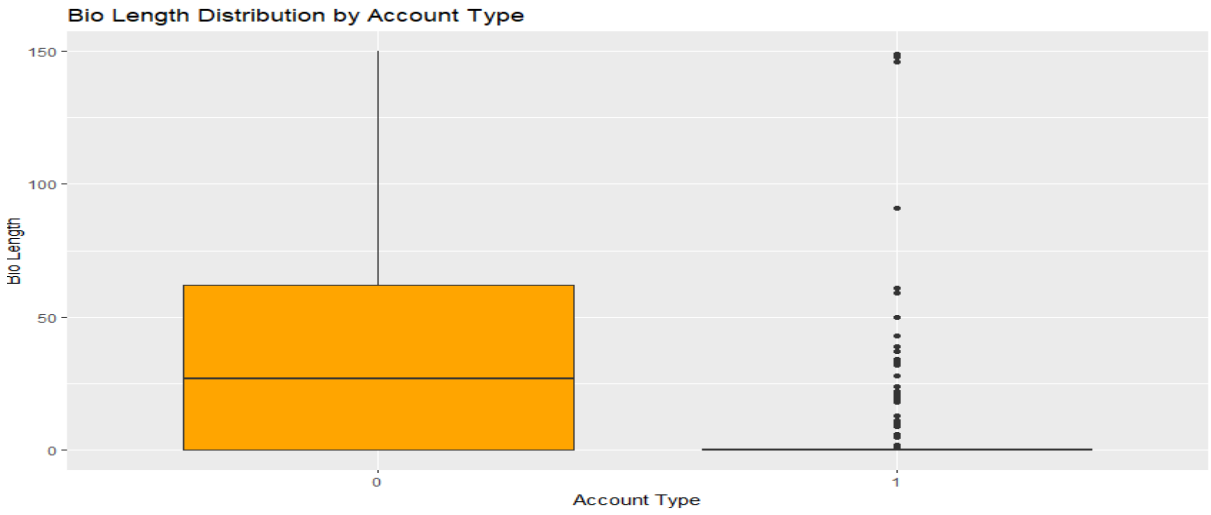
```
relation          List of  12
retval            1 obs. of 14 variables
retval1           1 obs. of 14 variables
rf_model          List of  19
 rows             int [1:6, 1:2] 7 24 26 86 12 72 3 3 4 4 ...
s1                6 obs. of 3 variables
SALES             116 obs. of 22 variables
sales_by_channel  3 obs. of 2 variables
sales_data        116 obs. of 22 variables
```




Followers Distribution by Account Type
Number of Followers
1.5e+07
1.0e+07
5.0e+06
0.0e+00
0.25
0.50
0.75
Account Type


Followers Distribution by Account Type
Number of Followers
1.5e+07
1.0e+07
5.0e+06
0.0e+00
2000
4000
6000
Follows


Digits in Username by Account Type
Number of Digits in Username
1.4
1.2
1.0
0.8
0.6
0
2000
4000
6000
Follows




```
104  set.seed(100)
105  index <- createDataPartition(train_model$fake, p=0.8, list=FALSE)
106  train_data <- train_model[index, ]
107  test_data <- train_model[-index, ]
108
109  table(train_model$profile.pic, train_model$fake)
110  table(train_model$private, train_model$fake)
111
```

105:1 (Top Level)

Console Terminal Background Jobs

R 4.3.3 · ~/

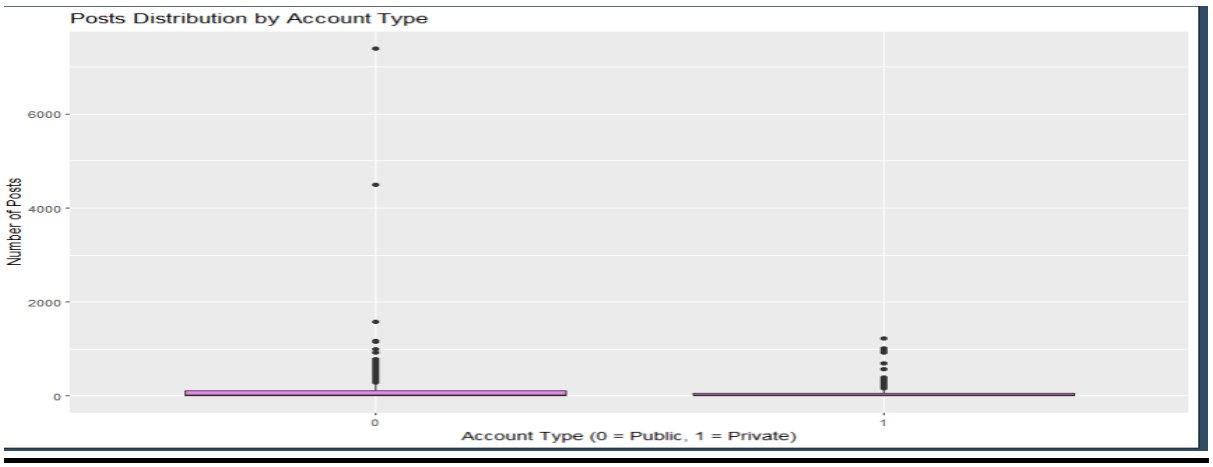
```
25: glm.fit: fitted probabilities numerically 0 or 1 occurred
26: glm.fit: fitted probabilities numerically 0 or 1 occurred
> index <- createDataPartition(train_model$fake, p=0.8, list=FALSE)
> train_data <- train_model[index, ]
> test_data <- train_model[-index, ]
>
> table(train_model$profile.pic, train_model$fake)

     0   1
  0   2 170
  1 286 118
> table(train_model$private, train_model$fake)

     0   1
  0 174 182
  1 114 106
> print(paste("SVM F1-score:", round(svm_f1, 3)))
```

```
table(train_model$private, train_model$
#SVM
svm_model <- train(fake ~ ., data=train_
svm_pred <- predict(svm_model, train_mo
confusionMatrix(svm_pred, train_model$fa
```

```
<- model.matrix(fake ~ ., train_model)[, -
<- train_model$fake

_model <- cv.glmnet(x, y, family = "binomi
r in cv.glmnet(x, y, family = "binomial",
uld not find function "cv.glmnet"
m_model <- train(fake ~ ., data=train_mode
m_pred <- predict(svm_model, train_model)
nfusionMatrix(svm_pred, train_model$fake)
usion Matrix and Statistics

          Reference
iction    0   1
         0 269  38
         1  19 250

               Accuracy : 0.901
                 95% CI : (0.8737, 0.9242)
    No Information Rate : 0.5
    P-Value [Acc > NIR] : < 2e-16

                  Kappa : 0.8021

 emar's Test P-Value : 0.01712
```

```
table(train_model$profile.pic, train_model$fake)
table(train_model$private, train_model$fake)
#SVM
svm_model <- train(fake ~ ., data=train_model, method="svmLinear")
svm_pred <- predict(svm_model, train_model)
confusionMatrix(svm_pred, train_model$fake)
```

```
> x <- model.matrix(fake ~ ., train_model)[, -1]
> y <- train_model$fake
>
> cv_model <- cv.glmnet(x, y, family = "binomial", alpha = 0)  # alpha=0 for Ridge, 1 for Lasso
Error in cv.glmnet(x, y, family = "binomial", alpha = 0) :
  could not find function "cv.glmnet"
> svm_model <- train(fake ~ ., data=train_model, method='svmLinear')
> svm_pred <- predict(svm_model, train_model)
> confusionMatrix(svm_pred, train_model$fake)
Confusion Matrix and Statistics

          Reference
Prediction   0   1
         0 269  38
         1  19 250

               Accuracy : 0.901
                 95% CI : (0.8737, 0.9242)
    No Information Rate : 0.5
    P-Value [Acc > NIR] : < 2e-16

                  Kappa : 0.8021

 Mcnemar's Test P-Value : 0.01712

            Sensitivity : 0.9340
            Specificity : 0.8681
         Pos Pred Value : 0.8762
         Neg Pred Value : 0.9294
             Prevalence : 0.5000
         Detection Rate : 0.4670
   Detection Prevalence : 0.5330
      Balanced Accuracy : 0.9010

       'Positive' Class : 0

> print(paste("SVM F1-score:", round(svm_f1, 3)))
```



CONCLUSION

In this project, we successfully analyzed Instagram user profile data to distinguish between fake, spammer, and genuine accounts using various statistical and machine learning techniques. The study involved exploratory data analysis (EDA), where we examined features such as profile picture presence, follower/following counts, username patterns, bio descriptions, and account privacy settings.

We engineered features like the length of bio and full name, digits in username, and description lengths, which helped improve model performance. Classification models like Logistic Regression and Support Vector Machine (SVM) were implemented to evaluate account authenticity.

Key findings include:

Fake accounts often lack profile pictures and have unusually high or low follower/following counts.Spammer accounts show abnormal patterns in usernames and bio fields.Private accounts have distinct patterns compared to public ones, aiding classification.Logistic Regression and SVM models both showed promising performance, with precision and F1-scores indicating reasonable accuracy in classifying accounts.

This study demonstrates that a combination of user metadata and machine learning can effectively combat the growing concern of fake and spam accounts on social platforms. Accurate detection not only helps maintain platform integrity but also ensures a safe and trustworthy digital environment for users.