Capstone 2 Fake account predictor

Problem Identification:

Many people complain about fake accounts and people passing as someone else on social media. This is a safety problem for users.
This is also a problem for the apps or the social network, as they lose users' trust and loyalty. This could be especially relevant to dating apps as the goal is to meet in person.

On the New York times website, Dec 2020, Jack Nicas wrote an article about fake accounts being a problem for social media. To site him: “despite removing billions of accounts, Facebook estimates that 5 percent of its profiles are fake, or more than 90 million accounts. He also stated that “manual fakes can be more pernicious than bots because they look more believable”.
https://www.nytimes.com/2020/12/08/technology/why-cant-the-social-networks-stop-fake-accounts.html
This is one example of a problem, another one is the case of Tinder. Women claimed to be conned out of hundreds of thousands by a Tinder user. https://www.washingtonpost.com/arts-entertainment/2022/02/06/tinder-bans-tinder-swindler-netflix-hayut-leviev/
Social networks and dating apps might want to take more preventive steps to protect users and restore trust in their products.

Goal:
The fake accounts will be identified as soon as possible. The app or the social network can take steps to verify a user or take other steps to eliminate fake accounts. This doesn’t have to be an identity verification but can be other ways of warning a person that he was identified and need to answer more questions.
A model will be created to predict whether an account is fake or not. The model will be evaluated for accuracy and prediction abilities. The best model will be applied in the industry as needed by apps and social networks. A client might want to build a system that will use the model for identifying accounts using their activity features.

Method:

Data

A publicly available dataset (on Kaggle) was used. It has 65325, user accounts identified as fake or not fake. The data will be analyzed and a model will be created. The model will allow us to predict is an account is fake or real.

The two values classifier was used: 2-class User classes: r (real/authentic user), f (fake user / bought followers)

Instagram:
https://www.kaggle.com/krpurba/fakeauthentic-user-instagram?select=user_fake_authentic_4class.csv

reference: K. R. Purba, D. Asirvatham and R. K. Murugesan, "Classification of Instagram fake users using supervised machine learning algorithms," International Journal of Electrical and Computer Engineering (IJECE), vol. 10, no. 3, pp. 2763-2772, 2020. The dataset was collected using web scraping from third-party Instagram websites, to capture their metadata and up to 12 latest media posts from each user. The collection process was executed from September 1st, 2019, until September 20th, 2019. The dataset contains authentic and fake users, filtered using human annotators. The authentic users were taken from followers of 24 private university pages (8 Indonesian, 8 Malaysian, 8 Australian) on Instagram. To reduce the number of users, they are picked using proportional random sampling based on their source university. All private users were removed, which is a total of 31,335 out of 63,795 users (49.11%). The final number of public users used in this research was 32,460 users.
I used a file with 65326 entrees and 18 features.

Features: This is the final list of features that were used in modeling.

1. pic — Picture availability | Value 0 if the user has no profile picture, or 1 if has
2. link — Link availability | Value 0 if the user has no external URL, or 1 if has
3. cap_zero_per — Percentage (0.0 to 1.0) of captions that has almost zero (<=3) length
4. no_image_per — Percentage (0.0 to 1.0) of non-image media. There are three types of media on an Instagram post, i.e. image, video, and carousel
5. loc_tag — Percentage (0.0 to 1.0) of posts tagged with location
6. class — 2-class User classes: r (real/authentic user), f (fake user / bought followers)
7. posts_a — Number of total posts that the user has ever posted-trimmed
8. flw_a — Number of followers-trimmed
9. flg_a — Number of following-trimmed
10. likes_a — Engagement rate (ER) is commonly defined as (num likes) divide by (num media) divided by (num followers)- trimmed
11. hash_a — Average number of hashtags used in a post- trimmed
12. cap_avg_a — The average number of characters of captions in media- trimmed
13. comment_r_a — Similar to ER like, but it is for comments- trimmed
14. post_interval_a — Number of total posts that the user has ever posted- trimmed.

Data wrangling:
The dataset was checked for size and cleaned before we use machine learning to answer our question. In the data wrangling process, we started with 18 features, and 3 were dropped because of uninformative and incomplete data. The three features that were dropped are the use of account for PR, the average use of keywords in PR, and the length of biography. They were dropped since more than 85% of the values were 0.

The 162 missing values were detected in form of “-1” where negative numbers were not expected. These rows dropped as they are a small proportion of data, have mostly duplicates and no significant information was lost.
The duplicates were also dropped. The sample size at the end of the wrangling process was 63587.
Further exploring the data and checking for outliers showed outliers in several features. To make sure that we are not losing important information, the class of these values was checked. The outliers belonged both to fake and real accounts.
The distribution of the data was also examined. Several features had a very skewed distribution. Extremely high values could lower the model's performance in the next stages. The solution was to trim the data at 99% . This was chosen over the interquartile range to preserve some of the higher values as they could be predicting fake accounts.
Preprocessing the data resulted in scaling the data. Since the features were on different scales, all data was scaled before modeling.
The categorical variables were dummy encoded. The target classes were: fake-1, and real-0

EDA

First step was visualizing the new distributions, after trimming. This revealed less skewed distributions but still skewed. This will be taken into account while modeling. I also expect the trimming to help the sample represent better the real data and fewer random values or mistakes, and eventually create a better model.
After the trimming, correlations between all features were plotted and a heat map was created. The correlations matrix showed that the features are not highly correlated with each other.
The feature “likes_rate” correlates with “comment_rates” which makes sense and will be taken into consideration while modeling. None of the other features were highly correlated.
In addition, I decided to check if the features are significantly different from each other in the real compared to fake accounts. Nine t-tests were performed and their p values were calculated. The results showed that all features were significantly different at a very low p-value of almost 0.
There were also two categorical features: the account having an external link to a different site” and whether the account had a profile picture. To check if the categorical features are different between fake accounts and real accounts chi square nonparametric test was performed. The results showed a significant difference at a very low p-value of almost 0. This result showed that real accounts had profile pictures and external links than fake accounts.

Model selection

Model	precision	recall	f1-score	ROC-AUC	Crossvalidated Train score	Crossvalidated Test score

					mean	mean
Logistic regression (best c)	0.81	0.81	0.81	0.8091	0.8751	0.8750
K Nearest Neighbors with parameters tuning	0.90	0.79	0.84	0.8537	0.8561	0.8519
Random Forest with parameters tuning	0.96	0.82	0.88	0.8921	0.9545	0.9522
Random Forest with threshold adjustment	0.91	0.87	0.89	0.8911	0.9544	0.9522
Ridge Classifier	0.81	0.81	0.81	0.8069	0.87419	0.87414
HistGradient Boosting Classifier: with parameters tuning	**0.94**	**0.83**	**0.89**	**0.8921**		
HistGradient Boosting Classifier: with threshold adjustment	0.91	0.87	0.89	0.8907	**0.9579**	**0.9564**
Support Vector Classifier: with parameters tuning	0.83	0.84	0.84	0.8340	**0.8336**	**0.8340**

I chose a simple model to start with and it was better than simply using the mean. Logistic regression misclassified about 19% of the data. This is not good enough if 19% of users will be fledged or have to go through a verification process. On the other hand, if the accounts are fake and misclassified, many will be allowed to continue taking advantage of the app or of people.

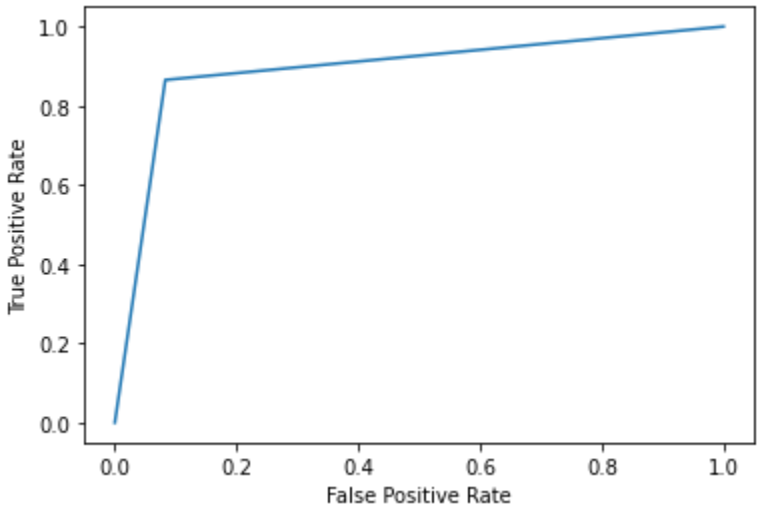
K nearest neighbors classifier, after tuning the parameters had similar results. I decided to try an ensemble method. Naive_bayes Bernoulli NB had even lower accuracy(0.739)

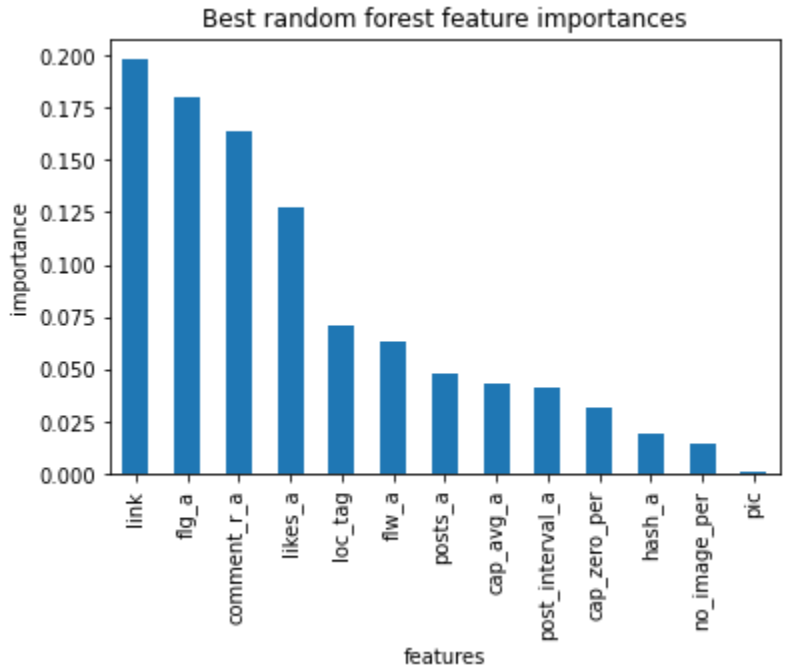
I chose to start with Random Forest and hyperparameters (max_depth, n_estimators, min_samples_leaf) tuning for best results.
Random Forest Classifier showed very good accuracy and precision, meaning true values were classified as such. But, the results were unbalanced towards higher precision and too low recall, with many false-negative cases. That means our classifier recognized the fake account cases well and falsely recognized real accounts as fake.
Since social media can pay a price if people are identified as fake accounts and users will be lost. Also to miss many fake accounts means people will be less safe. In this classification problem, we want a relatively balanced classification as it's ok if some portion of people will be misclassified. Most will go through a quick check-up process. Only a small proportion of those misclassified will be dissatisfied and we lose users. At the same time, it's ok if some fake accounts will be missed as social media also have the help of their users to detect fake accounts, a flagging system.
Unbalanced recall and precision can happen when a dataset is not balanced, but our dataset is balanced and I can't adjust it. I used a different method.
To create a better precision/recall balance, I modified the threshold at which the data points' probability of classification will be classified as positive. We needed fewer positives so the threshold was reduced from default 0.5 to 0.4. This showed high accuracy and still good and this time balanced precision vs recall rates.
After a good result with the Random Forest model, I decided to try another simple model the ridge classifier, because I had a small number of features. If a simpler model would show a good result that could be more effective and less computationally expensive. This model showed similar results to logistic regression and I continued to more complex models. Eventually, I tried another two models HistGradient Boosting Classifier and SVC (Support vector classifier). The SVC after parameter tuning didn't improve the prediction power and was computationally expansive.

Conclusion

The HistGradient Boosting Classifier from the experimental module was faster than standard Boosting models and had very similar results to Random Forest. This model also had a lower cross-validated standard deviation (, **Hist 0.0039** vs **RF 0.0058)**, meaning we have more stability in the predicted mean of our model. Since the models are similar the more stable model will be chosen.
Using this model in a social network, social media, and dating app security analysis can help predict fake accounts and use some protective measures before any harm is done.

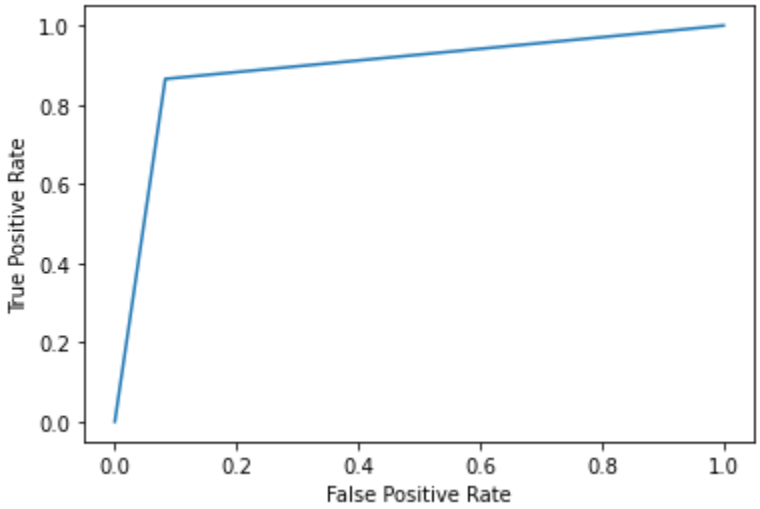
Random Forest ROC curve:





If we check the most important features we can see that having external links and following others are the best predictors of fake or true accounts. Possibly external links can support a real business that is using Instagram. It is also possible that following others is an activity that needs an investment of time from a real person that is engaged with other people. There is also a possibility of a nonlinear prediction, where too much following others is a sign of negative engagement with others. Of course, this is my assumption and this model's feature selection doesn't give us this information. To understand the relationships of these features with the type of account, a different kind of investigation of these features is needed.

HistGradient Boosting Classifier



https://www.python-graph-gallery.com/2-horizontal-barplot
https://seaborn.pydata.org/examples/part_whole_bars.html
https://www.delftstack.com/howto/seaborn/seaborn-horizontal-bar-plot/#:~:text=Horizontal%20Bar%20Graph%20Using%20Seaborn%20A%20bar%20graph,%28%29%20function%20to%20create%20a%20horizontal%20bar%20plot.