

SATHYABAMA

**INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)**

CATEGORY - 1 UNIVERSITY BY UGC

Accredited with Grade "A++" by NAAC | Approved by AICTE

www.sathyabama.ac.in

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

PROJECT VIVA VOCE

INSTAGENIE - AI POWERED COMMUNICATION TOOL

PROJECT STUDENTS -
HRITURAJ SAHA, 41110495.
HUSSAIN M, 41110496.

GUIDE -
Ms. G. KAVITHA, M.E., (Ph.D.)

AGENDA

- Abstract
- Objective(s)
- Literature Survey
- Proposed System
- Modules Explanation
- Hardware & Software Requirements.
- Results and Discussion
- Conclusion
- References
- Paper Publication Status

ABSTRACT

Instagram automation, particularly in the form of automated comments and likes, has become a popular strategy among marketers and influencers to enhance engagement and visibility on the platform. It is an innovative automated chatbot designed to streamline social media management by efficiently handling likes, comments, and posts on platforms like Instagram. The primary objective of this project is to enhance user engagement and presence on social media by automating routine tasks, thereby allowing users to focus on creating quality content and strategic interactions.

OBJECTIVE(S)

The objectives of the proposed work is to :-

- To streamline social media management by automating likes, comments, and post scheduling.
- It aims to enhance user engagement and online presence by reducing manual effort.
- By leveraging advanced natural language processing and machine learning, InstaGenie ensures authentic and contextually relevant interactions.

LITERATURE SURVEY

| SL. No | Author | Title of the paper | Method | Merits | Demerits |
|--------|---|--|--|--|---|
| 1. | Carta, S.; Corrigan, A.; Ferreira, A.; Podda, A.S.; Recupero, D.R. (2020) | A multi-layer and multi-ensemble stock trader using deep learning and deep reinforcement learning. | In this paper method starts by pre-processing data with hundreds of deep neural networks. | Adaptability :- Deep reinforcement learning allows the model to learn from its actions and adapt over time. | Computational Complexity :- Training deep neural networks and running simulations for reinforcement learning can be computationally intensive. |
| 2. | Barra, S.; Carta, S.M.; Corrigan, A.; Podda, A.S.; Recupero, D.R. (2020) | Deep learning and time series-to- image encoding for financial forecasting. | This method outperforms the buy-and-hold (B&H) strategy in a time frame where the latter provides excellent returns. | Multi-resolution Analysis :- The approach allows for the analysis of financial data across different time intervals within a single observation. | Complexity and Computational Cost :- Training an ensemble of CNNs on encoded images derived from time series data can be computationally intensive and time-consuming |

LITERATURE SURVEY

| SL. No | Author | Title of the paper | Method | Merits | Demerits |
|--------|---|--|---|--|---|
| 3. | Van Engelen, J.E.; Hoos, H.H. (2020) | Multi-DQN: An Ensemble of Deep Q-Learning Agents for Stock Market Forecasting. | Here we exploit a Q-learning agent trained several times with the same training data and investigate its ensemble behavior in important real-world stock markets. | Efficiency :- It requires less labeled data compared to supervised learning, making it more efficient in situations where obtaining labeled data is costly or time-consuming | Data Requirements :- Although semi-supervised learning reduces the need for labeled data, it does not eliminate this requirement entirely. |
| 4. | Carta, S.; Ferreira, A.; Podda, A.S.; Recupero, D.R.; Sanna, A (2019) | A survey on semi-supervised learning. | This proposes a new taxonomy of semi-supervised classification algorithms. | Handle Noisy Data :- Multi-DQN addresses this challenge by learning to maximize a return function over the training stage without relying on annotated data thus reducing the risk of overfitting to noisy data. | Data Requirements :- While the method aims to minimize reliance on annotated data, the effectiveness of Multi-DQN still depends on the quality and quantity of historical stock market data available for training. |

LITERATURE SURVEY

| SL. No | Author | Title of the paper | Method | Merits | Demerits |
|--------|--------------------------------|--|---|--|---|
| 5. | Meena, K.S.; Suriya, S. (2019) | A Survey on Supervised and Unsupervised Learning Techniques. | This survey paper gives an overview of supervised algorithms, namely, support vector machine, decision tree, naïve Bayes, KNN, and linear regression, and an overview of unsupervised algorithms, namely, K-means, agglomerative divisive, and neural networks. | Predictive Accuracy :- Supervised learning models are known for their high predictive accuracy because they learn from labeled training data. This makes them particularly effective for tasks where the output is known and needs to be predicted accurately. | Overfitting :- Supervised learning models can suffer from overfitting, where they perform well on training data but poorly on unseen data. This requires techniques like regularization and cross-validation to mitigate. |

LITERATURE SURVEY

| SL. No | Author | Title of the paper | Method | Merits | Demerits |
|--------|--|--|---|---|---|
| 6. | Michael F. Porter, David G. Novick, and Laura E. Beckman (2022) - IEEE Xplore. | ChatPapers: An AI Chatbot for Interacting with Academic Research | This paper introduces ChatPapers, an AI chatbot designed to assist researchers in navigating academic papers. The AI uses a large language model (LLM) to interpret and summarize academic publications, enabling researchers to interact with a wealth of academic information in a conversational manner. | Quick Access :- Facilitates quick access to key research insights by summarizing large amounts of text and offers an interactive experience for academic researchers, making it easier to identify the main points in complex research papers. | Limited Training Data :- The tool's effectiveness is limited to the quality of training data used to teach the LLM. If the input data is biased or outdated, the chatbot may produce inaccurate or misleading results. |

PROPOSED SYSTEM

- Instagenie, a hypothetical tool designed to automate comments and likes on social media platforms like Instagram, could potentially leverage the Time Frequency Decomposition (TFD) filter algorithm to enhance its functionality.
- The Time Frequency Decomposition filter algorithm is a powerful tool in signal processing, capable of analyzing signals in both time and frequency domains simultaneously.
- Automating comments and likes on Instagram using the TFD filter algorithm requires a combination of image recognition, NLP, and web automation skills.

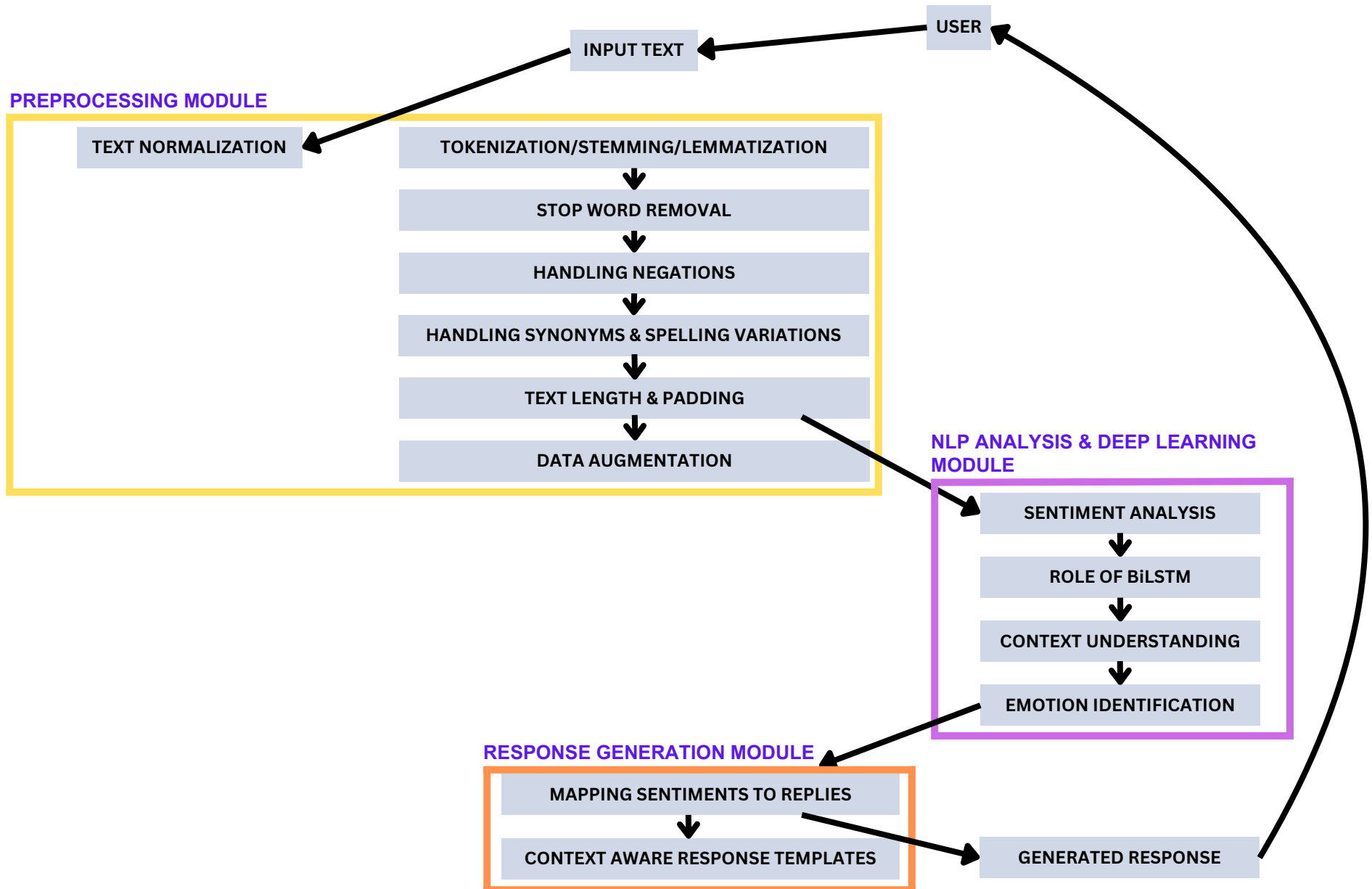
PROPOSED SYSTEM

Advantages -

- By automating routine tasks such as liking, commenting, and scheduling posts, InstaGenie significantly reduces the time and effort required for social media management, allowing users to focus on strategic activities.
- Automated interactions ensure a consistent level of engagement, which helps maintain a strong presence on social media platforms.
- InstaGenie allows users to tailor settings and interactions based on their specific needs and preferences. This customization ensures that automated actions align with the user's brand voice and objectives, providing a more personalized social media presence.

PROPOSED SYSTEM

SYSTEM ARCHITECTURE



MODULES EXPLANATION

MODULES -

Preprocessing Module -

The preprocessing module is an essential step in preparing raw text data for further analysis. It transforms user input into a more standardized, clean format, making it suitable for machine learning models and natural language processing (NLP) tasks. By performing operations like normalization, tokenization, stemming, and stop word removal, this module ensures that the text data is processed consistently, helping improve the accuracy and efficiency of subsequent analysis, such as sentiment or emotion detection.

MODULES EXPLANATION

NLP Analysis and Deep Learning Module -

The NLP (Natural Language Processing) Analysis and Deep Learning Module processes the cleaned text data from the preprocessing stage and extracts meaningful insights using advanced machine learning techniques. It leverages a Bi- Directional Long Short-Term Memory(BiLSTM) network to perform sentiment analysis and classify the text based on emotional tone. This module helps in understanding the context and intent of the user's input, allowing the system to generate accurate and relevant responses.

MODULES EXPLANATION

Response Generation Module -

The Response Generation Module is responsible for crafting appropriate replies based on the insights derived from the NLP Analysis and Deep Learning Module. It ensures that responses are context-aware, personalized, and aligned with the user's sentiment and emotion. By mapping identified sentiments to predefined responses and leveraging response templates, the module delivers timely and relevant communication.

MODULES EXPLANATION

Engagement Module -

The Engagement Module is designed to enhance user interaction and foster a sense of community on the platform. It automates actions such as liking posts and following users, reducing manual effort while improving user satisfaction and retention. By using predefined criteria, the module ensures that these actions are meaningful and relevant, encouraging active participation and positive engagement

HARDWARE & SOFTWARE REQUIREMENTS

Hardware Requirements -

Server Infrastructure :

1. Cloud-Based Servers :- To ensure scalability and high availability, InstaGenie should be hosted on cloud-based servers such as AWS, Google Cloud, or Microsoft Azure.
2. Processor :- Multi-core processors (e.g., Intel Xeon or AMD EPYC).
3. Memory (RAM) :- Minimum 16 GB RAM.
4. Storage :- Fast SSDs (Solid State Drives) are necessary for quick data access and reduced latency.

Software Requirements -

Operating System :

1. Server OS :- Linux-based operating systems (e.g., Ubuntu, CentOS).
2. Client OS :- Windows, macOS, and Linux.

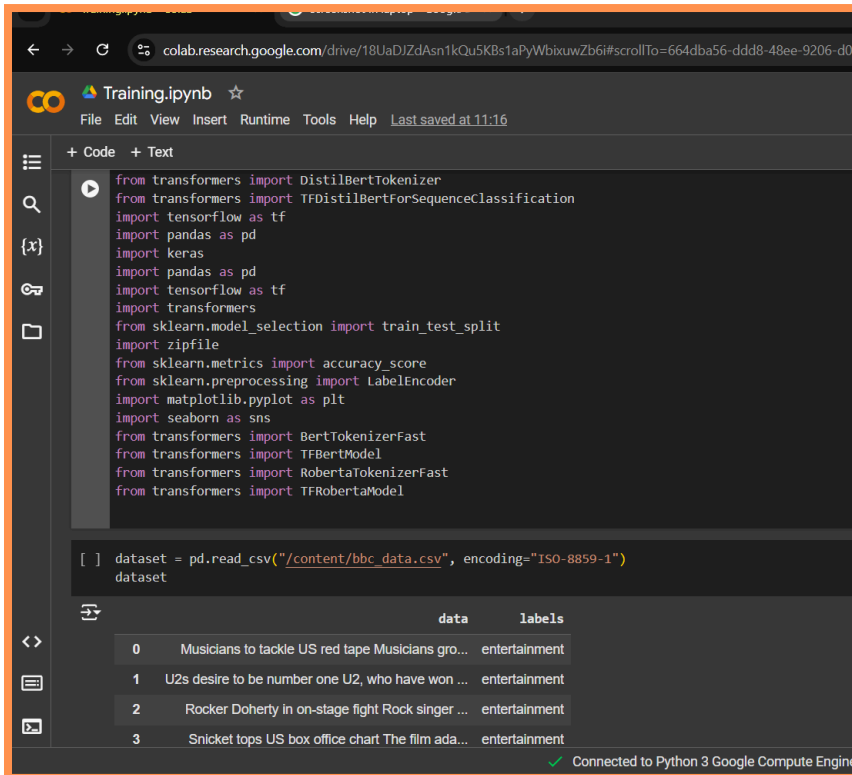
Backend Development : PyTorch, Scikit-learn, Node.js or Django.

Frontend Development : JavaScript, HTML/CSS.

Database Management : MySQL/PostgreSQL.

Machine Learning Frameworks : TensorFlow or PyTorch, NLP.

RESULTS AND DISCUSSION



The screenshot shows a Google Colab notebook titled "Training.ipynb". The code cell contains the following imports:

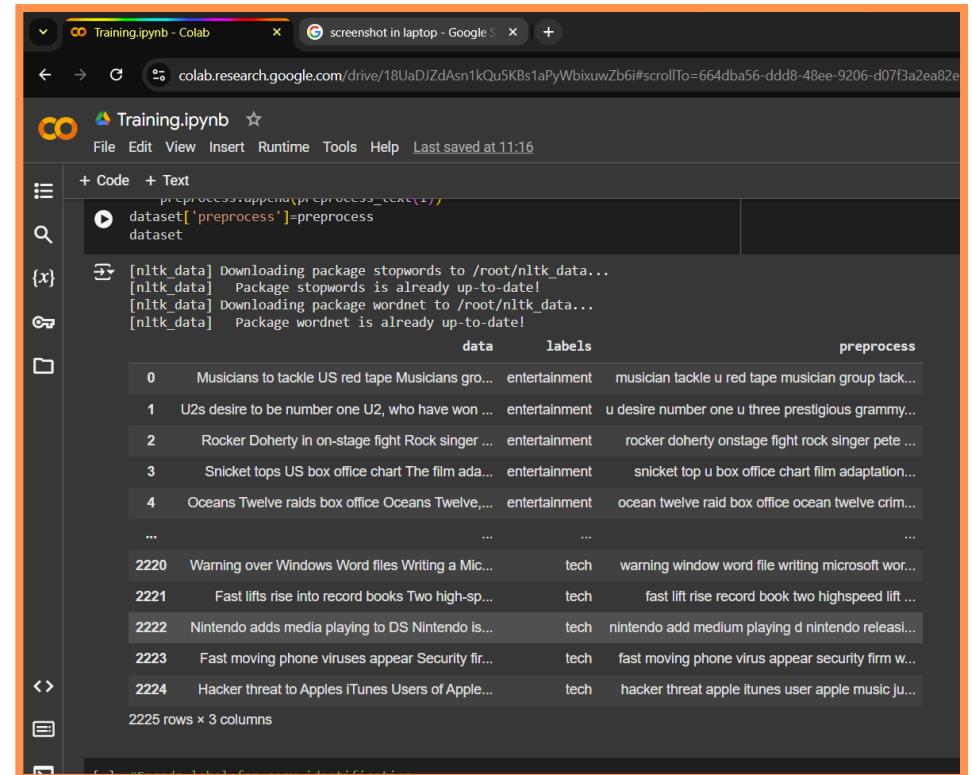
```
from transformers import DistilBertTokenizer
from transformers import TFDistilBertForSequenceClassification
import tensorflow as tf
import pandas as pd
import keras
import pandas as pd
import tensorflow as tf
import transformers
from sklearn.model_selection import train_test_split
import zipfile
from sklearn.metrics import accuracy_score
from sklearn.preprocessing import LabelEncoder
import matplotlib.pyplot as plt
import seaborn as sns
from transformers import BertTokenizerFast
from transformers import TFBertModel
from transformers import RobertaTokenizerFast
from transformers import TFRobertaModel
```

Below the code, a variable named 'dataset' is defined as a pandas DataFrame:

```
[ ] dataset = pd.read_csv("/content/bbc_data.csv", encoding="ISO-8859-1")
dataset
```

The output shows a preview of the dataset with columns 'data' and 'labels':

| | data | labels |
|---|--|---------------|
| 0 | Musicians to tackle US red tape Musicians gro... | entertainment |
| 1 | U2s desire to be number one U2, who have won ... | entertainment |
| 2 | Rocker Doherty in on-stage fight Rock singer ... | entertainment |
| 3 | Snicket tops US box office chart The film ada... | entertainment |



The screenshot shows the same Google Colab notebook with the following code cell:

```
preprocess = preprocessing.Preprocessor.from_instances([
    dataset["preprocess"] = preprocess
dataset
```

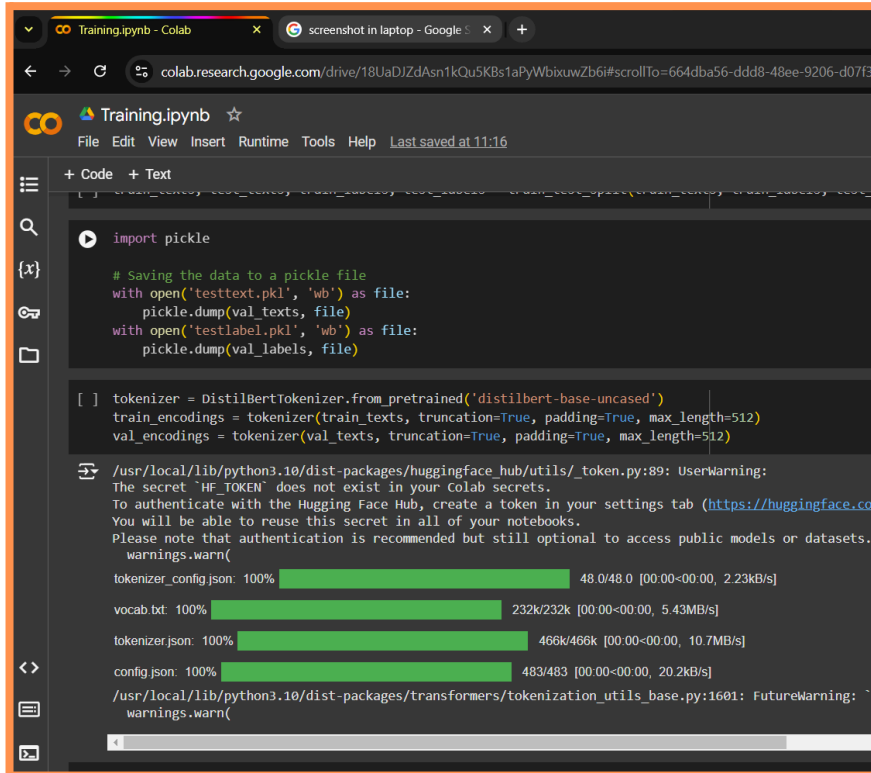
The output shows the dataset after preprocessing, with columns 'data', 'labels', and 'preprocess':

| | data | labels | preprocess |
|------|--|---------------|---|
| 0 | Musicians to tackle US red tape Musicians gro... | entertainment | musician tackle u red tape musician group tack... |
| 1 | U2s desire to be number one U2, who have won ... | entertainment | u desire number one u three prestigious grammy... |
| 2 | Rocker Doherty in on-stage fight Rock singer ... | entertainment | rocker doherty onstage fight rock singer pete ... |
| 3 | Snicket tops US box office chart The film ada... | entertainment | snicket top u box office chart film adaptation... |
| 4 | Oceans Twelve raids box office Oceans Twelve,... | entertainment | ocean twelve raid box office ocean twelve crim... |
| ... | ... | ... | ... |
| 2220 | Warning over Windows Word files Writing a Mic... | tech | warning window word file writing microsoft wor... |
| 2221 | Fast lifts rise into record books Two high-sp... | tech | fast lift rise record book two highspeed lift ... |
| 2222 | Nintendo adds media playing to DS Nintendo is... | tech | nintendo add medium playing d nintendo releasi... |
| 2223 | Fast moving phone viruses appear Security fir... | tech | fast moving phone virus appear security firm w... |
| 2224 | Hacker threat to Apples iTunes Users of Apple... | tech | hacker threat apple itunes user apple music ju... |

2225 rows x 3 columns

- To import various libraries from Hugging Face's `transformers`, TensorFlow, and other machine learning tools to work with pre-trained transformer models for sequence classification tasks, such as `DistilBERT`, `BERT`, and `RoBERTa`. It also includes data preprocessing, model training, and evaluation utilities like accuracy score and label encoding.
- To loads a CSV file named "bbc_data.csv" into a pandas DataFrame . The dataset likely contains text data, possibly news articles from the BBC, for further analysis or modeling. and preprossesing the dataet

RESULTS AND DISCUSSION

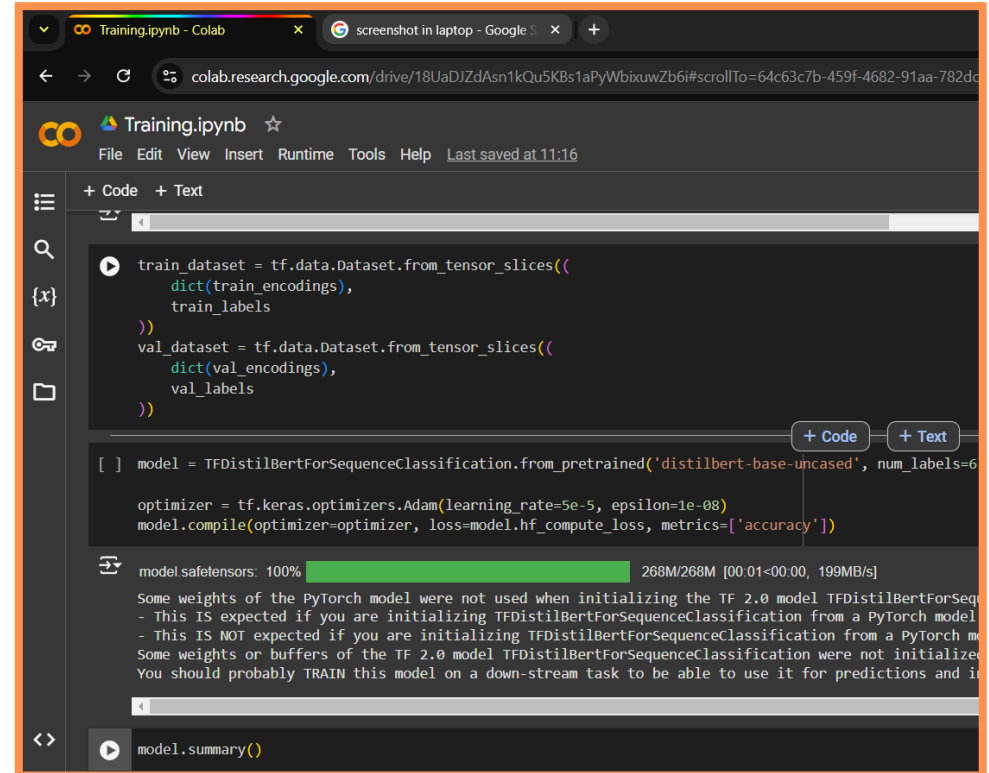


```
import pickle

# Saving the data to a pickle file
with open('testtext.pkl', 'wb') as file:
    pickle.dump(val_texts, file)
with open('testlabel.pkl', 'wb') as file:
    pickle.dump(val_labels, file)

[ ] tokenizer = DistilBertTokenizer.from_pretrained('distilbert-base-uncased')
train_encodings = tokenizer(train_texts, truncation=True, padding=True, max_length=512)
val_encodings = tokenizer(val_texts, truncation=True, padding=True, max_length=512)

/usr/local/lib/python3.10/dist-packages/huggingface_hub/utils/_token.py:89: UserWarning:
The secret `HF_TOKEN` does not exist in your Colab secrets.
To authenticate with the Hugging Face Hub, create a token in your settings tab (https://huggingface.co/settings/tokens).
You will be able to reuse this secret in all of your notebooks.
Please note that authentication is recommended but still optional to access public models or datasets.
warnings.warn(
tokenizer_config.json: 100% ██████████ 48.0/48.0 [00:00<00:00, 2.23kB/s]
vocab.txt: 100% ██████████ 232k/232k [00:00<00:00, 5.43MB/s]
tokenizer.json: 100% ██████████ 466k/466k [00:00<00:00, 10.7MB/s]
config.json: 100% ██████████ 483/483 [00:00<00:00, 20.2kB/s]
/usr/local/lib/python3.10/dist-packages/transformers/tokenization_utils_base.py:1601: FutureWarning: `
```



```
train_dataset = tf.data.Dataset.from_tensor_slices((
    dict(train_encodings),
    train_labels
))
val_dataset = tf.data.Dataset.from_tensor_slices((
    dict(val_encodings),
    val_labels
))

[ ] model = TFDistilBertForSequenceClassification.from_pretrained('distilbert-base-uncased', num_labels=6)

optimizer = tf.keras.optimizers.Adam(learning_rate=5e-5, epsilon=1e-08)
model.compile(optimizer=optimizer, loss=model.hf_compute_loss, metrics=['accuracy'])

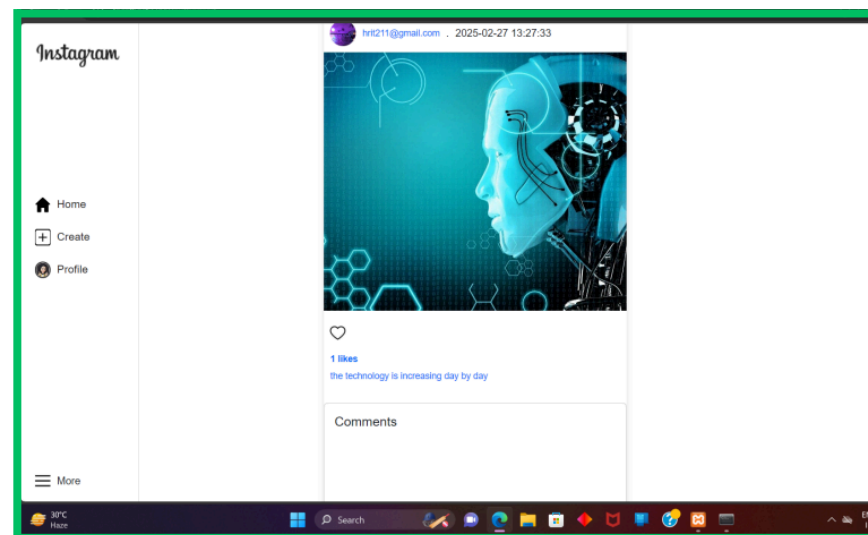
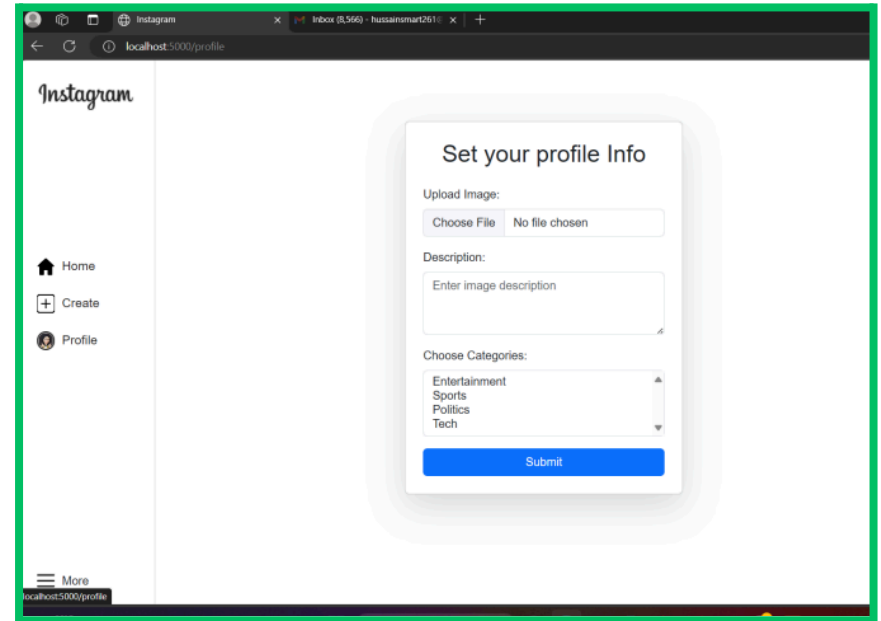
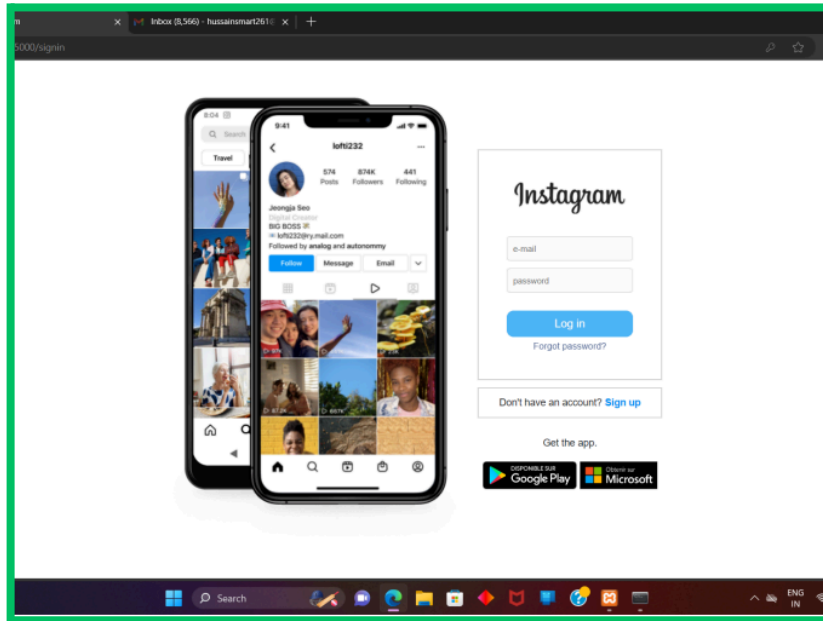
models.safetensors: 100% ██████████ 268M/268M [00:01<00:00, 199MB/s]

Some weights of the PyTorch model were not used when initializing the TF 2.0 model TFDistilBertForSeq
- This IS expected if you are initializing TFDistilBertForSequenceClassification from a PyTorch model
- This IS NOT expected if you are initializing TFDistilBertForSequenceClassification from a PyTorch m
Some weights or buffers of the TF 2.0 model TFDistilBertForSequenceClassification were not initializ
You should probably TRAIN this model on a down-stream task to be able to use it for predictions and i

model.summary()
```

- The code serializes the `val_texts` and `val_labels` variables into binary format and saves them as `testtext.pkl` and `testlabel.pkl` files using Python's `pickle` module. This allows for efficient storage and later loading of the data.
- The code creates TensorFlow `Dataset` objects for training (`train_dataset`) and validation (`val_dataset`) by converting the encoded text data (`train_encodings`, `val_encodings`) and their corresponding labels (`train_labels`, `val_labels`) into tensor slices. These datasets are used for efficient batching and model training in TensorFlow.

RESULTS AND DISCUSSION



CONCLUSION

Instagenie represents a significant advancement in the field of social media management by automating essential tasks such as likes, comments, and post scheduling. This innovative tool leverages cutting-edge natural language processing and machine learning technologies to ensure that interactions on social media are authentic, timely, and contextually relevant. By automating these routine yet crucial activities, InstaGenie significantly reduces the time and effort required to maintain an active and engaging social media presence.

The system can be further developed and expanded to address evolving requirements and challenges in social media interaction management :-
Integration of Multilingual Support - Expanding the system's capabilities to process and analyze text in multiple languages will increase its applicability to a global audience and **Enhanced Emotion Detection** - Incorporating more complex emotional classifications, such as anger, joy, or fear, will improve the system's ability to generate nuanced and empathetic responses.

REFERENCES

- Baltaci S, Ersoz AR (2022) Social media engagement, fear of missing out and problematic internet use in secondary school children. *Int Online J Educ Sci* 14.
- Bansal S, Gowda K, Kumar N (2022) A hybrid deep neural network for multimodal personalized hashtag recommendation. In: *IEEE transactions on computational social systems*.
- Gayberi, M.; Gunduz Oguducu, S. Popularity Prediction of Posts in Social Networks Based on User, Post and Image Features.
- Karthikeyan K, Wang Z, Mayhew S, Roth D (2019) Cross-lingual ability of multilingual BERT: an empirical study. In: *International conference on learning representations*.
- Kumar N, Baskaran E, Konjengbam A, Singh M (2021) Hashtag recommendation for short social media texts using word-embeddings and external knowledge. *Knowl Inf Syst* 63(1):175–198.
- Peláez, J.I.; Martínez, E.A.; Vargas, L.G. Products and services valuation through unsolicited information from social media. *Soft Comput.* 2020.

REFERENCES

- Pes, B. Ensemble feature selection for high-dimensional data: A stability analysis across multiple domains. *Neural Comput. Appl.* 2022.
- Rousidis, D.; Koukaras, P.; Tjortjis, C. Social media prediction: A literature review. *Multimed. Tools Appl.* 2022, 79, 6279–6311.
- Saia, R.; Carta, S. Evaluating the benefits of using proactive transformed-domain-based techniques in fraud detection tasks. *Future Gener. Comput. Syst.* 2023.
- Shao, C., & Ciampaglia, G. L. (2020). Proceedings of the National Academy of Sciences, 115(40), 9582-9587. Understanding the Impact of Social Media Bots on the Spread of Misinformation.
- Smith, A., & Jones, M. (2022). *Journal of Digital Ethics*, 5(2), 78-93. The Ethics of Automating Online Interactions: Implications for Social Media Engagement.
- Wang X, Zhang Y, Yamasaki T (2021) Earn more social attention: user popularity based tag recommendation system. In: Companion proceedings of the web conference 2020

PAPER PUBLICATION STATUS

PAPER PRESENTED AT 13TH INTERNATIONAL CONFERENCE ON CONTEMPORARY
ENGINEERING AND TECHNOLOGY (ICCET 2025)



JOURNAL - INTERNATIONAL JOURNAL OF SCIENCE AND INNOVATIVE ENGINEERING & TECHNOLOGY (IJSIET).
INDEXING - GOOGLE SCHOLAR.
STATUS - SUBMITTED AND UNDER REVIEW.
SUBMISSION DATE - 28/03/2025.

THANK YOU

We thank God, Our Department, Guide, Panel Members, Supportive Professors and all Technical and non Technical staff who helped us in our Project.