**Statistical Learning Report –**

**Supervised methods**

Matteo Hasa (56269A)

**Abstract**

Through the application of supervised machine learning techniques, this report shows the results of an analysis regarding the detection of fake Instagram accounts using the **LIMFADD** dataset from Kaggle. The objective of the analysis is to create a model that, given some characteristics, detects when an account is fake.

**Objective of the analysis**

Nowadays, as social media plays a fundamental role in our lives, it is becoming increasingly important and challenging to distinguish real/genuine profiles from fake ones.

This analysis has the purpose of creating a model that, given some characteristics, it’ll be helpful on detecting when an account is fake.

The dataset is “*LIMFADD: Instagram Multi-Class Fake Detection”* and can be found on Kaggle. It has the information of 15000 Instagram accounts, all divided in 11 variables, binomial, numerical and categorical.

We can divide the variables in:

* *Numerical variables:*
  + *Followers:* Number of followers of the account
  + *Following:* Number of accounts that the profile follows
  + *Following.Followers:* Ratio between the number of accounts followed and the number of followers
  + *Posts:* Total number of posts published by the account
  + *Posts.Followers:* Ratio between the number of posts and followers
  + *Mutual.Friends: Number of mutual connections with other users*
* *Binomial variables:*
  + *Bio:* Indicates whether the account has a biography text in its profile
  + *Profile.Picture:* Indicates whether the account has a profile picture
  + *External.Link:* Indicates the presence of an external link in the bio
  + *Threads:* Indicates whether the user has posted Threads
* *Categorical:*
  + *Labels:* True class of the account, divided into four categories: *Real, Spam,*
  + *Scam, and Bot.* It represents the target variable for classification

**Exploratory Data Analysis**

First of all, it’s important to know the dataset in every aspect.

In the beginning of the analysis, I wanted to know if there were any missing values and after checking it, I discovered that there weren’t.

Since there are a lot of binomial and a categorical variables, I checked if *Followers, Following* and *Posts* were normal.

Immagine che contiene diagramma, linea, Diagramma, testo

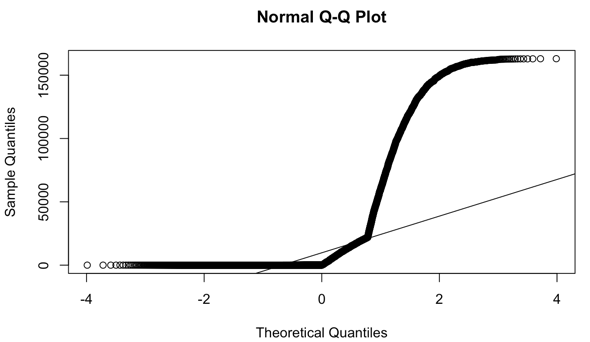
Il contenuto generato dall'IA potrebbe non essere corretto.

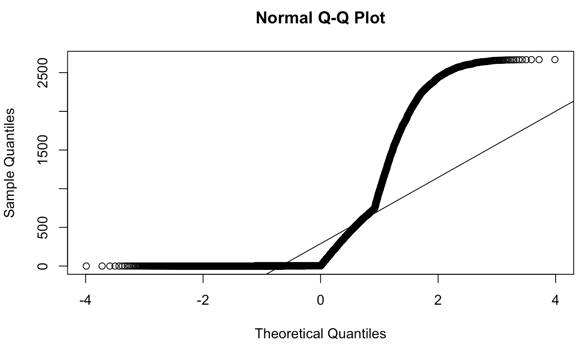
Figure 1: qqplot Followers  Figure 2: qqplot Following

Figure 3: qqplot Posts

These variables are highly skewed. Normal Q-Q plots confirmed that the distributions deviate strongly from normality. For example, the median number of followers is much lower than the mean, indicating a skew where a handful of accounts have tens of thousands of followers while many have only a few. I did not remove these outliers since they could be informative.

And then, I wanted to visualize the distribution of the binomial variables.

Immagine che contiene testo, schermata, diagramma, linea

Il contenuto generato dall'IA potrebbe non essere corretto.From *Figure 4*, we observe that

A large part of profiles have a

profile picture.

A lot of profiles also have

a bio. In contrast, relatively few profiles

have an external link in their bio, this

suggests that, it is potentially a

characteristic of certain account types.

Additionally, only a small portion of

profiles are associated with Threads,

which is expected as *Threads* is a new

feature that not all users use.

Figure 4: Distribution of Binomial variables

As we can observe, *External.Link* and

*Threads* seems to have similar distribution, maybe they are correlated.

Then I proceeded computing the correlation matrix and the plots.

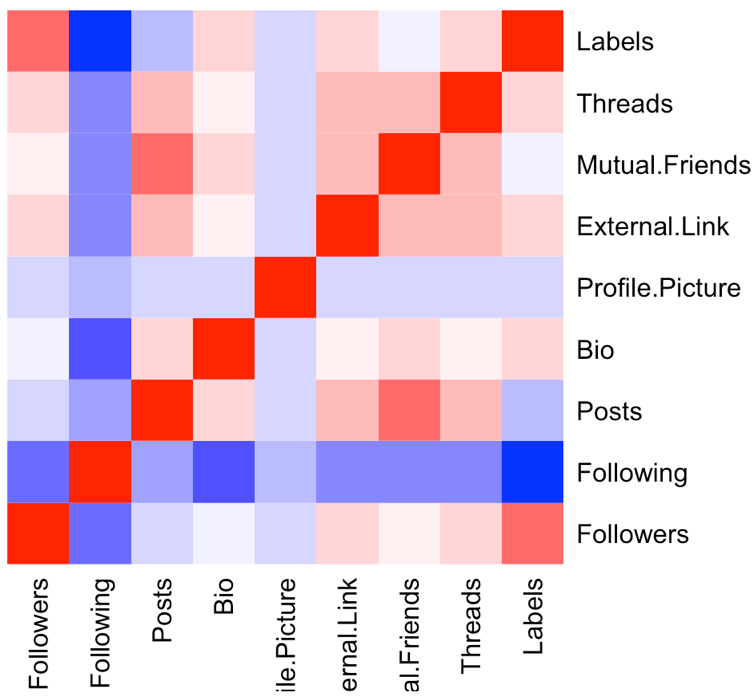
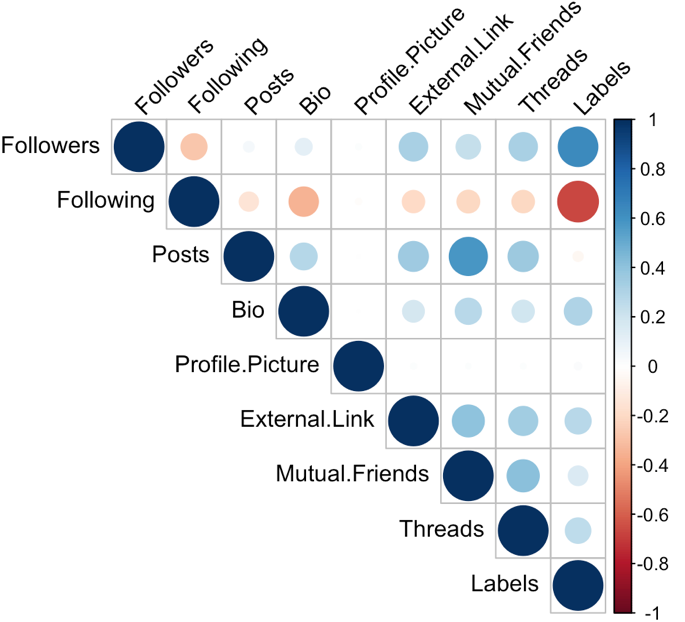


Figure 5: Correlation Plot

Figure 6: Correlation Heatmap

From *Figure 5,6* (correlation heatmap), we can interpret several patterns:

* ***Followers* vs. *Following*:** There is a **moderate negative correlation** between the number of followers and the number of accounts followed. This means that accounts that follow a great many others tend to have fewer followers themselves. This is often a characteristic of **bot accounts**, which might follow many users but are not followed back. Real popular accounts (like celebrities) often have the opposite characteristic: many followers but follow relatively few. This inverse relationship is a potential discriminator for account type.
* ***Posts* and *Mutual.Friends*:** These show a **strong positive correlation**. Profiles with many posts tend to have more mutual friends. Intuitively, an account that is active is likely a real user who has friends in common with others; fake accounts might post less or have no mutual connections.
* ***External.Link*:** The presence of an external link is moderately positively correlated with having many followers and with posting a lot. This may indicate that profiles that include a link are often businesses or influencers or conversely certain spam accounts.
* ***Bio* and *Threads*:** Both show a positive correlation with Label. This probably indicates that **real accounts more often have a bio filled and are a bit more likely to use Threads** compared to fake accounts. Many bot or scam profiles leave the bio blank and may not engage with new features like Threads.
* ***Profile.Picture*:** Interestingly, having a profile picture showed essentially **no correlation** with other features or with the label. This suggests that both real and fake accounts usually have profile images.

Finally, I inspected the distributions and values by account type. For example, genuine *Real* accounts tended to have: a moderate number of followers and followings, at least some mutual friends, usually a bio and profile pic, and rarely an external link. Fake *Bot* accounts often had extreme behavior like *Following very many accounts* while having very few followers themselves, zero mutual friends, and possibly missing bios. *Scam* accounts might differ by having external links and possibly also having many followings, etc. These observations from EDA align with our expectations and will be tested in the supervised models.

**Data Cleaning**

Before building models, the dataset was cleaned and preprocessed. Key data cleaning steps included:

* **Converting categorical to numeric:** I converted the binary Yes/No features into numerical dummy variables: "Yes" was mapped to 1 and "No" (or "N") to 0. This was done for *Bio*, *Profile.Picture*, *External.Link*, and *Threads*. The account Labels were also factorized: for multi-class analysis, the text labels (“Real”, “Bot”, “Spam”, “Scam”) were turned into factor levels 1–4.
* **Deleting variables:** The columns Following/Followers and Posts/Followers were dropped to avoid complications with missing or infinite values.
* **Splitting data:** For modeling, the dataset was split into a **training set (80%) and a test set (20%)**.

After these steps, the cleaned data was ready for modeling. All features are numerical or binary 0/1, which most algorithms can handle directly.

**Models application**

I applied different supervised learning methods to this classification problem: a decision tree, a random forest and cross validation.

**First Method: The decision tree**

I computed a training for a Decision Tree classifier with the optimal complexity parameter (cp) to control the tree size.

Immagine che contiene testo, Carattere, schermata, bianco e nero

Il contenuto generato dall'IA potrebbe non essere corretto.

🡪

Figure 7: Complexity Parameter (CP) Table

I chose the optimal complexity parameter (CP) by analyzing the cross-validation results in this table. It is important to choose the best CP in order to avoid overfitting and to maintain the model generalization.

The CP minimizing the cross-validated error (xerror) is the optimal one.

In particular, a CP of 0.00066667 was identified.

Immagine che contiene testo, schermata, diagramma, linea

Il contenuto generato dall'IA potrebbe non essere corretto.

Figure 8: Decision Tree

This is the final decision tree computed with the optimal cp.

The Decision Tree highlights follower and following counts as primary indicators for classifying Instagram accounts. Accounts with extremely low followers and high following are immediately separated as potentially non-genuine. Further splits involve posting activity, mutual friends, and bio presence, refining the classification and helping to distinguish real from fake, spam or scam accounts.

*Label* is divided in 4 classes:

* Class 1: **Bot**
* Class 2: **Real**
* Class 3: **Scam**
* Class 4: **Spam**

Then, using the “predict” function, I computed the confusion Matrix

Immagine che contiene testo, schermata, Carattere, numero

Il contenuto generato dall'IA potrebbe non essere corretto.

Figure 9: Confusion Matrix

As we can observe, the model achieved an overall accuracy of **97.4%**, with a 95% confidence interval ranging from **96.77% to 97.94%**.

The No Information Rate (NIR) was **25%**, indicating that random guessing would be much less effective.

The p-value (**< 2.2e-16**) confirms that the model’s accuracy is statistically significantly higher than random classification.

The Kappa statistic was **0.9653**, demonstrating almost perfect agreement between the predicted and actual classes.

From the confusion matrix:

* The model performed really well in classifying **Bot** and **Scam** accounts.
* Some confusion was observed between **Real** and **Spam** accounts, but the number of mistakes were very low.

These results highlight the strong predictive performance and robustness of the classification model across all account types.

Immagine che contiene testo, Carattere, schermata, bianco

Il contenuto generato dall'IA potrebbe non essere corretto.

Figure 10: Statistics by Class

The model shown excellent performance across all classes, achieving high sensitivity (recall), specificity, and balanced accuracy.

* **Sensitivity** ranged from **93.5% to 100%**, indicating the model’s strong ability to correctly identify each account type.
* **Specificity** is high (**above 98%**) across all classes, demonstrating that false positives were minimal.
* **Balanced Accuracy** for each class exceeded **96%**, confirming the robustness.

While minor confusion occurred between Real and Spam accounts, the overall detection rates closely matched the true prevalence of each class, highlighting the effectiveness of the classification approach.

**Second method: Random Forest**

A Random Forest model was trained using 500 trees and selecting 3 variables at each split.

Immagine che contiene testo, numero, Carattere, schermata

Il contenuto generato dall'IA potrebbe non essere corretto.

Figure 11: Importance of the variables

The variable importance analysis based on Mean Decrease Accuracy and Mean Decrease Gini identified the following key predictors:

* **Followers**
* **Following**
* **Posts**
* **Mutual Friends**
* Followed by **Bio**, **External.Link**, **Threads**, and **Profile.Picture**.

The importance measures confirmed that the number of followers and following behavior were crucial for accurately classifying Instagram accounts, while external features like bio presence and external links contributed moderately.

Then I computed the confusion Matrix of the model

Immagine che contiene testo, schermata, Carattere, numero

Il contenuto generato dall'IA potrebbe non essere corretto.

Figure 12: Confusion Matrix of the rf model

The model achieved an overall **accuracy of 97.53%** on the test set, with a 95% Confidence Interval between **96.91% and 98.06%** which is slightly better than the previous model.

The associated **Kappa statistic was 0.967**, indicating an almost perfect agreement beyond chance.

Compared to a No Information Rate of 25%, the model performance was statistically highly significant with p-value < 2.2e-16

Then I computed the Sensitivity, Specificity, and Balanced Accuracy

Immagine che contiene testo, Carattere, schermata, bianco

Il contenuto generato dall'IA potrebbe non essere corretto.

Figure 13: Statistics by class

Across the four classes (Bot, Real, Scam, Spam):

* **Sensitivity (Recall)** ranged from **94.5% to 100%**, showing strong ability to detect each class.
* **Specificity** remained above **98%** across all classes.
* **Balanced Accuracy** was extremely high for all classes, ranging from **96.8% to 99.7%**.

Class 3 (**Scam**) was perfectly classified with 100% sensitivity, while Class 2 (**Real**) showed slightly lower sensitivity due to minor confusion with Spam accounts.

The Random Forest model demonstrated excellent predictive ability and robustness across all account categories.

The most influential variables were linked to account popularity (Followers, Following) and activity patterns (Posts, Mutual Friends).

Despite slight misclassifications between Real and Spam classes, the overall classification quality remains very high.

**Third method: Cross-Validation**

To evaluate the stability and generalization performance of the Decision Tree model, two different cross-validation techniques were applied:

1. **10-Fold Cross-Validation (CV)**
2. **Leave-One-Out Cross-Validation (LOOCV)**

Both methods aimed to select the optimal **complexity parameter** (cp) by minimizing the **Root Mean Squared Error** (RMSE).

Immagine che contiene testo, schermata, ricevuta, Carattere

Il contenuto generato dall'IA potrebbe non essere corretto.Immagine che contiene testo, ricevuta, Carattere, schermata

Il contenuto generato dall'IA potrebbe non essere corretto.

Figure 14: Leave-One-Out-CV

Figure 15: 10 Fold Cross-Validation

10-Fold Cross-Validation **Results**

* The dataset was split into 10 folds, training on 90% and testing on 10% iteratively.
* The model was tuned across different cp values.
* Optimal **cp** selected is 0.**0662**, corresponding to the lowest RMSE.
* **Performance Metrics:**
  + RMSE: **0.5258**
  + R-squared: **0.7773**
  + MAE (Mean Absolute Error): **0.3608**

This suggests the model explains about **77.7% of the variance** and maintains a relatively low prediction error across folds.

Leave-One-Out Cross-Validation (LOOCV) **Results**

* Each observation was used once as a test set, with all remaining observations used as training data.
* LOOCV is computationally intensive but provides an almost unbiased estimate of model performance.
* **Optimal cp selected**: again **0.0662** (same as in 10-fold CV).
* **Performance Metrics:**
  + RMSE: **0.5285**
  + R-squared: **0.7871**
  + MAE: **0.3432**

The results from LOOCV are very similar to those from 10-fold CV, confirming the **robustness and consistency** of the selected Decision Tree model.

Then, using the optimal complexity parameter **cp = 0.0662**, I computed the final Decision Tree.

Immagine che contiene testo, schermata, Carattere, diagramma

Il contenuto generato dall'IA potrebbe non essere corretto.

Figure 16:Final Decision Tree

The final Decision Tree, optimized by cross-validation, shows a simple yet powerful structure that primarily relies on Followers and Following metrics. This confirms that basic user engagement measures are highly discriminative in distinguishing between genuine and fake Instagram profiles.

I tried also to build a random forest model using the caret::train() function, which allowed for automated cross-validation and hyperparameter tuning within a unified modeling framework. This approach simplifies model comparison and ensures consistency in evaluation strategies across different models.

Immagine che contiene testo, Carattere, schermata, bianco

Il contenuto generato dall'IA potrebbe non essere corretto.Immagine che contiene testo, schermata, Carattere, numero

Il contenuto generato dall'IA potrebbe non essere corretto.

Figure 17: Random Forest model

As we can see the result is really similar to the other model built using “*randomForest*”.

**Extra method: Binary Outcome and Logistic regression**

I tried to make a binary classification task by grouping Real accounts that I called “Genuine” versus all fake account types (Bot, Scam and Spam). However, this method introduced a high level of heterogeneity within the “Not Genuine” group. Since Bots, Scams, and Spam exhibit significantly different characteristics, the model struggled to find a simple decision boundary. As a result, the stepwise forward selection retained almost all variables as significant, indicating that each feature contributed to distinguishing at least one type of fake account, but no subset of features could effectively separate Genuine users from the diverse Not Genuine group.

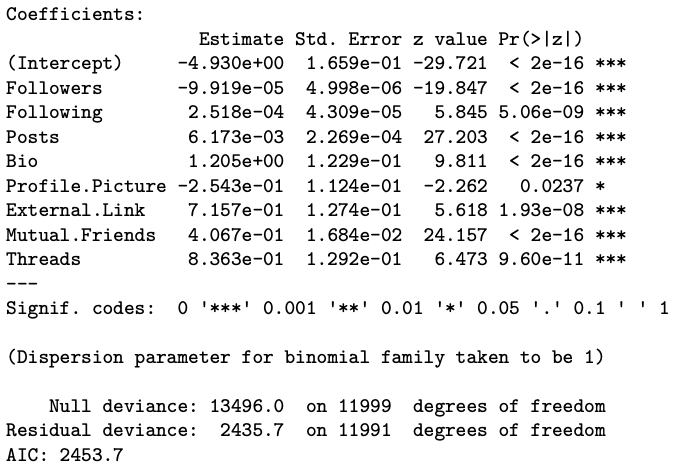


Immagine che contiene testo, schermata, Carattere, numero

Il contenuto generato dall'IA potrebbe non essere corretto.

Figure 18: Optimal cp and logistic regression

To fix this, I tried to make a second binary classification using only Real accounts versus Bot accounts. This approach was much cleaner, but when I computed the ridge, the lasso regression and the stepwise forward selection I obtained 2 different results: 4 coefficients for the ridge and the lasso and 7 for the stepwise forward selection.

Immagine che contiene linea, Diagramma, diagramma, testo

Il contenuto generato dall'IA potrebbe non essere corretto.

Immagine che contiene linea, diagramma, Diagramma, pendio

Il contenuto generato dall'IA potrebbe non essere corretto.

Figure 19: Lasso Regression

Figure 20: Ridge Regression

Immagine che contiene testo, schermata, Carattere, diagramma

Il contenuto generato dall'IA potrebbe non essere corretto.

Figure 21: Optimal cp

After building two different logistic regressions models, all variables were found to be statistically non-significant. However, this was not due to a lack of predictive power. Instead, it happened because the two classes were so easily distinguishable based on the available features that the logistic regression model struggled to compute reliable coefficient estimates. As a result, standard errors became extremely large, leading to non-significant p-values despite good classification performance.

Immagine che contiene testo, schermata, Carattere, documento

Il contenuto generato dall'IA potrebbe non essere corretto.Immagine che contiene testo, ricevuta, Carattere, bianco

Il contenuto generato dall'IA potrebbe non essere corretto.

Figure 22: 7 regressors model

Immagine che contiene testo, Carattere, ricevuta, bianco

Il contenuto generato dall'IA potrebbe non essere corretto.

Figure 23: 4 regressors model

**Conclusions**

The Random Forest was the best-performing model, almost perfectly identifying account genuineness.

The Decision Tree, while somewhat less accurate, offers a clear set of rules confirming that Follower and Following are crucial in detecting fake accounts.

The Cross-Validation performed a robust and powerful model finding in Followers and Following the most important variables.

The logistic regression (for binary classification) further highlights features like profile bios, mutual friends, and external links as critical factors.

All models agree on the fundamental traits that distinguish real and fake Instagram profiles, providing high predictive accuracy. These findings demonstrate the effectiveness of supervised learning in social media fraud detection and help prioritize which profile attributes are most important for verification processes.