Minor Project-II Report
On

"INSTAGRAM FAKE ID DETECTION"

Submitted in partial fulfillment of
The requirements for the 6th Semester Sessional Examination
of

BACHELOR OF TECHNOLOGY
IN
COMPUTER SCIENCE & ENGINEERING
By

NAMES
AMARNATH PANDA
SANCHITA SABAT
CHANDAN DANDIA
Registration No.
21UG010149
21UG010568
21UG010264

Under the able Supervision of
Ms. SANDHYARANI BISWAL
DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

GIET UNIVERSITY, GUNUPUR

2023 - 24

GIET UNIVERSITY,GUNUPUR

Dist. - Rayagada, Odisha-765022, Contact:- +91 7735745535,
06857-250170,172, Visit us:- **www.giet.edu**

Department of Computer Science & Engineering

CERTIFICATE

This is to certify that the project work entitled "INSTAGRAM FAKE ID DETECTION" is done by Name-AMARNATH PANDA , SANCHITA SABAT, CHANDAN DANDIA Regd. No.-21UG010149, 21UG010568, 21UG010264 in partial fulfillment of the requirements for the 6th Semester Sessional Examination of Bachelor of Technology in Computer Science and Engineering during the academic year 2023-24. This work is submitted to the department as a part of evaluation of 6th Semester Minor Project-II.

Project Supervisor *Class Teacher*

Project Coordinator, 3rd Year *HoD, CSE*

ACKNOWLEDGEMENT

We express our sincere gratitude to Ms. Sandhyarani Biswal of Computer science and engineering for giving us an opportunity to accomplish the project. Without his/her active support and guidance, this project report has not been successfully completed.

We also thank to our class teacher Mr. Pritam Raul for his/her constant support during the execution of our project.

We also thank Mr.Bhavani Sankar Panda, Project Coordinator, Dr. Sachikant Dash, Head of the Department of Computer Science and Engineering, Prof. (Dr.) Kakita Murali Gopal, Dy. Dean, Computational Science, SOET for their consistent support, guidance and help.

We also thanks to our friends, family members and others for their unconditional support during the project execution.

Amarnath Panda(21UG010149)
Sanchita Sabat (21UG010568)
Chandan Dandia(21UG010264)

ABSTRACT

The popularity of social media continues to grow, and its dominance of the entire world has become one of the aspects of modern life that cannot be ignored. The rapid growth of social media has resulted in the emergence of ecosystem problems. Hate speech, fraud, fake news, and a slew of other issues are becoming un-stoppable. With over 1.7 billion fake accounts on social media, the losses have al-ready been significant, and removing these accounts will take a long time. Due to the growing number of Instagram users, the need for identifying fake accounts on social media, specifically in Instagram, is increasing.

This process takes a long time if done manually by humans, we can now use machine learning to identify fake accounts thanks to the rapid development of machine learning. We can detect fake accounts on Instagram using machine learning by implementing the combination of image detection and natural language processing.

Fake ID detection algorithms may occasionally flag legitimate accounts as fake, leading to account suspensions or other penalties for innocent users. This can result in frustration and inconvenience for affected individuals.

By integrating these and potentially other techniques, Instagram could develop a robust system for detecting fake IDs on its platform, helping to maintain the integrity and trustworthiness of its user base. However, it's important to note that any such system would need to balance effectiveness with user privacy and transparency concerns.

Chapter: 1

INTRODUCTION

1.1 Purpose:

The primary goal of this project is to combat the proliferation of fake Instagram accounts, which can lead to various forms of abuse, including identity theft, spreading misinformation, and manipulation of user interactions. By implementing a robust detection system, the aim is to enhance user trust and confidence in the platform. Mitigating the proliferation of fake Instagram accounts, which not only jeopardizes user privacy and security but also undermines the credibility of the platform as a whole. Enhancing user confidence and trust in the authenticity of interactions and content on Instagram, fostering a safer and more enjoyable user experience.

1.2 Project Scope:

This project encompasses a comprehensive analysis of user behaviour, profile attributes, and other relevant factors to develop algorithms capable of identifying suspicious accounts accurately. It also includes the implementation of features to strengthen user authentication and profile verification processes. Implementing a multi-layered approach to account verification and authentication, including identity verification measures, two-factor authentication, and real-time monitoring of account activity. Collaborating with Instagram's existing security and moderation systems to integrate the proposed solution seamlessly and augment the platform's overall security posture.

1.3 Product Features:

In addition to the core functionality of fake account detection, the system will offer features such as two-factor authentication, real-time monitoring of user activity, and customizable alert mechanisms to notify users of potential security threats. Employing machine learning algorithms to analyse user behaviour patterns, engagement metrics, and content interactions to identify suspicious activity indicative of fake accounts.

Chapter: 2

WORK DONE IN THE RELATED AREA

2. 1 Research Papers on Social Media Account Verification Methods:

- Extensive literature review encompassing a wide range of methodologies and techniques employed for verifying the authenticity of social media accounts.
- Analysis of the strengths and limitations of various verification methods, including email verification, phone number verification, and identity document verification.
- Exploration of emerging trends such as biometric authentication and blockchain-based identity verification systems.

2.2 Machine Learning Studies for Fraud Detection in Online Platforms:

- Investigation into the application of machine learning algorithms, including supervised, unsupervised, and semi-supervised learning techniques, for detecting fraudulent activities on online platforms.
- Evaluation of feature engineering approaches, anomaly detection algorithms, and ensemble learning methods for improving detection accuracy and reducing false positives.

2.3 Existing Tools and Software Solutions for Detecting Fake Accounts:

- Comparative analysis of existing tools and software solutions designed for detecting fake accounts on social media platforms, such as Instagram, Facebook, and Twitter.
- Examination of features, performance metrics, and user feedback to assess the effectiveness and usability of these tools in real-world scenarios.
- Identification of gaps and areas for improvement in existing solutions, informing the design and development of novel detection mechanisms.

2.4 Case Studies Illustrating the Impact of Fake Accounts on Social Media:

- In-depth examination of real-world case studies showcasing the detrimental effects of fake accounts on user trust, platform integrity, and societal well-being.
- Analysis of high-profile incidents involving fake accounts, such as coordinated misinformation campaigns, identity thefts, and social engineering attacks.
- Quantitative and qualitative assessment of the economic, social, and psychological implications of fake accounts on individuals and communities.

2.5 User Behaviour Analysis Specific to Instagram:

- Empirical study of user behaviour patterns specific to Instagram, including posting frequency, content preferences, engagement metrics, and interaction dynamics.
- Identification of anomalies and deviations from normal behaviour that may indicate fraudulent or malicious activity, such as sudden spikes in follower count or abnormal engagement rates.
- Development of models and algorithms for detecting anomalous behaviour and distinguishing between legitimate and fake accounts based on behavioural cues.

2.6 Regulatory and Legal Frameworks for Combatting Fake Accounts:

- Review of existing regulatory frameworks and legal provisions governing the detection and prevention of fake accounts on social media platforms.
- Analysis of legal challenges and jurisdictional issues associated with combating fake accounts across different geographical regions and legal systems.
- Exploration of potential collaborations between government agencies, regulatory bodies, and technology companies to establish effective mechanisms for addressing the proliferation of fake accounts.

2.7 Ethical Considerations in Fake Account Detection and Moderation:

- Examination of ethical dilemmas and considerations inherent in the detection and moderation of fake accounts on social media platforms.
- Discussion of issues such as privacy infringement, algorithmic bias, and freedom of speech in the context of fake account detection.
- Development of ethical guidelines and best practices for balancing the need for user protection with respect for individual rights and freedoms.

Chapter: 3
SYSTEM ANALYSIS

3.1 User Requirements (SRS):

- Detailed specifications gathered through stakeholder interviews, surveys, and user feedback sessions.
- User personas representing various roles and their respective needs within the system.
- Functional requirements delineating the desired features and interactions for each user role.
- Non-functional requirements addressing performance, scalability, security, and usability aspects of the system.
- Use case scenarios illustrating how users will interact with the system to achieve their goals.
- Requirement prioritization based on business impact and user needs.
- Validation criteria to ensure that user requirements are met and validated during system testing.

3.2 Hardware Requirements:

- Detailed hardware specifications for server infrastructure, including CPU, RAM, storage capacity, and network bandwidth.
- Redundancy and fault-tolerance configurations to ensure high availability and data integrity.
- Scalability considerations to accommodate potential increases in user traffic and data volume.
- Environmental requirements for hosting the server infrastructure, including power, cooling, and physical security measures.
- Compliance with industry standards and best practices for data center management and hardware procurement.

3.3 Software Requirements:

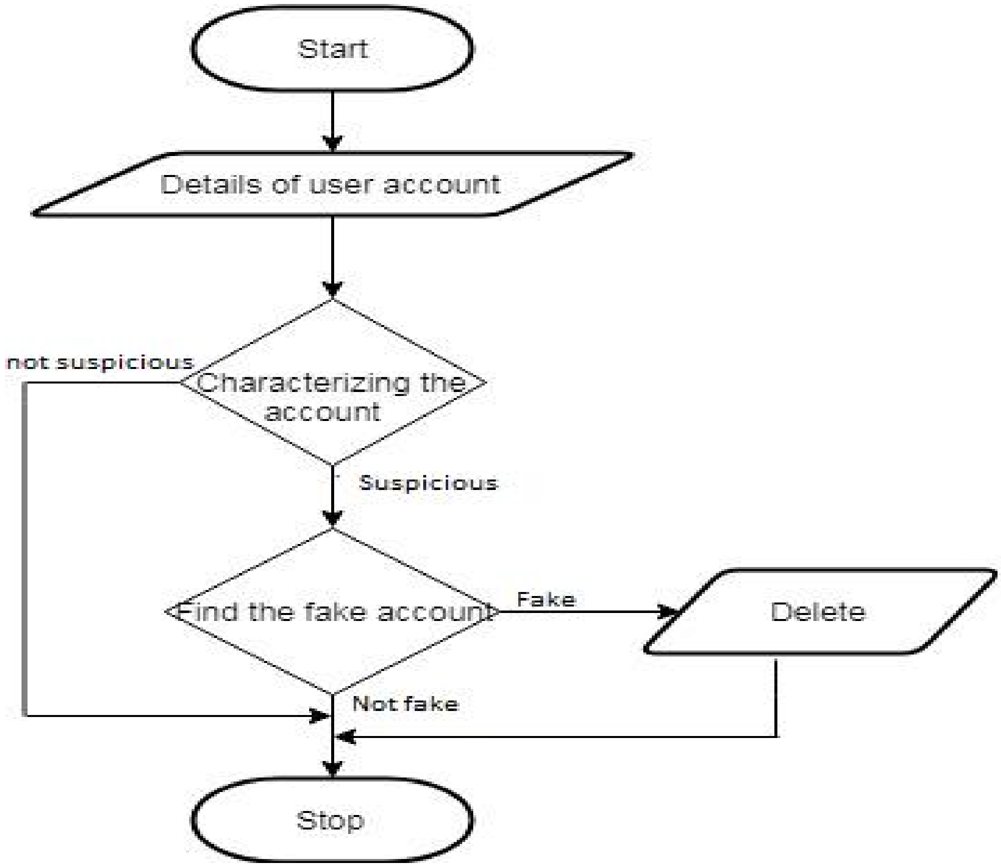
- Selection of programming languages, frameworks, and libraries based on project requirements, team expertise, and industry best practices.
- Database management system (DBMS) selection considering factors such as data volume, performance, and scalability requirements.
- Integration with third-party APIs and services for implementing specific functionality, such as identity verification and fraud detection.
- Compatibility requirements with operating systems, web browsers, and mobile devices to ensure broad accessibility.
- Security measures, including encryption protocols, access controls, and intrusion detection/prevention systems.
- Compliance with regulatory requirements, such as GDPR, HIPAA, and COPPA, governing data privacy and security.

Chapter:4

SYSTEM DESIGN AND SPECIFICATIONS

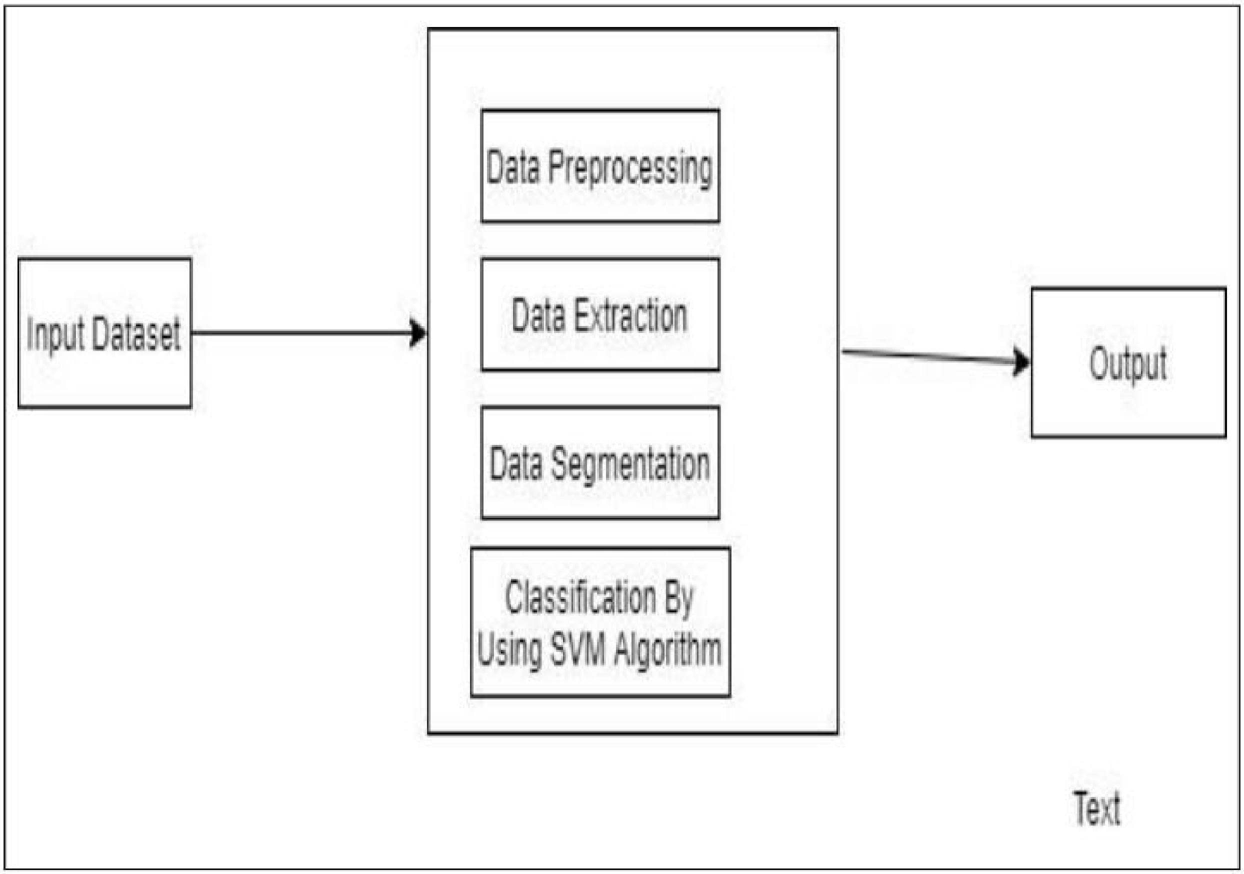
4.1 High Level Design (HLD)

- **4.1.1 Structure Chart:**
  - The structure chart provides a visual representation of the system's architecture, illustrating the modular decomposition of the system into distinct components and their interrelationships.
  - Each module is represented as a box, with arrows indicating the flow of data and control between modules.
  - Modules may be organized hierarchically, with higher-level modules representing major system functionalities and lower-level modules representing specific tasks or sub-functions.
  - The structure chart helps in understanding the overall organization of the system and facilitates communication among team members during the design and development phases.



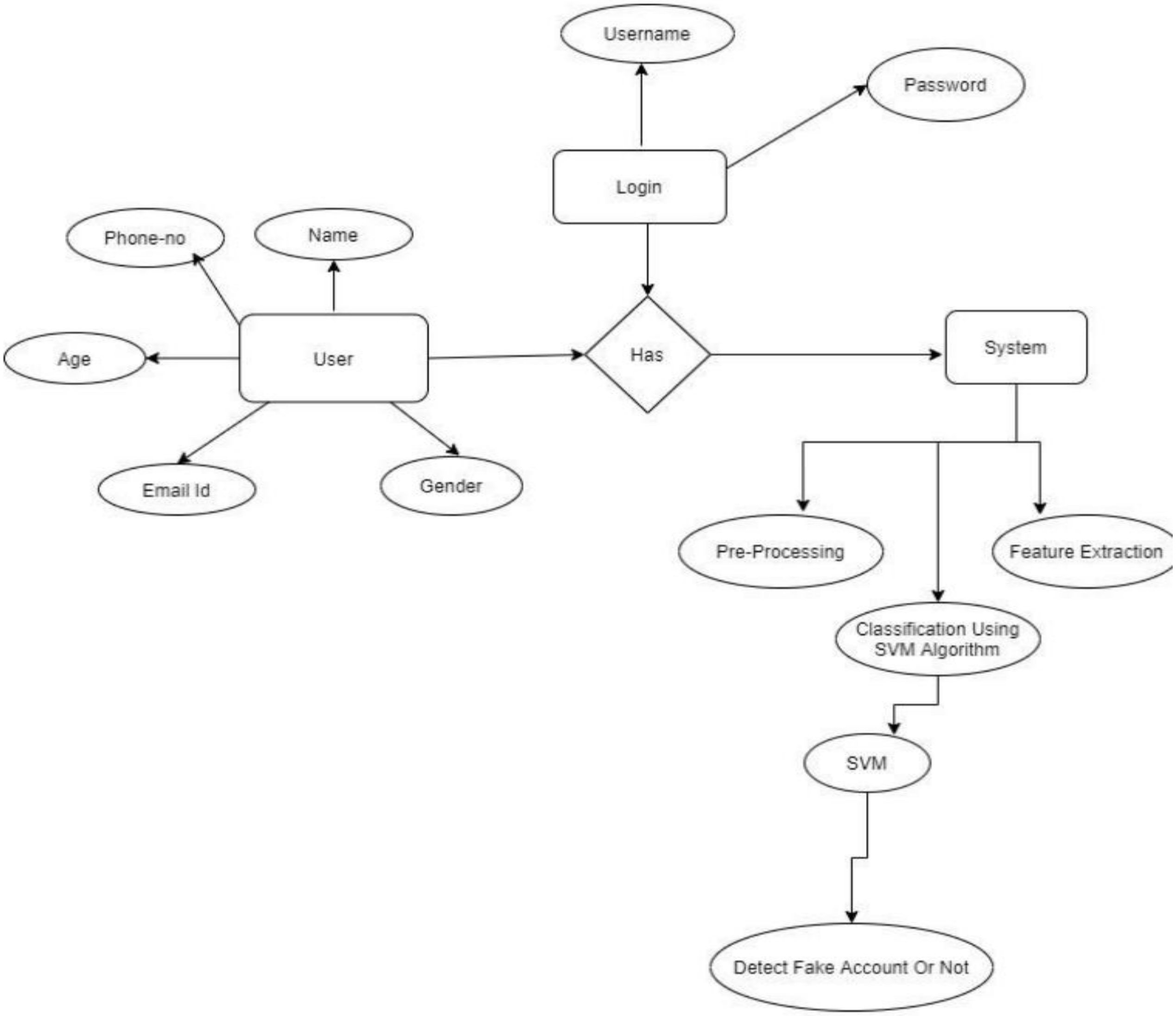
- **4.1.2 Data Flow Diagram (DFD):**

  - The DFD is a graphical representation of the flow of data within the system, depicting the movement of data from external sources through processes to storage and output destinations.
  - It consists of four main components: external entities, processes, data stores, and data flows.
  - External entities represent sources or destinations of data, such as users or external systems.
  - Processes represent transformations or operations performed on the data, such as data validation or calculation.
  - Data stores represent repositories where data is stored for later retrieval or processing.
  - Data flows represent the movement of data between components of the system, indicating the direction and nature of data flow.



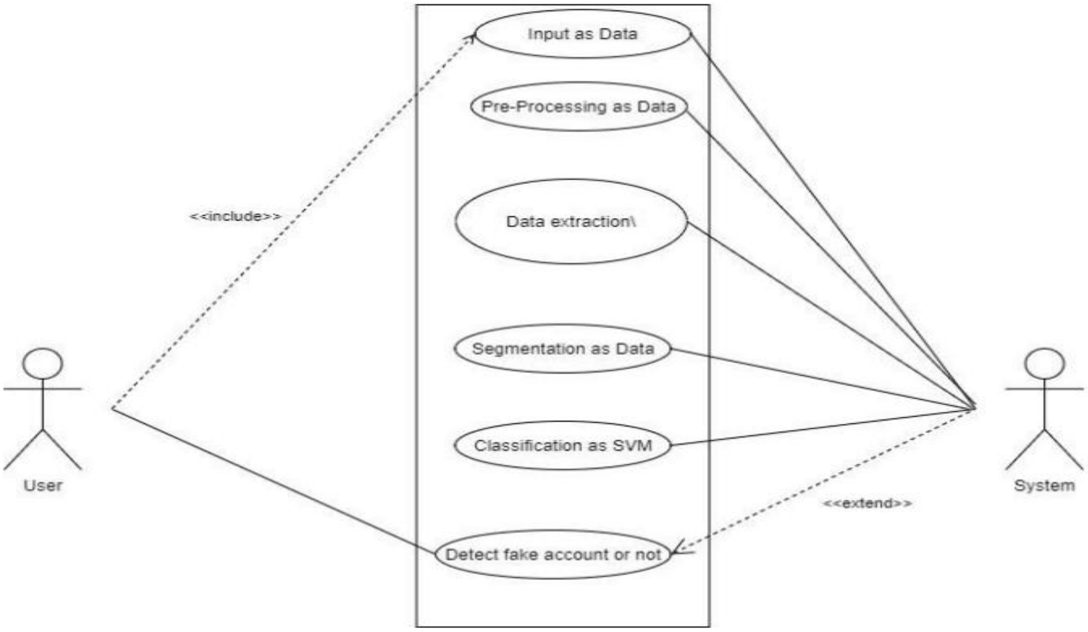
➢ 4.1.3 Entity-Relationship (E-R) Diagram:

- The E-R diagram is a data modelling technique used to represent the logical structure of the system's database.
- It consists of entities, which represent real-world objects or concepts, and relationships, which describe the associations between entities.
- Attributes are used to describe the properties or characteristics of entities, while keys are used to uniquely identify instances of entities.
- The E-R diagram provides a visual overview of the system's data schema, helping in the identification of entities, relationships, and constraints that need to be modeled in the database.



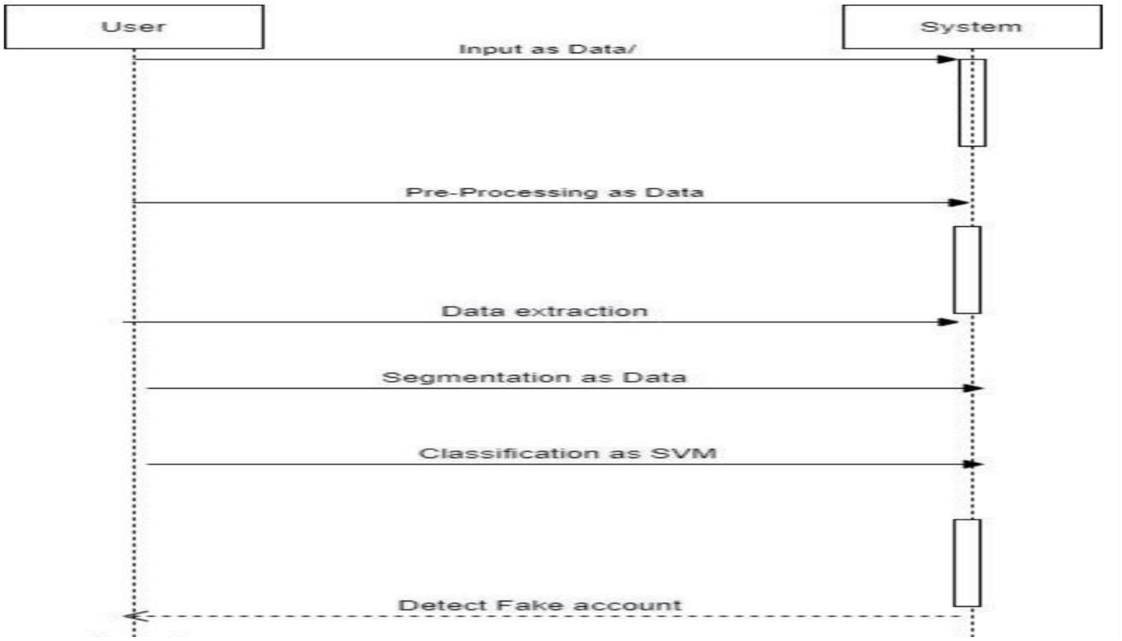
4.1.4 UML Diagrams:

- **Use Case Diagram:** A use case diagram depicts the interactions between users (actors) and the system, illustrating the various ways in which users can interact with the system to achieve specific goals or tasks.

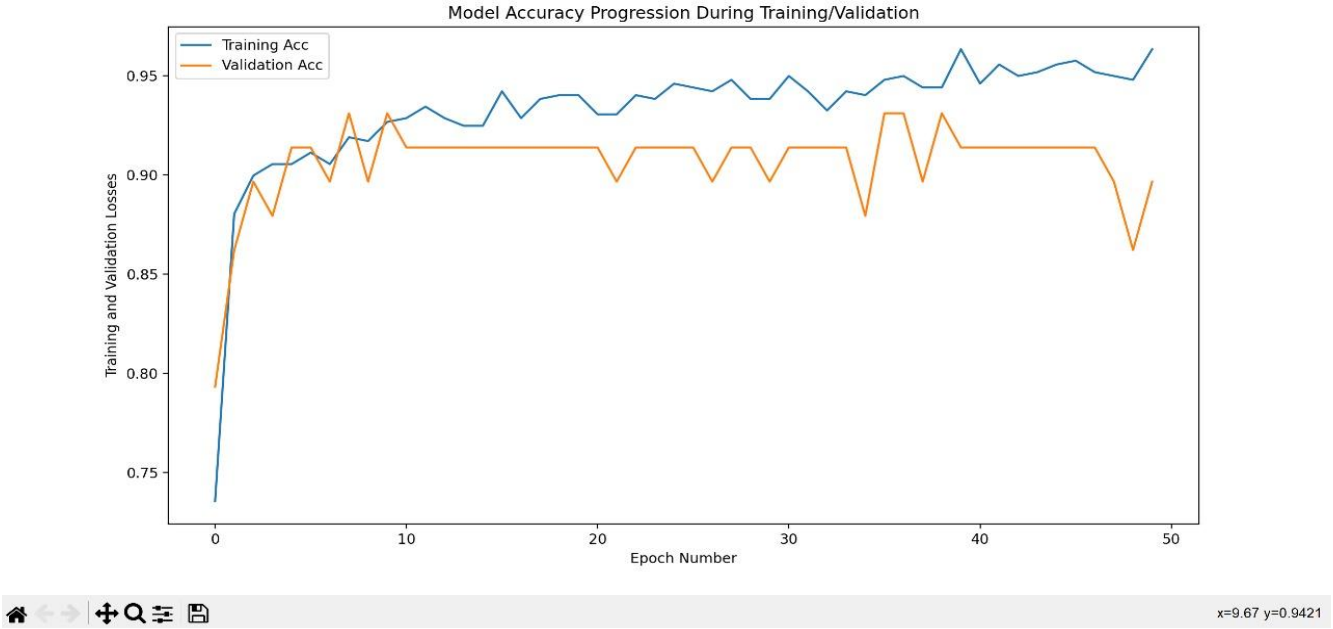


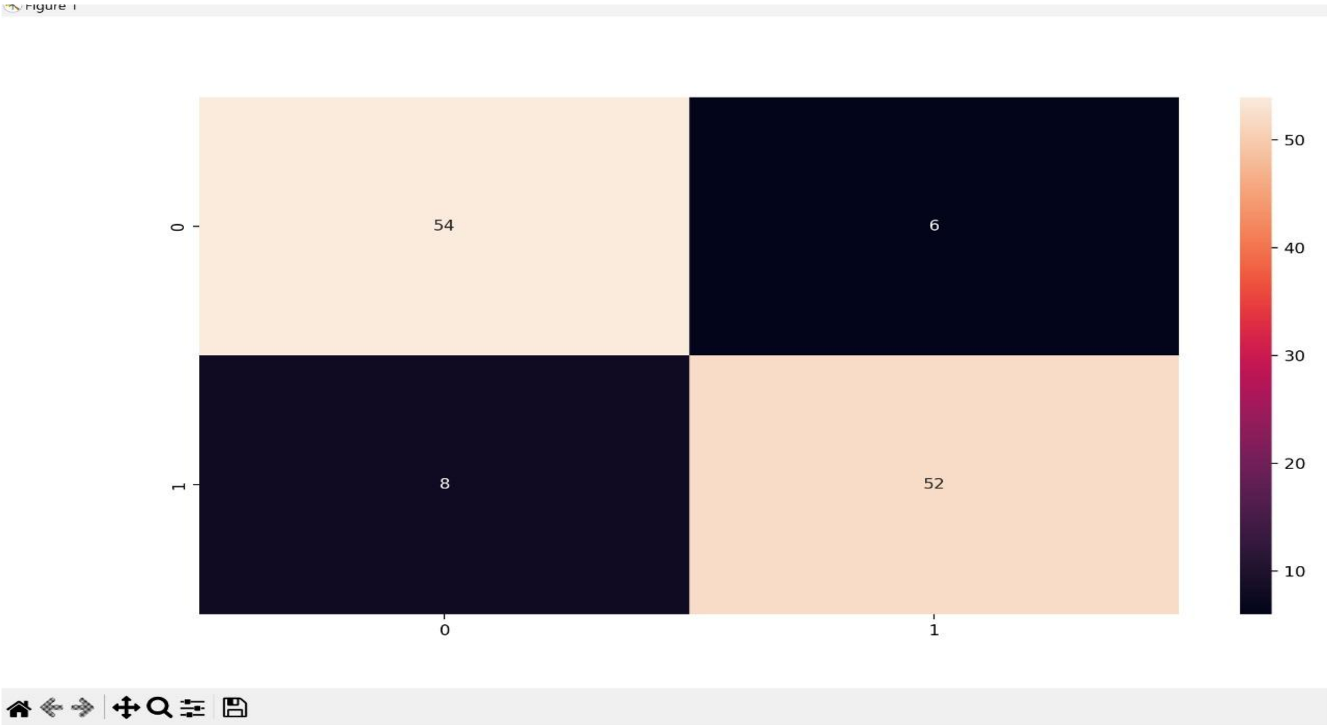
- **Class/Object Diagram:** A class/object diagram represents the static structure of the system, showing classes or objects, their attributes, and relationships between them.

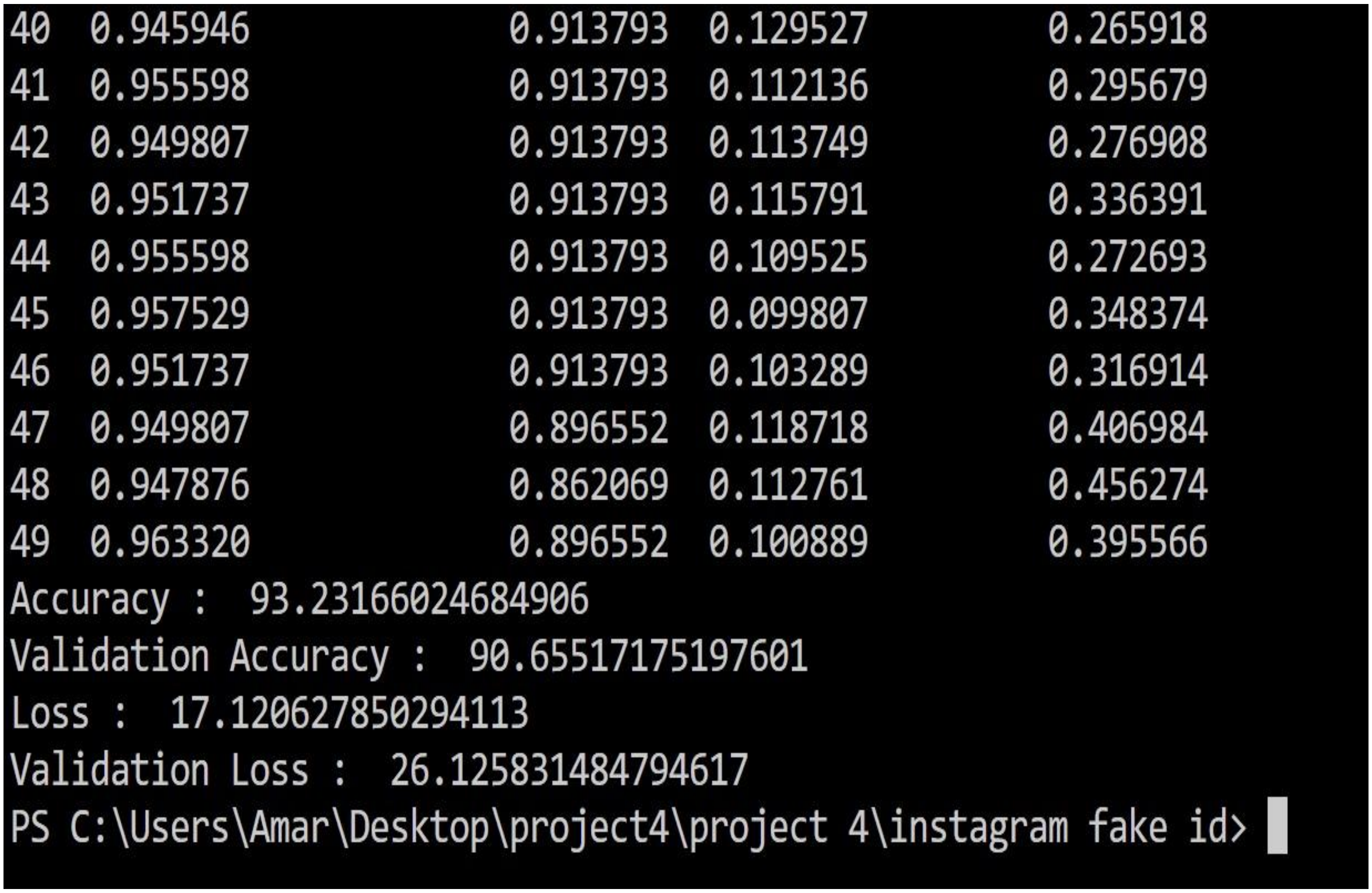
4.2 Low Level Design (LLD)



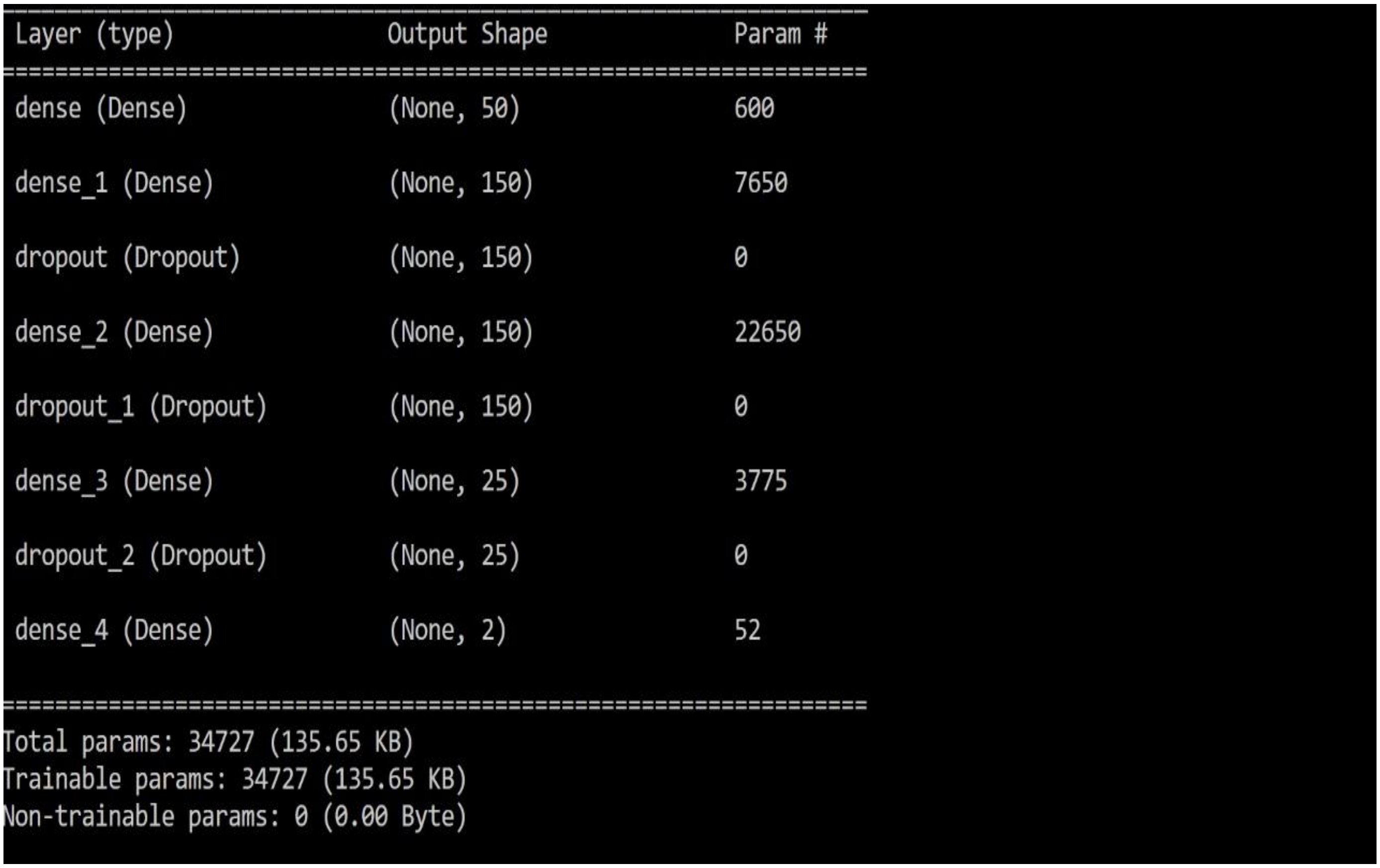
4.2.1 Screen Shot Diagram:



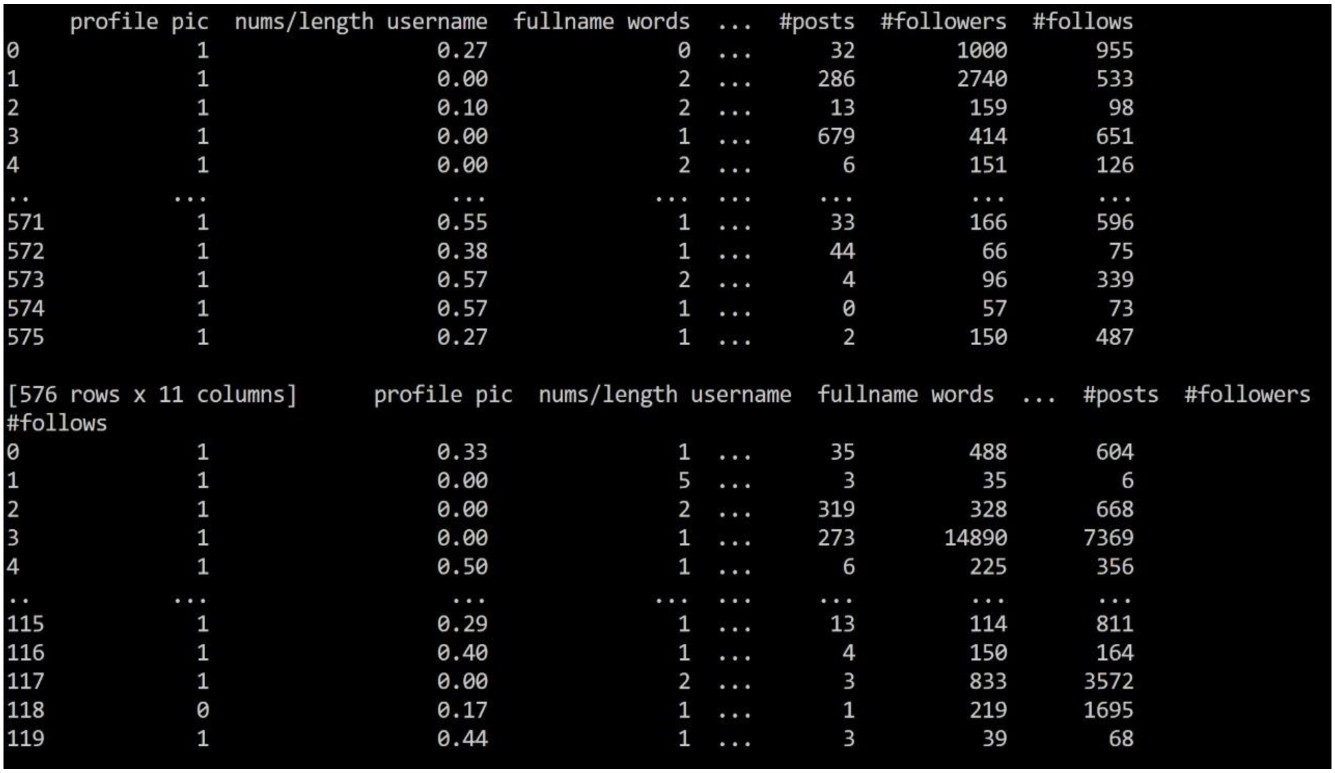




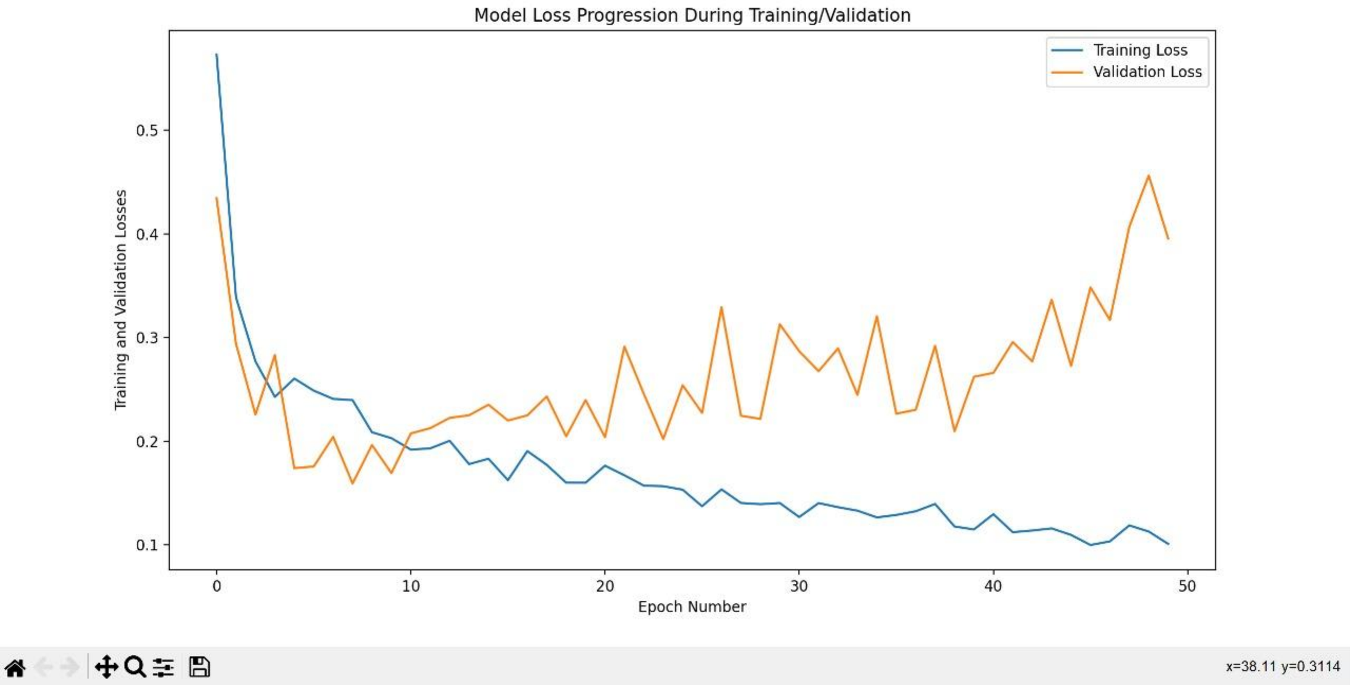
```
40  0.945946              0.913793  0.129527          0.265918
41  0.955598              0.913793  0.112136          0.295679
42  0.949807              0.913793  0.113749          0.276908
43  0.951737              0.913793  0.115791          0.336391
44  0.955598              0.913793  0.109525          0.272693
45  0.957529              0.913793  0.099807          0.348374
46  0.951737              0.913793  0.103289          0.316914
47  0.949807              0.896552  0.118718          0.406984
48  0.947876              0.862069  0.112761          0.456274
49  0.963320              0.896552  0.100889          0.395566
Accuracy :  93.23166024684906
Validation Accuracy :  90.65517175197601
Loss :  17.120627850294113
Validation Loss :  26.125831484794617
PS C:\Users\Amar\Desktop\project4\project 4\instagram fake id>
```

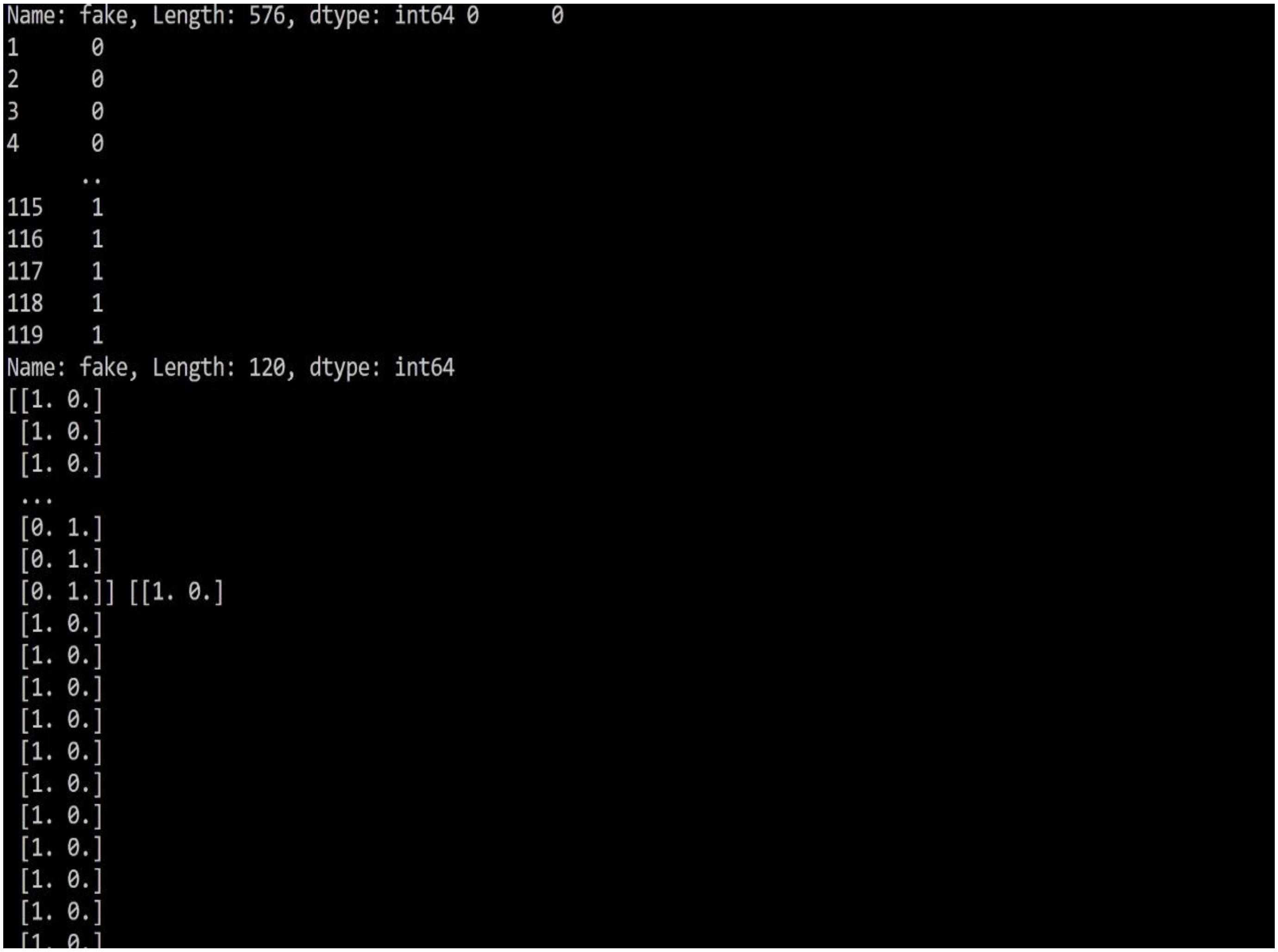


```
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense (Dense)               (None, 50)                600

 dense_1 (Dense)             (None, 150)               7650

 dropout (Dropout)           (None, 150)               0

 dense_2 (Dense)             (None, 150)               22650

 dropout_1 (Dropout)         (None, 150)               0

 dense_3 (Dense)             (None, 25)                3775

 dropout_2 (Dropout)         (None, 25)                0

 dense_4 (Dense)             (None, 2)                 52

=================================================================
Total params: 34727 (135.65 KB)
Trainable params: 34727 (135.65 KB)
Non-trainable params: 0 (0.00 Byte)
```



```
     profile pic  nums/length username  fullname words  ...  #posts  #followers  #follows
0              1                  0.27               0  ...      32        1000       955
1              1                  0.00               2  ...     286        2740       533
2              1                  0.10               2  ...      13         159        98
3              1                  0.00               1  ...     679         414       651
4              1                  0.00               2  ...       6         151       126
..           ...                   ...             ...  ...     ...         ...       ...
571            1                  0.55               1  ...      33         166       596
572            1                  0.38               1  ...      44          66        75
573            1                  0.57               2  ...       4          96       339
574            1                  0.57               1  ...       0          57        73
575            1                  0.27               1  ...       2         150       487

[576 rows x 11 columns]      profile pic  nums/length username  fullname words  ...  #posts  #followers
#follows
0              1                  0.33               1  ...      35         488       604
1              1                  0.00               5  ...       3          35         6
2              1                  0.00               2  ...     319         328       668
3              1                  0.00               1  ...     273       14890      7369
4              1                  0.50               1  ...       6         225       356
..           ...                   ...             ...  ...     ...         ...       ...
115            1                  0.29               1  ...      13         114       811
116            1                  0.40               1  ...       4         150       164
117            1                  0.00               2  ...       3         833      3572
118            0                  0.17               1  ...       1         219      1695
119            1                  0.44               1  ...       3          39        68
```





```
Name: fake, Length: 576, dtype: int64 0      0
1      0
2      0
3      0
4      0
      ..
115    1
116    1
117    1
118    1
119    1
Name: fake, Length: 120, dtype: int64
[[1. 0.]
 [1. 0.]
 [1. 0.]
 ...
 [0. 1.]
 [0. 1.]
 [0. 1.]] [[1. 0.]
 [1. 0.]
 [1. 0.]
 [1. 0.]
 [1. 0.]
 [1. 0.]
 [1. 0.]
 [1. 0.]
 [1. 0.]
 [1. 0.]
 [1. 0.]
 [1. 0.]
```

DATASETS:
insta_test.csv

profile pic	nums/length username	fullname words	nums/length fullname	name==username	description length	external URL	private
1	0.33	1	0.33	1	30	0	1
1	0	5	0	0	64	0	1
1	0	2	0	0	82	0	1
1	0	1	0	0	143	0	1
1	0.5	1	0	0	76	0	1
1	0	1	0	0	0	0	1
1	0	1	0	0	132	0	1
1	0	2	0	0	0	0	1
1	0	2	0	0	96	0	1
1	0	1	0	0	78	0	1
1	0	1	0	0	0	0	1
1	0.14	1	0	0	78	1	1
1	0.14	2	0	0	61	0	1
1	0.33	2	0	0	45	0	1
1	0.1	2	0	0	43	0	0
1	0	2	0	0	56	0	1
1	0.33	2	0	0	86	0	1
1	0	1	0	0	97	0	1
1	0	3	0	0	46	0	1
1	0	2	0	0	39	0	1
1	0.5	1	0	0	0	0	1
1	0	0	0	0	28	0	1
1	0.22	2	0	0	63	0	1
1	0	2	0	0	24	0	1
1	0.1	1	0	0	4	0	1
1	0	1	0	0	27	0	1
1	0	1	0	0	137	1	0
1	0	1	0	0	69	0	1
1	0.33	1	0	0	0	0	1
1	0	2	0	0	147	1	0
1	0	2	0	0	0	0	0
1	0	2	0	0	138	1	0
1	0	2	0	0	117	1	0
1	0	2	0	0	0	0	1
1	0.17	1	0	0	0	0	0
1	0	2	0	0	14	0	0
1	0	2	0.3	0	18	0	0
1	0	4	0	0	54	1	0

1	0	1	0	0	0	1	0
1	0.15	2	0	0	73	1	0
1	0	0	0	0	47	0	1
1	0	2	0	0	84	1	0
1	0	1	0	0	39	1	0
1	0.09	2	0	0	77	0	0
1	0	1	0	0	33	0	1
1	0	2	0	0	149	1	1
1	0	2	0	0	8	0	1
1	0	2	0	0	0	0	0
1	0	2	0	0	18	0	1
1	0	9	0	0	133	1	0
1	0	3	0	0	0	0	0
1	0	2	0	0	81	0	1
1	0	2	0	0	4	0	0
1	0	1	0	0	13	0	1
1	0	2	0	0	20	0	0
1	0	7	0	0	24	0	1
1	0	2	0	0	19	0	0
1	0.14	2	0	0	0	0	0
1	0	2	0	0	146	0	0
1	0	2	0	0	0	0	1
0	0.05	1	0	0	0	0	0
1	0.27	1	0	0	0	0	0
0	0.07	1	0	0	0	0	0
0	0	1	0	1	0	0	0
0	0	2	0	0	0	0	0
0	0.22	0	0	0	0	0	0
0	0	3	0	0	0	0	0
0	0	1	0	1	0	0	0
0	0	1	0	0	0	0	0
0	0.62	1	0.4	0	0	0	0
0	0	2	0	0	14	0	0
0	0.42	1	0	0	0	0	0
0	0.62	1	0.4	0	0	0	0

insta_train.csv

profile pic	nums/length username	fullname words	nums/length fullname	name==username	description length	external URL	private
1	0.27	0	0	0	53	0	0
1	0	2	0	0	44	0	0
1	0.1	2	0	0	0	0	1
1	0	1	0	0	82	0	0
1	0	2	0	0	0	0	1
1	0	4	0	0	81	1	0
1	0	2	0	0	50	0	0
1	0	2	0	0	0	0	0
1	0	0	0	0	71	0	0
1	0	2	0	0	40	1	0
1	0	2	0	0	54	0	0
1	0	2	0	0	54	1	0
1	0	2	0	0	0	1	0
1	0	2	0	0	103	1	0
1	0	2	0	0	98	1	0
1	0	3	0	0	46	0	0
1	0	3	0	0	0	0	0
1	0.29	3	0	0	48	0	0
1	0	2	0	0	63	1	0
1	0	2	0	0	106	1	0
1	0	2	0	0	40	0	0
1	0	1	0	0	35	1	1
1	0	2	0	0	30	0	0
1	0	1	0	0	27	0	0
1	0	0	0	0	0	0	0
1	0	1	0	0	109	1	1
1	0	6	0	0	0	0	1
1	0	2	0	0	132	1	0
1	0	2	0	0	126	1	0
1	0	2	0	0	122	0	1
1	0	2	0	0	138	0	1
1	0.13	0	0	0	0	0	1
1	0	2	0	0	50	0	1
1	0	2	0	0	35	0	0
1	0	2	0	0	56	1	0
1	0.18	2	0	0	9	0	0
1	0.33	0	0	0	0	0	1
1	0	2	0	0	81	0	0
1	0	2	0	0	134	0	1
1	0	2	0	0	0	0	0
1	0	0	0	0	2	0	0
1	0	2	0	0	0	1	0

1	0	2	0	0	23	0	0
1	0	2	0	0	138	1	0
1	0	4	0	0	35	0	0
1	0	3	0	0	93	0	0
1	0	2	0	0	4	0	1
1	0	2	0	0	1	0	1
1	0	1	0	0	4	0	0
1	0	0	0	0	23	0	0
1	0	3	0	0	91	1	0
1	0	2	0	0	57	0	0
1	0	1	0	0	108	1	0
1	0	2	0.12	0	30	1	0
1	0	0	0	0	82	0	0
1	0.12	1	0	0	12	1	0
1	0	2	0	0	54	0	0
1	0	1	0	0	0	0	1
1	0	1	0	0	12	0	0
1	0.12	2	0	0	0	0	0
1	0	1	0	0	3	0	0
1	0	1	0	0	39	1	0
1	0.19	2	0	0	0	0	0
1	0	1	0	0	68	1	0
1	0	2	0	0	129	1	0
1	0	2	0	0	57	1	0
1	0	2	0	0	64	0	0
1	0	1	0	0	42	0	0
1	0	2	0	0	71	1	0
1	0	3	0	0	0	0	0
1	0.33	2	0	0	70	0	0
1	0	1	0	0	74	0	0
1	0	1	0	0	8	0	1
1	0	2	0	0	35	0	0
1	0.2	2	0	0	0	0	1
1	0	1	0	0	0	0	1
1	0	2	0	0	28	0	1
1	0	2	0	0	18	0	1
1	0	3	0	0	28	0	0
1	0.33	1	0	0	36	0	0
1	0	2	0	0	2	0	0
1	0.06	2	0	0	11	0	1
1	0	3	0	0	70	1	0
1	0	2	0	0	29	0	0
1	0	2	0	0	24	1	0
1	0	3	0	0	21	1	0
1	0	1	0	0	81	0	0

1	0	1	0	0	34	1	1
1	0	0	0	0	40	0	0
1	0	1	0	0	12	0	0
1	0	2	0	0	0	0	1
1	0	2	0	0	59	0	1
1	0	1	0	0	15	0	1
1	0	2	0	0	54	0	0
1	0	1	0	0	16	0	0
1	0	2	0	0	73	0	1
1	0	1	0	0	24	0	0
1	0	0	0	0	0	0	0
1	0	0	0	0	26	1	0
1	0	3	0	0	0	0	0
1	0	2	0	0	0	0	1
1	0	0	0	0	0	0	1
1	0	3	0	0	0	0	1
1	0	1	0	0	28	0	1
1	0	2	0	0	55	1	0
1	0	1	0	0	140	1	0
1	0	0	0	0	122	1	0
1	0	2	0	0	113	1	0
1	0	2	0	0	38	0	1
1	0	2	0	0	0	0	0
1	0	2	0	0	23	0	0
1	0	1	0	0	0	0	0
1	0.2	1	0	0	89	0	1
1	0.44	1	0	0	30	0	1
1	0	1	0	0	0	0	0
1	0.1	1	0.1	0	0	0	0
1	0	1	0	0	0	0	0
1	0	1	0.08	0	12	0	1
1	0	0	0	0	123	1	0
1	0.24	1	0.24	0	0	0	0
1	0.14	2	0	0	0	0	1
1	0	0	0	0	0	0	1
1	0	2	0	0	40	0	0
1	0.29	2	0	0	0	0	0
1	0	2	0	0	33	0	1
1	0	0	0	0	0	0	1
1	0	2	0	0	5	0	1
1	0	1	0	0	0	0	1
1	0.18	2	0	0	23	0	1
1	0	1	0	0	35	1	0
1	0	4	0	0	150	1	0
1	0	1	0	0	26	0	1

1	0	2	0	0	149	0	1
1	0	3	0	0	129	0	0
1	0	2	0	0	0	0	0
1	0	0	0	0	18	0	1
1	0	2	0	0	74	1	0
1	0	2	0	0	0	1	1
1	0	1	0	0	59	0	0
1	0	3	0	0	148	0	0
1	0	0	0	0	0	1	0
1	0.29	1	0	0	15	0	0
1	0	12	0	0	46	0	1
1	0	2	0	0	5	0	1
1	0	1	0	0	98	0	0
1	0	2	0	0	55	0	0
1	0	2	0	0	19	1	0
1	0.36	5	0	0	71	0	0
1	0.22	2	0	0	133	0	0
1	0.08	2	0	0	150	1	0
1	0	2	0	0	43	0	1
1	0.15	2	0	0	37	0	1
1	0	1	0	0	35	0	1
1	0	2	0	0	87	0	0
1	0.14	2	0	0	59	0	0
1	0	2	0	0	0	0	0
1	0	3	0	0	0	0	0
1	0	2	0	0	9	0	1
1	0	2	0	0	12	0	1
1	0.1	0	0	0	0	0	0
1	0.09	1	0	0	0	0	0
1	0.15	3	0	0	95	1	1
1	0	1	0	0	0	0	0
1	0	2	0	0	46	0	1
1	0	3	0	0	123	0	1
1	0	6	0	0	117	1	0
1	0	1	0	0	26	0	1
1	0	0	0	0	0	0	1
1	0	2	0	0	58	0	1
1	0	1	0	0	0	0	0
1	0	1	0	0	30	0	1
1	0	3	0	0	62	1	0
1	0.45	1	0.25	0	137	1	0
1	0	2	0	0	149	1	0
0	0	0	0	0	14	0	1
1	0	2	0	0	19	0	1
1	0	10	0	0	131	1	0

1	0	2	0	0	0	0	0	
1	0.09	5	0	0	5	0	0	
1	0.09	2	0	0	0	0	0	
1	0	2	0	0	11	0	1	
1	0	2	0	0	0	0	1	
1	0	2	0	0	27	1	0	
1	0.38	2	0	0	10	1	0	
1	0	2	0	0	72	1	0	
1	0	2	0	0	3	0	1	
1	0	3	0	0	51	0	1	
1	0	1	0	0	44	0	1	
1	0	2	0	0	0	0	0	
1	0	2	0	0	73	1	0	
1	0.18	1	0	0	70	0	1	
1	0	2	0	0	35	0	0	
1	0	2	0	0	13	0	1	
1	0	4	0.25	0	105	0	1	
1	0	1	0.33	0	91	0	0	
1	0.14	2	0	0	0	0	1	
1	0	2	0	0	48	0	0	
1	0	2	0	0	48	0	0	
1	0	2	0	0	126	1	1	
1	0	2	0	0	0	1	1	
1	0	2	0	0	53	0	0	
1	0	0	0	0	8	0	1	
1	0	1	0	0	67	0	0	
1	0	1	0	0	20	0	0	
1	0.08	2	0	0	26	1	0	
1	0	0	0	0	86	0	1	
1	0	1	0	0	51	0	1	9
1	0.11	4	0	0	26	1	0	1
1	0	0	0	0	18	0	1	13
1	0	2	0	0	96	1	0	14
1	0	2	0	0	17	0	0	6
1	0	2	0	0	0	0	1	7
1	0	2	0	0	62	1	0	40
1	0	2	0	0	86	0	0	1
1	0	2	0	0	148	1	0	99
1	0	2	0	0	1	0	1	1
1	0.22	3	0	0	39	0	0	2
1	0.36	2	0	0	35	0	0	
1	0	0	0	0	103	0	0	2
1	0	2	0	0	0	1	0	41
1	0	2	0	0	61	0	1	21
1	0.15	1	0	0	44	0	0	15

1	0	1	0	0	0	0	0
1	0	1	0	0	0	0	1
1	0	1	0	0	0	0	1
1	0	1	0	0	112	0	1
1	0.18	1	0	0	123	0	0
1	0	3	0	0	24	0	1
1	0	2	0	0	34	0	0
1	0	1	0	0	19	0	0
1	0	2	0	0	0	0	0
1	0	2	0	0	42	0	0
1	0.18	3	0	0	50	0	0
1	0.22	5	0	0	67	0	0
1	0.17	2	0	0	134	1	0
1	0	2	0	0	101	0	0
1	0	2	0	0	0	0	0
1	0	3	0	0	0	0	1
1	0	2	0	0	17	0	1
1	0	2	0	0	0	0	1
1	0	2	0	0	32	0	1
1	0	2	0	0	0	0	0
1	0	1	0	1	80	1	1
1	0	2	0	0	2	0	1
1	0	2	0	0	0	0	1
1	0	1	0	0	146	1	0
1	0.19	0	0	0	0	0	0
1	0	0	0	0	0	1	1
1	0.18	1	0	0	0	0	0
1	0.31	1	0	0	0	0	1
1	0	0	0	0	6	0	0
1	0.15	1	0	0	0	0	0
1	0	2	0	0	0	0	1
1	0.22	1	0.14	0	0	0	1
1	0	1	0	0	64	0	0
1	0	1	0	0	0	0	0
1	0.24	0	0	0	0	0	1
1	0	1	0	0	0	0	1
1	0	2	0	0	49	1	0
1	0	2	0	0	23	0	0
1	0	2	0	0	120	1	0
1	0.14	2	0	0	34	0	0
1	0	3	0	0	25	0	1
1	0	2	0	0	0	0	1
1	0	2	0	0	12	0	0
1	0.17	2	0	0	0	0	1
1	0	2	0	0	9	0	1

1	0.08	2	0	0	1	0	0
1	0	2	0	0	18	0	1
1	0	2	0	0	34	0	1
1	0	3	0	0	23	0	1
1	0	1	0	0	19	0	1
1	0	2	0	0	139	0	0
1	0	2	0	0	13	0	1
1	0	2	0	0	50	0	0
1	0	3	0	0	46	1	1
1	0	1	0	0	30	0	1
1	0.3	2	0	0	26	0	1
1	0	0	0	0	0	0	1
1	0	1	0	0	0	0	1
1	0.11	1	0	0	0	0	0
0	0	1	0	0	27	0	1
1	0.07	2	0	0	37	1	1
1	0	1	0	0	31	0	1
1	0	1	0	0	20	0	1
1	0	2	0	0	7	0	0
1	0	2	0	0	0	0	0
1	0	2	0	0	0	0	1
0	0.22	1	0	0	0	0	0
0	0.38	1	0	0	0	0	0
0	0.43	1	0	0	0	0	1
1	0	0	0	0	0	0	1
1	0.5	3	0	0	24	0	1
0	0.31	2	0	0	0	0	0

Chapter:5

COADING

```
# Importing Libraries and Dataset

import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
import seaborn as sns

import tensorflow as tf
from tensorflow import keras
from tensorflow.keras.layers import Dense, Activation, Dropout
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.metrics import Accuracy

from sklearn import metrics
from sklearn.preprocessing import LabelEncoder
from sklearn.metrics import classification_report,accuracy_score,roc_curve,confusion_matrix

# Load the training dataset
instagram_df_train=pd.read_csv('insta_train.csv')
instagram_df_train

# Load the testing data
instagram_df_test=pd.read_csv('insta_test.csv')
instagram_df_test

instagram_df_train.head()
instagram_df_train.tail()

instagram_df_test.head()
instagram_df_test.tail()

#Performing Exploratory Data Analysis EDA

# Getting dataframe info
instagram_df_train.info()

# Get the statistical summary of the dataframe
instagram_df_train.describe()

# Checking if null values exist
instagram_df_train.isnull().sum()
```

```
# Get the number of unique values in the "profile pic" feature
instagram_df_train['profile pic'].value_counts()

# Get the number of unique values in "fake" (Target column)
instagram_df_train['fake'].value_counts()

instagram_df_test.info()

instagram_df_test.describe()

instagram_df_test.isnull().sum()

instagram_df_test['fake'].value_counts()

# Perform Data Visualizations

# Visualize the data
sns.countplot(instagram_df_train['fake'])
plt.show()

# Visualize the private column data
sns.countplot(instagram_df_train['private'])
plt.show()

# Visualize the "profile pic" column data
sns.countplot(instagram_df_train['profile pic'])
plt.show()

# Visualize the data
plt.figure(figsize = (20, 10))
sns.distplot(instagram_df_train['nums/length username'])
plt.show()

# Correlation plot
plt.figure(figsize=(20, 20))
cm = instagram_df_train.corr()
ax = plt.subplot()
sns.heatmap(cm, annot = True, ax = ax)
plt.show()

sns.countplot(instagram_df_test['fake'])

sns.countplot(instagram_df_test['private'])
sns.countplot(instagram_df_test['profile pic'])

# Preparing Data to Train the Model

# Training and testing dataset (inputs)
X_train = instagram_df_train.drop(columns = ['fake'])
X_test = instagram_df_test.drop(columns = ['fake'])
X_train

X_test

# Training and testing dataset (Outputs)
y_train = instagram_df_train['fake']
y_test = instagram_df_test['fake']

y_train

y_test

# Scale the data before training the model
from sklearn.preprocessing import StandardScaler, MinMaxScaler
scaler_x = StandardScaler()
X_train = scaler_x.fit_transform(X_train)
X_test = scaler_x.transform(X_test)

y_train = tf.keras.utils.to_categorical(y_train, num_classes = 2)
y_test = tf.keras.utils.to_categorical(y_test, num_classes = 2)

y_train

y_test

# print the shapes of training and testing datasets
X_train.shape, X_test.shape, y_train.shape, y_test.shape

Training_data = len(X_train)/( len(X_test) + len(X_train) ) * 100
Training_data

Testing_data = len(X_test)/( len(X_test) + len(X_train) ) * 100
Testing_data

# Building and Training Deep Training Model
```

```
import tensorflow.keras
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Dropout

model = Sequential()
model.add(Dense(50, input_dim=11, activation='relu'))
model.add(Dense(150, activation='relu'))
model.add(Dropout(0.3))
model.add(Dense(150, activation='relu'))
model.add(Dropout(0.3))
model.add(Dense(25, activation='relu'))
model.add(Dropout(0.3))
model.add(Dense(2,activation='softmax'))

model.summary()

model.compile(optimizer = 'adam', loss = 'categorical_crossentropy', metrics = ['accuracy'])

epochs_hist = model.fit(X_train, y_train, epochs = 50,  verbose = 1, validation_split = 0.1)

import tensorflow.keras
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Dropout

model = Sequential()
model.add(Dense(50, input_dim=11, activation='relu'))
model.add(Dense(150, activation='relu'))
model.add(Dropout(0.3))
model.add(Dense(25, activation='relu'))
model.add(Dropout(0.3))
model.add(Dense(25, activation='relu'))
model.add(Dropout(0.3))
model.add(Dense(2, activation='softmax'))
model.summary()

# Access the Performance of the model

print(epochs_hist.history.keys())

plt.plot(epochs_hist.history['loss'])
plt.plot(epochs_hist.history['val_loss'])

plt.title('Model Loss Progression During Training/Validation')
plt.ylabel('Training and Validation Losses')
plt.xlabel('Epoch Number')
plt.legend(['Training Loss', 'Validation Loss'])
plt.show()

predicted = model.predict(X_test)

predicted_value = []
test = []
for i in predicted:
    predicted_value.append(np.argmax(i))

for i in y_test:
    test.append(np.argmax(i))

print(classification_report(test, predicted_value))

plt.figure(figsize=(10, 10))
cm=confusion_matrix(test, predicted_value)
sns.heatmap(cm, annot=True)
plt.show()


from dataset_load import instagram_df_train,instagram_df_test
from import_libs import *

instagram_df_train.head()
instagram_df_train.tail()

instagram_df_test.head()
instagram_df_test.tail()

# Getting dataframe info
instagram_df_train.info()

# Get the statistical summary of the dataframe
instagram_df_train.describe()

# Checking if null values exist
instagram_df_train.isnull().sum()



# Get the number of unique values in the "profile pic" feature
instagram_df_train['profile pic'].value_counts()
```

```
# Get the number of unique values in "fake" (Target column)
instagram_df_train['fake'].value_counts()

instagram_df_test.info()

instagram_df_test.describe()

instagram_df_test.isnull().sum()

instagram_df_test['fake'].value_counts()

# Perform Data Visualizations

# Visualize the data
sns.countplot(instagram_df_train['fake'])
plt.show()

# Visualize the private column data
sns.countplot(instagram_df_train['private'])
plt.show()

# Visualize the "profile pic" column data
sns.countplot(instagram_df_train['profile pic'])
plt.show()

# Visualize the data
plt.figure(figsize = (20, 10))
sns.distplot(instagram_df_train['nums/length username'])
plt.show()

# Correlation plot
plt.figure(figsize=(20, 20))
cm = instagram_df_train.corr()
ax = plt.subplot()
sns.heatmap(cm, annot = True, ax = ax)
plt.show()

sns.countplot(instagram_df_test['fake'])

sns.countplot(instagram_df_test['private'])

sns.countplot(instagram_df_test['profile pic'])
```

```
from dataset_load import instagram_df_test,instagram_df_train
from import_libs import *

# Preparing Data to Train the Model

# Training and testing dataset (inputs)
X_train = instagram_df_train.drop(columns = ['fake'])
X_test = instagram_df_test.drop(columns = ['fake'])

print(X_train,X_test)


# Training and testing dataset (Outputs)
y_train = instagram_df_train['fake']
y_test = instagram_df_test['fake']

print(y_train,y_test)


# Scale the data before training the model
from sklearn.preprocessing import StandardScaler, MinMaxScaler
scaler_x = StandardScaler()
X_train = scaler_x.fit_transform(X_train)
X_test = scaler_x.transform(X_test)

y_train = tf.keras.utils.to_categorical(y_train, num_classes = 2)
y_test = tf.keras.utils.to_categorical(y_test, num_classes = 2)

print(y_train,y_test)



# print the shapes of training and testing datasets
X_train.shape, X_test.shape, y_train.shape, y_test.shape

Training_data = len(X_train)/( len(X_test) + len(X_train) ) * 100


Testing_data = len(X_test)/( len(X_test) + len(X_train) ) * 100


print(Training_data, Testing_data)
```

```
from import_libs import *

train_data_path = 'datasets/Fake-Instagram-Profile-Detection-main/insta_train.csv'
test_data_path = 'datasets/Fake-Instagram-Profile-Detection-main/insta_test.csv'

pd.read_csv(test_data_path)


train_data_path = 'datasets/Insta_Fake_Profile_Detection/train.csv'
test_data_path = 'datasets/Insta_Fake_Profile_Detection/test.csv'

pd.read_csv(train_data_path)

# Load the training dataset
instagram_df_train=pd.read_csv(train_data_path)
instagram_df_train

# Load the testing data
instagram_df_test=pd.read_csv(test_data_path)
instagram_df_test
```

```
from import_libs import *

from dataset_load import instagram_df_train,instagram_df_test

from data_preprocess_model import X_train,y_train,X_test,y_test

# Building and Training Deep Training Model
import tensorflow.keras
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Dropout

model = Sequential()
model.add(Dense(50, input_dim=11, activation='relu'))
model.add(Dense(150, activation='relu'))
model.add(Dropout(0.3))
model.add(Dense(150, activation='relu'))
model.add(Dropout(0.3))
model.add(Dense(25, activation='relu'))
model.add(Dropout(0.3))
model.add(Dense(2,activation='softmax'))
model.summary()

model.compile(optimizer = 'adam', loss = 'categorical_crossentropy', metrics = ['accuracy'])

epochs_hist = model.fit(X_train, y_train, epochs = 50,  verbose = 1, validation_split = 0.1)
```

```
from main_model import model, epochs_hist
from data_preprocess_model import X_test,y_test
from import_libs import *


predicted = model.predict(X_test)

predicted_value = []
test = []
for i in predicted:
    predicted_value.append(np.argmax(i))

for i in y_test:
    test.append(np.argmax(i))

print(classification_report(test, predicted_value))

plt.figure(figsize=(10, 10))
cm=confusion_matrix(test, predicted_value)
sns.heatmap(cm, annot=True)
plt.show()

# Access the Performance of the model

print(epochs_hist.history.keys())

plt.plot(epochs_hist.history['loss'])
plt.plot(epochs_hist.history['val_loss'])

plt.title('Model Loss Progression During Training/Validation')
plt.ylabel('Training and Validation Losses')
plt.xlabel('Epoch Number')
plt.legend(['Training Loss', 'Validation Loss'])
plt.show()


plt.plot(epochs_hist.history['accuracy'])
plt.plot(epochs_hist.history['val_accuracy'])



plt.title('Model Accuracy Progression During Training/Validation')
plt.ylabel('Training and Validation Losses')
plt.xlabel('Epoch Number')
plt.legend(['Training Acc', 'Validation Acc'])
plt.show()


dicts = {
   'Accuracy' : epochs_hist.history['accuracy'],
   'Validation_Accuracy' : epochs_hist.history['val_accuracy'],
   'Loss' : epochs_hist.history['loss'],
   'Validation Loss' : epochs_hist.history['val_loss']




}

model_training_progress = pd.DataFrame(dicts)
model_training_progress

print(model_training_progress)

def get_avg(lst):
   return sum(lst) / len(lst)


print("Accuracy : ", get_avg(model_training_progress['Accuracy']) * 100)

print("Validation Accuracy : ", get_avg(model_training_progress['Validation_Accuracy']) * 100)


print("Loss : ", get_avg(model_training_progress['Loss']) * 100)
print("Validation Loss : ", get_avg(model_training_progress['Validation Loss']) * 100)
```

Chapter:6

CONCLUSION AND LIMITATIONS

6.1 Conclusion:

The development of a robust fake Instagram ID detection system marks a significant stride towards enhancing user security and trust on the platform. By leveraging advanced algorithms and analysis techniques, coupled with user authentication and verification mechanisms, the system aims to mitigate the risks associated with fake accounts, including identity theft and the spread of misinformation. Through a multi-layered approach encompassing behavioral analysis, identity verification, and real-time monitoring, the system empowers users to identify and report suspicious activity, fostering a safer and more trustworthy online environment. Furthermore, ongoing collaboration with law enforcement agencies and continuous refinement of detection algorithms ensure the system's effectiveness in combating evolving threats posed by malicious actors.

6.2 Limitations:

Despite the system's comprehensive approach and advanced capabilities, several limitations warrant consideration:

6.2.1 False Positives/Negatives:

The detection algorithms may occasionally generate false positives, flagging legitimate accounts as suspicious, or false negatives, failing to identify certain fake accounts.

6.2.2 Resource Constraints:

The system's effectiveness may be limited by resource constraints, including computational resources for processing large volumes of data and storage capacity for maintaining historical records.

6.2.3 Scalability Challenges:

As user traffic and data volume increase, scalability challenges may arise, necessitating upgrades to server infrastructure and optimization of processing algorithms.

Chapter: 7

REFERENCES

- https://www.geeksforgeeks.org
- https://www.kaggle.com
- https://www.youtube.com
- https://chat.openai.com