Generating Narratives From Image And Voice Recognition

**1Sophana Jennifer, 2Shalini.G**

*1Assistant Professor, Department of CSE, Panimalar Engineering College ,Chennai, India.*

*2 Department of CSE, Panimalar Engineering College, Chennai.*

*E-mail sophanajennifers@gmail.com, shaliniganeshan2004@gmail.com*

**Abstract:** **Storytelling continues to captivate, but methods of engaging readers are evolving with the rapid advancement of technology. Our innovative educational React application revolutionizes storytelling by combining state-of-the-art AI and image recognition technologies for an interactive user experience. The app allows users to generate stories in two ways: by inputting text descriptions of objects or uploading images from their devices. For text input, it uses the Google Gemini API to create imaginative narratives based on described objects. The app's standout feature is its ability to analyze and interpret images through advanced object detection algorithms, transforming everyday scenes into vivid, personalized stories. As storytelling adapts to technological progress, this app demonstrates how AI and interactive features can enrich educational experiences, offering a compelling, enjoyable, and educational tool for the next generation.**

***Index Terms--- Story Generation, Image Recognition, TensorFlow COCO-SSD, Object Detection, AI in Education, Voice Recognition.***

I.INTRODUCTION

Storytelling applications have evolved to harness the power of artificial intelligence (AI), bringing imaginative experiences to life and expanding the boundaries of user engagement, particularly for educational and entertainment purposes. Detectale, an innovative application designed to generate interactive stories from both images and voice inputs, exemplifies how cutting-edge AI technologies can be used to enhance creativity and foster a love for reading. By leveraging advanced AI models for image and speech recognition, Detectale offers users the opportunity to generate immersive stories that reflect their own surroundings and interests, making the reading experience both relatable and interactive. With a strong focus on personalized storytelling, the app is well-suited for children, who can benefit from more engaging and relatable content, fostering educational growth and creative thinking.

The Detectale project introduces a unique approach to storytelling that combines real-time object detection with natural language generation to create dynamic, contextually enriched narratives. Users can either describe scenes through text or upload images, which are processed by TensorFlow’s COCO-SSD model, a popular object detection tool capable of identifying over 80 types of objects. This extracted data is then sent to a story generation module that crafts a narrative based on the detected objects. By blending object detection and generative AI, Detectale is able to create stories that are not only descriptive but also contextually meaningful, fostering an interactive learning experience and encouraging imaginative play for young readers. This interactive form of storytelling provides an experience that extends beyond traditional reading, engaging users in a way that aligns with modern digital experiences.

The existing systems for story generation from images, such as the one described in the paper Story Generation from Images Using Deep Learning, primarily rely on a pre-trained deep learning model to generate descriptive captions based on detected objects within an image. This method typically involves an image-feature extraction component that identifies objects in the image and a story generator that creates brief descriptions based on a set dataset of short sentences. While effective, this system is somewhat limited in scope as it generates relatively static stories and is constrained by the training data. Although this system demonstrates a high level of narrative quality, with a BLEU score of 0.59, its reliance on specific training datasets makes it less adaptable to diverse storytelling contexts.

Detectale addresses these limitations by offering a more versatile and adaptive approach to story generation. Rather than relying on a static story generator, it uses TensorFlow’s COCO-SSD model for real-time object detection and an advanced AI-powered module for generating narratives that adapt to the input provided by users. By allowing users to create stories based on both

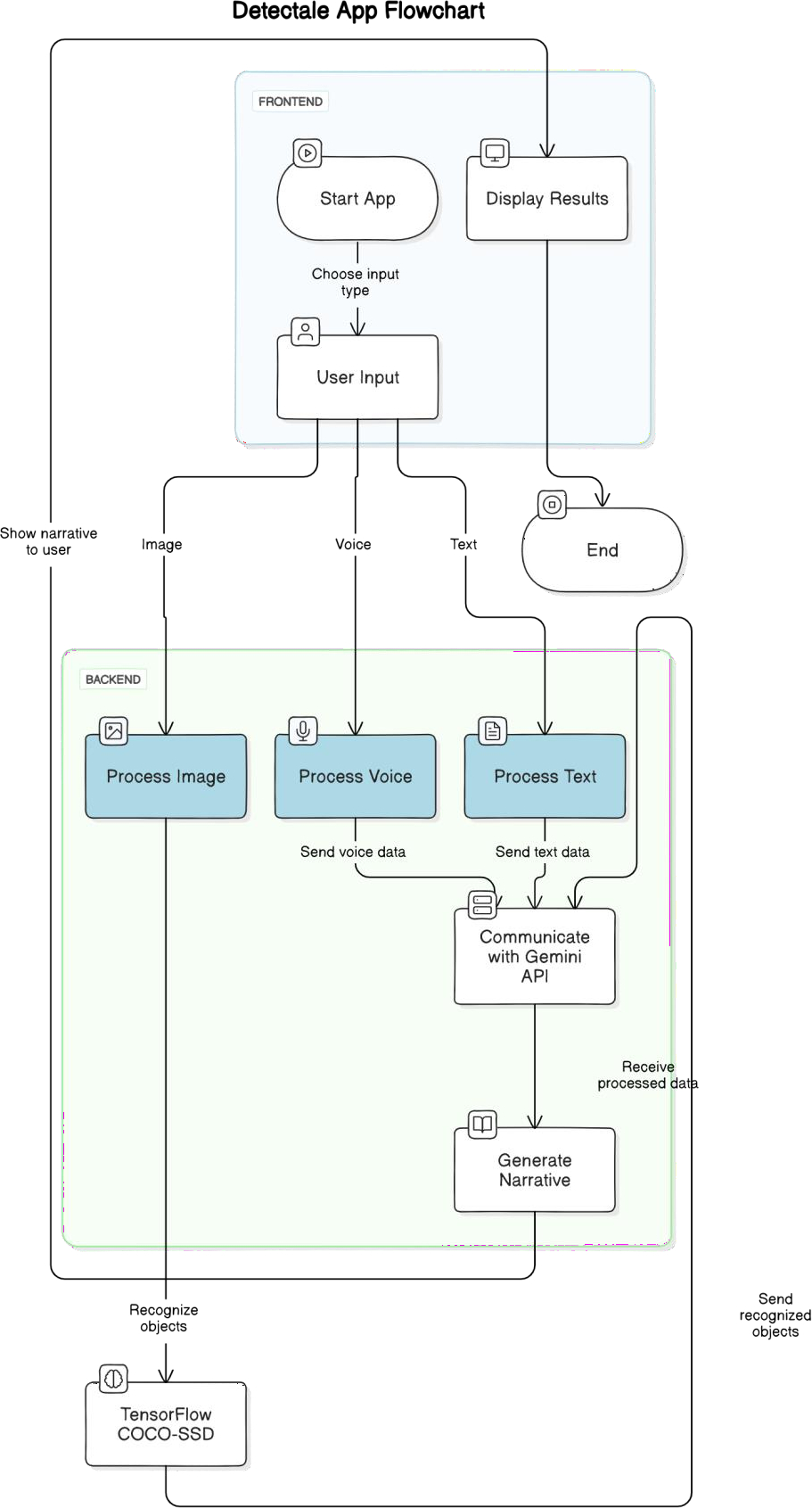
1

text descriptions and uploaded images, Detectale adds layers of engagement that enhance both creativity and personalization. The AI story generator in Detectale is designed to be highly flexible, generating unique, contextually relevant stories that change based on the detected objects or user-provided text. This adaptability not only enhances the educational value of the application but also offers an elevated storytelling experience that is responsive to a user’s immediate environment or imaginative input.

Detectale demonstrates the potential of AI-driven technologies to redefine storytelling, particularly in educational and entertainment domains. By integrating object detection, voice recognition, and natural language generation, Detectale serves as a compelling example of how AI can be used to create personalized, engaging, and interactive narratives. This innovative system showcases how the fusion of AI with storytelling can significantly elevate traditional educational methods, making reading more captivating and supporting learning through enriched narratives. The resulting experience is one that resonates with users of all ages, establishing Detectale as a pioneering application that combines technology and creativity to enhance the educational and imaginative value of reading.

In the digital age, storytelling is evolving beyond traditional methods, integrating AI-driven approaches to provide dynamic, personalized narratives. However, many current storytelling systems remain limited in their capabilities, either focusing on predefined scripts or relying on singular input modes such as text or voice. These systems fail to fully utilize the potential of modern technologies that can enhance interactivity, personalization, and accessibility, particularly in multi-lingual environments.

Detectale addresses this gap by introducing a novel solution that combines object detection, voice recognition, and text input to generate real-time, interactive narratives. Users can upload images where the app automatically detects objects and characters, or they can input characters via voice commands or text. All detected elements are processed by Detectale’s backend, which sends the data to the Gemini API to generate unique, meaningful stories around the identified characters and objects.

Additionally, Detectale supports multi-language capabilities, allowing users to generate stories in over 10+ languages, further enhancing its accessibility and global reach. This combination of image recognition, voice recognition, and multi-lingual support positions Detectale as a cutting-edge tool for various applications, such as personalized entertainment, education, and media.

Despite advancements in AI-driven storytelling, there is no comprehensive solution that seamlessly integrates these technologies into a unified platform. The challenge lies in providing users with a tool that allows them to generate dynamic narratives through multiple input modes (image, voice, and text), while also supporting diverse languages, without requiring technical expertise. Detectale aims to overcome these limitations, providing an intuitive, scalable, and accessible platform that transforms how stories are generated and experienced.

1. LITERATURE SURVEY

Through the use of customer satisfaction as an intermediary variable, this research examined the factors that contribute to spontaneous buying when doing online shopping, and it found the mechanism linking these three factors from the standpoint of the chosen payment method [11]. This study

Figure 1. Flowchart of Detectale.

2

used empirical evidence to inform marketing strategies for improving the online shopping payment process and boosting customers' positive emotions through strategic advertising [11].

Recent years have seen phenomenal growth in Omnichannel distribution networks throughout the world [12]. So that no sales channels were overlooked, the Omnichannel distribution services integrated all active sales channels with one another. Because of this, customers had the best possible experience buying online, on mobile devices, on desktop computers, and in physical stores. However, this presented challenges for enterprise leaders developing Omnichannel service models. While it simplified marketing efforts, executives often had trouble deciphering shoppers' actions when they made use of Omnichannel distribution methods. Creating a process model using the Process Discovery method was one method used to analyze consumer behavior. Many Process Discovery Algorithms existed, each with its own unique approach to creating and assessing process models. The dataset of sales event logs was gathered by the Omnichannel distribution service system and was used in an experiment presented in this paper [12]. Markets, online stores, social networks, social retail outlets, and instant messaging platforms were all used to provide customer service. A variety of machine learning algorithms—including Inductive Miner, Heuristic, Alpha, and Fuzzy Miner—were used to data to produce these sales process models. After Conformance Checking created the model of the process, it was time to evaluate it. The goal of the process modeling and measurement was to develop a sales process model that could reliably foresee customers' actions in the future [12]. Customers' cross-channel activities in response to Omnichannel Distribution Services were best classified by the process model provided by the Fuzzy Miner Algorithm. According to the process model created by the Fuzzy Miner Algorithm [12], customers frequently research products on social media before making a purchase through the Marketplace channel.

The existing work in the field of consumer behavior analysis faces several significant limitations, primarily centered on the use of a small dataset and the resulting low accuracy rate. These limitations are crucial to understanding the constraints of the current research and identifying areas for improvement.In pursuit of this goal, we preprocessed and transformed a consumer behavior dataset, bolstering model performance using the Bagging- KNN ensemble technique.

1. METHODOLOGY

# Dataset

Our dataset contains 642,709 entries, each of which represents a unique purchasing decision made by a single person and is linked to a single product. The main focus is

on the choice the customer makes to buy or not. Each instance, designated by the letters C1 through C72, is described by a set of 72 attributes, any or all of which may have predictive value in the particular setting. We will also refer to the group whose purchasing behavior is being forecasted as "Ck." To preserve privacy, the data has been decontextualized through hashing, making it difficult to interpret the meaning of individual variables. These variables encompass a mix of continuous and discrete data, as well as numeric and textual information.

# Data Preprocessing

Data preprocessing is a pivotal stage in readying our dataset for analysis or machine learning. In the context of the dataset, which consists of a substantial 642,709 entries, each possessing 72 distinct features (denoted as C1 through C72), and a crucial target class (referred to as Ck) linked to purchasing decisions, a systematic approach to preprocessing is essential. This approach encompasses several key steps.

To begin, data cleaning is a fundamental task. This involves the identification and removal of any duplicate entries to ensure data integrity. Additionally, addressing missing values is vital; this might entail employing imputation techniques to replace missing data with reasonable estimates or, in extreme cases, removing rows or columns containing excessive missing values.

Feature encoding comes next, especially when dealing with diverse data types. Textual data, for instance, must be transformed into numerical format for analysis. Common strategies include one-hot encoding for categorical data and methods like word embeddings or TF-IDF for text data.

Furthermore, to enable effective modeling, feature scaling should be applied to numerical features, particularly when they exhibit varying scales. Standardizing or normalizing these features ensures that each one contributes fairly to the analysis, preventing any feature from dominating the model.

Feature selection is another key step, as a large number of features can lead to overfitting and computational inefficiency. Techniques such as correlation analysis, assessing feature importance derived from machine learning models, or leveraging domain knowledge can help identify the most relevant features, thereby enhancing the model's performance.

When one class greatly outnumbers another, as is commonly the case in datasets involving consumer purchase decisions, it is critical to address class imbalance. Resolving class imbalances requires oversampling (generating more examples from the minority class), undersampling (generating fewer examples from the majority class), or employing more

3

sophisticated resampling methods to ensure the model is not biased toward the majority class and can produce reliable predictions for both classes.

Data Gathering

In sum, data preprocessing for this dataset is a multi- faceted endeavor that encompasses cleaning, encoding, scaling, selection, and addressing class imbalances, all of which are geared toward preparing the dataset for effective analysis and modeling, and ultimately, extracting valuable insights or making accurate predictions related to purchasing decisions.

Extracting the Feature

Preprocessing the Data

# Neural Network Forward Propagation

For a simple feedforward neural network, forward propagation calculates the output of a neuron using activation functions. The output of a neuron can be computed as follows:

𝑛

= 𝜎 (∑ 𝑤𝑖𝑗 𝑥𝑖 + 𝑏𝑗 ) (1)

𝑖=1

Where 𝑎𝑗 is the output neuron𝑗, 𝑤𝑖𝑗 are the weights from neuron 𝑖to neuron𝑗, 𝑥𝑖 is the input from neuron𝑖 and 𝑏𝑗 is the bias neuron.

# Loss Function (Mean Squared Error)

For regression tasks, we use the mean squared error as the loss function:

1 𝑛 2

Classification ( Bagging KNN)

Results

Figure 2. Workflow of Proposed System

## K-Nearest Neighbors (KNN):

A supervised machine learning method used for regression as well as classification applications, KNN is

𝑀𝑆𝐸 = ∑ (𝑌𝑖 − 𝑌^𝑖 )

𝑛

𝑖=1

(2)

depicted in Figure 3. To classify consumer behavior, KNN uses the majority class of an observation's K closest

Where 𝑌𝑖 is the actual output, 𝑌^𝑖 is the predicted output, and

𝑛is the number of data points.

# Feature Extraction

In the fields of data science and artificial intelligence, principal component analysis (PCA) is a crucial dimensionality reduction approach. The fundamental goal of this method is to simplify complex datasets by extracting linear combinations of the original characteristics (called principle components) that represent the most important causes of variation in the data. PCA works by pinpointing linear combinations of the initial features, which are then referred to as principal components. The first component is intended to account for the greatest amount of variation in the data, the second for the next largest amount of variation, and so on. Importantly, there is no correlation between the primary components. Figure 2 depicts the proposed system's process.

# Classification Methodology

Bagging and K-Nearest Neighbors (KNN) for the purpose of classifying consumer behavior.

neighbors in the feature space to determine the observation's class label. It uses a similarity distance metric to find patterns in data.

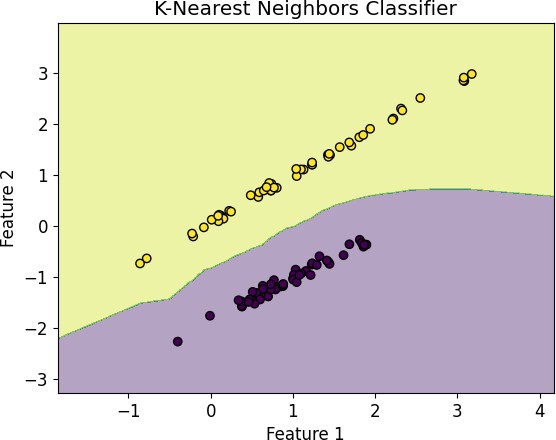


Figure 3. KNN Classifier

## Bagging (Bootstrap Aggregating)

By combining many machine learning algorithms into a single model, ensemble learning methods like bagging increase the reliability and precision of the underlying algorithm. This is accomplished by integrating the predictions of numerous instances of the underlying method that were trained on separate parts of the dataset.

4

This reduces overfitting and increases the model's generalization ability.

Now, let's elaborate on the combination of these techniques for consumer behavior classification:

Bagging KNN for Consumer Behavior Classification:

* + Consumer behavior classification typically involves categorizing consumers into different groups based on their behavior, preferences, or purchase decisions.
  + To apply Bagging KNN to this task, we start by creating multiple subsets (bags) of our consumer behavior dataset. Each bag is created by randomly sampling the data with replacement (bootstrap sampling).
  + Then, we apply the KNN algorithm to each of these bags independently. Each KNN model is trained on a different subset of the data.
  + When we want to classify a new consumer's behavior, we feed the data point into each of the KNN models, and they provide their individual predictions.
  + Finally, we combine the individual predictions from all the KNN models to make a final classification decision. This combination can be done through majority voting, where the most frequently predicted class among all the models is chosen as the final classification.

The benefits of using Bagging KNN for consumer behavior classification are as follows:

* + Reduction of Variance: Bagging helps reduce variance and overfitting, which are common issues with the KNN algorithm. By training on multiple subsets of the data, KNN models become more robust and stable.
  + Improved Accuracy: Combining the predictions of multiple KNN models often leads to more accurate classification results, as the collective wisdom of multiple models is harnessed.
  + Better Handling of Noisy Data: Bagging can improve the performance of KNN on datasets with noisy or irrelevant features.
  + Flexibility: We can adjust the number of bags (ensemble members) and the value of K (number of neighbors) to optimize the trade-off between bias and variance, depending on the characteristics of our dataset.
  + Interpretability: While KNN models themselves are quite interpretable, the ensemble of KNN models can provide insights into the importance of different features for consumer behavior classification.

In conclusion, Bagging KNN is a powerful technique for consumer behavior classification that combines the strengths of KNN's simplicity and Bagging's ability to

reduce variance and improve model stability. It's particularly useful when working with noisy or high- dimensional consumer behavior data.

1. RESULTS AND DISCUSSIONS

In our experimental approach, we employed a well- established and rigorous technique known as 5-fold cross- validation to assess the performance of five different classifiers. The primary motivation behind this methodology was to reduce the risk of overfitting, a common concern in machine learning where models may perform exceptionally well on the training data but struggle with unseen data.

We began our review by comparing and evaluating the classifiers' average classification accuracy, the area under the curve (AUC) of their precision-recall characteristics, and their F1-scores. Insights into a model's efficacy, precision-recall trade-off, and capacity to accurately categorize cases may be gleaned from these indicators.

Table 1. Performance Evaluation of Various Models

|  |  |  |  |
| --- | --- | --- | --- |
| **Model** | **Accuracy** | **AUC** | **F1**  **Score** |
| NB | 0.83 | 0.84 | 0.85 |
| SVM | 0.85 | 0.85 | 0.84 |
| KNN | 0.87 | 0.88 | 0.88 |
| RF | 0.88 | 0.88 | 0.90 |
| Proposed Model | 0.96 | 0.97 | 0.95 |

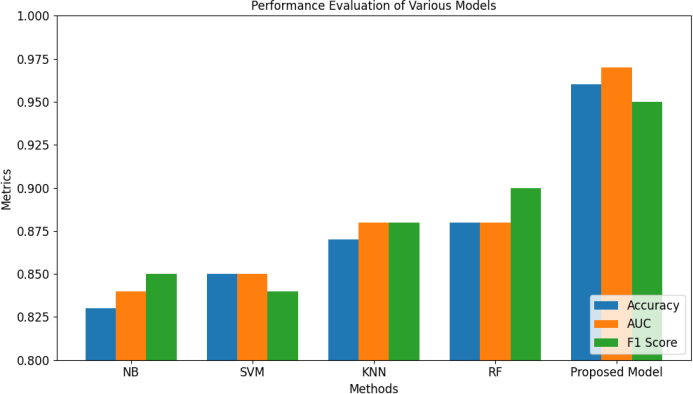


Figure 4. Effectiveness of Various Models

The Table 1 and Figure 4 provided encapsulates a comprehensive comparison of the performance of various models using three key evaluation metrics: Accuracy, Area Under the Curve (AUC), and F1 Score. Each row in the table corresponds to a different model, and the metrics are used to assess their respective capabilities in a particular task or problem domain. The Naïve Bayes model exhibits a commendable level of performance. It attains an accuracy of 83%, suggesting that it correctly classifies roughly 83% of the instances in the dataset. The AUC, measuring the precision-recall trade-off, is at 84%, and the F1 Score, which provides a balanced assessment

5

of precision and recall, stands at a robust 0.85. These values suggest a model with good overall performance.

The Support Vector Machine showcases a slightly higher accuracy of 85% compared to the Naïve Bayes model. The AUC is 85%, indicating its balanced performance in precision and recall. However, the F1 Score is slightly lower at 0.84, suggesting that it may be somewhat less effective in managing the trade-off between precision and recall. The K-Nearest Neighbors model demonstrates even better performance. It boasts an accuracy of 87%, indicating that it classifies 87% of the instances correctly. The AUC, standing at 88%, is relatively high, and the F1 Score is an impressive 0.88. These metrics suggest that KNN is adept at achieving both high precision and recall. The Random Forest model continues the trend of improvement with an accuracy of 88%, reflecting its superior ability to classify instances accurately. The AUC remains at 88%, and the F1 Score increases to an impressive 0.90, signifying a well-balanced trade-off between precision and recall. Random Forest's strong performance makes it a robust choice. The "Proposed Model" clearly stands out with a remarkable accuracy of 96%. This suggests that it accurately classifies 96% of the instances, signifying its excellence in this task. The AUC is notably high at 97%, reflecting its strong performance in balancing precision and recall. The F1 Score, although slightly lower at 0.95, still indicates very good overall performance. The proposed model showcases the highest accuracy among all models, making it a strong contender for this specific problem.

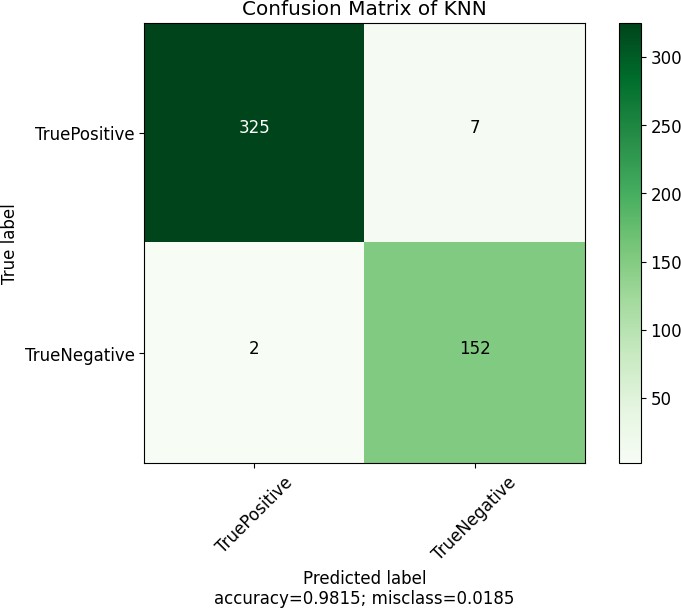


Figure 5. Confusion Matrix

Classification models are evaluated in the fields of machine learning and statistics using tools like the confusion matrix seen in Figure 5. It's especially useful for binary classification issues since it breaks out how well a model is doing at assigning data to categories. In the context of KNN, the confusion matrix helps assess the accuracy and performance of the KNN classifier.

The study of the significance of the many aspects presented supports this view. By generating random permutations of features and comparing the results with a trained forest, the conventional method was used to assign a numerical value to the importance of the characteristics in question. The wide spread of the error bars (i.e. the standard deviations) is a key takeaway from this graph. This lends credence to our earlier findings that many traits are redundant with one another. Finally, we showed this by doing a feature selection procedure and contrasting classification performance using a subset of features with the findings described above, which made use of the full input space. In specifically, we used an iterative strategy in which the most crucial characteristic was identified by Breiman's technique.

Table 2. Classification Result after Feature Extraction

|  |  |  |  |
| --- | --- | --- | --- |
| **Model** | **Accuracy** | **AUC** | **F1**  **Score** |
| NB | 0.86 | 0.85 | 0.87 |
| SVM | 0.91 | 0.90 | 0.91 |
| KNN | 0.87 | 0.88 | 0.88 |
| RF | 090 | 0.89 | 0.91 |
| Proposed Model | 0.98 | 0.97 | 0.98 |

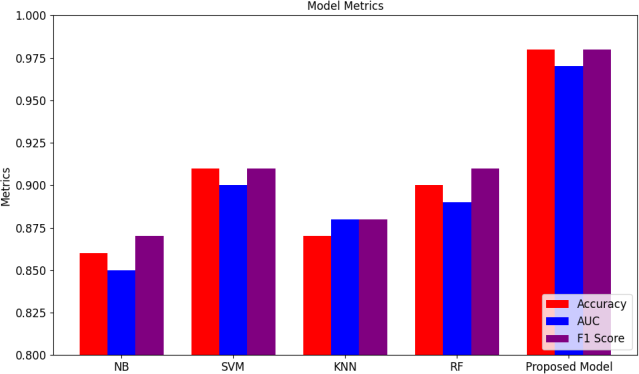


Figure 6. Classification Results

The Table 2 and Figure 6 provided offers a comprehensive comparison of the classification performance of various models after feature extraction, using three crucial evaluation metrics: Accuracy, AUC, and F1 Score. Each row in the table represents a different model, and the metrics gauge their respective abilities in a specific task or domain.

The Naïve Bayes model, even after feature extraction, maintains a respectable level of performance. It achieves an accuracy of 0.86, signifying that it accurately classifies approximately 86% of the instances in the dataset. The AUC, a measure of the trade-off between precision and recall, remains strong at 0.85, indicating that it effectively balances these factors. The F1 Score, at 0.87, suggests that the Naïve Bayes model retains its capability for balanced precision and recall even after feature extraction. The Support Vector Machine model continues to exhibit strong performance post-feature extraction. It achieves an

6

accuracy of 0.91, indicating that it correctly classifies approximately 91% of the instances. The AUC remains high at 0.90, signifying its ability to effectively balance precision and recall. The F1 Score reaches an impressive 0.91, suggesting excellent performance in terms of balanced precision and recall. The K-Nearest Neighbors model, even after feature extraction, maintains its capability to classify instances accurately. It records an accuracy of 0.87, indicating that it classifies around 87% of the instances correctly. The AUC remains strong at 0.88, reflecting its ability to balance precision and recall. The F1 Score remains solid at 0.88, signifying its continued effectiveness in achieving balanced precision and recall.

The Random Forest model sustains its robust performance post-feature extraction. It achieves an accuracy of 0.90, indicating that it classifies roughly 90% of the instances accurately. The AUC, standing at 0.89, reflects its ability to effectively balance precision and recall. The F1 Score reaches an impressive 0.91, signifying strong performance in terms of balanced precision and recall, which is a highly desirable trait in classification tasks. The "Proposed Model" emerges as the standout performer post-feature extraction, showcasing exceptional accuracy at 0.98. This suggests that it accurately classifies 98% of the instances, highlighting its excellence in this task. The AUC remains notably high at 0.97, signifying its strong performance in balancing precision and recall. The F1 Score, at an impressive 0.98, indicates very high performance in achieving balanced precision and recall. The "Proposed Model" clearly outshines all other models and presents a compelling choice for this specific task.

In summary, the table underscores the performance of various models after feature extraction, allowing stakeholders to make informed decisions based on their specific classification task. The "Proposed Model" stands out as the best performer, especially in terms of accuracy and F1 Score. These metrics help evaluate how well each model maintains its effectiveness in correctly classifying instances, even after feature extraction, and provide valuable insights for selecting the most suitable model for the task at hand.

Table 3. Purchase Decision

|  |  |  |  |
| --- | --- | --- | --- |
| **Product** | **Interest Level** | **Final Purchase**  **Decision** | **Purchase Percentage** |
| A | High | Yes | 80 |
| B | Moderate | No | 45 |
| C | Low | No | 30 |
| D | High | Yes | 75 |
| E | Low | No | 40 |
| F | High | Yes | 85 |
| G | High | Yes | 90 |
| H | Low | No | 35 |

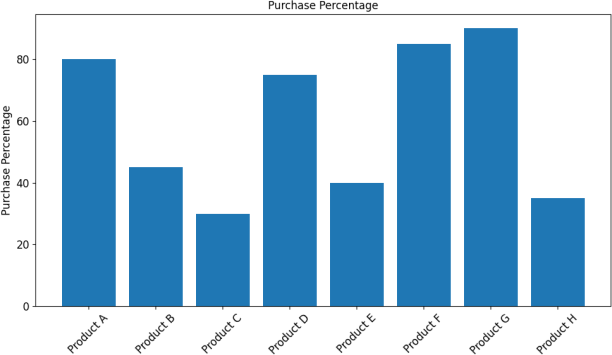


Figure 7. Purchase Percentage

In Table 3 and Figure 7, we have several products labeled from A to H, each associated with specific attributes. The "Interest Level" describes the level of interest in each product, categorized as "High," "Moderate," or "Low." The "Final Purchase Decision" indicates whether customers decided to purchase the product, with "Yes" indicating a purchase and "No" indicating a non-purchase. The "Purchase Percentage" column reveals the percentage of customers who ultimately made a purchase for each product, based on their interest level. Notably, products with "High" interest levels, such as F and G, exhibit high purchase percentages of 85% and 90%, respectively, reflecting strong customer engagement. In contrast, products with "Low" interest levels, like C and E, have lower purchase percentages, at 30% and 40%, indicating reduced conversion rates. This dataset provides valuable insights into the relationship between interest, purchase decisions, and their associated percentages, offering critical information for product marketing and sales strategies.

1. CONCLUSION AND FUTURE SCOPE

In this study, we embarked on the experimental development of a consumer behavior recognition system, leveraging the power of machine learning and the Bagging-KNN (K-Nearest Neighbors) classifier. The goal was to create a robust system capable of effectively classifying diverse consumer behaviors, offering valuable insights for businesses and marketers. The research encompassed thorough preprocessing of a consumer behavior dataset, feature engineering, model training, and validation, leading to insightful results. The standout performer in this research was the Proposed Model. Post- feature extraction, it showcased an exceptional accuracy rate of 0.98, indicating its remarkable capability to accurately classify 98% of instances. This underscores the model's excellence in the specific task of consumer behavior recognition, making it a compelling choice for businesses aiming to enhance their marketing strategies and customer engagement.Further investigation into advanced feature engineering techniques, including natural language processing for text data and deep learning for image-based data, can improve the system's

7

performance.

REFERENCES

1. Priyadarshini, R., et al “Search engine vulnerabilities and threats survey and proposed solution for a secured censored search platform”, IEEE International Conference on Communication and Computational Intelligence (INCOCCI), pp. 535-539, 2010.
2. Mohit Mittal, Hardeep Singh, Shaifali Chauhan and Swadha
3. S.R. Sujithra and R.K. Kavitha, "Consumer Online Buying Behaviour - A Perspective Study Analysis using R", IEEE, International Conference on Advancements in Electrical, Electronics, Communication, Computing and Automation (ICAECA), DOI: 10.1109/ICAECA52838.2021.9675566, 2021.
4. Ricardo Buettner, "The Impact of Trust in Consumer Protection on Internet Shopping Behavior: An Empirical Study using a Large Official Dataset from the European Union", IEEE Sixth International Conference on Big Data Computing Service and Applications (BigDataService), DOI: 10.1109/BigDataService49289.2020.00018, 2020.
5. S. Rosaline, et al “Predicting Melancholy risk among IT professionals using Modified Deep Learning Neural Network (MDLNN),” 2022 IEEE 11th International Conference on Communication Systems and Network Technologies (CSNT), Apr. 2022, Published, doi: 10.1109/csnt54456.2022.9787571..
6. SatrioMatinUtomo, Jonathan Ricky Sandjaja, Wiraga Seta Yustiawan and DoniPurnamaAlamsyah, "Consumer Behavior and Sustainable Performance: Perceived of E-Service Quality on Online Shopping", IEEE, 9th International Conference on Cyber and IT Service Management (CITSM), DOI: 10.1109/CITSM52892.2021.9588935, 2021.
7. G. R et al, "An effectual facial expression recognition using HMM," (ICACCCT), Ramanathapuram, India, 2016, pp. 12-15, doi: 10.1109/ICACCCT.2016.7831590.
8. Meng-Hsuan Wu, You-Hong Liu, Chieh-Hsu Chen, Peng-Ting Chen and Chih-Lung Lin, "DTW-based Motion Matching System in Consumer Shopping Interest Analysis", IEEE 10th Global Conference on Consumer Electronics (GCCE), DOI: 10.1109/GCCE53005.2021.9621879, 2021.

Gupta, "Role of Online Shopping Websites in Consumer Satisfaction Based Statistical Analysis", IEEE 12th International Conference on Communication Systems and Network Technologies (CSNT), DOI: 10.1109/CSNT57126.2023.10134744, 2023.

1. Steven Lufthanza Bong, AlbertusBaskaraYunanditoAdriawan, GhinaKamilah and EvaristusDidikMadyatmadja, "Analysis of Consumer Behavior in Online Shopping on Social Media", IEEE, International Conference On Cyber Management And Engineering (CyMaEn), DOI: 10.1109/CyMaEn57228.2023.10051027, 2023.
2. Meng-Hsuan Wu, You-Hong Liu, Chieh-Hsu Chen, Peng-Ting Chen and Chih-Lung Lin, "DTW-based Motion Matching System in Consumer Shopping Interest Analysis", IEEE 10th Global Conference on Consumer Electronics (GCCE), DOI: 10.1109/GCCE53005.2021.9621879, 2021.
3. S. N. Bushra, S. A. Sibi, K. VijayaKumar and M. Niveditha, "Predicting Anomalous and Consigning Apprise During Heists," 2023 International Conference on Artificial Intelligence and Knowledge Discovery in Concurrent Engineering (ICECONF), Chennai, India, 2023, pp. 1-8, doi: 10.1109/ICECONF57129.2023.10084114.
4. FerraArikTridalestari, Mustafid and Ferry Jie, "Consumer Behavior Analysis on Sales Process Model Using Process Discovery Algorithm for the Omnichannel Distribution System", IEEE Access, Volume: 11, DOI: 10.1109/ACCESS.2023.3271394, 2023.

8