Name-Based Human Gender Classification using Machine Learning

Barkha A. Wadhvani[1], Shreya Jagani[2], Kreesh Rajani[3], Hardik Magatarpara[4], Darshan Chauhan[5]
[1]*Assistant Professor, School of Engineering, P P Savani University, India*
[2,3,4]*Student, Dept. of Information Technology, School of Engineering, P. P. Savani University, India*
[5]*Director of Durvasa Infotech, India*

Abstract: The origin, gender, and other characteristics of a person can be inferred from his name. A person's name (both given and middle) can be used as a form of identification since it contains linguistic patterns that can be used to determine someone's gender. Numerous applications, including marketing and data analysis, make extensive use of first-name-based gender prediction. This study aims to investigate the accuracy of machine learning algorithms for predicting gender based on first name, taking into account the influence of cultural and historical factors. A sample of 95,024 names from various countries and regions was used to collect data, and a supervised learning model was then developed and trained. Our results demonstrates that machine learning offers a promising approach for gender prediction based on the first name but also highlights the importance of considering cultural and historical factors during algorithm development.

Keywords: Machine learning, Deep Learning, Gender Prediction, Multinomial Naive Bayes, LSTM, Natural Language Processing

1. Introduction

In the fields of Artificial Intelligence, Natural Language Processing, and Natural Language Understanding, gender identification based on a person's name has been a hot topic for more than a decade for several years [1]. Numerous applications, including marketing, social media analysis, and individualized recommendation systems, can benefit from using first names to infer gender [2]. No human being could ever make a 100% accurate prediction of a person's gender based solely on their name, and the difficulty of this endeavour could vary widely across geographic areas. This study examines the application of machine learning techniques to gender prediction based on first names. Using machine learning algorithms, we present a gender prediction model based on the classification of first names. The model is trained using a dataset of first names and their associated gender designations. The dataset was compiled using a variety of resources, such as government databases and online directories. The model derives a number of features from the first names, including the length of the name, the frequency of certain letters, and the presence of particular substrings. Various metrics, such as accuracy, precision, recall, and F1 score, are utilized to assess the performance of our model. Our results demonstrate that the machine learning model outperforms and achieves a high degree of accuracy in gender prediction from first names.

Following is the structure of this paper: In Section 2, we discuss previous studies on gender prediction from first names. The third section describes the dataset utilized in our experiments as well as the feature extraction techniques. Our machine-learning model for gender prediction based on first names is described in Section 4. In Section 5, we describe evolution metrics and outcomes. In Section 6, we conclude the paper and discuss directions for future research in the field.

2. Related Work

In recent years many scientists have tried to analyze the demographic breakdown of a population of Twitter users. Features related to gendered language use, such as the use of gendered pronouns and adjectives, are particularly important for accurate gender identification [3]. SVM and random forests compared for gender identification and found that SVM performs the best overall, but that different models may perform better for specific subsets of the data. The author proposes a multiscale decision fusion approach for gender recognition and combines information from multiple sources and multiple scales, such as facial features and wavelet transforms, to improve the accuracy of gender recognition [4]. To contribute to the development of more accurate and robust methods for gender recognition of pedestrians in surveillance videos, which can have important practical implications in areas such as security and public safety [5]. To identify several future research directions such as exploring more complex linguistic features and incorporating contextual information to improve prediction accuracy [6]. [7] demonstrated the effectiveness of the proposed method in predicting gender based on Indonesian names. It also provides insights into the linguistic characteristics of Indonesian names that are most indicative of gender. [8] analyzes the most informative characters and character combinations for predicting gender in each language. For example, in English, the most indicative characters for male names are "r", "k", and "l", while for female names, they are "a", "e", and "i". In Portuguese, the most indicative characters for male names are "r", "s", and "o", while for female names, they are "a", "e", and "i". Out of all machine learning algorithms, SVM performed the best. Using a convolutional neural network (CNN) model, [9] proposed a way for predicting a user's age and gender based on a single image. This has broad implications in areas including advertising, surveillance, and healthcare. The authors of [10] address gender disparities in academic publishing and promote diversity and inclusion in urban land science research.

3. Dataset and Processing

In this section, we describe the dataset and the pre-processing techniques we applied to it. To accurately predict gender from human first names, we trained our model using a publicly available dataset.

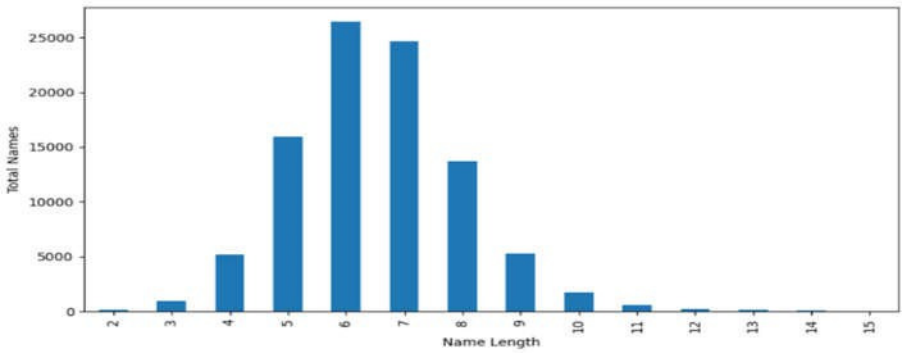


Figure 1. Distribution of Name length

3.1 Dataset

Kaggle's "names_dataset" was utilized [11]. The dataset contains 95,024 rows with various first names, of which 63.77 percent are female and 36.23 percent are male. The dataset consists of two primary attributes: first name and sex. Table 1 displays a sample of five entries from the dataset. The distribution of name length is depicted in Figure 1.

Index	Name	Sex
0	Mary	F
1	Anna	F
2	Emma	F
3	Edward	M
4	David	M

Table 1. Sample Dataset

3.2 Encoding

Encoding is the process of transforming data into a format readily understood and processed by a machine-learning model. Encoding is used to convert the first name data into a numerical format that can be input into a machine-learning model. Raw text data cannot be understood and processed by machine learning models, which is one of the primary justifications for encoding. Machine learning models can process and learn from text data by encoding it into a numerical format.

The count vectorizer [12] is used to convert text data into a numerical representation that can be used as input for a machine-learning model. This method generates a matrix in which each row represents a single data point and each column represents a feature (in this instance, a particular word from the first name data). This method creates a numerical value in the matrix for each occurrence of each word in each data point. The resulting matrix can then be used as input for a machine learning model. The Count Vectorizer is a feature extraction technique that transforms text data into a sparse matrix of token counts, where each row represents a document and each column represents a unique word in the vocabulary [13]. Then, for each document (in this case, each first name), the document is represented as a vector of word counts, where each dimension corresponds to a distinct vocabulary term. As an illustration, the word "Marry" appears in the first name at index 0 in the dataset, and the count vectorizer assigns it the index 74552 in the vocabulary. As "Mary" only appears once in the given name, its tally in the vector representation of the given name is 1. Table 2 shows the result of count vectorizer for 5 sample Names. Encoding is an essential stage in preparing data for machine learning models as a whole. Without encoding, models would not be able to process unstructured text data, nor would they be able to learn from it or make predictions.

Name	Sex	Result of Count Vectorizer		
		Index	Vocabulary Index	Count
Mary	F	1	74552	1
Anna	F	2	30448	1
Emma	F	3	58311	1
Edward	M	4	4802	1
David	M	5	63650	1

Table 2. Result of count vectorizer for 5 sample Names

Encoding of Sentence Piece Byte Pairs Tokenizer is a form of tokenizer employed in natural language processing that acquires a subword vocabulary from a given corpus. Based on the byte pair encoding algorithm, it can handle any form of text, including those written in complex scripts, such as Chinese and Japanese. The tokenizer learns the most prevalent sub-words from the given corpus before segmenting words into sub-words. The output is a collection of sub-words that can represent any given language's term. To optimize the sub-word

vocabulary for a particular use case, the hyperparameters, including vocabulary size and minimal frequency, can be modified. It is frequently employed in machine learning applications like text classification, machine translation, and language modelling. We tokenized and encoded the first names in the datasets using the Sentence Piece Byte Pair Encoding Tokenizer. The tokenizer is trained using the training procedure in conjunction with a list of text files containing the training data and the hyperparameters. Then, the tokenizer's encoding procedure is applied to each first name in the datasets, yielding a list of tokens for each name. Using the join method with a space delimiter, these tokens are combined into a single string and preserved in the lists. The resultant encoded tokens are used as inputs for training and testing gender classification machine-learning algorithms for gender classification. TF-IDF Vectorizer is a text feature extraction technique that generates a matrix of TF-IDF features from a corpus of documents. It functions by transforming unprocessed text into a set of normalized feature vectors based on the term frequency-inverse document frequency metric. The term frequency quantifies the frequency of a word within a document, whereas the inverse document frequency quantifies the rarity of the word throughout the corpus. The TF-IDF Vectorizer algorithm computes the product of these two metrics to produce a metric that captures the relative importance of each word in a document based on its frequency in the corpus. This vectorization process can serve as input to machine learning algorithms for sentiment analysis, topic modelling, and document classification, among other applications.

4. Methodology

In the undertaking of name prediction, we utilised two distinct machine learning algorithms to study and predict based on the characteristics of the names. All of these algorithms operated at the character level and included Multinomial Nave Bayes and Long Short-Term Memory (LSTM). Figure 2 illustrates how these algorithms were utilized in the process of prediction.

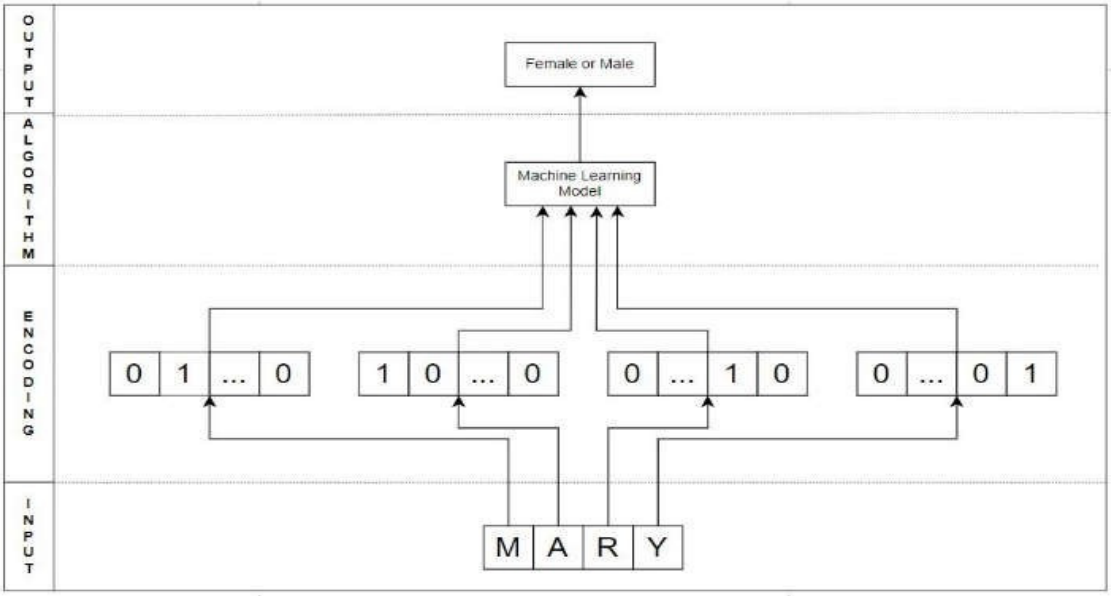


Figure 2. Character level machine learning classification

4.1 Multinomial Naïve Bayes

Multinomial Nave Bayes [14] is a widely employed probabilistic classification algorithm for text classification tasks. The Multinomial Naive Bayes algorithm is used exclusively for text classification tasks in which input data is typically represented as word frequencies. In multinomial naive Bayes, input data is represented as a frequency vector of each word's occurrence in a given document. Based on the frequency vector, the algorithm

then determines the probability of each class. It calculates the probability of each class given the input data using the Maximum A Posteriori (MAP) estimator. The algorithm implies that the frequency of each word in a document is independent of the frequency of other words, a simplification known as the "Naive Bayes assumption". In spite of this assumption, the multinomial naive Bayes algorithm has been demonstrated to be effective for a variety of text classification tasks. Numerous applications, including spam filtering, sentiment analysis, and topic classification, make extensive use of the multinomial Naive Bayes algorithm. A multinomial Nave Bayes classifier is a suitable [11] sentiment analysis technique. It is a straightforward and effective algorithm capable of processing large datasets and high-dimensional input data. However, it may perform poorly on datasets with highly correlated features and may be susceptible to overfitting. A multinomial model's progression is depicted in Figure 3. The stages involved in developing a multinomial model for text classification are outlined in the diagram above. Following the preparation of the dataset, Byte Pair Encoding (BPE) is applied for tokenization. The subsequent step is to use TF-IDF Vectorizer to convert text to numeric values. Finally, a multinomial Naive Bayes classifier is applied to the text data for training and prediction.

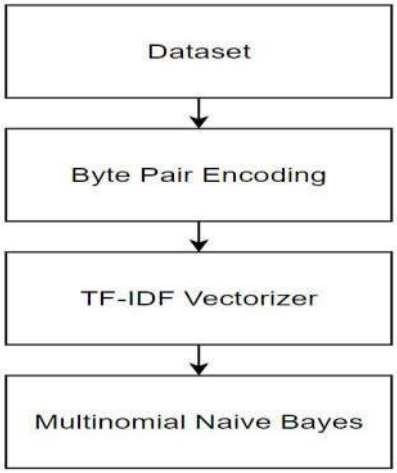


Figure 3. Model based on Multinomial naïve bayes

4.2. Long-Short-Term Memory

Long-Short-Term Memory (LSTM) is a form of recurrent neural network (RNN) capable of handling long-term dependencies and mitigating the vanishing gradient problem [15]. LSTMs are effective for sequential and time-series data due to their ability to store and propagate information over extended periods. In the LSTM architecture , input, output, and forget gates control the information transit throughout the network. The memory cell is responsible for long-term memory maintenance, whereas the input and forget gates regulate the passage of information into and out of the cell. The output gate governs the flow of data to the subsequent network layer. The LSTM architecture is trained using backpropagation through time, in which the error signal is propagated backward through the network in order to update the weights. Utilizing memory cells and gates, the gradient is maintained over time. Common applications of LSTM networks include speech recognition, language modelling, and machine translation. In addition, they have been applied to tasks in natural language processing like sentiment analysis and named entity recognition. The input sequence is fed into the network, followed by a series of LSTM cells that process the input and sustain the memory, as depicted in the LSTM flowchart. The output of the final LSTM cell is then processed by a dense layer to generate the final output. Using a loss function and optimization algorithm, the network is trained to minimize the difference between the predicted and actual output. During inference, the network receives new input and generates output based on the training data's learned patterns. Figure 4 depicts the flowchart illustrating the processes that we followed to implement LSTM. Collecting and preparing the data for processing is the initial step. Byte Pair Encoding is used to tokenize the input text and convert it to numeric data. The tokenized sequences that result are then converted

into integer sequences that the model can process. Integer sequences are padded with zeros to ensure that their lengths are identical. Training the prepared data on a LSTM layer for sequence classification is incorporated into the model architecture.

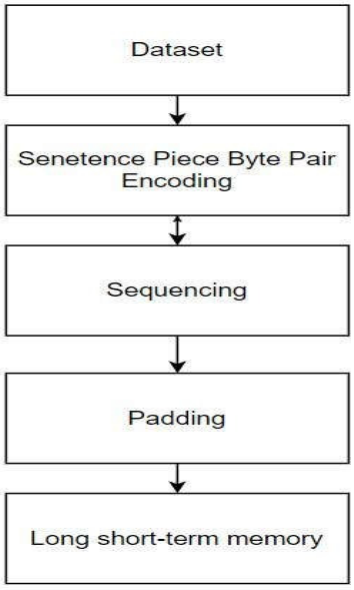


Figure 5. Model based on Long Short-Term Memory

5. Experimentation and Evaluation

In this section, we will discuss the metrics that were used to evaluate the model, and the various results that were obtained.

5.1. Metrics for Assessment

The efficacy of a machine-learning model is measured using evaluation metrics. The three most common evaluation metrics are accuracy, precision, and recall [16]. Accuracy is the degree to which the model accurately anticipates the class of the data points. The precision of a model is determined by calculating the proportion of accurate predictions relative to the total number of predictions made.

$$\text{Accuracy} = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}} \quad (1)$$

Precision quantifies how many of the anticipated positive instances actually occur. It is the ratio between the number of true positives (TP) and the sum of true positives and false positives (FP).

$$\text{Precision} = \frac{TP}{TP + FP} \quad (2)$$

Recall is a statistic that indicates how many true positives have been successfully anticipated. It is determined by comparing the number of positive results (TP) to the combined positive and negative results (TP+FN).

$$\text{Recall} = \frac{TP}{TP + FN} \quad (3)$$

The F1 score is a single number that represents both precision and recall equally well because it is the harmonic mean of the two.

$$\text{F1 score} = \frac{2 * (\text{precision} * \text{recall})}{\text{precision} + \text{recall}} \quad (4)$$

5.2. Outcomes

We used a database that included both male and female names. We have trained models with 80% of the data and test them with the remaining 20%. Following their tokenization with the Sentence Piece Byte Pair Encoding Tokenizer, the names are encoded with the category function. The model is built with the Adam optimizer and binary cross-entropy loss, with precision serving as the metric for success. The training of the model consists of 10 epochs with a batch size of 12. The LSTM-trained model has a 92.37% accuracy rate on the training set. Figure 5 shows the accuracy and validation results of training an LSTM model. The graphic displays two lines, representing the training loss and the training accuracy of the model, respectively. Loss reduces and accuracy increases with more epochs (training iterations). The model is learning to predict the correct output for the input data, which is good. The horizontal black dotted line on the y-axis denotes a 0.55 threshold for model accuracy or loss.

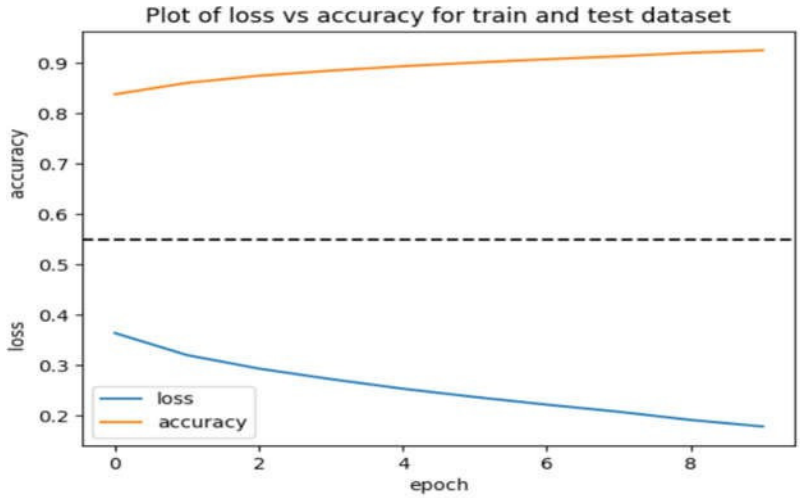


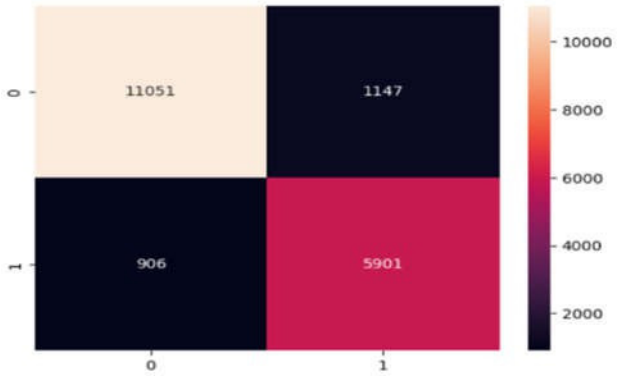
Figure 5. Loss vs Accuracy for training and test dataset

Table 3 compares Multinomial Naïve Bayes and LSTM classification performance. Precision, recall, F1-score, and overall accuracy are used for both M and F classes. LSTM exceeds Multinomial Naïve Bayes in all evaluation measures for both male and female classes. The Multinomial Naïve Bayes model has 72% accuracy, while the LSTM model has 92%. Thus, LSTM is better for categorization.

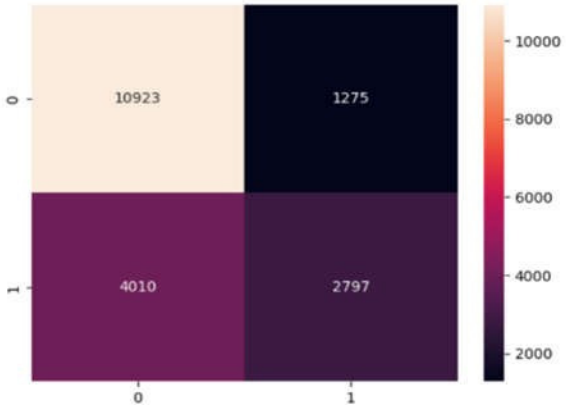
Model	**Precision**		**Recall**		**F1-score**		**Accuracy**
	F	**M**	**F**	**M**	**F**	**M**	
Multinomial Naïve Bayes	0.73	0.69	0.9	0.41	0.81	0.51	0.72
LSTM	0.84	0.92	0.87	0.91	0.85	0.92	0.92

Table 3. Model Outcome

A confusion matrix shows the number of correct and wrong predictions to evaluate a classifier model. We measured the deep learning model's performance using the confusion matrix to compare strategies. Figures 6(a) and (b) show the performance metrics. Results from the LSTM model were shown to be better than those from the Multinomial naive bayes model, as measured by the confusion matrix. When applying the aforementioned encoding word method, all models performed admirably. However, the LSTM model excelled with an impressive 92% accuracy.



(a) LSTM Model



(b) Multinomial naïve bayes Model

Figure 6 Confusion matrix (a) LSTM (b) Multinomial naïve bayes

6. Conclusion

Automated gender classification based on a person's name could aid researchers, policymakers, and businesses in gaining insight into systemic gender bias and making more targeted product recommendations. Gender identification based just on a person's name is another task that is difficult for any human and practically impossible to complete accurately. Furthermore, no single person can possibly master every variety of human language. This research compared the accuracy of two methods for determining a person's gender from their first name: LSTM and the Multinomial naive bayes. Findings from this study indicate that deep learning models can accurately estimate a person's gender from their first name, with the LSTM model providing the best performance (an accuracy of 92%) in this regard. To keep this research's ambition alive, however, we will need faster, more accurate models and more varied datasets that include names for many more language families.

References

[1] Mukherjee, Rajesh, et al. "Human gender classification based on hand images using deep learning." Artificial Intelligence: First International Symposium, ISAI 2022, Haldia, India, February 17-22, 2022, Revised Selected Papers. Cham: Springer Nature Switzerland, 2023.

[2] Ansari, Fatemeh Sajadi, et al. "Classifiers-Based Personality Disorders Detection." Smart Applications and Data Analysis: 4th International Conference, SADASC 2022, Marrakesh, Morocco, September 22–24, 2022, Proceedings. Cham: Springer International Publishing, 2023.

[3] Ikae, Catherine, and Jacques Savoy. "Gender identification on Twitter." Journal of the Association for Information Science and Technology 73.1 (2022): 58-69.

[4] Alexandre, Luís A. "Gender recognition: A multiscale decision fusion approach." Pattern recognition letters 31, no. 11 (2010): 1422-1427.

[5] Cai, Lei, Jianqing Zhu, Huanqiang Zeng, Jing Chen, Canhui Cai, and Kai-Kuang Ma. "HOG-assisted deep feature learning for pedestrian gender recognition." Journal of the Franklin Institute 355, no. 4 (2018): 1991-2008.

[6] Jia, Jizheng, and Qiyang Zhao. "Gender prediction based on Chinese name." In Natural Language Processing and Chinese Computing: 8th CCF International Conference, NLPCC 2019, Dunhuang, China, October 9–14, 2019, Proceedings, Part II 8, pp. 676-683. Springer International Publishing, 2019.

[7] Septiandri, Ali Akbar. "Predicting the gender of Indonesian names." arXiv preprint arXiv:1707.07129 (2017).

[8] Rego, Rosana CB, Verônica ML Silva, and Victor M. Fernandes. "Predicting Gender by First Name Using Character-level Machine Learning." arXiv preprint arXiv:2106.10156 (2021).

[9] Abu Nada, Abdullah M., Eman Alajrami, Ahmed A. Al-Saqqa, and Samy S. Abu-Naser. "Age and Gender Prediction and Validation Through Single User Images Using CNN." (2020).

[10] Chen, Tzu-Hsin Karen, and Karen C. Seto. "Gender and authorship patterns in urban land science." Journal of Land Use Science 17.1 (2022): 245-261.

[11] https://www.kaggle.com/datasets/monukhan/gender-prediction-by-using-name.

[12] Eshan, Shahnoor C., and Mohammad S. Hasan. "An application of machine learning to detect abusive bengali text." In 2017 20th International conference of computer and information technology (ICCIT), pp. 1-6. IEEE, 2017.

[13] Vijayaraghavan, Sairamvinay, Ye Wang, Zhiyuan Guo, John Voong, Wenda Xu, Armand Nasseri, Jiaru Cai, Linda Li, Kevin Vuong, and Eshan Wadhwa. "Fake news detection with different models." arXiv preprint arXiv:2003.04978 (2020).

[14] Harzevili, Nima Shiri, and Sasan H. Alizadeh. "Mixture of latent multinomial naive Bayes classifier." Applied Soft Computing 69 (2018): 516-527.

[15] Yu, Yong, et al. "A review of recurrent neural networks: LSTM cells and network architectures." *Neural computation* 31.7 (2019): 1235-1270.

[16] Handelman, Guy S., et al. "Peering into the black box of artificial intelligence: evaluation metrics of machine learning methods." *American Journal of Roentgenology* 212.1 (2019): 38-43.