**Detecting and Informing Digital Deception**

submitted in partial fulfillment of the requirement

for the award of the Degree of

**Bachelor of Technology**

in

**Computer Science & Engineering (Data Science)**

by

**Gauri Gughe**

**Harshvardhan Pawar**

**Nikhil Prajapati**

under the guidance of

**Prof. Siddhartha Chandra**

****

**Department of Computer Science & Engineering**

Bharatiya Vidya Bhavan’s

Sardar Patel Institute of Technology

(Autonomous Institute Affiliated to University of Mumbai)

Munshi Nagar, Andheri-West, Mumbai-400058

University of Mumbai

**Project Approval Certificate**

This is to certify that the Project entitled **“Enhancing Accessibility for People with Disabilities”** by Ms. **Gauri Ghuge**, Mr. **Harshvardhan Pawar**  and Mr. **Nikhil Prajapati** is found to be satisfactory and is partially approved for the award of Degree of Bachelor of Technology in Computer Engineering from University of Mumbai.

**External Examiner** **Internal Examiner**

**(signature) (signature)**

**Name: Name:**

**Date : Date :**

**Seal of the Institute**

**Contents**

1 Abstract…………………………………………. 4

2 Introduction……………………………………... 5

3 Objective………………………………………... 6

4 Literature Survey……………………………….. 7

5 Architecture Diagrams…………………………..11

6 Methodology…………………………………….13

7 Results & Discussion……………………………25

8 Conclusion………………………………………28

**Abstract**

The project endeavors to address accessibility challenges encountered by individuals with visual impairments by introducing a comprehensive platform offering image description and text-to-speech capabilities. Leveraging cutting-edge technologies such as React, Python, and AI models specialized for audio description and text-to-speech conversion, the objective is to generate complete narrations of both image objects and text. Prioritizing simplicity and user customization, the project underscores the significance of continuous user feedback, iterative development, and collaboration with accessibility experts to ensure inclusivity and usability. Through this endeavor, it aspires to bridge the accessibility gap and empower individuals with disabilities to engage more seamlessly with academic content.

**Introduction**

Individuals with visual or hearing impairments often face challenges in accessing online content. This project aims to tackle these challenges by developing a platform that provides real-time audio descriptions, text-to-speech, and sign language translation.Visually impaired students encounter significant challenges in learning mathematics due to their limitations in reading and writing mathematical formulas. Typically, these students rely on human readers to access and interpret mathematical course materials. To tackle this challenge creating an AI model to analyze visual content. Generate concise and descriptive audio narrations for images. Text-to-Speech (TTS) Convert text content on web pages and documents into natural-sounding speech.Offer multiple voice options and adjustable speech With an Indian accent for speech.

**Objective**

* To utilize object detection using YOLO for identifying objects within images.
* To employ OCR for extracting text from each identified object.
* To utilize the extracted features as input for a transformer model, generating image captions.
* To pass the generated captions as prompts to a large language model (LLM), generating narrations.
* To convert the narrations to text-to-speech format.
* To read aloud the generated narrations

**Literature Survey**

1. **Title:** Image to Audio Conversion to Aid Visually impaired People by CNN **(IEEE)**

**Author:** Sivaganesan D, Venkateshwaran M, Dhinesh S P

**Inference:** The proposed model uses CNNs and LSTMs to convert images to audio, with preprocessing, encoding, decoding, and postprocessing steps. CNN-LSTM architecture focuses on visual feature recognition, with hyperparameter adjustments. Post Processing refines audio output for accurate representation of image content.

**Limitation:** Limitations include dataset bias, varying audio quality affecting accuracy, limited handling of complex image content, unaddressed real-time performance feasibility, and potential usability issues with the user interface.

1. **Title:** Online Handwritten Mathematical Expression Recognition and Applications: A Survey (IEEE)

**Author:** Dmytro Zhelezniakov, Viktor Zaytsev, Olga Radyvonenko

**Inference:** The paper surveys techniques in Online Handwritten Mathematical Expression Recognition (OHMER), discussing challenges, applications, performance evaluation, and future directions, reflecting the authors' diverse backgrounds.

**Limitation: T**he paper lacks depth in analysis and experimental validation, restricting its coverage of recent advancements and niche areas in OHMER. It could exhibit bias towards favored methodologies, potentially overlooking promising alternatives. As a survey paper, it may not introduce novel techniques, limiting its contribution to OHMER research. Additionally, it may overgeneralize findings without considering specific system nuances or application challenges

1. **Title:** Audio Rendering of Mathematical Equations (ResearchGate)

**Author:** Sai Krishna Rallabandi, Kishore Prahallad, Priyanka Srivastava.

**Inference:** The paper introduces a novel approach converting mathematical equations into audio, emphasizing accessibility for visually impaired individuals. It outlines text-to-speech synthesis techniques tailored for mathematical notation. Evaluation likely includes metrics on comprehension, accuracy, and user satisfaction. Future directions may focus on improving speech synthesis naturalness, enhancing expressiveness, and extending applicability to complex mathematical structures.

**Limitation:** The paper lack comparisons, focus narrowly on specific math expressions, and lack diverse user testing. Technical constraints and long-term effectiveness may also be overlooked.

1. **Title:** Indian Text to Speech Systems: A Short Survey (IEEE)

**Author:** Jayashree Nair, Akhila Krishnan, Vrinda S

**Inference**: The paper likely focuses on Indian TTS systems, with authors from Amrita Vishwa Vidyapeetham suggesting expertise in computer science and linguistics. Described as a "short survey," it provides a concise overview of Indian TTS systems and their contributions to digital inclusivity and linguistic diversity.

**Limitations:** The paper lack depth in analyzing individual TTS systems and could potentially overlook international advancements by focusing solely on Indian systems. Affiliations may introduce bias towards specific systems, and methodological details such as selection criteria or evaluation methodology may be lacking. Findings may also be specific to the time of publication, limiting generalizability.

1. **Title:** Mathspeak: An Audio Method for Presenting Mathematical Formulae to Blind Students (ResearchGate)

**Author:** Azadeh Nazemi, Iain Murray, Nazanin Mohammadi

**Inference**: "Mathspeak" aids blind students with audio conversion of LaTeX math formulas, prioritizing conceptual preservation and navigation. However, dataset bias and audio quality variations may impact accuracy. Optimizing real-time performance and user interface enhancements are crucial.

**Limitations:** "Mathspeak" struggle with complex mathematical content like graphs or diagrams, limiting its utility in some applications.

1. **Title:** Textual Description for Mathematical Equation (IEEE)

**Author:** Ajoy Mondal, C V Jawahar

**Inference**: The paper likely develops a method for automatically generating textual descriptions of mathematical equations to improve accessibility. Authors detail the technical implementation, likely involving machine learning or natural language processing techniques, with experimental results evaluating performance. Validation against benchmark datasets may be conducted, with significant implications for enhancing accessibility to mathematical content and advancing language technology in digital learning environments.

**Limitations:** The proposed method may struggle with accurately describing complex equations, limiting its applicability. It might have performance issues with large expressions and lack user customization options. Evaluation may be biased towards certain equations, overlooking real-world shortcomings.

1. **Title:** Image classification using Deep learning (ResearchGate)

**Author:** M Manoj Krishna, M Neelima, M Harshali

**Inference**: The study likely explores deep learning for image classification, involving B.Tech students and a professor from KLEF's ECE Department. They employ frameworks like TensorFlow or PyTorch, using CNNs to classify images and evaluate models using metrics like accuracy. Findings and contact details may foster collaboration.

**Limitations:** The study faces constraints including a narrow dataset, increasing the risk of overfitting due to limited diversity. Comparisons with contemporary methods are absent, potentially limiting insights into the proposed approach's effectiveness. Findings may lack broader applicability beyond the specific dataset and task, and validation procedures might lack rigor, impacting result reliability.

1. **Title:** Sign Language Recognition (IEEE)

**Author:** Anup Kumar, Karun Thankachan, Mevin M. Dominic

**Inference**: The paper explores sign language recognition, aiming to develop a system for interpreting gestures into textual or spoken language. Authors from NIT Calicut use machine learning or computer vision techniques, potentially employing deep learning architectures like CNNs or RNNs. They contribute expertise to assistive technology for the hearing-impaired community.

**Limitations:** The study faces challenges like limited dataset variability and overfitting due to resource constraints, impacting performance. Real-world variability and hardware limitations may further limit system effectiveness and scalability. Evaluation scope may overlook certain gestures, reducing practical utility.

1. **Title:** FormulaNet: A Benchmark Dataset for Mathematical Formula Detection

**Author:** Felix M. Schmitt-Koopmann, Elaine M. Huang, Hans-Peter Hutter, Thilo Stadelmann, Alireza Darvishy

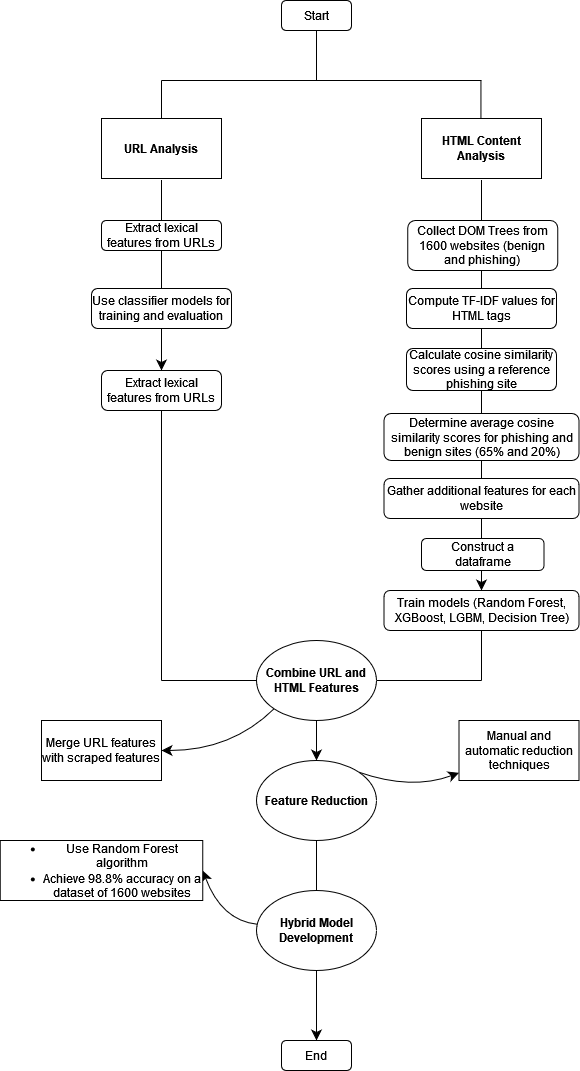
**Inference**: FormulaNet, with over 46,000 pages and nearly 1 million labeled formulas, is the largest publicly available dataset. Its automated, consistent labeling process ensures high-quality labels, including both inline and display formulas, essential for scientific document understanding.

**Limitations:** FormulaNet's formulas are solely from LaTeX source code for high-energy physics papers on arXiv.org, limiting representation. Its labeling pipeline is tailored for LaTeX, potentially hindering applicability to other formats. Model accuracy and robustness still require improvement**.**

**Architecture Diagrams**

**Block/Architecture Diagram :**

**Flow Diagram:**

****

**Methodology**

1. Data Collection:

The data for the proposed work is obtained from various sources. Data for Detecting Phishing websites using URL analysis is obtained from Kaggle Dataset consisting of around 650,000 URLs belonging to 4 classes. For working on HTML content analysis, DOM trees of websites are required which are still live (as many of the flagged websites get taken down after reporting), sources like PhishTank.org & Openphish.com are used. Finally, a combined and standardized dataset is generated out of all these 3 sources. A dataset for 1600 websites is generated as of now, which would be scaled to around 50,000 websites.

1. Data Pre-processing:

The data collected is then pre-processed and combined in a standardized manner including the target variable isPhishing, and the URL which is recorded. Initially, data cleaning procedures are applied to address issues such as missing values, incorrect data formats, and duplicates, which could otherwise introduce biases or inaccuracies in the analysis.

1. Feature Extraction

In the feature extraction process for the phishing detection project, a comprehensive set of features is extracted from both the lexical aspects of URLs and the content of HTML pages. For lexical URL features, key characteristics are captured to reveal patterns indicative of phishing behavior. These features include domain-related attributes like domain length, presence of hyphens, and domain age, which can provide insights into the authenticity of the website. Additionally, URL-specific features such as URL length, presence of special characters, and the number of subdomains are extracted to identify anomalies or suspicious patterns commonly associated with phishing URLs.

In HTML content analysis, the focus is on extracting features that reflect the structure and content of web pages. One prominent technique involves calculating TF-IDF (Term Frequency-Inverse Document Frequency) scores for words in the HTML content. This process assigns weights to terms based on their frequency in a specific page relative to their occurrence across all pages, highlighting terms that are more distinctive to a particular page. This information is then used to compute cosine similarity between the TF-IDF vectors of a given webpage and a base phishing website, enabling a quantitative comparison to detect similarities or deviations.

Beyond textual analysis, structural features of HTML pages are also extracted. This includes counting the number of login forms (num\_login\_forms) present on a webpage, as phishing pages often mimic legitimate login interfaces to harvest credentials. The analysis extends to identifying hyperlinks and external scripts embedded within the HTML, as these elements can reveal connections to external domains or potentially malicious sources. Furthermore, attributes related to the layout and composition of the HTML, such as the presence of hidden elements, obfuscated code, or redirection mechanisms, are considered as additional features to assess the likelihood of phishing activity.

**Implementation Description**

**Using URL Analysis:**

Lexical Features were extracted from URL which included features like length of URL, whether URL is httpSecure or not, presence of shortening service, digit count, consisting ip address, presence of special characters, etc.

Classifier models were used to train and evaluate, with random forest giving the best result of around 97%

**Using HTML Content Analysis:**

We began by collecting DOM Trees from 1600 websites, encompassing both benign and phishing sites. Next, we computed TF-IDF values for HTML tags within these trees. One phishing site was selected as a reference to calculate cosine similarity scores for all sites. We then determined the average cosine similarity scores for phishing and benign sites, which were 65% and 20%, respectively, in reference to the chosen phishing site. Additional features such as 'num\_forms', 'has\_login\_form', 'num\_scripts', 'has\_external\_scripts','num\_hyperlinks', 'num\_external\_hyperlinks', 'num\_js\_events' were gathered for each website. Using these features, we constructed a dataframe and trained random forest, XGBoost, LGBM, Decision Tree models.

Then we combined the previously made URL Features with the newly extracted (scrapped) features.

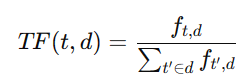
After that we used reduction techniques for feature reduction manually as well as automatically and then developed a hybrid model by using the Random Forest algorithm which gave an accuracy of 98.8% on a dataset of 1600 websites.

* **TF-IDF (Term Frequency-Inverse Document Frequency):**

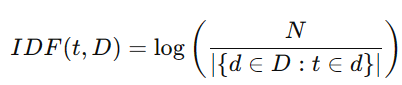
**TF-IDF** is a statistical measure used to evaluate the importance of a term in a document relative to a collection of documents (corpus). It consists of two parts:

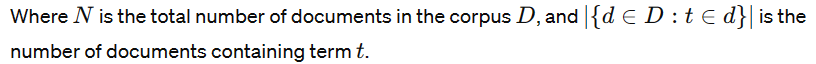
Term Frequency (TF): Measures how frequently a term appears in a document. It is calculated as the ratio of the number of times a term

occurs in a document to the total number of terms in the document.

****

**Inverse Document Frequency (IDF):** Measures the rarity of a term across the entire corpus. It is calculated as the logarithm of the ratio of the total number of documents to the number of documents containing the term.





**TF-IDF Score:** The TF-IDF score combines the TF and IDF components to determine the importance of a term within a document relative to the entire corpus.

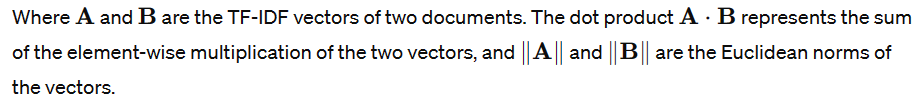
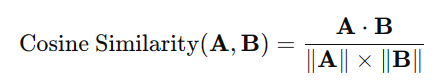


The TF-IDF score for term 𝑡 in document 𝑑 from corpus D.

* **Cosine Similarity:**

Cosine similarity is a metric used to determine the similarity between two vectors in a multi-dimensional space. Cosine similarity scores for HTML tag TF-IDF vectors are calculate using this method

The cosine similarity between two vectors is calculated as the cosine of the angle between them. It ranges from -1 (completely opposite) to 1 (exactly the same), with 0 indicating no similarity.



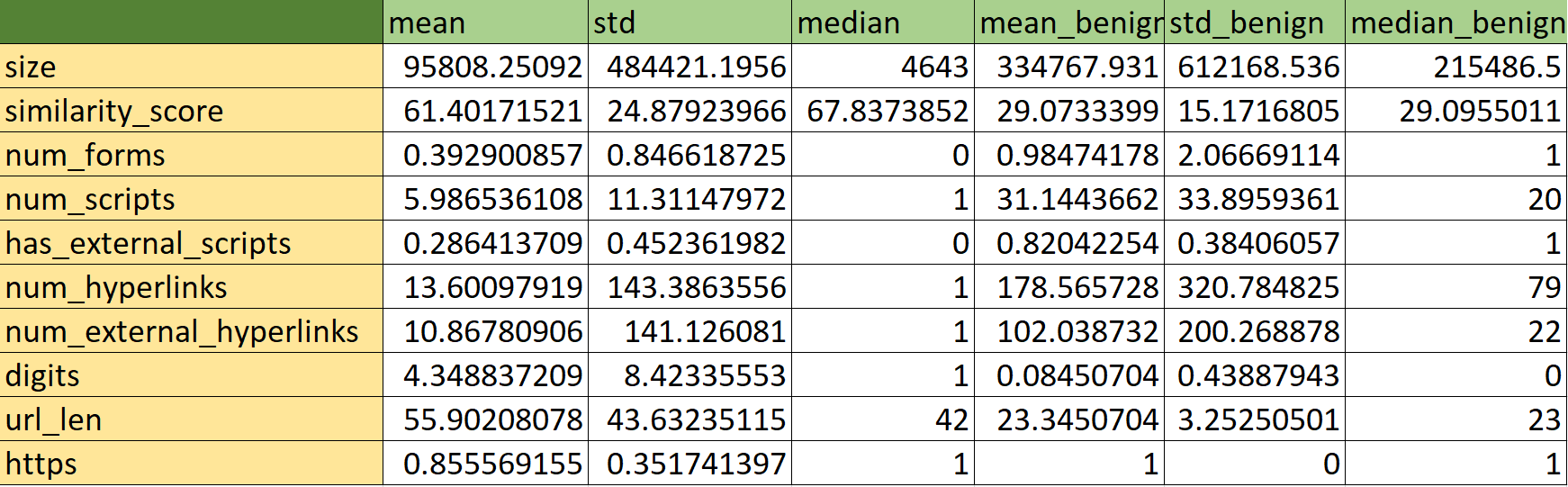
**Feature Extraction Justification :**

**Technique used to shortlist significant features :**

* First we calculated the mean, median and standard deviation of all numerical features for both benign and phishing.
* Then we calculated the percentage difference of their respective mean, median and standard deviation.
* The features with high percentage differences were shortlisted and considered as significant features for training.

The following are the mean median and standard deviation values for phishing and benign websites in the created dataset:

We can see considerable difference in the mean, median and standard deviation values of the below features between phishing and benign websites



For Benign Websites, % boolean composition:  
***has\_login\_form 2.230047***

***has\_external\_scripts 82.042254***

***Shortining\_Service 8.568075***

***having\_ip\_address 0.000000***

***https 100.000000***

For Phishing Websites, % boolean composition:  
***has\_login\_form 6.731946***

***has\_external\_scripts 28.641371***

***Shortining\_Service 3.671971***

***having\_ip\_address 0.000000***

***https 85.556916***

We can see by analyzing the percentage boolean composition that shortening and having\_ip\_address features have negligible presence when compared, hence not including these two features.

Features ‘has\_external\_scripts’ and ‘https’ show significant difference, hence adding them as distinct features for training.

**Initial Features used :**

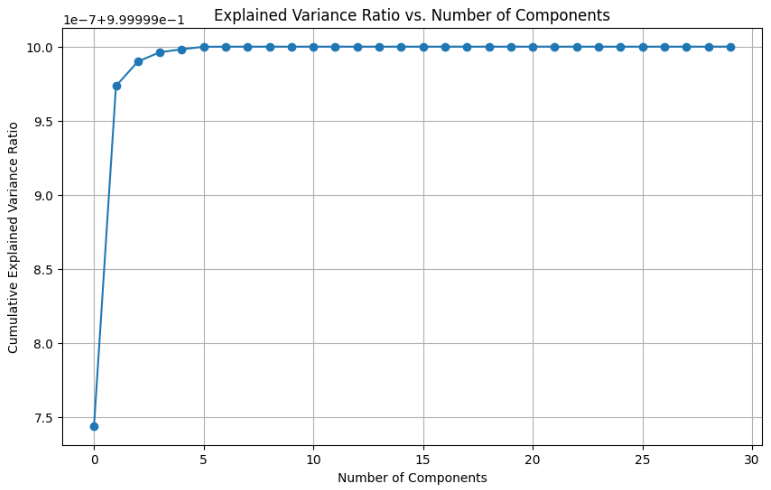
size, similarity\_score, num\_forms, has\_login\_form, num\_scripts, has\_external\_scripts, num\_hyperlinks, num\_external\_hyperlinks

Num\_js\_events, isPhishing, “@”, “?”, “ -”, “=” , “,” , “#” , “%” , “+” , “$” , “!” , “\*” , “.” , “//” , digits, url\_len, https, letters, Shortining\_Service, having\_ip\_address

**After Reduction of Non-Significant Features :**

|  |  |
| --- | --- |
| **Feature** | **Description** |
| size | It indicates the size of the DOM tree |
| similarity\_score | It is the cosine similarity score of the TF-IDF vectors of all the tags of the DOM tree between the given website and the genuine phishing websites. |
| num\_forms | It indicates the number of form tags present in a DOM tree. |
| num\_scripts | It indicates the number of occurrences of the script tag in a DOM tree. |
| num\_hyperlinks | It indicates the number of hyperlinks present in a DOM tree. |
| num\_external\_hyperlinks | It indicates the links to external domains serve as part of a redirection chain. |
| digits | It indicates the number of digits (0-9) present in the url of the website. |
| url\_len | It indicates the length of the url of a website. |
| https | It indicates whether the website is secured using https or not. (Presence of SSL certificate) |
| has\_external\_script | It indicates whether the DOM tree has external script tag or not. |

Also, PCA was carried out for automatic dimensionality reduction and feature extraction:



We can see that after 5 components, the variance is constant. Thus, with ***number\_of\_components=5***, we get high variance with as few number of features as possible.  
Hence choosing number\_of\_components=5

PCA for 5 components:  
***PC1: size, num\_hyperlinks, num\_external\_hyperlinks***

***PC2: num\_hyperlinks, num\_external\_hyperlinks, num\_scripts***

***PC3: num\_external\_hyperlinks, url\_len, letters***

***PC4: url\_len, letters, similarity\_score***

***PC5: num\_scripts, similarity\_score, url\_len***

These were the features extracted through PCA.

The analysis yielded promising results, suggesting success in both manual feature selection and PCA-derived feature extraction.

Several features chosen manually, such as those related to the number of elements in the DOM tree (size, num\_forms, num\_scripts, num\_hyperlinks) and URL properties (letters, url\_len), also appeared in the PCA results. This overlap indicated that PCA captured some of the underlying structure the manual selection aimed to target. In other words, PCA appeared to be aligned with the understanding of what features might be important for phishing detection.

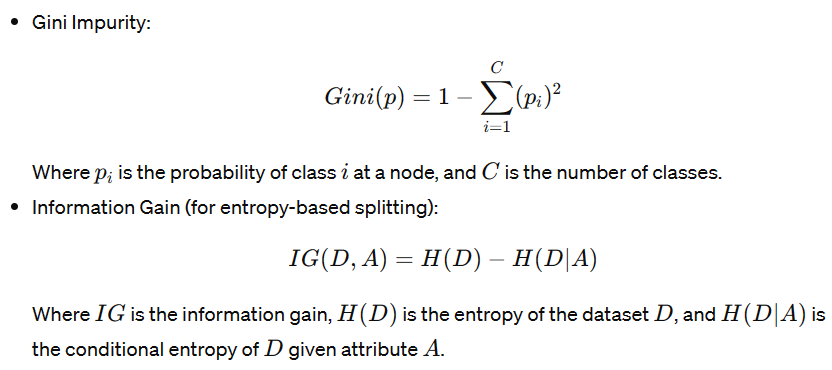
Furthermore, the observation that the explained variance plateaued after 5 components was a positive sign. PCA successfully reduced the dimensionality of the data while retaining most of the relevant information (variance). This suggested that the 5 principal components effectively represented the core patterns that the potentially larger set of manually chosen features aimed to capture.

**Model Evaluation:**

The model evaluation phase encompassed an assessment of various machine learning models, namely Decision Tree, Random Forest, XGBoost, and LightGBM. These models were selected for their adeptness in managing intricate datasets, capturing intricate nonlinear patterns, and delivering precise predictions across diverse analytical scenarios.

1. **Decision Tree:**

* A Decision Tree is a supervised learning algorithm used for both classification and regression tasks. It creates a tree-like structure where each internal node represents a feature, each branch represents a decision based on that feature, and each leaf node represents the outcome or prediction.
* Decision trees are built recursively by selecting the best feature to split the data at each node, typically based on criteria like Gini impurity or information gain.

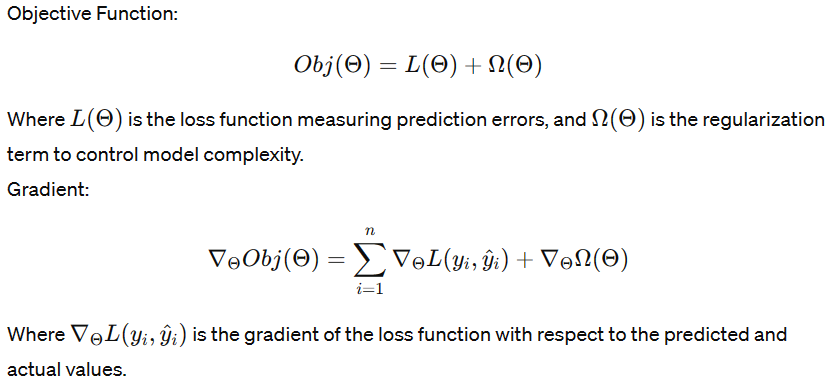


1. **Random Forest:**

* Random Forest is an ensemble learning technique that builds multiple decision trees during training and outputs the mode of the classes (classification) or the mean prediction (regression) of individual trees.
* It introduces randomness by using bootstrap sampling to create multiple datasets for training each tree and randomly selecting a subset of features for each split in the trees.
* Random Forest combines the decision tree formulas with averaging or voting across multiple trees. Each tree's prediction is combined to form the final prediction.

1. **XGBoost (Extreme Gradient Boosting):**

* XGBoost is an ensemble learning algorithm that uses a gradient boosting framework. It builds a strong model by sequentially adding weak learners (typically decision trees) that correct errors made by previous models.
* It optimizes a differentiable loss function by adding new trees that minimize the loss.



1. **LightGBM:**

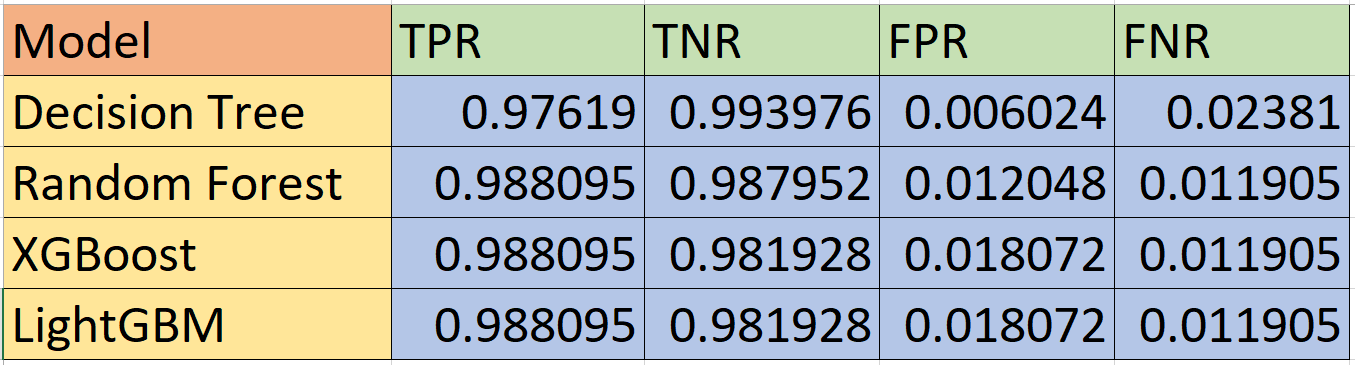
* LightGBM is another gradient boosting framework that focuses on achieving high efficiency and low memory usage. It uses a histogram-based approach for tree building and leaf-wise tree growth strategy.
* LightGBM optimizes the loss function using a gradient-based boosting algorithm similar to XGBoost but with optimizations for speed and memory.
* Similar to XGBoost, LightGBM uses objective functions and gradients specific to the chosen loss function.

**Results & Discussion:**

Evaluating using different models & their KPIs:

****

Model Classification Rates (TPR, TNR, FPR, FNR)

****

LGBM uses a technique called gradient boosting which incorporates regularization to reduce model complexity and prevent overfitting. Random forests don't have built-in regularization, so they can be more prone to overfitting, which can lead to inconsistent results.

For testing on further websites, we then proceeded to use LGBM classifier model

**Testing on different phishing as well as Benign Websites:**

|  |  |  |
| --- | --- | --- |
|  | Phishing Website | Real Website |
| Site Photo |  |  |
| Prediction |  |  |
| Site Photo |  |  |
| Prediction |  |  |
| Site Photo |  |  |
| Prediction |  |  |

**Conclusion**

In conclusion, this project successfully developed a robust phishing detection system by leveraging a combination of URL analysis and HTML content analysis techniques. By extracting lexical features from URLs and computing TF-IDF values for HTML tags, we constructed a comprehensive feature set for classification. Through the utilization of various machine learning algorithms and feature reduction techniques, we achieved a high accuracy rate of 98.8% on a dataset comprising 1600 websites. This demonstrates the effectiveness of the hybrid model in accurately distinguishing between benign and phishing websites, thereby contributing to enhancing cybersecurity measures for internet users.