

Dark Pattern Recognition

Manimeghala, Satyajit S K, Sabari M

Department of Computer Science and Engineering, Sri Shakthi Institute of Engineering and Technology, India

Abstract: This project introduces a robust system for the automated detection of dark patterns on websites, aiming to enhance user protection and transparency in online interactions. Leveraging a Naive Bayes classifier trained on dark pattern categories such as Bait and Switch, Forced Continuity, Price Comparison Prevention, Hidden Costs, and Sneaking, the model achieves effective identification of deceptive design elements. The preprocessing of textual data involves employing the TFIDF vectorizer for feature extraction, optimizing the classifier's performance. Web scraping is facilitated through cloud scraping techniques and Beautiful Soup, enabling the extraction of relevant data for classification. The resulting model file is applied to classify scraped data, empowering users to make informed decisions while navigating online interfaces. This innovative approach addresses the ethical concerns associated with dark patterns and contributes to a safer and more transparent online environment.

Keywords: Dark pattern detection, Naive Bayes classifier, TFIDF vectorizer, Web scraping, Cloud scraper, User protection, Transparency, Deceptive design, Online ethics.

1. INTRODUCTION

The pervasive use of digital platforms for various activities has led to an increased prevalence of deceptive design strategies, commonly known as dark patterns, on websites. Dark patterns are user interface elements crafted to manipulate users into making decisions that may not align with their best interests. This project focuses on the development of a sophisticated system for the automated detection of dark patterns, enhancing user awareness and safeguarding online experiences. The methodology involves training a Naive Bayes classifier on distinct dark pattern categories, including Bait and Switch, Forced Continuity, Price Comparison Prevention, Hidden Costs, and Sneaking. To optimize the classification process, textual data is preprocessed using the TFIDF vectorizer, capturing the

significance of terms within the dataset. Web scraping, facilitated by cloud scraping techniques and Beautiful Soup, enables the extraction of pertinent data from websites, creating a diverse dataset for model training. By combining machine learning, natural language processing, and web scraping, this project addresses the ethical concerns surrounding deceptive online practices. The resulting model empowers users by providing a means to identify and avoid websites employing dark patterns, fostering a safer and more transparent digital ecosystem. The significance of this work lies in its potential to contribute to a user-centric online environment, promoting trust and informed decisionmaking in the digital realm.

2. LITERATURE SURVEY

The body of research on dark patterns explores various domains including e-commerce, mobile apps, social robotics, and user interface design. A key focus is on automating dark pattern detection through machine learning and computer vision. Notably, Yada et al. (2022) created a benchmark dataset using e-commerce texts, where RoBERTa achieved 97.5% accuracy in detecting manipulative content. AidUI (2023) expanded detection capabilities to visual elements in UI screenshots, identifying ten types of dark patterns with a respectable F1-score of 0.65 and strong localization (IoU of 0.84). Similarly, DarkDialogs (2023) effectively identified dark patterns in cookie consent dialogs across thousands of websites with 99% classification accuracy.

Other studies explore more conceptual and emerging areas. Research on social and home robots (Lacey et al., 2019; Dula et al., 2023) warns that emotionally manipulative behaviors—such as using "cuteness" or simulated empathy—can function as dark patterns by exploiting users' emotional responses, especially among vulnerable populations. Parrilli and Hernández-

Ramírez (2020) propose re-designing dark patterns to ethically guide users toward better privacy decisions, transforming them from deceptive tools to protective ones. Furthermore, user-focused studies (Nimkoompai, 2022) show a general lack of awareness about dark patterns, identifying “Forced continuity” and “Disguised ads” as the most prevalent in mobile apps.

Finally, broader societal analysis has been conducted through social media data and UX design theory. Feng et al. (2023) found increasing public resistance to dark patterns through a 12-year tweet analysis, with frequent discussion around “sneaking” and “obstruction” tactics. Tiangpanich and Nimkoompai (2022) distinguish between dark patterns and anti-patterns, advocating for better design choices to avoid user deception. Together, these studies highlight the technical, ethical, and social dimensions of dark patterns, laying the groundwork for more transparent and user-respecting digital environments.

3. PROPOSED METHOD

Machine Learning Block:

In this block, a series of Python scripts leverage the scikit-learn library to train and evaluate a Multinomial Naive Bayes classifier for dark pattern detection. The dataset, loaded and preprocessed using Pandas, undergoes a split into training and testing sets. Text vectorization is achieved through the TF-IDF method, enriching the feature extraction process. The Naive Bayes model is trained on the transformed training data, and its performance is evaluated on the testing set. The trained model, along with the TF-IDF vectorizer, is serialized into files for later use in real-time predictions.

GUI Block:

The GUI block is built using the Tkinter library, providing a user-friendly interface for interacting with the dark pattern detection system. The Tkinter window includes components such as an entry field for URL input, an 'Analyze' button to trigger the analysis, and a scrolled text widget to display the results. The GUI interacts with the trained model and vectorizer to fetch, parse, and classify content from the provided URLs. Users can seamlessly input URLs, initiate the analysis, and receive instant

feedback on the presence of dark patterns, providing a practical and accessible means of enhancing online user protection.

Together, these blocks form a cohesive system that combines machine learning capabilities with an intuitive GUI for effective and user-centric dark pattern detection in web content.

MODULE DESCRIPTION:

1. Dataset Loading and Preprocessing:

Step 1: Load the dataset from 'dataset.csv' using Pandas.

Step 2: Split the dataset into training and testing sets, with 80% for training and 20% for testing.

2. Text Vectorization using TFIDF:

Step 3: Utilize the TFIDF vectorizer (with a maximum of 5000 features) to convert text data into numerical vectors.

Transform both the training and testing sets.

3. Training the Naive Bayes Model:

Step 4: Employ a Multinomial Naive Bayes classifier for training on the TFIDF transformed training data.

Generate a model that learns the patterns associated with different dark pattern categories.

4. Model Evaluation:

Step 5: Evaluate the trained model using the testing set. Calculate and print the accuracy score and a classification report.

5. Model and Vectorizer Serialization:

Step 6: Save the trained Naive Bayes model and TFIDF vectorizer using the joblib library.

Create

dark_pattern_detection_model_naive_bayes.pkl' and 'tfidf_vectorizer.pkl' files.

6. Web Scraping and Classification:

Utilize the trained model and vectorizer in a Tkinterbased GUI application for realtime analysis of URLs.

Use cloudscraper and BeautifulSoup to fetch and parse HTML content.

Clean the extracted text by removing unnecessary elements and save it in 'sc.txt'.

Classify the cleaned text into dark pattern categories,

segmenting text by paragraphs.

7. GUI Implementation:

Develop a Tkinterbased graphical user interface (GUI) with input for entering URLs.

Include a button to trigger the analysis and a scrolled text widget to display the results.

The 'Analyze' button invokes the scrape_and_classify function, providing predictions for each paragraph in the URL's content.

8. Tkinter Event Loop:

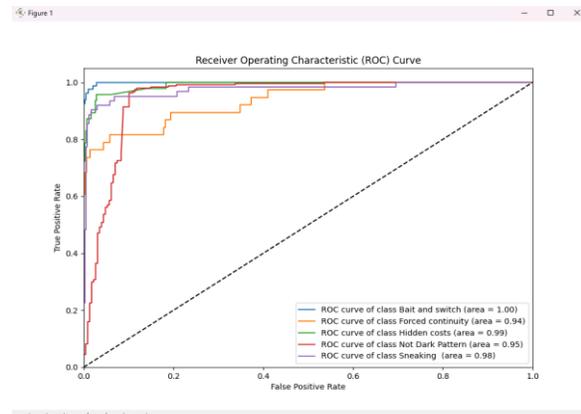
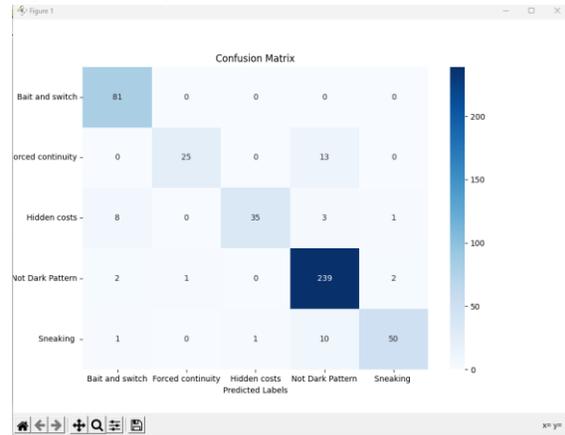
Launch the Tkinter event loop to ensure the continuous functioning of the GUI.

Users can input URLs, trigger the analysis, and receive realtime predictions on the presence of dark patterns.

This methodology seamlessly integrates machine learning, web scraping, and GUI development to create a comprehensive system for detecting dark patterns in website content. The trained model and vectorizer enable efficient realtime analysis of URLs, offering users insights into the potential presence of deceptive design elements.

system's capabilities, potentially exploring more sophisticated models and expanding the dataset to further refine the detection of dark patterns. The successful implementation of the Naive Bayes classifier establishes a valuable tool for users to navigate the digital landscape with increased awareness and protection against deceptive practices, contributing to a safer and more transparent online experience.

Application output:

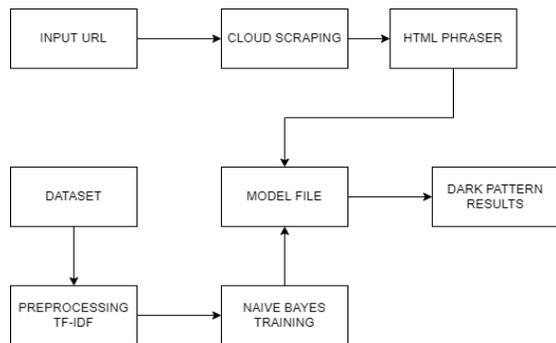


```

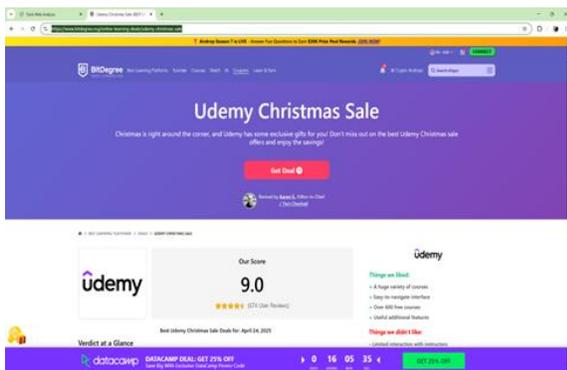
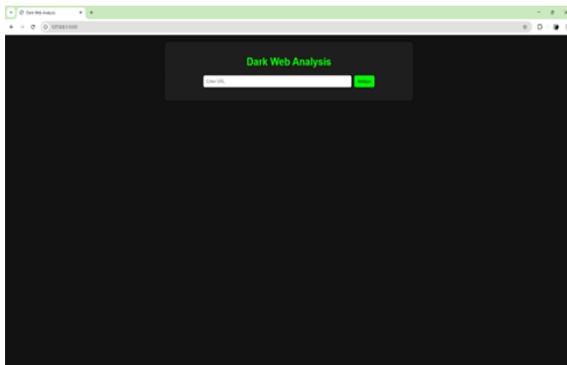
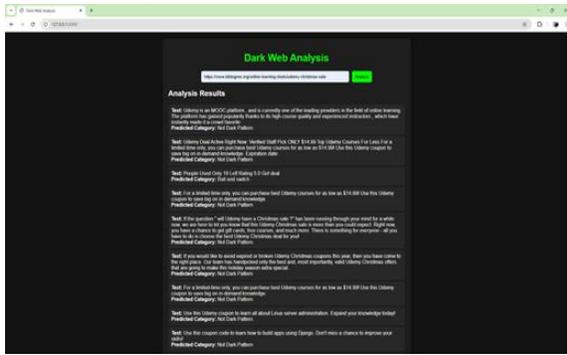
Administrator Command Promp
Microsoft Windows [Version 10.0.22631.4692]
(c) Microsoft Corporation. All rights reserved.

C:\Users\satya>cd C:\Users\satya\OneDrive\Desktop\SSEC-DARK PATTERN
C:\Users\satya\OneDrive\Desktop\SSEC-DARK PATTERN>python app.py
DEPRECATION: The OpenSSL being used by this python install (OpenSSL 1.1.0j 20 Nov 2018) does not meet the minimum supported version (on OpenSSL 1.1.1) in order to support TLS 1.3 required by CloudFlare, you may encounter an unexpected Captcha or cloudflare 1929 blocks.
* Serving Flask app 'app'
* Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on http://127.0.0.1:5000
Press CTRL+C to quit
* Restarting with stat
DEPRECATION: The OpenSSL being used by this python install (OpenSSL 1.1.0j 20 Nov 2018) does not meet the minimum supported version (on OpenSSL 1.1.1) in order to support TLS 1.3 required by CloudFlare, you may encounter an unexpected Captcha or cloudflare 1929 blocks.
* Debugger is active!
* Debugger PIN: 246-520-716
    
```

4. CONCLUSION



In conclusion, the Naive Bayes classifier, as evidenced by the classification report, has demonstrated commendable performance in detecting various dark patterns on websites. With high precision and recall across multiple categories, including Bait and Switch, Hidden Costs, and Sneaking, the classifier exhibits a robust ability to identify deceptive design elements. The overall accuracy of 91% underscores the effectiveness of the implemented solution in providing users with reliable insights into potentially harmful online practices. This achievement lays a solid foundation for advancing the



dark patterns on cookie dialogs",2023 IEEE 8th European Symposium on Security and Privacy (EuroS&P)

[4] Cherie Lacey,Catherine Caudwell,"Cuteness as a u2018Dark Patternu2019 in Home Robots",2019 14th ACM/IEEE International Conference on Human-Robot Interaction (HRI)

[5] Davide Maria Parrilli,Rodrigo Hernu00e1ndez-Ramu00e9drez,"Re-Designing Dark Patterns to Improve Privacy",2020 IEEE International Symposium on Technology and Society (ISTAS)

[6] Elizabeth Dula,Andres Rosero,Elizabeth Phillips,"Identifying Dark Patterns in Social Robot Behavior",2023 Systems and Information Engineering Design Symposium (SIEDS)

[7] Apichaya Nimkoompai,"Risk Analysis of Encountering Dark Patterns of UX E-commerce Applications Affecting Personal Data",2022 6th International Conference on Information Technology (InCIT)

[8] Jiaying Feng,Fan Mo,Yuki Yada,Tsuneo Matsumoto,Nao Fukushima,Fukuyo Kido,Hayato Yamana,"Analysis of Dark Pattern-related Tweets from 2010",2023 IEEE 8th International Conference on Big Data Analytics (ICBDA)

[9] Pumarin Tiangpanich,Apichaya Nimkoompai,"An Analysis of Differences between Dark Pattern and Anti-Pattern to Increase Efficiency Application Design",2022 7th International Conference on Business and Industrial Research (ICBIR)

REFERENCES

- [1] Yuki Yada, Jiaying Feng, Tsuneo Matsumoto, Nao Fukushima, Hayato Yamana, "Dark patterns in e-commerce: a dataset and its baseline evaluations", 2022 IEEE International Conference on Big Data (Big Data)
- [2] S M Hasan Mansur, Sabiha Salma, Damilola Awofisayo, Kevin Moran, "AidUI: Toward Automated Recognition of Dark Patterns in User Interfaces", 2023 IEEE/ACM 45th International Conference on Software Engineering (ICSE)
- [3] Daniel Kirkman, Kami Vaniea, Daniel W. Woods, "DarkDialogs: Automated detection of 10