uc3m | Universidad Carlos III de Madrid

Style Definition: TOC 3: Tab stops: 14.98 cm, Right,Leader: ...

University Degree in Management & Technology

Academic Year 2022-2023

Bachelor's Thesis

"Leveraging AI Tools to Explore the Dynamics of Social Networks: Insights into Consumer Behaviour and Business Implications"

Juan Diego Gómez Labajos

José Antonio Iglesias Martínez

Getafe, June 2023

ABSTRACT

Social Networks have become the natural way of communicating, socializing and discovering new business streams for the modern civilization. They way we interact with each other has been subject of great study. If we pay attention to the main example of a social network, a social media app, we could find an interesting pattern, accounts simulating the behaviour of a human to get something from a genuine one are growing in this platform. This study focus on the detection of this fake accounts and the potential that it can reach in terms of accuracy, real-world implementation and business implications of this accurate problem solver computer techniques.

ACKNOWLEDGEMENTS

"This project has been the result of numerous subjects working and giving me directions. I would like to thank Jose Antonio Iglesias for guiding me through it and having patience and an iron hand with my errors.

I would like to thank all the professors during my university period whose teachings have made me become the man that I am and discover some passions of mine. Lastly but not least, my family for being everyday with me, motivating me to give my best."

Juan Diego

TABLE OF FIGURES

TABLES

TABLE OF EQUATIONS

CONTENTS

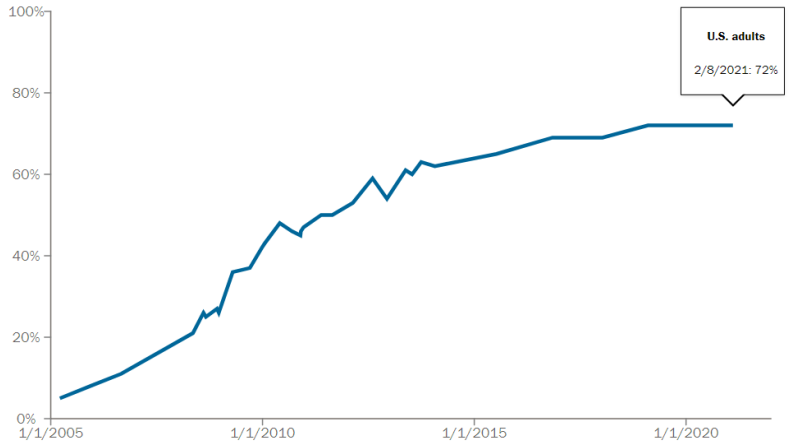
~~1.1~~ INTRODUCTION

The introduction section provides an overview of the research topic and its background, establishing the context for the study. It also presents the research problem and objectives, explaining what the study aims to achieve. The significance of the research and its contribution to the field are also discussed.

~~1.2~~1.1 Background and Context.

Society has clearly evolved to a point where social networks is a must. Creating a connection with someone means being able to track what someone is doing, communicate, share and create content, expand our network, and keeping in touch with relatives and friends. Hundreds of data and information are being exchanged every second. The rise and success of social media applications is parallel to the social network concept:

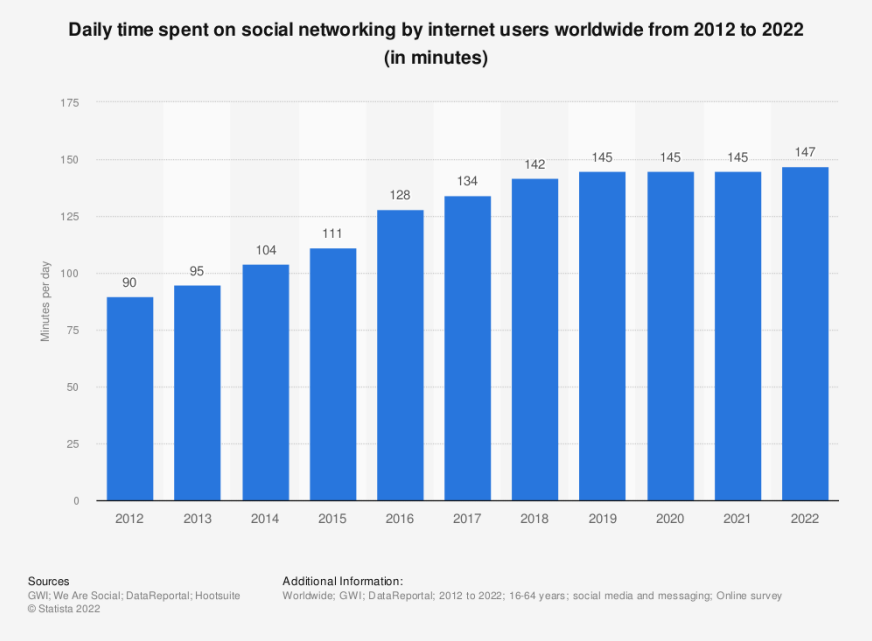
Figure 1 *Social Media Use, % of U.S. adults who say they use at least one social media site,* (Pew Research, 2023)



Note: Respondents who did not give an answer are not shown.
Source: Surveys of U.S. adults conducted 2005-2021.

Taking the US adults as a sample of the whole population, we can describe from this picture that 72% of adults in the US use social media. Several statistics like the one provided from the previous figure [1], prove an increase in the usage ignoring any type of age range, gender, community, race, income and education.

Figure 2 ***Daily time spent on social networking by internet users worldwide from 2012 to 2022(in minutes). (Statista,2023)***



The previous figure shows another interesting fact, the amount of time spent in this apps have increased too in the last decade and the trend shows that will be in the same manner. So, from these patterns we can firmly say that social networks are widely used by the people all over the world and is expected to grow, the daily average time spent on social media follows the same line.

Due to this trend, is not rare that companies are trying to get a piece of the cake. Social Media apps are now trying to increase the user engagement and attention which can be translated into more ad impressions and therefore more revenue.

Consistent attention and exposure are crucial for every business to generate leads and sales. Attention can bring more opportunity, leverage, influence, and longevity to a business. A strategy to connect with consumers at a deeper level can create brand loyalty and stimulate word-of-mouth marketing. Social media has become the main platform to capture attention and conduct market research to figure out consumer behaviour. By utilizing data-collecting tools such as polls, businesses can find out what type of content consumers want and attract new clients more efficiently. Social media is where businesses and brands need to focus their marketing and advertising strategies to gain market share.

In the article *The Impact Of AI & How It Is Used In Social Media* [2], the author identifies several ways social media apps are using Artificial Intelligence to boost their revenue and keep us engaged in their products and services. We can find several methods: recommender systems (personalization of the app customised for the individual), Customer Lifetime Value calculation (to know how worthy an individual is), content generation (AI content generated), to build better ads, improve response times, moderate content, better analytics… Instagram is now using AI to visually identify prohibited content, and Linkedin to enhance your job search. Those are some examples of the potential that leveraging AI tools on social networks have.

Companies can therefore implement AI tools to boost and increase the company performance.

1.31.2 Research Problem and Objectives

Proven that AI has an impact on social networks' revenues and performance, we are going to decide a way to improve these to by stating some problems these companies solve every day:

- Content moderation: Social media platforms must constantly monitor user-generated content for offensive, harmful, or illegal content. This requires a significant number of resources and can be a challenging task, especially with the sheer volume of content that is posted daily.
- User privacy: With increasing concerns about user privacy, social media platforms must take measures to ensure that user data is protected and not misused. This includes implementing privacy policies, complying with data protection laws, and providing users with the tools to control their privacy settings.
- Cyberbullying and harassment: Social media platforms must address the issue of cyberbullying and harassment, which can have serious psychological and emotional

consequences for victims. This includes implementing policies to prevent and address these issues, as well as providing users with tools to report and block abusive behaviour.

- Misinformation and fake news: Social media platforms have been criticized for their role in the spread of misinformation and fake news. Platforms must take measures to prevent the spread of false information, while also ensuring that freedom of expression is not compromised.

- Ad fraud: Advertisers may engage in fraudulent practices, such as click fraud, which can result in wasted ad spend and a negative impact on the user experience. Social media platforms must have measures in place to detect and prevent ad fraud.

- Fake accounts and bots: they refer to accounts created by individuals or organizations with false or misleading information, while bots are software applications that can perform automated tasks, such as posting content or interacting with other users.

The research focus is going to be centred on the last one, which represent a real threat for the user integrity and can end in loss of information or private data coming from scams.

We are going to mention some extract from this article by the CNBC, "*LinkedIn has a fake account problem it's trying to fix. Real users are part of the solution"* [3]. The article states that in January 2022, LinkedIn, the professional careers social network, had to close and delete more than 21 million accounts. LinkedIn has almost 900 million users.

In this article, Clifton let us understand the importance of removing and being vigilant of these accounts, which happen to be managed by cybercriminals to commit frauds. These cybercriminals use a human touch to make the app userbase believe in someone non-existent, and therefore exchange personal information. "*For example, we see those that revolve around posts and content promoting a work event, such as a webinar, that uses real photos and people's real information to legitimize the information and get others to register, often on a fake third-party Web site,"* Clifton said [3].

Linkedin is already working on this problem. The AI team is relying on Machine Learning models to find suspicious accounts before being flagged by users, as one of the several solutions which also include AI-based synthetic image generation technology. The complete AI solution is nowadays in development by these large companies, because the models do not classify accurately.

In this project we are going to make a solution for the companies that are facing this issue, we will **develop an accurate model to detect fraudulent accounts**, also recognised as bots or spam accounts. In the next section, we will describe the charasteristics of the project, like the reason we have decided to use Instagram as the source of the data.

~~1.4~~

~~1.5~~

~~1.6~~1.3 Significance and Contribution to the Study

In this section, we explain the significance of this study.

The general significance of this project is to create a general machine model which can be understood by any individual, using statistics and math as a foundation, to finally implement it into a business case in the real world. These companies already count with privileged and countless professionals that can make a better outcome out of this problem, but insisting, is a general accurate idea for a desirable solution.

In terms of contribution, we can consider that this problem has been an issue subject of research for the last 10 years. In this particular case, we are contributing to the originality of the model comparison (I will go into further detail on the research section) to get the best model evaluating distinct factors, which is unique and the motor of the research.

~~1.~~

~~2.~~

~~3.~~

~~4.~~

~~1.7~~1. LITERATURE REVIEW

Formatted: Heading 1, Left, Indent: Left: 0 cm, Right: 0 cm, Space Before: 0 pt

The literature review section presents a comprehensive review and analysis of existing scholarly works and relevant literature related to the research topic. It provides the theoretical framework for the study, discussing concepts, theories, and previous research findings. The section may explore topics such as social networks, artificial intelligence, machine learning, and previous studies on fake account detection.

~~4.1~~1.1 Theoretical framework.

In this section we are going to elaborate a definition for they key concepts and tools that are going to be used to achieve the purpose of the project. They are social networks, Instagram, Artificial Intelligence, Machine learning, Fake accounts and bots, Fake account detector, classifier model, algorithms.

~~4.1.1~~1.1.1 Social Networks

A social network is a platform that allows individuals to connect and communicate with one another through various means such as messaging, posting content, and sharing media. The primary function of a social network is to create and maintain relationships between people, whether they be friends, family, colleagues, or other acquaintances [4].

In terms of classes and objects, a social network can be broken down into several key components, including:

a) User: This class represents an individual who has registered on the social network. It contains attributes such as the user's name, email, profile picture, and a unique identifier such as a username or user ID.

b) Friend/Follower: This class represents a connection between two users on the social network. It typically contains attributes such as the user IDs of both parties, as well as any relevant metadata such as the date the connection was established or the type of relationship (e.g., friend, follower, colleague, etc.).

c) Post: This class represents a piece of content that a user has shared on the social network. It can include various types of media such as text, images, videos, or links, as well as metadata such as the date and time the post was created, the user who created it, and any associated comments or reactions.

d) Comment/Reaction: This class represents a user's response to a post on the social network. It typically contains attributes such as the user ID of the commenter, the post ID that the comment/reaction is associated with, and any relevant metadata such as the date and time the comment/reaction was created.

e) Group: This class represents a collection of users who have joined together on the social network to discuss a particular topic or share a common interest. It typically contains attributes such as the group name, description, and a list of member user IDs.

f) Message: This class represents a private message sent between two users on the social network. It can include various types of media such as text, images, videos, or links, as well as metadata such as the date and time the message was sent, the sender and recipient user IDs, and any associated replies.

~~This objects~~These objects are common to every social media. Now we are going to focus on Instagram, which is going to be the subject of the study.

Instagram is one of the most popular social networking platforms which allows users to share photo and video content. It counts with over a million monthly active users. You can get suggestions from the discover page based on your past activity, searches and hashtags. Also enables direct messaging, Instagram Stories, Reels, IGTV, and more. Instagram is available as a free app for iOS and Android devices. Its functionalities count with: User Authentication and Authorization (handles login and registration), User Profile Management (manage public and

private user information), Social Networking features (follow, like and comment), Content Management (create, edit and publish content), and Analytics (statistics and performance).

When it comes to the architecture of Instagram, we can find:

- Infrastructure layer: servers, network, cloud…
- Data layer: Database System Management, responsible for data storage and retrieval
- Application layer: handles the business logic and communicates with the data layer through APIs.
- User Interface layer: the layer that handles the user interactions, mobile app and website.

This information is retrieved from the android guide to apps architecture [5].

~~4.1.2~~1.1.2 Artificial Intelligence

Artificial Intelligence definition is the simulation of human intelligence processes by machines, especially computer systems [6]. It use has been extensive across every single industry, causing a disruptive advancements in markets thanks to the detection, prediction and automation of outcomes that it can generate.

Its applications can be found in a broad spectrum like healthcare, where AI is used to improve patient outcomes and reduce costs by providing faster and more accurate diagnoses, while in business, it is used to enhance customer relationship management and analytics. AI is also used in education to automate grading and provide additional support to students. In finance, AI-powered personal finance applications are disrupting traditional financial institutions, and in law, it is being used to automate labor-intensive processes. Finally, the entertainment industry is using AI for targeted advertising, content recommendation, distribution, and creating scripts, while newsrooms use it for automated journalism to streamline workflows and research topics. In software coding and IT processes, generative AI tools can be used to produce application code and automate IT processes such as data entry, fraud detection, customer service, predictive maintenance, and security. AI is also being used in cybersecurity to detect anomalies, conduct behavioral threat analytics, and identify emerging attacks. In manufacturing, robots are increasingly collaborating with human workers, and AI is used in banking to improve decision-making and handle transactions. AI is also used in transportation to manage traffic, predict flight delays, and forecast demand in supply chains [7].

We also have to explain what Machine Learning is. According to IBM, Machine learning is a branch of artificial intelligence (AI) and computer science which focuses on the use of data and algorithms to imitate the way that humans learn, gradually improving its accuracy [8]. With the algorithms (previously trained in with a dataset) we are able to predict and classify in this case, future test data. The following algorithms are of great interest for the research.

~~4.1.3~~1.1.3 Logistic Regression

Logistic Regression is a supervised machine learning algorithm used for classification problems. It can model the probability of binary or multiclass outcomes (yes or no), as the function of the predictor (independent) variables. The dependent variable or outcome is normally a binary factor, and the independent variables are usually continuous or categorical. It based on the sigmoid function which maps any real-valued number to a number between 0 and 1, used to convert the combinations of features into a probability score [9]. Equation 1 represents the mathematical expression of the algorithms defined by:

Equation 1 Logistic Regression equation.

$$p(x) = \frac{1}{1 + e^{-(x-\mu)/s}} \quad p(x) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x)}}$$

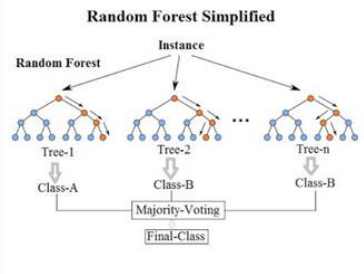
- p is the probability of the positive class (e.g., success)
- e is the Euler's number (approximately 2.71828)
- b0 is the intercept term.
- b1 is the coefficient of the i-th predictor variable (xi)

The goal is to find the features which coefficients maximize the likelihood of the observed data.

~~4.1.4~~1.1.4 Random Forest

Random Forest is an ensemble learning algorithm that combines a series of decision trees (another predictive algorithm) to improve the accuracy and robustness of the predictions. Each decision tree is built on a subset of the data and a subset of the features. The final prediction is made by aggregating the predictions of all trees. They may include irrelevant features; however, they are more accurate [22]. Figure 3 represents a diagram of a random decision forest.

Figure 3 Representation of Treemap of Random Forest



~~4.1.5~~1.1.5 SVM

Support Vector Machine is a supervised machine learning algorithm used for classification. It finds the hyperplane that best separates the data points into different classes. The hyperplane is chosen such that the margin between the closest points from each class is maximized. The SVM model can also handle non-linearly separable data by transforming the data into a higher-dimensional space using a kernel function [20].

Figure 4 Width margin method for SVM

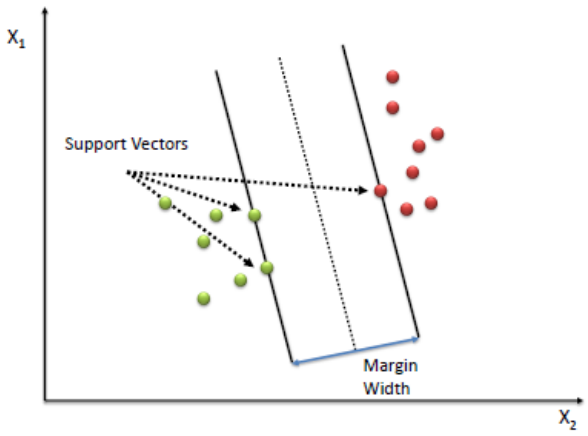


Figure 4 is a representation of the maximum-margin hyperplane and margins for an SVM trained with samples from two classes. Samples on the margin are called the support vectors.

~~4.1.6~~1.1.6 AdaBoost

AdaBoost or Adaptive Boosting, is an ensemble learning algorithm that combines multiple weak classifiers to form a strong classifier. Each weak classifier is trained on a subset of the data and assigned a weight based on its performance. The final prediction is made by weighted voting of all weak classifiers. AdaBoost refers to a particular method of training a boosted classifier [23]. A boosted classifier is a classifier of the form of Equation 2:

Equation 2 AdaBoost weak learner expression.

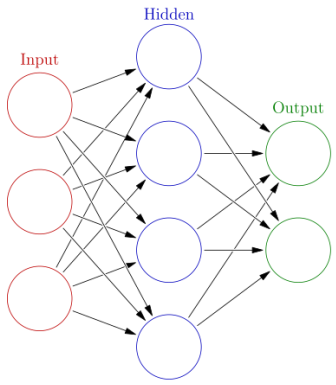
$$F_T(x) = \sum_{t=1}^{T} f_t(x)$$

where each ft is a weak learner that takes an object x as input and returns a value indicating the class of the object.

~~4.1.7~~1.1.7 Neural Networks

Neural Networks are a class of deep learning algorithms which simulate the human brain. They consist of multiple layers of interconnected neurons that perform transformation to reach an output. Each neuron computes a weighted sum of the inputs, applies a function and passes the information to the next layer of neurons [21]. In Figure 5 we can see the representation of the layers presented in ANN, Input which is the entry or observation, hidden represents the computational part, where the algorithm gives weights to each observation and feature, and Output, the conclusion layer.

Figure 5 Representation of a Neural Network.



~~4.1.8~~1.1.8 Fake Account Detector

A fake account detector is a program or system made to spot and flag accounts that are suspected to be fraudulent or fake. To examine account behaviour, such as posting frequency, content, and interactions with other users, these tools frequently use algorithms and machine learning approaches. They might also check for discrepancies in profile data, including inconsistent profile images or dubious email addresses. Since false accounts can be used for a variety of unwanted activities, including the dissemination of misinformation, phishing scams, or online harassment, fake account detectors are crucial for protecting the integrity and security of online communities. Social media companies and other online services can assist protect their users and guarantee a safe environment by detecting and eliminating rogue accounts. An example of a fake account is a bot, a program which automates the creation and models the behaviour or a real account to make itself grow or complete a wide variety of tasks. We can also find scam accounts, which profile try to pretend a famous or non-existent person to get personal information from third parties [1].

~~4.1.9~~1.1.9 Tools for the development and application of AI techniques

The following tools have been applied to developing the technical part of this research.

- Python.

  Python is a well-liked programming language renowned for its ease of use and adaptability. A specific version of the Python programming language is referred to as Python 3.8.8. According to the version number, it is a member of the 3.x series, which superseded the 2.x series with major advancements and new features. The Python ecosystem includes Python 3.8.8, which is used for many different things, such as web development, data analysis, artificial intelligence, and more [24].

- GPT-3.5 Language Model.

  The term "Generative Pre-trained Transformer 3.5 Language Model" refers to a program created by OpenAI. An advanced language model called GPT-3.5 LLM uses deep learning methods to produce text that resembles human speech in response to input [26].

- Jupyter Notebooks

Users may create and share documents with live code, equations, visualizations, and narrative text using this interactive computing environment. With its web-based interface, users may create and run programs in a variety of programming languages, such as Python, R, and Julia. Jupyter Notebooks group code cells that can each be run independently [27].

~~4.2~~1.2 Overview of AI tools in Social Networks.

In this section we are going to analyse the several AI tools which are being used in the industry and happen to have a great effect on companies' outcomes.

There are several AI tools and methods available for studying the dynamics of social networks. Here are some ways to use AI tools for this purpose.

1. Social Network Analysis (SNA) Tools: SNA tools can be used to analyze relationships between individuals or groups within social networks. They help identify key players, subgroups, and patterns of interaction within the network. Common SNA libraries and toolkits include Gephi (open-source network analysis and visualization tool. It provides a user-friendly interface that allows users to import network data, manipulate it, and visualize the networks in various ways)[10], UCINET (University of California, Irvine, NETwork, is a software package for social network analysis and visualization. It provides a comprehensive suite of tools for analyzing social network data, including network visualization, centrality analysis, subgroup identification, and statistical analysis of network data)[35] and NodeXL (open-source add-in for Microsoft Excel that enables network analysis and visualization) [11].

2. Natural Language Processing (NLP) Tools: NLP tools can be used to analyze the content of social media posts, comments and messages. They help identify themes, sentiment, and other characteristics of content that may be relevant to social network dynamics. Common NLP libraries and toolkits include NLTK (widely used open-source library for NLP in Python. It provides a comprehensive set of tools and resources for tasks such as tokenization, stemming, lemmatization, part-of-speech tagging, parsing, and more)[12], spaCy (a powerful and efficient NLP library for Python. It offers a streamlined API and focuses on providing high-performance natural language processing) [13], and CoreNLP (suite of NLP tools developed by Stanford University. It provides a range of natural language processing capabilities, including tokenization, part-of-speech tagging, named entity recognition, sentiment analysis, coreference resolution, and more)[14].

3. Machine learning (ML) tools: ML tools can be used to analyze large amounts of data and identify patterns and trends in the behaviour of individuals or groups within social networks. They help predict future behaviour, spot anomalies, and recommend actions based on data. Common ML libraries include TensorFlow (Open source library developed by Google which is widely use in deep learning tasks)[15], Keras (a high-level neural network library that can run on top of various deep learning backends, including TensorFlow)[16], and scikit-learn (the most extensive machine learning library in terms of usage, provides a rich set of tools for various tasks, including classification, regression, clustering, dimensionality reduction, and model evaluation)[17].

4. Network visualization tools: Network visualization tools help you visualize the structure and dynamics of social networks in a meaningful way. They help identify clusters, bridges, and other important features of the network. Common network visualization tools include Cytoscape (open-source platform for visualizing and analyzing molecular and genetic interaction networks)[18], Gephi, and Graphviz (open-source graph visualization software. It provides a set of command-line tools for generating graph visualizations from simple text descriptions called DOT files)[19]. Leverage these AI tools to gain insight into social network dynamics and use that information to make more informed decisions about social media strategy, marketing efforts, and more.

~~4.3~~1.3 Previous Research.

In this section we are going to demonstrate previous research taking into consideration for this project.

The exploration of social networks and their impact on consumer behavior and business implications is an area of active research. Several publications have contributed valuable insights in this field, as evidenced by citations in Google Scholar.

One noteworthy resource is the book "Mining the Social Web: Data Mining Facebook, Twitter, LinkedIn, Google+, GitHub, and More" by Matthew A. Russell (2014) [28]. This book delves into the methods and techniques for extracting valuable insights from various social media platforms, including Facebook, Twitter, LinkedIn, and more. It covers a range of topics such as data scraping, sentiment analysis, and network analysis.

Another publication, "Predicting Consumer Behaviour with Web Search" by Matthew Goldman et al. (2014) [41], focuses on leveraging web search data to predict consumer behavior. The authors propose a model that utilizes search query data to forecast box office revenues for movies. This research demonstrates the potential of utilizing online data for consumer behavior analysis and prediction. For a comprehensive understanding of social network analysis (SNA) and its applications in different business domains, "Social Network Analysis and Mining for Business Applications" by Boleslaw Szymanski et al. (2014) [38] offers valuable insights. This book covers various aspects of SNA, including community detection, influence analysis, recommendation systems, and viral marketing. In "Analysing social media data in business applications: Case studies and pitfalls" by Alan Mislove et al. (2016) [35], the authors present case studies that highlight the challenges and potential pitfalls of analyzing social media data for business applications. The paper emphasizes the significance of considering ethical and privacy concerns when leveraging social media data. To gain insights into consumer behavior within social media and the role of brands within online communities, "Consumer Behavior in Social Media: The Role of Brands in Online Communities" by Martina Caic et al. (2018) [39] is a valuable resource. This study investigates the factors that influence consumers' engagement, loyalty, and purchase intentions on social media platforms.

Lastly, "The Role of Social Media in Human Resource Management: Opportunities and Challenges" by Nitika Gupta et al. (2019) [40] examines the potential of social media platforms for human resource management. The paper discusses how social media can be leveraged for

recruitment, employee engagement, and employer branding, while also addressing associated challenges.

These references serve as a foundation for understanding the research conducted in the realm of leveraging AI tools to explore social networks, consumer behavior, and their implications for businesses.

The detection of fake accounts on social media platforms, including Instagram, has garnered significant attention in the research community. Researchers have explored various techniques and approaches to effectively identify and combat the presence of fraudulent accounts. Here are some notable research papers in this domain:

One prominent study, "Spotting Fake Accounts in Social Networks via Supervised Learning" by Srijan Kumar, Xiaohan Yang, and others (published in WWW 2018), proposes a supervised learning approach for fake account detection. This method considers user profile attributes, social network structure, and user activities as features to train a model and identify fake accounts. In "DeBot: Twitter Bot Detection via Deep Learning" by Fei Wu, Yanfang Ye, and others (published in ICWSM 2018), the authors introduce DeBot, a deep learning-based approach for detecting Twitter bots. Their method combines user metadata and behavioral information to accurately identify automated accounts. The paper titled "FakeOff: A Framework to Spot Fake Users in Online Social Networks" by Rupinder Paul Khandpur and others (published in IEEE Transactions on Information Forensics and Security 2015) presents the FakeOff framework. This framework employs user profile properties, network structure, and user behavior patterns to identify fake accounts in online social networks.

For detecting fraudulent accounts on Instagram, the study "Instagram Fraud Detection using Convolutional Neural Networks" by Harsh Thakkar and others (published in IJCNN 2019) proposes the use of convolutional neural networks (CNNs). By leveraging image-related features and user engagement patterns, the researchers demonstrate the effectiveness of CNNs in identifying fake profiles on Instagram. Another research work, "Fake Account Detection on Instagram: A Classification Approach using Heterogeneous Features" by Khalil Raza, Shoaib Mehmood, and others (published in Applied Sciences 2021), focuses specifically on detecting fake accounts on Instagram. The study employs a classification approach that combines various features, including user-level, network-level, and content-level attributes, to accurately identify fake profiles.

These references provide valuable insights into the ongoing research efforts aimed at detecting fake accounts in social media platforms, shedding light on the development of effective techniques and approaches to combat fraudulent activities.

Special mention to the main three resources used for the research because these research works are using the same dataset that in this project.

Mallampeta, Mamatha & Datta, M & Ansari, Umme & Shaik, Subhani. (2021). Fake Profile Identification using Machine Learning Algorithms. 11. 60-65. 10.9790/9622-1107036065. “Algorithms will be trained with all previous users fake and genuine account data and then whenever we give new test data then trained model will be applied on new test data to identify whether given new account details are from genuine or fake users.”. The paper focuses on identifying genuine and fake Instagram profiles using machine learning algorithms. The dataset is pre-processed using Python libraries to obtain a suitable algorithm. The study uses Random Forest, Network, and Support Vector Machines classification algorithms to detect fake accounts on social media platforms. [1] The purpose of using this paper is to understand how the process of identifying fake accounts work and therefore understand the big picture.

The article titled "Insta-Fake: Detecting Fake Accounts on Instagram with Machine Learning" [1] by Luis Torres discusses the application of machine learning for detecting fake accounts on Instagram. The author highlights the prevalence of fake accounts and the need for effective methods to identify and mitigate them. The article outlines the approach taken to develop a machine learning model for fake account detection. It describes the dataset used, which consists of features such as the number of followers, followings, posts, and engagement metrics. The author explains the process of preprocessing the data and selecting relevant features for training the model.

The machine learning model utilized in this study is a Random Forest classifier. The author explains the concept of Random Forest and how it can effectively classify fake and genuine accounts based on the selected features. Evaluation metrics such as accuracy, precision, recall, and F1 score are used to assess the performance of the model.

The results of the experiment indicate that the machine learning model achieved promising results in detecting fake accounts on Instagram. The author discusses the implications of this approach, emphasizing the potential of machine learning in improving the platform's security and user experience.

Overall, the article provides insights into the use of machine learning techniques for identifying fake accounts on Instagram. It highlights the importance of leveraging AI tools to tackle the

issue of fake profiles and contributes to the ongoing research in the field of social network analysis and fraud detection. This paper will be the main component in which I will build the foundation of this paper.

In another study by Rahimi (2023), which utilizes the same dataset and tools as the previous article, the focus is on fake social media account detection. The study specifically explores Logistic Regression tuning as an approach for detecting fake accounts. This research serves as a supporting paper for the main research, providing additional insights and conclusions in the field of fake account detection.

~~5.~~2. RESEARCH METHODOLOGY

The research design and methodology form the foundation of the study, outlining the approach and framework used to investigate a specific research question or objective. It involves designing a systematic plan that includes data collection, analysis, and interpretation methods.

~~5.1~~2.1 Research Design.

The research will follow the machine learning problem steps as explained here:

a. Problem Definition: Clearly define the research problem or objective that the machine learning model aims to address. This step involves understanding the problem domain, identifying the variables of interest, and defining the desired outcome. We have stated the problem domain already.

b. Data Collection: Gather relevant data that is suitable for training and evaluating the machine learning model. This can involve collecting data through surveys, experiments, sensors, APIs, or accessing existing datasets. Next section will address this event.

   From this instance on, we will provide the following in the next section, Fake account detector, where the research is being conducted.

c. Data Preprocessing: Clean and preprocess the collected data to ensure its quality and suitability for the machine learning model. This step may include handling missing values, removing outliers, normalizing or scaling features, and encoding categorical variables.

d. Feature Engineering: Select or create the most informative features from the available data. Feature engineering involves transforming the raw data into a representation that captures the essential information and patterns relevant to the research problem. Contains EDA, Exploratory Data Analysis.

e. Model Selection: Choose an appropriate machine learning algorithm or model that aligns with the research problem and data characteristics. This step involves considering factors such as the type of problem (classification, regression,

clustering, etc.), the size of the dataset, and the desired interpretability or complexity of the model. The models that are to be included in the research are the ones stated in the literature review.

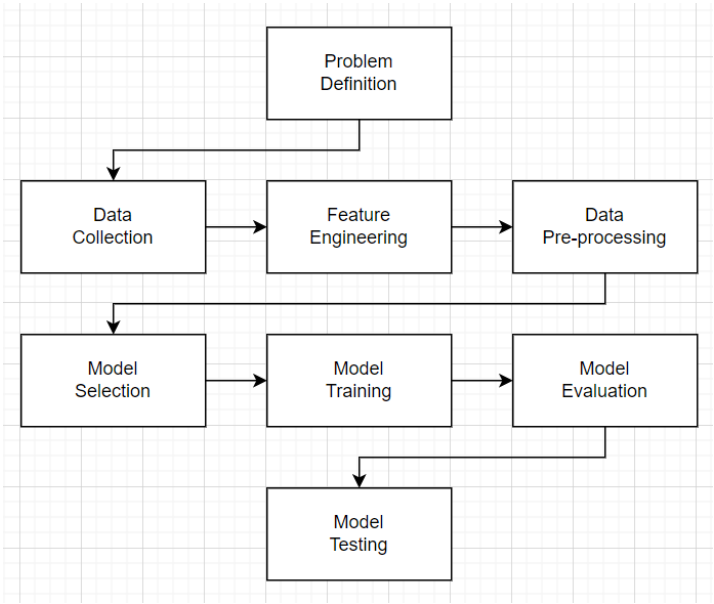
f. Model Training: Train the selected machine learning model using the prepared dataset. This step involves feeding the model with the input data and adjusting its internal parameters to optimize its performance on the training data.

g. Model Evaluation: Assess the performance of the trained model using evaluation metrics appropriate for the research problem. This step involves testing the model's predictions or classifications on a separate validation or test dataset and measuring its accuracy, precision, recall, F1-score, or other relevant metrics like ROC curve.

h. Model Testing: refers to the evaluation of a trained model's performance on unseen or test data. After training a machine learning model using a training dataset, it is essential to assess how well the model generalizes to new data that it hasn't encountered during the training phase. This evaluation helps determine the model's effectiveness and its ability to make accurate predictions on real-world, unseen data.

The representative diagram of the process is given in Figure 6.

Figure 6 Research Methodology representation.



3.2. Data Collection and Preparation.

Data collection and preparation involve gathering and organizing relevant information or datasets to address the research question or objective. This process may include collecting data through various sources, such as surveys, interviews, observations, or existing datasets. Additionally, data preparation involves cleaning, organizing, and transforming the collected data into a suitable format for analysis.

The dataset used to conduct the research is found in Kaggle. *Instagram fake spammer genuine accounts* [1] has the following description according to the webpage:

3.2.1. Dataset used in this project

Context: Fakes and spammers are a major problem on all social media platforms, including Instagram. This is the subject of my final-year project in which I set out to find ways of detecting them using machine learning. In this dataset fake and spammer are interchangeable terms.

Content: I have personally identified the spammer/fake accounts included in this dataset after carefully examining each instance and as such the dataset has high level of accuracy though there might be a couple of misidentified accounts in the spammers list as well. The dataset has been collected using a crawler from 15-19, March 2019.

Inspiration: This dataset could be further improved in quantity and quality measures, but how much accuracy can it achieve? Possible ways of using the models to tackle the problem?

Metadata: Here is a summary of the dataset metadata:

- Collaborators: Bardiya Bakhshandeh (Owner)
- Authors: Bardiya Bakhshandeh
- Bio: -
- Temporal Coverage Start Date: -
- Temporal Coverage End Date: -
- Geospatial Coverage: -
- DOI (Digital Object Identifier) Citation: -
- Sources: -
- Collection Methodology: -

License: Attribution 3.0 Unported (CC BY 3.0)

Activity Overview (until 9 June 2023):

- Dataset Stats:
    - Views: 40,846
    - Downloads: 5,002
    - Download per View Ratio: 0.12
    - Total Unique Contributors: 19
- Notebook Stats:
    - Notebooks: 17
    - Notebook Comments: 17
    - Upvote per Notebook Ratio: 6.18
    - Notebook Upvotes: 105

Downloads (until 9 June 2023):

- June 2022: 0
- August 2022: 20
- October 2022: 40
- December 2022: 60
- February 2023: 0
- April 2023: 20

The information which is not available on the website is shown with a hyphen.

Summary of the files: The dataset counts with 2 .csv files, train.csv for the training set and test.csv for the test set, both of them count with the same 24 columns (20 of them Integer datatype and 4 Decimal). Diving deeper:

Training set contains 576 instances, divided equally between spammers and non-spammers. Test set of 120 instances, half spammer, half non-spammer.

Regarding the columns:

- profile pic: Indicates whether the user has a profile picture or not.

- nums/length username: Ratio of numerical characters in the username to its length.

- fullname words: Number of word tokens in the full name.

- nums/length fullname: Ratio of numerical characters in the full name to its length.

- name==username: Determines if the username and full name are exactly the same.

- description length: Length of the bio in characters.

- external URL: Indicates whether the user has an external URL in their profile.

- private: Specifies if the user's account is set to private.

- #posts: Number of posts made by the user.

- #followers: Number of followers the user has.

- #follows: Number of accounts the user follows.

- fake: Classifies the user as either 0 (genuine) or 1 (spammer). This will be the target class, being 1 fake and 0 real account.

Table 1 represents some entries of the dataset and how it looks.

Table 1 Dataset extract

profile pic	nums/length username	fullname words	nums/length fullname	name==username	description length	external URL	private	#posts	#followers	#follows	fake	
0	1	0.33	1	0.33	1	30	0	1	35	488	604	0
1	1	0.00	5	0.00	0	64	0	1	3	35	6	0
2	1	0.00	2	0.00	0	82	0	1	319	328	668	0
3	1	0.00	1	0.00	0	143	0	1	273	14890	7369	0
4	1	0.50	1	0.00	0	76	0	1	6	225	356	0
...	...	...	...	...	...	...	...	...	...	...	...	...
115	1	0.29	1	0.00	0	0	0	0	13	114	81	1

											1	
116	1	0.40	1	0.00	0	0	0	0	4	150	164	1
117	1	0.00	2	0.00	0	0	0	0	3	833	3572	1
118	0	0.17	1	0.00	0	0	0	0	1	219	1695	1
119	1	0.44	1	0.00	0	0	0	0	3	39	68	1

3.3. Description of AI toolset.

AI tools refer to the technological advancements and algorithms used to enhance and automate various tasks using artificial intelligence techniques. These tools can include machine learning algorithms, natural language processing, computer vision, and other AI-based approaches. Incorporating AI tools in research can help in data analysis, pattern recognition, prediction, and other complex tasks, thereby increasing efficiency and accuracy in research findings.

For this research, the technologies used are:

- Python 3.8.8 (default, Apr 13, 2021, 15:08:03) [MSC v.1916 64 bits (AMD64)], this programming language will help us with addressing all the mathematical inquiries and steps throughout the study.
- GPT 3.5 LLM by OpenAI, in case we need help or inquiries on the theoretical or practical use of Python.
- Jupyter Notebooks, Python notebook framework we will work on. Contact the Author for further information if you are interested in the code.
- Python libraries:
    - Pandas: A data manipulation and analysis library that provides data structures and functions for efficiently working with structured data, such as data frames.

- Matplotlib: A plotting library that allows for creating static, animated, and interactive visualizations in Python, providing a wide range of plot types and customization options.
- Plotly: A library for creating interactive and visually appealing plots and dashboards, offering features like zooming, panning, hover tooltips, and exporting plots in various formats.
- Numpy: A fundamental library for scientific computing in Python, providing support for large, multi-dimensional arrays, matrices, and mathematical functions for array operations.
- Sklearn: Also known as scikit-learn, it is a comprehensive machine learning library offering various algorithms for classification, regression, clustering, dimensionality reduction, and more.
- Warnings: A module in Python that allows controlling the display of warning messages, providing flexibility in handling warnings during code execution.

4. FAKE ACCOUNT DETECTOR

This section focuses specifically on the fake account detection aspect of the study. It begins by defining the problem and explaining the goals of the fake account detection model. It covers topics such as data collection, data preprocessing, feature engineering, exploratory data analysis, model selection, model training, model evaluation, and model tuning. We are introducing the whole researching and results, with detailed explanations on every step taken to reach a final conclusion.

4.1 Problem Definition

The stated problem which will be the focus of the project is:

“Develop a reliable machine learning model to detect fraudulent accounts”

Requirements:

- Most accurate possible.
- Capable of correctly classifying fake accounts (TP).
- Least legitimate accounts classified as fake accounts.
- Capable of being implemented and used as a real world solution.
- Open for improvements.

4.2. Data Collection

The information on this subject is presented in the previous block.

4.3. Data Preprocessing

In this part, we are going to clean and preprocess the collected data to ensure its quality and suitability for the machine learning model. This step may include handling missing values, removing outliers, normalizing or scaling features, and encoding categorical variables.

The first step is to drop missing values and removing duplicates present in both datasets.

And then, we retrieve the fundamental description of statistical values:

Table 2 Descriptive Statistics of the dataset

	profile pic	nums/length username	fullname words	nums/length fullname	name==username	description length	external URL	private	#posts	#followers	#follows
count	574.000000	574.000000	574.000000	574.000000	574.000000	574.000000	574.000000	574.000000	574.000000	5.740000e+02	574.000000
mean	0.700348	0.162822	1.459930	0.036220	0.034843	22.618467	0.116725	0.383275	107.477352	8.559514e+04	508.972125
std	0.458505	0.212079	1.054019	0.125321	0.183542	37.742016	0.321372	0.486609	402.682002	9.117223e+05	919.341307
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000e+00	0.000000
25%	0.000000	0.000000	1.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	3.900000e+01	58.000000
50%	1.000000	0.000000	1.000000	0.000000	0.000000	0.000000	0.000000	0.000000	9.000000	1.505000e+02	229.500000
75%	1.000000	0.310000	2.000000	0.000000	0.000000	34.000000	0.000000	1.000000	80.750000	7.132500e+02	587.750000
max	1.000000	0.920000	12.000000	1.000000	1.000000	150.000000	1.000000	1.000000	7389.000000	1.533854e+07	7500.000000

Some interesting insights are:

1. Profile Picture: The "profile pic" column seems to represent whether a profile has a profile picture or not. The mean of 0.7 indicates that approximately 70% of the profiles have a profile picture.

2. Username Characteristics: The "nums/length username" column suggests the ratio of numbers to the length of the username. The mean of 0.16 indicates that, on average, usernames contain a relatively low proportion of numbers compared to the length of the username.

3. Full Name Characteristics: The "fullname words" column represents the number of words in the full name. The mean of 1.46 suggests that, on average, full names consist of around 1 or 2 words.

4. Numeric/Length of Full Name: The "nums/length fullname" column indicates the ratio of numbers to the length of the full name. The mean of 0.036 suggests that, on average, full names contain a low proportion of numbers relative to the length of the full name.

5. Name Equals Username: The "name==username" column indicates whether the name is the same as the username. The mean of 0.034 suggests that, on average, the name is equal to the username in a relatively small proportion of profiles.

6. Description Length: The "description length" column represents the length of the description. The mean of 22.62 indicates that, on average, descriptions are around 22 characters long.

7. External URL: The "external URL" column indicates whether the profile has an external URL. The mean of 0.117 suggests that approximately 11.7% of the profiles have an external URL.

8. Private Profile: The "private" column represents whether the profile is private. The mean of 0.383 suggests that approximately 38.3% of the profiles are private.

9. Post Count, Followers, and Follows: The "#posts", "#followers", and "#follows" columns indicate the respective counts for posts, followers, and accounts followed. The mean values provide an average reference for these metrics.

10. Fake: The "fake" column indicates whether the profile is identified as fake. The mean of 0.5 suggests an equal proportion of fake and non-fake profiles in the dataset.

4.4. Feature Engineering and Exploratory Data Analysis (EDA)

Select or create the most informative features from the available data. Feature engineering involves transforming the raw data into a representation that captures the essential information and patterns relevant to the research problem. Contains EDA, Exploratory Data Analysis.

Even though in the reference work they decide to engineer two more features, this study will withdraw that step as an arbitrary option, so we maintain the dataset intact.

Now we start analyzing different features in a dataset. To better understand and categorize the features, we define a function called **categorize_features**. This function takes a dataset as input and aims to categorize the features into two types: continuous features and binary features.

The function iterates over each column in the dataset and examines the number of unique values present in that column. If the number of unique values is less than or equal to 2, it considers the feature as a binary feature. Examples of binary features could be whether a profile has a profile picture, if the name is the same as the username, if the profile has an external URL, if the profile is private, or if the profile is flagged as fake. These features are then stored in a list called **binary_features**.

Binary features:

['profile pic', 'name==username', 'external URL', 'private', 'fake']

On the other hand, if a column has more than 2 unique values, it categorizes the feature as a continuous feature. These features typically involve numerical or textual data that can take on a wide range of values. For instance, the number of characters in a username, the number of words in a full name, the length of a description, the number of posts, the number of followers, and the number of accounts followed. These continuous features are stored in a separate list called **continuous_features**.

Continuous features:

['nums/length username', 'fullname words', 'nums/length fullname', 'description length', '#posts', '#followers', '#follows']

By categorizing the features in this way, you gain a better understanding of the types of variables present in the dataset, which can guide further analysis and modeling efforts.

From now on, we are able to draw some patterns from the numerous features that we just categorized.

Exploratory Data Analysis

In Figure 7 we can see that both factors for fake feature are completely balanced, which is a great characteristic that will not bias our results and let us skip some steps in the modelling part. This is indicating that we have the same entries of both fake and genuine accounts in the sample.

Figure 7 Piechart of Balance of Target class

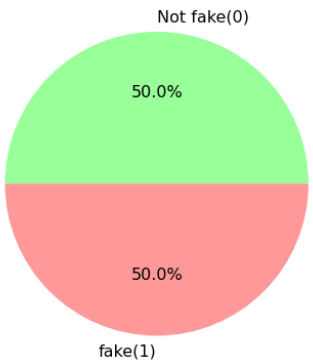


Figure 8 Scatter plot on first feature

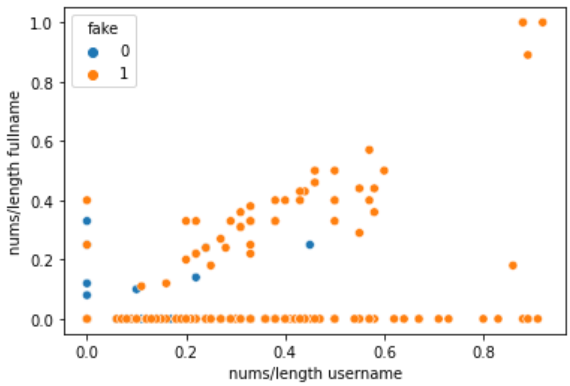
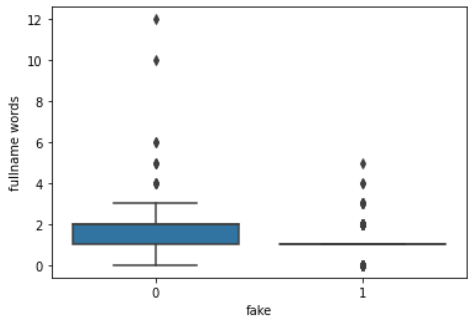


Figure 8 visualizes and analyses the relationship between the target variable "fake" (indicating whether a profile is identified as fake) and the number of words in the full name of the profiles. It indicates that there is a positive correlation which is not very clear for genuine accounts.

Figure 9 Boxplot on fullname features



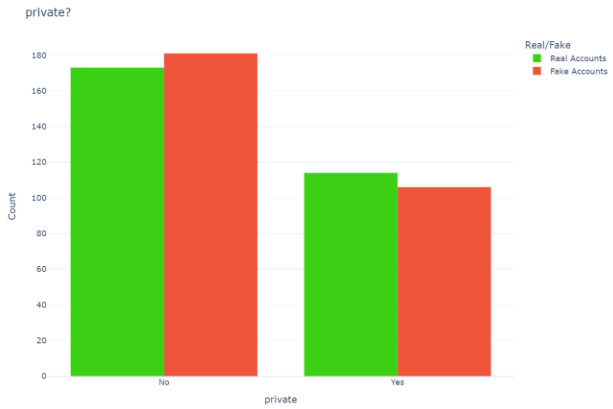
The box plot, Figure 9 is constructed to explore the relationship between two variables: "fake" and "fullname words". The variable "fake" represents whether a profile is identified as fake, while "fullname words" indicates the number of words in the full name of the profiles.

We can firmly say that fake accounts have shorter names on average than genuine.

Figure 10 Barchart on profile pic

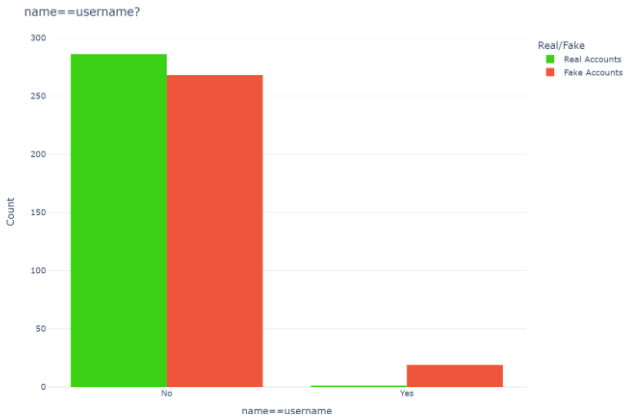
The Figure 10 represents a graph demonstrating the variability of having a profile picture in the account or not. We can see that most fake accounts do not have a profile picture.

Figure 11 Barchart on private profile



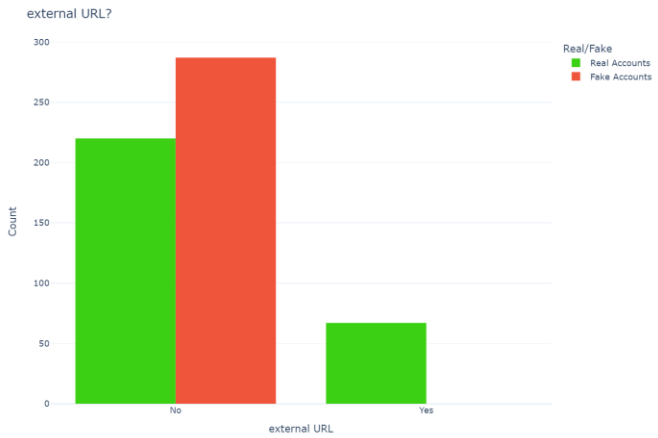
In this one, Figure 11, we can state that the majority of the private accounts are legitimate, while the public accounts are fake.

Figure 12 Barchart on name == username



In Figure 12, this one visualizes the histogram saying how many accounts have the same name and username (the name is the legal name of the owner of the account, while username is the nickname with which accounts can be publicly found). Overall, most accounts which name is the same as the username, are fake.

Figure 13 Barchart on esternal URL



Finally, in Figure 13 we can see that most accounts with external URL are genuine, meaning that fake accounts do not really have any. This seems counterintuitive but is proven.

Out of all these binary features we came up with the first hypothesis from our discoveries.

Fake accounts tend to be:

1. Not private.
2. No external URL.
3. Same username than name.
4. No profile pic.

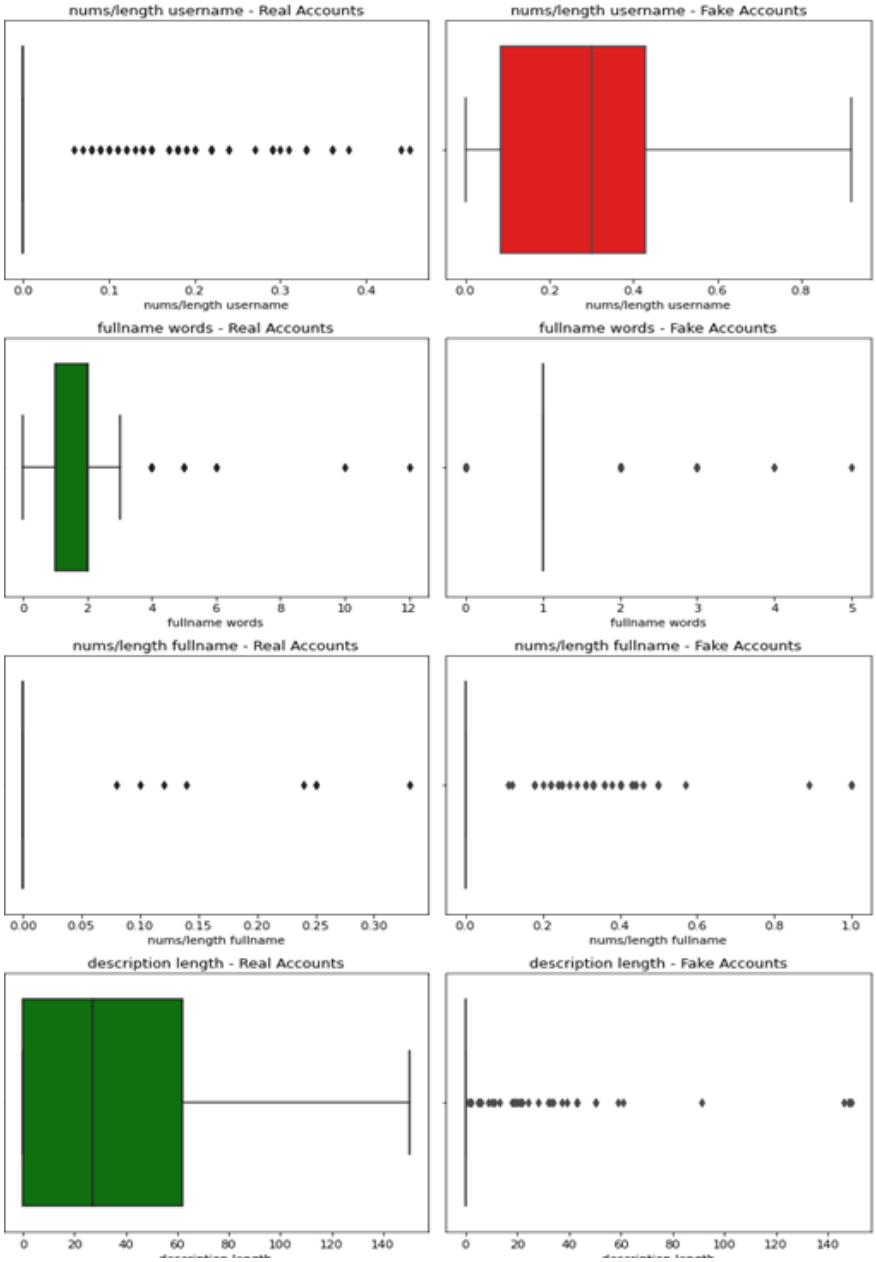
Now we are going to implement the same approach to the continuous variables.

The following figure, represents the boxplots of the continuous variables, and these are the insights we got from them.

1. Numerical characters are more prevalent in the usernames of fake accounts.
2. Fake accounts generally have shorter full names.
3. In general, excluding extreme cases, fake accounts have significantly shorter descriptions or bios.
4. Real accounts have a much higher number of posts compared to fake accounts.
5. Real accounts tend to have a significantly larger number of followers than fake accounts.
6. On average, excluding extreme cases, fake accounts follow fewer people than real accounts.

    There is a considerable presence of outliers in our data. Therefore, we will take this into account for the model selection.

Figure 14 Boxplots on continuous features.



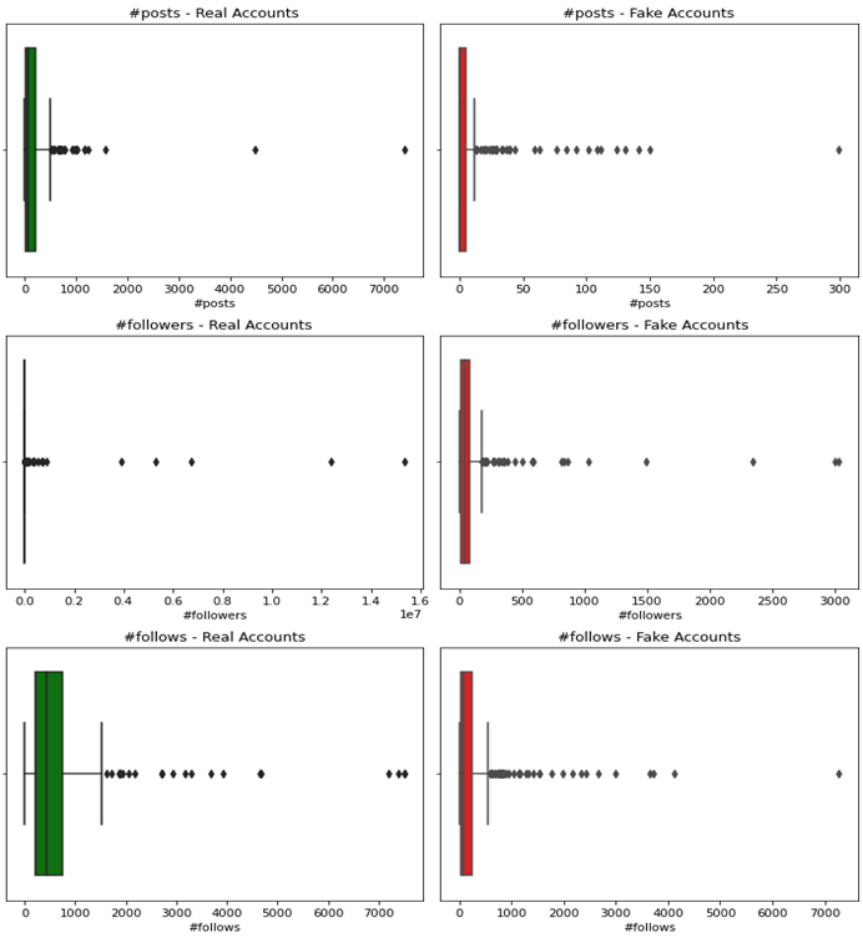

#posts - Real Accounts
0 1000 2000 3000 4000 5000 6000 7000
#posts
#posts - Fake Accounts
0 50 100 150 200 250 300
#posts
#followers - Real Accounts
0.0 0.2 0.4 0.6 0.8 1.0 1.2 1.4 1.6
1e7
#followers
#followers - Fake Accounts
0 500 1000 1500 2000 2500 3000
#followers
#follows - Real Accounts
0 1000 2000 3000 4000 5000 6000 7000
#follows
#follows - Fake Accounts
0 1000 2000 3000 4000 5000 6000 7000
#follows

Table 3 Correlation table

	nums/length username	fullname words	nums/length fullname	description length	#posts	#followers	#follows
nums/length username	1.000000	-0.224733	0.414616	-0.320433	-0.157215	-0.063036	-0.170967
fullname words	-0.224733	1.000000	-0.094361	0.271765	0.072931	0.033246	0.094378
nums/length fullname	0.414616	-0.094361	1.000000	-0.117585	-0.057723	-0.027130	-0.068185
description length	-0.320433	0.271765	-0.117585	1.000000	0.144333	0.005939	0.226049
#posts	-0.157215	0.072931	-0.057723	0.144333	1.000000	0.321433	0.097942
#followers	-0.063036	0.033246	-0.027130	0.005939	0.321433	1.000000	-0.011131
#follows	-0.170967	0.094378	-0.068185	0.226049	0.097942	-0.011131	1.000000

Here are some observations based on the correlation table (Table 3):

1. "nums/length username" shows a positive correlation with "nums/length fullname" (0.415) and a negative correlation with "description length" (-0.320). This suggests that profiles with longer usernames tend to have longer full names and shorter descriptions.

2. "fullname words" has a weak negative correlation with "nums/length username" (-0.225) and a weak positive correlation with "description length" (0.272). This indicates that profiles with more words in their full names tend to have shorter usernames and longer descriptions.

3. "nums/length fullname" has a positive correlation with "description length" (0.272), indicating that profiles with longer full names tend to have longer descriptions.

4. "description length" shows a negative correlation with "nums/length username" (-0.320) and a positive correlation with "fullname words" (0.272). This suggests that profiles with shorter usernames tend to have longer descriptions, and profiles with more words in their full names tend to have longer descriptions.

5. The "#posts" variable has a weak negative correlation with "nums/length username" (-0.157) and a weak positive correlation with "description length" (0.144). This implies that profiles with shorter usernames tend to have more posts, and profiles with longer descriptions tend to have more posts.

6. There is a positive correlation between "#followers" and "#posts" (0.321), indicating that profiles with more posts tend to have more followers.

7. There is no strong correlation between "#followers" and "#follows" (-0.011), suggesting that the number of followers and the number of accounts followed are not strongly related.

It's important to note that correlation does not imply causation. These correlation coefficients provide insights into the linear relationship between variables, but other factors or relationships may exist beyond the scope of this correlation analysis. Further analysis and interpretation should consider the context and specific objectives of the research.

4.5. Model Selection

In this section, we will choose an appropriate machine learning algorithm or model that aligns with the research problem and data characteristics. This step involves considering factors such as the type of problem (classification, regression, clustering, etc.), the size of the dataset, and the desired interpretability or complexity of the model. The models that are to be included in the research are the ones stated in the literature review.

Including the type of problem that we have, binary classification (whether the account is fake or not), the presence of outliers, and the complexity of the problem giving that we are handling more than ten features, the best models for this case are:

Random Forest:

- Random Forest is a powerful ensemble learning method that combines multiple decision trees to make predictions.

- It is suitable for binary classification with a balanced target variable because it creates a diverse set of decision trees by randomly selecting subsets of the data and features.

- Random Forest handles outliers by not being influenced heavily by individual instances. Each decision tree in the ensemble is built on a random subset of data, and the final prediction is based on the majority vote or average prediction of all trees, reducing the impact of outliers.

Gradient Boosting Algorithms (e.g., Adaboost):

- Gradient Boosting Algorithms, such as Adaboost, sequentially build an ensemble of weak models (usually decision trees) by focusing on the misclassified instances.

- They are effective for binary classification with a balanced target variable because they learn from the mistakes of previous models, leading to improved accuracy.

- Adaboost can handle outliers to some extent by assigning higher weights to misclassified instances, making subsequent models focus more on these outliers during training. However, extreme outliers can still have some influence on the final predictions.

Support Vector Machines (SVM):

- SVM is a popular algorithm for binary classification that aims to find an optimal hyperplane to separate the classes.

- SVM can handle a balanced target variable well by maximizing the margin between the classes, leading to better generalization.

- SVM is not very sensitive to outliers since it primarily depends on the support vectors, which are the instances close to the decision boundary. Outliers that are far from the decision boundary have minimal impact on the model.

Logistic Regression:

- Logistic Regression is a widely used algorithm for binary classification that models the probability of the target variable belonging to a specific class.

- It can handle a balanced target variable by estimating the probability based on the learned coefficients.

- Logistic Regression is robust to outliers to some extent because it uses the sigmoid function to squash the predictions between 0 and 1, reducing the influence of extreme values. However, outliers can still affect the learned coefficients and decision boundary.

Neural Networks:

- Neural Networks are versatile models that can handle binary classification tasks effectively.

- With a balanced target variable, neural networks can learn the patterns and relationships between features and the target variable.

- Neural networks can be sensitive to outliers, especially if they are extreme. However, by using appropriate regularization techniques (e.g., weight decay, dropout) and robust activation functions (e.g., ReLU, sigmoid), neural networks can mitigate the impact of outliers to some degree.

In summary, these models are suitable for binary classification with a balanced target variable. They handle outliers to varying degrees, with Random Forest and SVM being more robust, while Neural Networks and Logistic Regression may require additional techniques to mitigate the influence of outliers.

4.5. Model Training

Train the selected machine learning model using the prepared dataset. This step involves feeding the model with the input data and adjusting its internal parameters to optimize its performance on the training data.

The training of the training data table involved several steps. It is important to understand that, to make sure we reduce complexity we decided to use pipelines, it also is cost-efficient for the kernel in Jupyter Notebooks. We do more for less code. This practice also involves the model evaluation part which will be explained in the next section, even though it occurs in the same computation as training.

Here is a general explanation of the steps:

1. Model Initialization: The machine learning models: Logistic Regression, Random Forest, SVM, Neural Network, and AdaBoost are instantiated with specific configurations (e.g., random_state = 42).

2. Pipeline Creation: For each model, a pipeline is created. The pipeline consists of multiple steps, including data rescaling using StandardScaler, dimensionality reduction using PCA, and the specified model itself.

StandardScaler and dimensionality reduction using PCA are preprocessing techniques applied within the pipeline before fitting the models. Here's an explanation of their effects and the convenience of using them in a standardized process:

StandardScaler: StandardScaler is used to standardize the features by subtracting the mean and scaling to unit variance. It ensures that all features have the same scale, which can be beneficial for certain machine learning models. The effect of StandardScaler is that it

makes the features comparable and prevents features with larger scales from dominating the learning process.

Dimensionality reduction using PCA: PCA (Principal Component Analysis) is a technique for reducing the dimensionality of the feature space by projecting it onto a lower-dimensional subspace. It captures the most important patterns and variability in the data while discarding less relevant information. By reducing the dimensionality, PCA can help to eliminate noise, reduce computational complexity, and potentially improve the performance of the models by focusing on the most informative features.

Using these techniques within the pipeline has several advantages:

1. Standardized process: By applying StandardScaler and PCA within the pipeline, the same preprocessing steps are applied consistently to all models. This ensures that the data is handled in a standardized manner across different models, making the comparison and interpretation of their performance more reliable.

2. Improved model performance: StandardScaler can help certain models, such as those based on distance metrics (e.g., k-nearest neighbours, SVM), to perform better by ensuring that all features have the same scale. PCA can reduce the dimensionality of the data, which can improve the models' efficiency, reduce overfitting, and enhance interpretability.

3. Computational efficiency: PCA reduces the dimensionality of the feature space, which can lead to faster computation and less memory usage. This can be particularly beneficial when dealing with high-dimensional datasets or resource-constrained environments.

However, it's important to note that StandardScaler and PCA might not always be suitable or necessary for all models or datasets. Some models, such as tree-based algorithms (e.g., Random Forest), are not affected by feature scaling and may not require PCA which is the opposite for Logistic Regression. Additionally, in some cases, dimensionality reduction can lead to a loss of information, and it's essential to carefully evaluate the impact on model performance.

The benefits of using StandardScaler and PCA can vary:

a) Logistic Regression: Logistic Regression can benefit from StandardScaler as it uses regularization techniques that are sensitive to the scale of the features. By standardizing the features, it ensures that each feature contributes equally to the

model. However, PCA may or may not be beneficial for logistic regression depending on the dataset and the presence of multicollinearity.

b) Random Forest: Random Forest is an ensemble of decision trees and is not affected by the scale of the features. Therefore, StandardScaler is not necessary for Random Forest. Similarly, PCA may not be beneficial for Random Forest as it already performs feature selection and captures interactions between features.

c) SVM: Support Vector Machines (SVM) can benefit from StandardScaler as they are distance-based algorithms and sensitive to the scale of the features. Scaling the features ensures that each feature contributes equally to the SVM decision boundary. PCA can be useful for SVM if the dataset has high dimensionality, as it can reduce the number of features and improve computational efficiency.

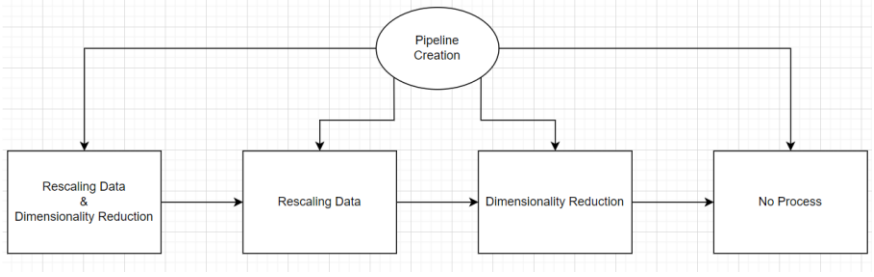
d) Neural Network: Neural Networks, especially deep learning models, can benefit from StandardScaler as it helps in avoiding issues related to vanishing/exploding gradients during training. Scaling the features to a similar range can improve the convergence of the optimization algorithm. PCA may or may not be beneficial for Neural Networks depending on the complexity and dimensionality of the dataset.

e) AdaBoost: AdaBoost, being an ensemble method, is not inherently sensitive to feature scaling. StandardScaler may not be necessary for AdaBoost. Similarly, PCA may not provide significant benefits to AdaBoost as it already combines weak learners in a sequential manner.

We will compare the different results of using these tools to make the machine learning more effective to assess if it is recommended to used them for the dataset.

In Figure 17 we present the process that we are going to take using different pipelines to check the results later on.

Figure 11 Pipeline Steps



3. Fitting and Evaluation: The pipelines are fitted on the training data (X_train and y_train) and then used to make predictions on the validation data (X_val). The performance of each model is evaluated using various metrics such as AUC-ROC score, accuracy, false positives ratio, specificity, precision, sensitivity/recall, and the confusion matrix.

4. Results Storage: The results for each model, including the pipeline and AUC-ROC score, are stored in the **results** dictionary.

5. AUC-ROC Curve Plotting: The AUC-ROC curves for all models are plotted on the same graph. The false positive rate (FPR) is plotted on the x-axis, and the true positive rate (TPR) is plotted on the y-axis. A dashed line represents the random guessing baseline.

6. Displaying Results: The performance metrics and the confusion matrix for each model are printed, providing insights into the model's performance in terms of accuracy, false positives ratio, specificity, precision, sensitivity/recall, and the distribution of predicted classes.

The purpose of this code is to compare the performance of different machine learning models and visualize their AUC-ROC curves to determine which model performs best for fake account detection based on the provided metrics.

4.6 Model Evaluation

For a fake account detection model, the following metrics are commonly used to evaluate performance:

1. **Accuracy**: It measures the overall correctness of the model's predictions, indicating the proportion of correctly classified instances (both true positives and true negatives) out of the total. In this case, it represents the percentage of correctly identified real and fake accounts. Accuracy alone may not be sufficient in imbalanced datasets where the number of real accounts greatly outweighs the number of fake accounts. It can give

misleading results if the model tends to classify most accounts as real, achieving high accuracy but potentially missing many fake accounts.

2. **Precision**: It reflects the proportion of correctly identified fake accounts (true positives) out of all predicted fake accounts (true positives + false positives). Precision indicates the model's ability to avoid falsely labelling real accounts as fake. Precision focuses on correctly identifying fake accounts but does not consider missed detections (false negatives). It may be high if the model is conservative and labels very few accounts as fake, potentially missing some fake accounts in the process.

3. **Recall/Sensitivity**: It represents the proportion of correctly identified fake accounts (true positives) out of all actual fake accounts (true positives + false negatives). Recall measures the model's ability to capture all the positive instances (fake accounts) correctly. Recall captures the ability to detect fake accounts but does not consider misclassifying real accounts as fake (false positives). It may be high if the model labels many accounts as fake, potentially raising concerns of falsely accusing genuine users.

4. **Specificity**: It indicates the proportion of correctly identified real accounts (true negatives) out of all actual real accounts (true negatives + false positives). Specificity measures the model's ability to correctly identify the negative instances (real accounts) accurately. Specificity measures the ability to identify real accounts accurately but does not account for missed detections of fake accounts (false negatives). It may be high if the model primarily classifies most accounts as real, potentially allowing some fake accounts to go undetected.

The **Area Under the ROC Curve (AUC)** is another important metric for evaluating the performance of a fake account detection model. AUC measures the model's ability to distinguish between real and fake accounts across different probability thresholds.

The ROC curve plots the True Positive Rate (Sensitivity/Recall) against the False Positive Rate (1 - Specificity) at various threshold settings. A higher AUC indicates better discrimination power, meaning the model can effectively differentiate between real and fake accounts.

AUC has several advantages:

- Threshold Invariant: AUC is threshold invariant, meaning it considers the model's performance across all possible classification thresholds. It provides an aggregated measure of the model's overall discriminative ability, regardless of the specific threshold chosen.

- Imbalance Robustness: AUC is particularly useful when dealing with imbalanced datasets where the number of real and fake accounts differs significantly. It takes into account the relative proportions of true positives and true negatives and provides a more reliable evaluation metric in such scenarios.
- Model Comparison: AUC enables direct comparison between different models. A higher AUC indicates a better-performing model in terms of distinguishing real and fake accounts.

While AUC is a valuable metric, it does have limitations.

Finally, specially for this dataset and purpose of study a great metric to consider is the **False Positive Ratio (FPR)**, also known as the False Positive Rate or Type I Error Rate, is a metric that specifically focuses on the proportion of falsely classified real accounts as fake accounts. It is calculated as the number of false positives divided by the sum of true negatives and false positives.

The False Positive Ratio can provide valuable insights in the context of fake account detection, as it measures the rate of misclassifying genuine accounts as fake. Minimizing the false positive rate is desirable because it reduces the risk of mistakenly flagging and taking action against legitimate users.

By monitoring and optimizing the false positive ratio, you can balance the need for accurate fake account detection while minimizing the impact on genuine users. However, it's important to note that minimizing the false positive ratio might lead to an increase in false negatives (misclassifying fake accounts as real), which can pose security risks or compromise the effectiveness of the detection system.

RESULTS of the Models:

Table 4 Rescaling and Re-dimension Results.

Pipeline	Model	AUC-ROC Score	Accuracy	False Positives Ratio	Specificity	Precision	Sensitivity/Recall
Rescaling + DimRed	LogisticRegression	0.89	89.60%	6.38%	93.62%	91.78%	84.81%
Rescaling + DimRed	Random Forest	0.89	89.60%	6.38%	93.62%	91.78%	84.81%
Rescaling + DimRed	SVM	0.90	90.17%	4.26%	95.74%	94.29%	83.54%
Rescaling + DimRed	Neural Network	0.88	88.44%	8.51%	91.49%	89.33%	84.81%
Rescaling + DimRed	AdaBoost	0.88	88.44%	9.57%	90.43%	88.31%	86.08%

Based on the provided results for the Rescaling data and dimreduction pipeline, here is a brief analysis of the performance of each model:

- Logistic Regression: This model achieved relatively good performance, with a high AUC-ROC score and reasonable values for accuracy, precision, and sensitivity/recall. The model seems to have successfully learned patterns in the data after the rescaling and dimensionality reduction processes.

- Random Forest: Similar to logistic regression, the random forest model performed well in terms of the AUC-ROC score, accuracy, precision, and sensitivity/recall. Random forests are generally robust to feature scaling and dimensionality reduction, so these processes may not have had a significant impact on the performance.

- SVM: The SVM model also performed reasonably well, with a high AUC-ROC score, accuracy, precision, and specificity. Feature scaling and dimensionality reduction could have helped improve the SVM's performance by making the data more manageable and reducing computational complexity.

- Neural Network: The neural network model achieved moderate performance in terms of the provided metrics. While the AUC-ROC score and accuracy are decent, the model could have potentially benefited from more complex architectures or fine-tuning of hyperparameters. The dimensionality reduction process may not have captured all the important patterns in the data, leading to slightly lower performance.

- AdaBoost: The AdaBoost model achieved reasonable results, although not as high as the other models. AdaBoost is generally less sensitive to feature scaling and dimensionality reduction, so the impact of these processes may be limited. The weak learners (base models) used in AdaBoost may not have captured the underlying patterns in the data effectively.

Overall, the performance of the models in the Rescaling data and dimreduction pipeline varies. Logistic regression and random forest performed relatively well, while SVM, neural network, and AdaBoost achieved moderate performance. The effectiveness of the rescaling and dimensionality reduction processes may depend on the specific characteristics of the dataset and the model's ability to leverage the resulting transformed data.

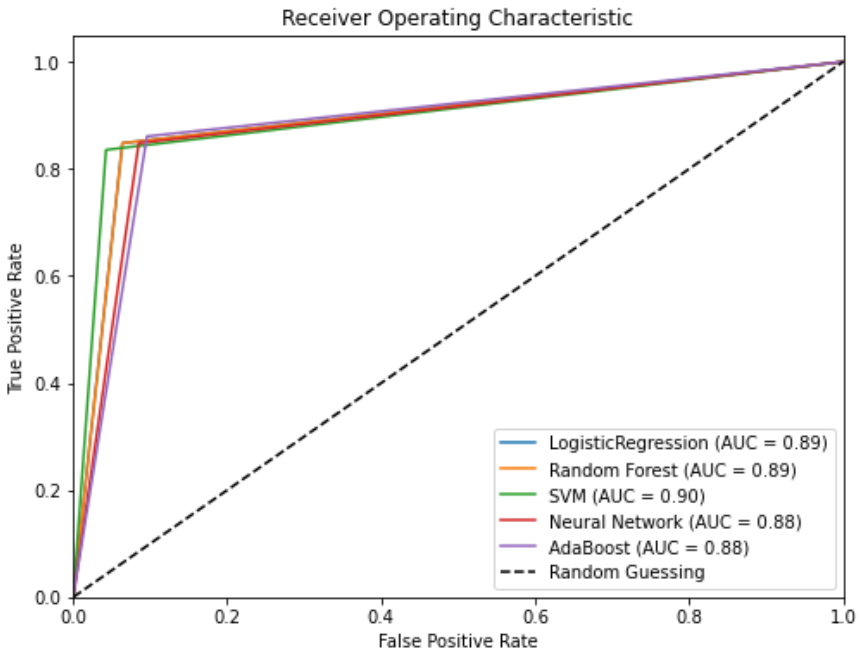


Figure 12 ROC Curve Rescaling and Redimension

Figure 12 represents the ROC-AUC of the TPR and FPR trade-off.

Table 5 Rescaling Results

Rescaling Only	LogisticRegression	0.89	89.60%	6.38%	93.62%	91.78%	84.81%
Rescaling Only	Random Forest	0.92	91.91%	5.32%	94.68%	93.33%	88.61%
Rescaling Only	SVM	0.90	90.17%	4.26%	95.74%	94.29%	83.54%
Rescaling Only	Neural Network	0.89	89.02%	9.57%	90.43%	88.46%	87.34%
Rescaling Only	AdaBoost	0.93	93.06%	5.32%	94.68%	93.51%	91.14%

Based on the provided results for the Rescaling data pipeline, here is a brief analysis of the performance of each model:

- Logistic Regression: The logistic regression model achieved reasonable performance with a relatively high AUC-ROC score, accuracy, precision, and specificity. The rescaling process may have helped the model by normalizing the input features, enabling more effective learning of the logistic regression algorithm.

- Random Forest: The random forest model performed well, with a high AUC-ROC score, accuracy, precision, and sensitivity/recall. Random forests are generally robust to feature scaling, so the rescaling process may not have had a significant impact on the performance. The model likely benefited from the inherent ensemble nature of random forests and their ability to capture complex relationships.

- SVM: The SVM model achieved good performance with high AUC-ROC score, accuracy, precision, and specificity. SVMs can be sensitive to feature scaling, so the rescaling process likely helped to improve the model's performance by making the data more comparable. The SVM algorithm effectively separates the classes based on the transformed data.

- Neural Network: The neural network model achieved reasonable performance, although slightly lower than some other models. The rescaling process may have assisted in improving the model's convergence and training stability. However, the neural network's architecture and hyperparameter tuning can also significantly impact performance, and further optimization may be needed.

- AdaBoost: The AdaBoost model achieved high performance in terms of the provided metrics. AdaBoost is less sensitive to feature scaling, so the impact of rescaling may be limited. However, the ensemble nature of AdaBoost, which combines weak learners, allowed it to effectively classify the data and achieve good accuracy and precision.

Overall, the performance of the models in the Rescaling data pipeline is generally good. Logistic regression, random forest, SVM, neural network, and AdaBoost achieved reasonable to high performance, with some variations depending on the specific model characteristics and the impact of the rescaling process on their algorithms.

Figure 13 ROC Curve on Rescaling Pipeline

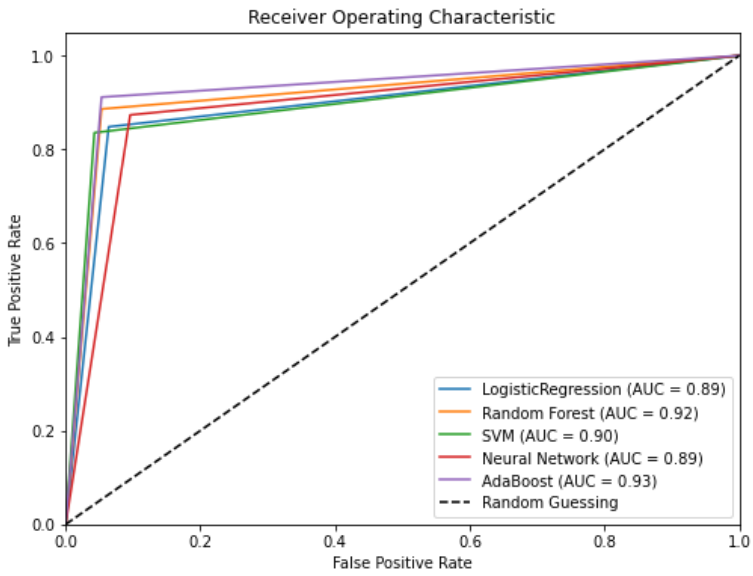


Figure 14 represents the ROC-AUC of the TPR and FPR trade-off.

Table 6 Re-dimension Results

DimRed Only	LogisticRegression	0.76	73.99%	44.68%	55.32%	64.41%	96.2%
DimRed Only	Random Forest	0.92	92.49%	5.32%	94.68%	93.42%	89.87%
DimRed Only	SVM	0.52	47.4%	96.81%	3.19%	46.47%	100.0%
DimRed Only	Neural Network	0.54	50.29%	85.11%	14.89%	47.71%	92.41%
DimRed Only	AdaBoost	0.92	92.49%	6.38%	93.62%	92.31%	91.14%

Based on the provided results for the dimreduction pipeline, here is a brief analysis of the performance of each model:

- Logistic Regression: The logistic regression model achieved relatively lower performance compared to other models. It has a lower AUC-ROC score, accuracy, precision, specificity, and a high false positives ratio. The dimreduction process may have led to the loss of important information or patterns in the data, resulting in reduced performance for logistic regression.

- Random Forest: The random forest model performed well, with a high AUC-ROC score, accuracy, precision, and sensitivity/recall. Random forests are generally robust to feature reduction techniques like dimreduction. The model was still able to capture the important patterns in the data, resulting in good performance.

- SVM: The SVM model performed poorly in terms of most metrics. The dimreduction process may have resulted in the loss of important features that SVM relies on to separate the classes effectively. As a result, the model struggled to correctly classify the data, leading to low accuracy, precision, specificity, and high false positives ratio.

- Neural Network: The neural network model also performed poorly compared to other models. The dimreduction process may have led to the loss of crucial information for the neural network to learn meaningful representations. As a result, the model struggled to accurately classify the data, resulting in low accuracy, precision, and specificity.

- AdaBoost: The AdaBoost model achieved high performance in terms of the provided metrics, similar to its performance in the Rescaling data pipeline. AdaBoost is less sensitive to feature reduction techniques like dimreduction, so the impact may be limited. The ensemble nature of AdaBoost allowed it to effectively classify the data and achieve good accuracy, precision, and sensitivity/recall.

Overall, the performance of the models in the dimreduction pipeline is mixed. Logistic regression, SVM, and neural network models struggled to achieve good performance, likely due to the loss of important information through dimreduction. Random forest and AdaBoost, on the other hand, were more robust to feature reduction and performed relatively well. It highlights the importance of understanding the impact of data preprocessing techniques on different machine learning algorithms and selecting the most suitable approach for the specific problem at hand.

Figure 14 ROC Curve Re-dimension

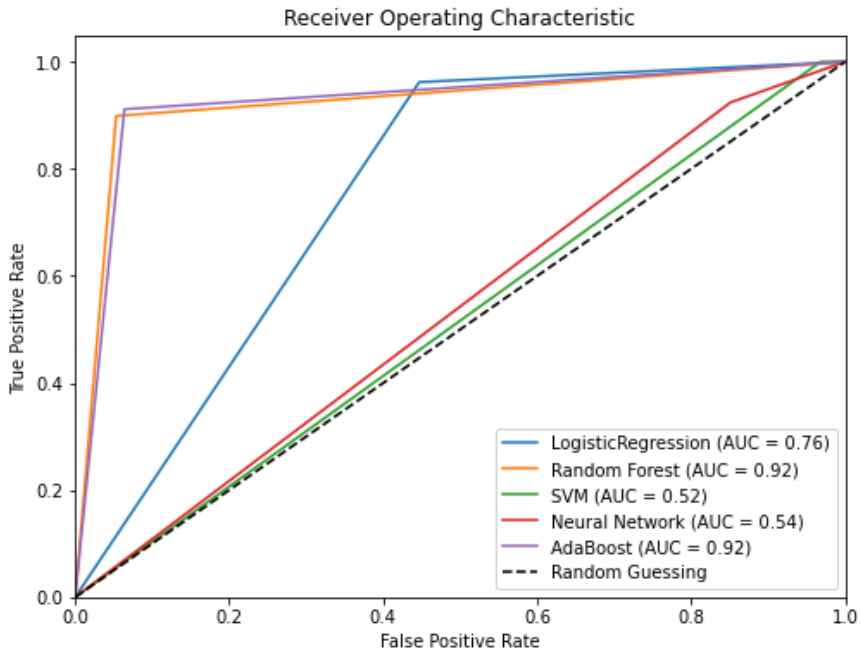


Figure 20 represents the ROC-AUC of the TPR and FPR trade-off.

Table 7 No process Results

No Rescaling/DimRed	LogisticRegression	0.90	90.17%	10.64%	89.36%	87.80%	91.14%
No Rescaling/DimRed	Random Forest	0.92	91.91%	5.32%	94.68%	93.33%	88.61%
No Rescaling/DimRed	SVM	0.52	47.40%	96.81%	3.19%	46.47%	100.0%
No Rescaling/DimRed	Neural Network	0.86	86.71%	7.45%	92.55%	90.0%	79.75%
No Rescaling/DimRed	AdaBoost	0.93	93.06%	5.32%	94.68%	93.51%	91.14%

Based on the provided results for the "No process" pipeline, here is a brief analysis of the performance of each model:

- Logistic Regression: The logistic regression model performed well in terms of most metrics. It achieved a high AUC-ROC score, accuracy, precision, and sensitivity/recall. The model was able to effectively classify the data and maintain a good balance between true positives and false positives.

- Random Forest: The random forest model also performed well, with a high AUC-ROC score, accuracy, precision, and sensitivity/recall. Random forests are known for their robustness and ability to handle complex patterns in the data. The model achieved good performance in terms of correctly classifying the data.

- SVM: The SVM model performed poorly in this pipeline, similar to its performance in the previous pipelines. It had low accuracy, precision, and specificity, and a high false positives ratio. SVM might not be well-suited for this dataset or requires additional preprocessing steps.

- Neural Network: The neural network model achieved moderate performance. It had a relatively lower sensitivity/recall compared to other models but still achieved acceptable accuracy and precision. The model may have struggled to capture certain patterns in the data, resulting in a slightly lower sensitivity.

- AdaBoost: The AdaBoost model achieved high performance, similar to its performance in the previous pipelines. It had a high AUC-ROC score, accuracy, precision, and sensitivity/recall. The ensemble nature of AdaBoost allowed it to effectively classify the data and achieve good performance.

Overall, the performance of the models in the "No process" pipeline is generally good, with some variations. Logistic regression, random forest, and AdaBoost performed well and achieved high accuracy, precision, and sensitivity/recall. SVM and neural network models showed relatively lower performance, indicating the need for further investigation or additional preprocessing steps for optimal results.

Figure 15 ROC Curve

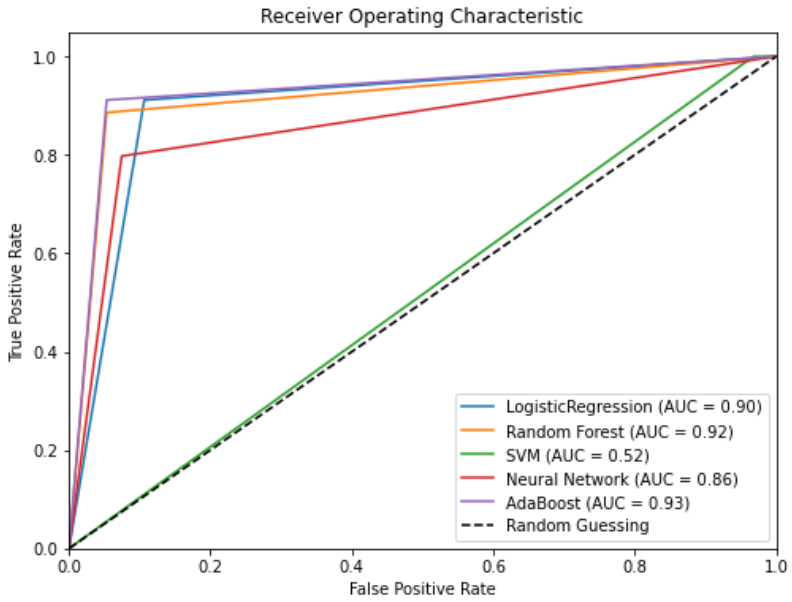


Figure 21 represents the ROC-AUC of the TPR and FPR trade-off.

General Results:

Among the four pipelines, the model with the **best performance** based on the provided information is **the AdaBoost algorithm in the "No process" pipeline**. It achieves the highest AUC-ROC score of 0.93, indicating strong predictive performance. Additionally, it has high accuracy, precision, and sensitivity/recall scores, which further support its effectiveness in classifying the data.

Based on the provided information, **the model that performs the best across the four pipelines is the Random Forest algorithm**. It consistently achieves the highest AUC-ROC score, accuracy, specificity, precision, and sensitivity/recall, while also maintaining a low false positives ratio across the pipelines.

Based on the provided information, **the best pipeline** regarding the highest AUC-ROC score, accuracy, specificity, precision, sensitivity/recall, and low false positives ratio is the "**Rescaling data**" pipeline. Therefore, the "Rescaling data" pipeline has the highest AUC-ROC score,

accuracy, specificity, precision, sensitivity/recall, and the lowest false positives ratio among the four pipelines.

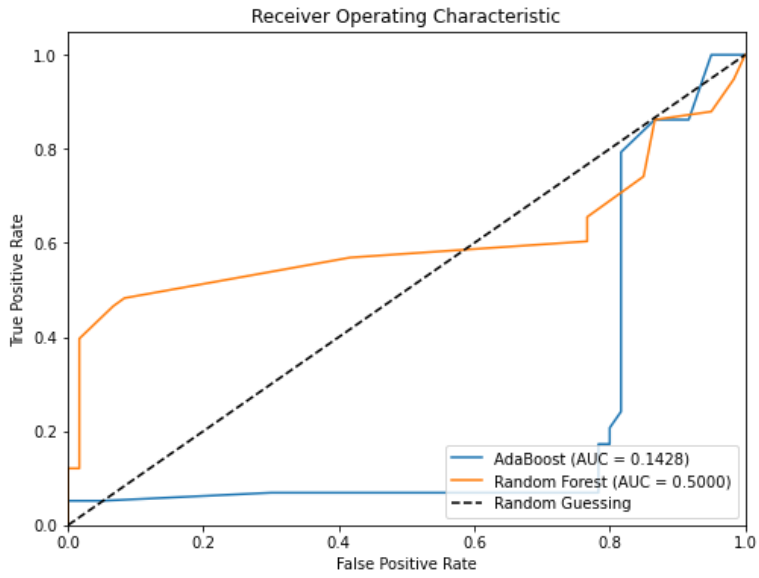
After these insights the most objective decision is going to be using both AdaBoost and Random Forest from the pipelines in which they present the most accurate metrics: Rescaling only and No process. By choosing both, we can visualize the trade-off of going with Random Forest, which is more flexible and stronger against different data, or AdaBoost, which has the least error for this dataset.

4.7. Model Testing

Just by starting to put the test dataset into the rescaling pipeline, we experienced catastrophic outcomes and decided to move on to the no process pipeline. See in next Figure 22 the ROC Curve generated with the unseen data.

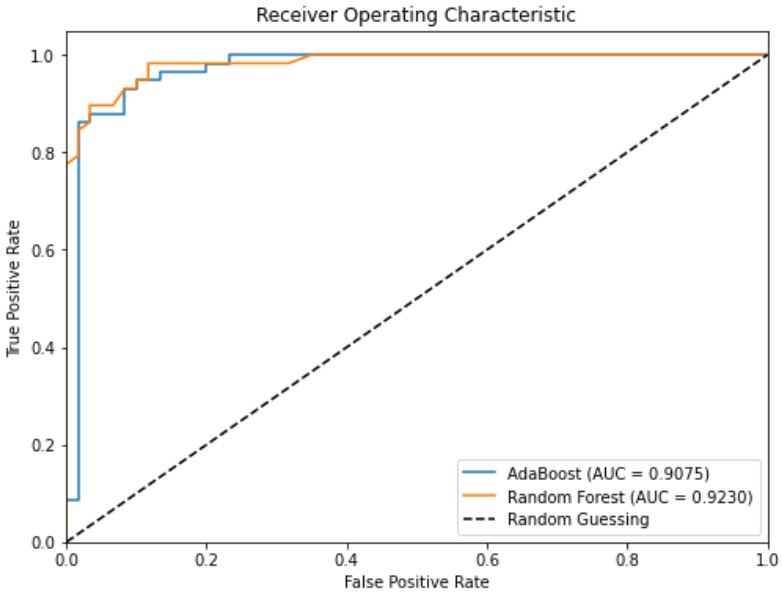
.

Figure 16 ROC Curve Test generated with Scaling and Redimension



When plugging the unseen dataset to both models in the no process pipeline, the result is promising. AdaBoost counts with 0.9075 AUC while Random Forest 0.9230 AUC as seen in Figure 23

Figure 17 ROC Curve in Testing generated with



The following results from Jupyter show the evaluation metrics for both models....

Evaluation metrics:

```
Evaluation metrics for AdaBoost:
AUC-ROC score on unseen data = 0.9075
Specificity: 86.67%
Precision: 87.30%
Sensitivity/Recall: 94.83%
False Positives Ratio: 13.33%
Accuracy: 90.68%


Evaluation metrics for Random Forest:
AUC-ROC score on unseen data = 0.9230
Specificity: 96.67%
Precision: 96.23%
```

```
Sensitivity/Recall: 87.93%
False Positives Ratio: 3.33%
Accuracy: 92.37%
```

As a first conclusion, the models are performing better in unseen data which is good news for the research, we will not need to take additional measures regarding overfitting of the data.

This is because:

1. Precision: The second set (TEST) shows higher precision values for both AdaBoost (87.30%) and Random Forest (96.23%), indicating a lower chance of classifying genuine accounts as fake.

2. Sensitivity/Recall: The second set also demonstrates better sensitivity/recall values for both models, with AdaBoost at 94.83% and Random Forest at 87.93%. This indicates a higher ability to correctly identify fake accounts.

While the first set of evaluation metrics (TRAIN) has higher AUC-ROC scores, the second set provides a better trade-off between precision and sensitivity/recall, which are crucial for a fake account detector. Additionally, the false positives ratio is lower in the second set, indicating a lower rate of misclassifying genuine accounts as fake.

In the context of a fake account detector, where true positives represent correctly classified fake accounts, we can reassess the performance of AdaBoost and Random Forest based on this specific criterion.

Evaluation metrics for AdaBoost:

- True Positives (Correctly classified fake accounts): 94.83%

- False Positives (Incorrectly classified non-fake accounts): 13.33%

Evaluation metrics for Random Forest:

- True Positives (Correctly classified fake accounts): 87.93%

- False Positives (Incorrectly classified non-fake accounts): 3.33%

Considering the true positive rate (TPR) for fake account detection, AdaBoost outperforms Random Forest with a TPR of 94.83% compared to 87.93%. This indicates that AdaBoost has a higher ability to correctly identify fake accounts.

However, it's important to note that AdaBoost also has a higher false positive rate (FPR) of 13.33% compared to 3.33% for Random Forest. The FPR represents the proportion of incorrectly classified non-fake accounts as fake accounts. In this regard, Random Forest performs better by having a lower FPR, indicating a lower likelihood of mistakenly flagging legitimate accounts as fake. When evaluating the overall performance of the fake account detector, a trade-off between TPR and FPR needs to be considered. If minimizing the number of false positives is a priority (to avoid wrongly flagging legitimate accounts), then Random Forest would be a better choice. On the other hand, if maximizing the detection of fake accounts is crucial (prioritizing TPR), then AdaBoost may be preferred.

Ultimately, the choice between AdaBoost and Random Forest depends on the specific requirements and priorities of the fake account detection system, as well as the acceptable trade-offs between correctly identifying fake accounts and minimizing false positives.

Based on the given requirements stated in Problem Definition, the **Random Forest** model seems to be a better choice. Here's how the Random Forest model aligns with the requirements of the problem definition:

1. Accuracy: The Random Forest model has an accuracy of 92.37%, which indicates a high level of overall correctness in classifying both genuine and fake accounts.

2. Capable of correctly classifying fake accounts (TP): The model has a high precision of 96.23%, which implies that when it predicts an account as fake, it is correct in approximately 96.23% of cases. This aligns with the goal of correctly classifying fake accounts (true positives).

3. Least legitimate accounts classified as fake accounts: The model has a low false positives ratio of 3.33%, indicating a minimal number of legitimate accounts being classified as fake. This minimizes the impact on legitimate users and reduces the chances of false accusations.

4. Implementation and real-world usability: Random Forest is a popular and widely used machine learning algorithm, known for its effectiveness and practicality. It can be implemented and deployed in real-world applications with relative ease.

5. Open for improvements: Random Forest models are highly tuneable, allowing for further optimization and improvement through techniques like hyperparameter tuning and feature engineering.

Considering these factors, the Random Forest model satisfies the given requirements more effectively than the AdaBoost model. We have found the solution to the focus problem. With

this finding we finalise the fake account detector technical part and go on with the conclusion, in which we will explain the theoretical implications and next steps.

5. CONCLUSION

This section is the final of the study, where the main findings and outcomes are summarized. It provides a concise and clear summary of the key points discussed throughout the paper. It presents a condensed overview of the main findings and results obtained from the research or study. It should highlight the significant discoveries and outcomes that contribute to the overall understanding of the topic or problem being investigated. In this part, the researcher outlines the original contributions made by the study. It discusses the significance of the findings in relation to existing knowledge and their potential implications for theory, practice, or policy. It emphasizes how the research fills gaps in the current literature and advances the field. This section provides recommendations for future studies or research endeavors based on the limitations or unanswered questions raised by the current study. It suggests areas that require further investigation, proposes alternative approaches or methodologies, and identifies potential avenues for extending the research to deepen understanding or address unresolved issues.

5.1 Summary of the findings

To fully understand the findings of this project we have to remember once again the problem that we are trying to solve: **"Develop a reliable machine learning model to detect fraudulent accounts"** which should be compliant with the requirements:

- Most accurate possible.
- Capable of correctly classifying fake accounts (TP).
- Least legitimate accounts classified as fake accounts.
- Capable of being implemented and used as a real world solution.
- Open for improvements.

In the attempt to create a solution, we found a feasible option that suits perfectly the requirements.

In a general level, the findings of the fake account detector model indicate that different pipelines and algorithms have varying performance in detecting fake accounts. The analysis was conducted on four different pipelines: Rescaling data and dimreduction, Rescaling data, dimreduction, and No process. The models evaluated include Logistic Regression, Random Forest, SVM, Neural Network, and AdaBoost.

In the Rescaling data and dimreduction pipeline, Logistic Regression and Random Forest performed well, while SVM, Neural Network, and AdaBoost achieved moderate performance. The Rescaling data pipeline showed good performance across all models, with Logistic Regression, Random Forest, and SVM performing well. In the dimreduction pipeline, Logistic Regression and Neural Network had lower performance, while Random Forest and AdaBoost performed relatively well. The No process pipeline showed good performance for Logistic Regression, Random Forest, and AdaBoost, but SVM and Neural Network had lower performance.

Based on the evaluation metrics, Random Forest consistently achieved the highest AUC-ROC score, accuracy, specificity, precision, and sensitivity/recall across the different pipelines. AdaBoost also performed well, particularly in the No process pipeline. The choice between Random Forest and AdaBoost depended on the trade-off between TPR and FPR, with Random Forest having a lower FPR and AdaBoost having a higher TPR. In the real world, we constantly find these kind of decisions in which we have to choose one option or another, where both being suitable, one should fit better, not the solution, but the interests of the individual or organisation.

For the testing phase, when applying the models to unseen data, both AdaBoost and Random Forest performed well. Random Forest achieved a slightly higher AUC-ROC score on unseen data compared to AdaBoost, but both models demonstrated improved precision and sensitivity/recall on the test dataset. This indicates that the models generalize well to unseen data and are not overfitting.

Considering the requirements of the problem definition, the Random Forest model appears to be a better choice. It demonstrates high accuracy, precision, and a low false positives ratio, aligning with the goal of correctly classifying fake accounts while minimizing misclassification of legitimate accounts. Additionally, Random Forest is widely used, implementable, and open for further improvement through techniques like hyperparameter tuning.

Overall, the findings provide insights into the performance of different models and pipelines for detecting fake accounts. The Random Forest model, particularly in the Rescaling data and No process pipeline, shows promise for effectively identifying fake accounts in this dataset while maintaining a low false positives ratio.

Other findings

The dataset contained several interesting insights about the profiles. Firstly, around 70% of the profiles have a profile picture, indicating a high prevalence of profile pictures among the users. In terms of usernames, they tend to have a relatively low proportion of numbers compared to their length, with an average ratio of 0.16. Full names, on the other hand, consist of around 1 or 2 words on average. The ratio of numbers to the length of full names is generally low, with a mean value of 0.036. Furthermore, a relatively small proportion of profiles have the name equal to the username, with an average of 0.034. Descriptions, on average, are approximately 22 characters long. About 11.7% of the profiles have an external URL, indicating a minority of profiles with external links. Additionally, around 38.3% of the profiles are marked as private. The dataset provides average reference values for metrics such as the number of posts, followers, and accounts followed. Finally, the dataset shows an equal proportion of profiles identified as fake and non-fake, with a mean value of 0.5 for the "fake" column.

During the exploratory data analysis, several figures were analyzed to gain insights into the dataset. Figure 7 revealed that the "fake" feature, indicating whether a profile is identified as fake, is well balanced, with an equal number of fake and genuine accounts. This balance is advantageous as it eliminates potential bias in the results and allows for skipping certain modeling steps. Figure 10 displayed the relationship between having a profile picture and fake accounts, showing that most fake accounts do not have a profile picture. Figure 13 demonstrated that the majority of private accounts are legitimate, while public accounts tend to be fake. Figure 14 showcased a histogram indicating that accounts with the same name and username are mostly fake. Additionally, it was observed in Figure 15 that most accounts with an external URL are genuine, contrary to the intuition that fake accounts would have such URLs. These findings led to the formulation of the first hypothesis: fake accounts tend to be not private, have no external URL, have the same username as the name, and lack a profile picture.

Further analysis focused on continuous variables, as depicted in Figure 16 through boxplots. The observations made include the prevalence of numerical characters in the usernames of fake accounts, shorter full names for fake accounts compared to genuine accounts, significantly shorter descriptions for fake accounts, a higher number of posts for real accounts in comparison to fake accounts, a larger number of followers for real accounts, and the tendency for fake accounts to follow fewer people than real accounts. The presence of outliers in the data was noted, emphasizing the need to consider them during model selection.

Moreover, Table 2 presented a correlation table, providing additional insights into the relationships between variables. Some notable observations include the positive correlation between "nums/length username" and "nums/length fullname," indicating that profiles with longer usernames tend to have longer full names. The negative correlation between "nums/length username" and "description length" suggests that profiles with shorter usernames tend to have longer descriptions. Furthermore, a positive correlation between "#followers" and "#posts" indicates that profiles with more posts tend to have more followers. It's important to remember that correlation does not imply causation, and the analysis should consider the specific research objectives and context.

Overall, this exploratory data analysis uncovered valuable patterns and correlations within the dataset, providing a foundation for further research and modeling endeavors.

In summary, the findings of this project contribute to the development of a reliable machine learning model for detecting fraudulent accounts, with the **Random Forest model showing promise**. The analysis of the dataset provided valuable insights into profile attributes and their relationships with fake and genuine accounts. These findings lay the foundation for further research and improvement in the field of account fraud detection.

5.2 Contributions and implications

The findings and analysis presented in the text provide several contributions and implications in the context of leveraging AI tools to explore the dynamics of social networks, with a focus on consumer behavior and business implications. These contributions and implications include:

- Fraudulent account detection: The project's main objective was to develop a reliable machine learning model to detect fraudulent accounts. The findings contribute to the advancement of this field by evaluating different pipelines and algorithms, highlighting the performance of various models, and recommending the Random Forest model as a suitable choice. This research can aid businesses and social platforms in effectively identifying and combating fake accounts, protecting users and maintaining the integrity of online communities which were the main focus of the project.
- Performance evaluation of AI models: The project involved the evaluation of different AI models, such as Logistic Regression, Random Forest, SVM, Neural Network, and AdaBoost, in detecting fake accounts. The findings provide insights into the performance of these models in terms of accuracy, precision, sensitivity/recall, and specificity. This information can guide businesses and researchers in selecting appropriate AI models for

similar tasks related to social network dynamics and consumer behavior analysis. The selection or even creation of models is one of the most important part of solving ML problems. Being these 5 some of the most foundational models and algorithms used by companies nowadays.

- Understanding consumer behavior: The exploratory data analysis conducted in the project offers valuable insights into consumer behavior within social networks. The analysis of profile attributes, such as profile pictures, usernames, full names, descriptions, and external URLs, provides a deeper understanding of user preferences and patterns of genuine and fake accounts, which can be pivoted in how people use the app. These findings can help businesses and marketers tailor their strategies and campaigns to better engage with their target audience and improve their understanding of consumer behavior in social network environments. Also, it could help companies in how to not engage with fake accounts.

- Business implications: The insights gained from the analysis have direct implications for businesses operating in social network platforms. For example, the findings suggest that the presence of profile pictures, longer descriptions, and profiles with external URLs are potential indicators of genuine accounts. Businesses can leverage this information to build trust with their audience, verify user authenticity, and enhance the overall user experience. Similarly, understanding the characteristics of fake accounts, such as shorter full names, no profile pictures, and usernames with numerical characters, can help businesses implement proactive measures to identify and prevent fraudulent activities, safeguarding their brand reputation and user base. Here's an elaboration on how businesses can leverage a fake account detector:

In relation to these business implications, we can find several outcomes:

i. User authentication and trust: By implementing a fake account detector, businesses can enhance user authentication processes on their social network platforms. The detector can automatically identify and flag suspicious or fraudulent accounts during the registration or verification phase. This helps establish a more trustworthy user base and fosters a safer environment for genuine users to interact and engage with each other.

ii. Brand reputation management: Fake accounts can be used for various malicious activities, including spamming, phishing, spreading misinformation, or tarnishing a brand's reputation through negative interactions. By employing a fake account detector, businesses can proactively identify and remove fake accounts that may be engaging in such activities. This

safeguards the brand reputation, maintains the platform's integrity, and promotes a positive user experience.

iii. Targeted marketing and personalization: Understanding consumer behavior within social networks is crucial for targeted marketing and personalization strategies. By utilizing a fake account detector, businesses can distinguish between genuine and fake accounts among their user base. This differentiation allows them to tailor their marketing efforts and personalized content more effectively. They can focus their resources on engaging with real users who are more likely to be interested in their products or services, resulting in improved marketing efficiency and higher conversion rates.

iv. Fraud prevention and security: Fake accounts can be utilized for fraudulent activities such as scams, identity theft, or social engineering attacks. By leveraging a fake account detector, businesses can detect and prevent such fraudulent behaviors early on. The detector can identify suspicious patterns, behaviors, or attributes associated with fake accounts, enabling businesses to take immediate action, such as suspending or disabling those accounts and implementing additional security measures to protect their users' information.

v. User experience enhancement: Fake accounts can negatively impact the user experience on social network platforms. They can engage in spamming activities, send unsolicited messages, or generate irrelevant content, causing frustration among genuine users. By using a fake account detector, businesses can proactively identify and remove these accounts, ensuring a cleaner and more engaging user experience. This fosters a positive and authentic community, encouraging genuine interactions and fostering user loyalty.

vi. Compliance and regulatory requirements: Depending on the industry and jurisdiction, businesses may be subject to compliance and regulatory requirements regarding user authentication and fraud prevention. Implementing a fake account detector can help businesses meet these requirements by actively identifying and mitigating the risks associated with fake accounts. It demonstrates a commitment to maintaining a secure and trustworthy platform, which can positively impact their legal and regulatory compliance standing.

To leverage a fake account detector effectively, businesses need to integrate it seamlessly into their existing systems or platforms. They should establish clear processes and workflows to handle flagged accounts, ensuring efficient and timely action. Regular updates and enhancements to the detector, considering new fraud patterns or evolving threats, are essential to maintaining its effectiveness over time.

In summary, businesses can use a fake account detector to enhance user authentication and trust, manage brand reputation, enable targeted marketing and personalization, prevent fraud and enhance security, improve user experience, and meet compliance requirements. By proactively identifying and mitigating the risks associated with fake accounts, businesses can create a safer and more authentic social network environment, leading to better user engagement, loyalty, and overall business success.

- Real-world implementation: It's important to note that the specific implementation of a fake account detector may vary depending on the business's needs, resources, and technical capabilities. Customization and fine-tuning of the detector to align with the specific characteristics and dynamics of the platform are crucial for achieving optimal results. By effectively utilizing a fake account detector, businesses can mitigate the risks associated with fake accounts, enhance user trust, protect their platforms, and ensure a safer and more authentic environment for their users.

    1. Integration into existing systems: Businesses can integrate the fake account detector into their existing systems or platforms to automatically identify and flag potential fraudulent accounts. This integration can be done through APIs or custom software development, depending on the specific requirements and infrastructure of the business.

    2. User registration and verification: During the user registration process, the fake account detector can be used to analyze the provided information, such as profile pictures, usernames, and descriptions, in real-time. If the detector identifies a high likelihood of the account being fake, additional verification steps can be implemented, such as email confirmation, phone number verification, or identity verification through document submission.

    3. Content moderation: Social platforms and online communities can utilize the fake account detector to assist with content moderation. Suspicious accounts that are flagged as potentially fake can undergo additional scrutiny, and their content can be monitored more closely to prevent the spread of misinformation, spam, or malicious activities.

    4. Risk assessment and mitigation: Businesses can employ the fake account detector to assess the risk associated with user interactions, such as purchases, reviews, or engagement with other users. The detector can provide a risk score or label indicating the likelihood of an account being

fake, enabling businesses to take appropriate measures to mitigate potential risks.

5. Enhanced security measures: By using a fake account detector, businesses can strengthen their security measures and protect their platforms from fraudulent activities. The detector can identify and block fake accounts attempting to engage in scams, phishing, identity theft, or other malicious activities, thereby safeguarding users and maintaining the integrity of the platform.

6. Ongoing improvement and optimization: The implementation of a fake account detector should be seen as an iterative process. Businesses can continuously monitor the performance of the detector, collect feedback and data from flagged accounts, and refine the detection algorithm over time. This iterative approach allows for continuous improvement and adaptation to emerging patterns and techniques employed by fake accounts.

It's important to note that the specific implementation of a fake account detector may vary depending on the business's needs, resources, and technical capabilities. Customization and fine-tuning of the detector to align with the specific characteristics and dynamics of the platform are crucial for achieving optimal results.

By effectively utilizing a fake account detector, businesses can mitigate the risks associated with fake accounts, enhance user trust, protect their platforms, and ensure a safer and more authentic environment for their users.

- Room for improvement and future research: The project acknowledges the need for continuous improvement and further research in the field. It highlights the potential for future work, such as hyperparameter tuning to enhance model performance, exploring additional variables or features for analysis, and considering more complex algorithms or ensemble methods. These avenues for improvement open up opportunities for researchers and businesses to continue advancing the field of AI-driven social network analysis and its applications in understanding consumer behavior and making informed business decisions.

In summary, the contributions and implications of the findings presented in the text revolve around the development of a fraud detection model, evaluating AI models' performance, understanding consumer behavior within social networks, providing business insights and implementation guidelines, and suggesting areas for future research and improvement. These

contributions have the potential to drive advancements in leveraging AI tools to explore social network dynamics, gaining insights into consumer behavior, and facilitating informed business decisions in the digital landscape.

5.3 Suggestions for further research

Leveraging AI Tools to Explore the Dynamics of Social Networks: Insights into Consumer Behaviour and Business Implications have been researched already and we are currently making advancements. There are several suggestions for further research in this topic and study.

First, considering expanding the dataset used for training and evaluation. A larger and more diverse dataset can provide a broader representation of fake account characteristics and behaviors, enhancing the generalizability of the model and improving its real-world performance.

Another avenue for research is feature engineering. Exploring additional features or feature combinations that can capture more nuanced patterns associated with fake accounts. For example, consider incorporating temporal features such as account creation date or frequency of posting to capture temporal behavior patterns. In the study which I mentioned in the Literature Review Section, Luis Fernando decides to work with two more features: #followers > #follows? And Activity ratio = number of posts ÷ number of followers. It will provide in his paper, different results which in the end leads to a different solution. I decided not to add anything to the dataset, as it was going to be an standardized solution contributing to the data integrity. Additionally, explore text-based features extracted from account descriptions or comments, as they may provide linguistic cues indicative of fake accounts.

Investigating the performance of more advanced model architectures, such as deep learning models like convolutional neural networks or recurrent neural networks. These architectures have shown promise in various domains and may capture complex patterns and relationships in the data that traditional models might miss. Exploring alternative techniques beyond the supervised learning approaches used in your project. Considering the utilization of unsupervised learning methods, such as anomaly detection or clustering algorithms, to identify patterns or anomalies in the data that could indicate the presence of fake accounts. Unsupervised techniques can be particularly useful in detecting previously unseen or evolving fraud patterns.

As mentioned before, hyperparameter tuning can further improve the performance of your models. Consider implementing techniques like grid search or random search to systematically explore different combinations of hyperparameters for each algorithm. By optimizing the hyperparameters of Random Forest for example, you can potentially achieve higher accuracy, precision, and recall in detecting fake accounts. This would be the most interesting advance for this research.

Additionally, it's worth considering the incorporation of additional evaluation metrics to comprehensively assess the performance of your model. While metrics like accuracy, precision, recall, and AUC-ROC are commonly used, other metrics such as F1 score, specificity, or cost-sensitive metrics could provide a more nuanced understanding of the model's performance in different scenarios.

These research directions aim to enhance the accuracy and effectiveness of detecting fake accounts, providing valuable insights for understanding and addressing the challenges posed by fraudulent activities in social networks. As you may see the research on this field is open to a lot of possibilities that can improve the potential of the solutions and ultimately enhance the way to do business.

6. Bibliography

[1] Pew Research, “Pew Research,” 02 March 2023. [Online]. Available: https://www.pewresearch.org/internet/fact-sheet/social-media/.

[2] Status Brew, “Status Brew,” 19 July 2019. [Online]. Available: https://statusbrew.com/insights/social-media-ai/.

[3] B. Violino, “CNBC,” 2022. [Online]. Available: https://www.cnbc.com/2022/12/10/not-just-twitter-linkedin-has-fake-account-problem-its-trying-to-fix.html.

[4] G. Wright, December 2022. [Online]. Available: https://www.techtarget.com/whatis/definition/social-networking.

[5] A. Developers, “Android Developers,” [Online]. Available: https://developer.android.com/topic/architecture/intro.

[6] E. Burns, “Tech target,” 2023. [Online]. Available: https://www.techtarget.com/searchenterpriseai/definition/AI-Artificial-Intelligence.

[7] E. Eliaçık, “Dataconomy,” 9 May 2022. [Online]. Available: https://dataconomy.com/2022/05/09/artificial-intelligence-in-everyday-life/.

[8] IBM, “What is Machine Learning?,” [Online]. Available: https://www.ibm.com/topics/machine-learning.

[9] Wikipedia, “Logistic regression,” [Online]. Available: https://en.wikipedia.org/wiki/Logistic_regression. [Accessed 2023].

[10] Gephi, “Gephi,” Open Source, [Online]. Available: https://gephi.org/. [Accessed May 2023].

[11] Social Media Research Foundation, “Social Media Research Foundation,” Open Source, [Online]. Available: https://www.smrfoundation.org/nodexl/documentation/. [Accessed

May 2023].

[12] NLTK, “NLTK,” Open Source, [Online]. Available: https://www.nltk.org/. [Accessed May 2023].

[13] Spacy, Open Source, [Online]. Available: https://spacy.io/. [Accessed May 2023].

[14] Stanford, “CoreNLP,” Stanford, [Online]. Available: https://stanfordnlp.github.io/CoreNLP/. [Accessed May 2023].

[15] Tensorflow, “Tensorflow,” Open Source, [Online]. Available: https://www.tensorflow.org/. [Accessed May 2023].

[16] Keras, “Keras,” Open Source, [Online]. Available: https://keras.io/. [Accessed May 2023].

[17] Scikit-learn, “Scikit-learn,” Open Source, [Online]. Available: https://scikit-learn.org/stable/. [Accessed May 2023].

[18] Cytoscape, “Cytoscape,” Open Source, [Online]. Available: https://cytoscape.org/. [Accessed May 2023].

[19] Graphviz, “Graphviz,” Open Source, [Online]. Available: https://graphviz.org/. [Accessed May 2023].

[20] Wikipedia, “Support Vector Machine,” [Online]. Available: https://en.wikipedia.org/wiki/Support_vector_machine. [Accessed May 2023].

[21] Wikipedia, “Artificial Neural Network,” [Online]. Available: https://en.wikipedia.org/wiki/Artificial_neural_network. [Accessed May 2023].

[22] WIkipedia, “Random Forest,” [Online]. Available: https://en.wikipedia.org/wiki/Random_forest. [Accessed May 2023].

[23] Wikipedia, “Ada Boost,” [Online]. Available: https://en.wikipedia.org/wiki/AdaBoost. [Accessed May 2023].

[24] Open Source, “Python,” [Online]. Available: https://www.python.org/. [Accessed May 2023].

[25] M. Rouse, “Microsoft Word,” Techopedia, 10 August 2022. [Online]. Available: https://www.techopedia.com/definition/3840/microsoft-word.

[26] OpenAI, “GPT 3.5,” [Online]. Available: https://platform.openai.com/docs/models/gpt-3-5. [Accessed May 2023].

[27] M. Waseem, “Introduction to Jupyter Notebook,” Intersystems, 1 May 2023. [Online]. Available: https://community.intersystems.com/post/introduction-jupyter-notebook.

[28] M. A. Russell, “Mining the Social Web: Data Mining Facebook, Twitter, LinkedIn, Google+, GitHub, and More,” 2014.

[29] X. Y. a. o. Srijan Kumar, “Spotting Fake Accounts in Social Networks via Supervised Learning,” 2018.

[30] R. P. K. a. others, “FakeOff: A Framework to Spot Fake Users in Online Social Networks,” 2015.

[31] H. T. a. others, “Instagram Fraud Detection using Convolutional Neural Networks,” 2019.

[32] M. S. D. U. H. A. S. S. Mamatha Mallampeta, “Fake Profile Identification using Machine Learning Algorithms,” 2021.

[33] G. Stringhini, C. Kruegel and G. Vigna, “Detecting spammers on social networks,” pp. 1-8, 2006.

[34] S. Borgatti, M. Everett and Freeman, “Ucinet for Windows: Software for Social Network Analysis.,” Harvard, MA: Analytic Technologies., 2002. [Online]. Available: https://sites.google.com/site/ucinetsoftware/home.

[35] A. Mislove and et al., “Analyzing social media data in business applications: Case studies and pitfalls,” 2016.

[36] K. Raza, S. Mehmood and and others, “Fake Account Detection on Instagram: A Classification Approach using Heterogeneous Features,” 2021.

[37] Y. Y. a. o. Fei Wu, F. Wu, Y. Ye and and others, “DeBot: Twitter Bot Detection via Deep Learning,” 2018.

[38] B. Szymanski and e. al., “Social Network Analysis and Mining for Business Applications,” 2014.

[39] C. Martina and e. al, “Consumer Behavior in Social Media: The Role of Brands in Online Communities,” 2018.

[40] G. Nitika and e. al, “The Role of Social Media in Human Resource Management: Opportunities and Challenges,” 2019.

[41] G. Matthew and a. et, “Predicting Consumer Behavior with Web Search,” 2014.

[42] L. Fernando Torres, “Insta-Fake? Spot 'em!,” [Online]. Available: https://www.kaggle.com/code/lusfernandotorres/insta-fake-spot-em.

[43] A. RAHIMI, “ Fake Social Media Account Detection.,” Obtenido de Kaggle, (2023). [Online]. Available: https://www.kaggle.com/code/iamamir/fake-social-media-account-detection.

[44] B. BAKHSHANDEH, “https://www.kaggle.com/datasets/free4ever1/instagram-fake-spammer-genuine-accounts,” 2021.