Fake Profile Prediction for Instagram

Aman Pareek and Ishan Kohli

Abstract

Fake and spam accounts are a major problem in social media. Many social media influencers use fake Instagram accounts to create an illusion of having so many social media followers. Fake accounts can be used to impersonate or catfish other people and be used to sell fake services/products. Facebook said it blocked 4.5 billion accounts in the first nine months of the year, and that it caught more than 99 percent of those accounts before users could flag them. That number of accounts — equivalent to nearly 60 percent of the world's population — is mind-boggling. Data used in this project was collect using a crawler from 15-19 March 2019 and spammer/fake accounts were identified by the person who collected it. In this dataset fake and spammer are interchangeable terms. We analysed the dataset with classification techniques like logistic regression, binary classification tree and also ridge regression. And both ridge regression and binary classification tree gave same accuracy.

Dataset Description

Testing and Training dataset are already in 2 separate files, training set has data of 576 profiles(or accounts) and testing set has 120 instances. There are total 12 attributes(or features) for both sets.
The variable '**fake**' is response variable; "1" means profile is spammer and "0" mean it is genuine.

Aa Attribute	Description
profile.pic	user has profile picture (1) or not (0)
nums.length.username	ratio of number of numerical chars in username to its length
fullname.words	full name in word tokens
nums.length.fullname	ratio of number of numerical characters in full name to its length
name..username	are username and full name literally the same
description.length	bio length in characters
external.URL	has external URL (1) or not (0)
private	private (1)or not (0)
X.posts	number of posts
X.followers	number of followers
X.follows	number of follows
fake	class (0 genuine, 1 spammer)

Mathematics behind the models used

1. Logistic Regression

It's a statistical algorithm similar to Linear Regression in that it uses independent variables to forecast the dependent variable, but the dependent variable must be categorical. The dependent

variable would still be categorical, regardless of whether the independent variables are numeric or categorical.
Logistic regression is a mathematical model that models conditional probability using the Logistic Function (also known as sigmoid function) such as:

$$F(x) = \frac{1}{1+e^{-x}} = \frac{e^x}{e^x+1}$$

We measure the conditional probability of the dependent variable Y given the independent variable X in binary regression. P(Y=1|X) or P(Y=0|X) are two ways to write it. The end goal in logistic regression is to minimize the cross-entropy error, that is,

$$E_{in}(w) = \frac{1}{N}\sum_{n=1}^{N} \log\left(1 + e^{-y_n w^T x_n}\right)$$

2. Ridge Regression

Ridge regression is also known as L2 regularization and it is used for variable selection and at the same time to reduce variance in the model.
It is very similar to linear regression in the sense that it minimizes

$$RSS + \lambda\left\|\hat{\beta}\right\|_2^2.$$

whereas linear regression just minimizes RSS. The constant $\lambda \geq 0$ is called the tuning parameter (in practice found via cross-validation) and the second term in the above equation is the penalty that is imposed on the error RSS for choosing a large λ. Very large λ, the minimization would force the coefficient of λ to shrink towards 0. So, if some parameters become 0, then the corresponding terms get dropped from the model.

3. Decision Tree (Binary)

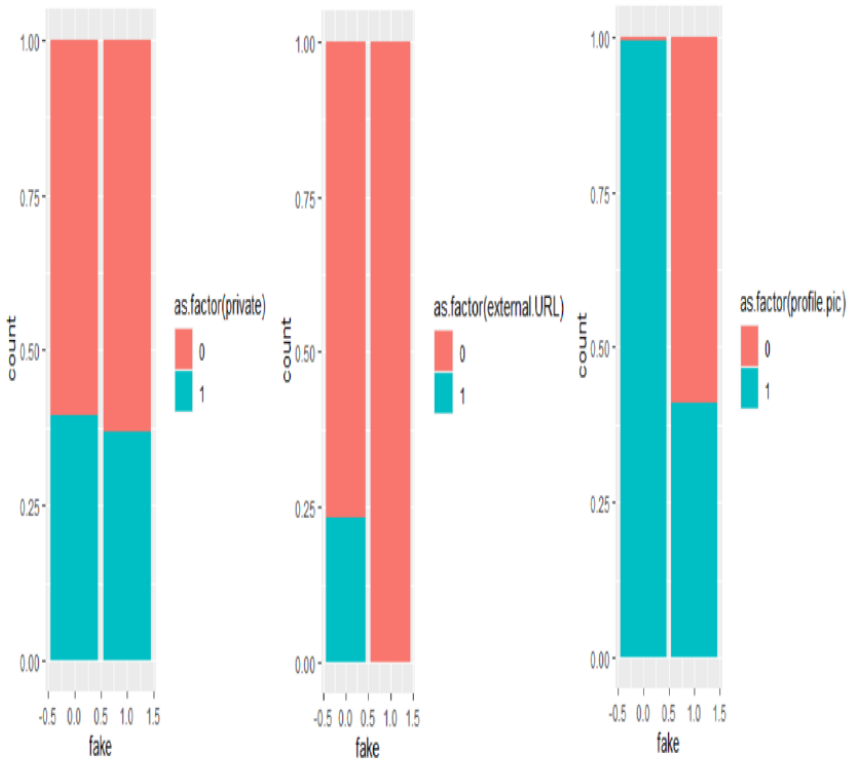
The decision tree is a type of supervised learning algorithm that can be used in both regression and classification problems. It works for both categorical and continuous input and output variables. Any decision tree is built by partitioning the feature space into rectangles (or boxes). In a classification scenario, the value (label) returned by the decision tree will be the most frequently occurring response in that region. And instead of RSS we use Gini index and cross-entropy for the choice of cuts. Gini Index for binary classification

$$p_{m0}(1-p_{m0}) + p_{m1}(1-p_{m1})$$

Experimental Analysis

Data Visualization

Since the data was nicely formatted, after loading the data we made few random plots just to see some relation between variables.



(i) (ii) (iii)

The response variable is on X-axis for all 3 plots.
(i) Fraction of profiles that are private. It can be seen that a fraction of private accounts is almost equal for both fake and genuine profiles.

(ii) Fraction of profiles having an external URL in their bio. Clearly, no fake profile has an external URL in the bio while close to 25% of genuine profiles have an external URL.

(iii) Fraction of profiles having profile pictures for both classes. We can clearly see that 99% of the genuine profiles have a profile picture. On the other hand, only 40% of the fake profiles have it.

Models

Now we take a look at model building and making prediction on testing dataset. Following modes were built:

(i) Logistic Regression

```
model_lr=glm(fake~profile.pic+nums.length.username+X.followers+X.follows+X.posts,data=train, family = binomial)
probs_test=predict(model_lr,test,type="response")
```

(ii) Ridge Regression

```
library(glmnet)
set.seed(123)
x= model.matrix(train$fake~.,train)[,-1]
y=train$fake
cv.ridge = cv.glmnet(x, y, alpha = 0, family = "binomial")
l1.model = glmnet(x, y, alpha = 0, family = "binomial",lambda = cv.ridge$lambda.min)
x.test = model.matrix(test$fake ~.,data=test)[,-1]
probabilities = predict(l1.model, newx = x.test)
```

(iii) Classification Tree

```
library(tree)
tree_mod=tree(as.factor(fake)~.,train)
tree_pred=predict(tree_mod,test,type="class")
```

Result Analysis

we calculated accuracy for all the models using the confusion matrix and it was found out that both the Classification tree and Ridge regression models have the same score of 0.891.

Logistic Regression

```
> probs=predict(model_lr,type ="response")
> probs_test=predict(model_lr,test,type="response")
> pred_test=rep("0",120)
> pred_test[probs_test>.5]="1"
> table(pred_test,test$fake)

pred_test  0  1
        0 51 10
        1  9 50
> mean(pred_test==test$fake)
[1] 0.8416667
```

Ridge Regression

```
> x.test = model.matrix(test$fake ~.,data=test)[,-1]
> probabilities = predict(l1.model, newx = x.test)
> predicted.classes <- ifelse(probabilities > 0.5, "1", "0")
> observed.classes <- test$fake
> table(predicted.classes,observed.classes)
                observed.classes
predicted.classes  0  1
                0 56  9
                1  4 51
> mean(predicted.classes == observed.classes)
[1] 0.8916667
```

Classification Tree

```
> tree_pred=predict(tree_mod,test,type="class")
> table(tree_pred, test$fake)

tree_pred  0  1
        0 57 10
        1  3 50
> mean(tree_pred==test$fake)
[1] 0.8916667
```

References

1. http://www.sthda.com/english/articles/36-classification-methods-essentials/149-penalized-logistic-regression-essentials-in-r-ridge-lasso-and-elastic-net/#:~:text=Penalized Logistic

Regression Essentials in R%3A Ridge%2C Lasso and Elastic Net,-kassambara | 11%2F03&text=Penalized logistic regression imposes a,is also known as regularization.

2. https://www.kaggle.com/free4ever1/instagram-fake-spammer-genuine-accounts?select=test.csv
3. An Introduction to Statistical Learning: With Applications in R

Github link: https://github.com/apth3hack3r/Fake-Insta-Profile-Detector