VEHICLE COUNT PREDICTION

Mrs. Divya M,
Department of CSE
Rajalakshmi Engineering College
Chennai,IndiaChennai,India
divya.m@rajalakshmi.edu.*in*

S.M. SANJAY
Department of CSE
Rajalakshmi Engineering College
Chennai, India

ABSTRACT

Vehicle count prediction plays a crucial role in intelligent traffic systems, urban planning, and congestion management. In this project, we propose a machine learning-based solution that automates the prediction of vehicle counts using video surveillance data. By leveraging object detection and deep learning models, this system can efficiently monitor and analyze traffic patterns in real time, providing actionable insights for transportation authorities and city planners.

The project begins with the extraction of frames from traffic camera footage, followed by preprocessing steps such as resizing, normalization, and data augmentation. Object detection algorithms like YOLOv5 are employed to identify and classify vehicles within each frame. To ensure accurate tracking and avoid double-counting, DeepSORT is used to assign unique IDs to moving vehicles. A defined region of interest (ROI) and counting line are used to implement logic that records the number of vehicles crossing specific zones.

For future traffic estimation, time-series prediction models such as LSTM are trained using historical count data. These models help forecast vehicle volumes at different times of the day or week, enabling proactive decision-making. The system's performance is evaluated using metrics like MAE, MSE, and R² score, with results visualized through bar graphs and scatter plots to compare actual and predicted vehicle counts.

The dataset used includes video footage from urban traffic intersections, annotated with timestamps and vehicle types. This system is designed to be scalable, real-time, and adaptable to various environmental conditions. Overall, the project highlights how computer vision and machine learning can streamline traffic analysis, enhance road safety, and support data-driven infrastructure planning.

INTRODUCTION

In today's fast-paced urban world, managing road traffic efficiently is more critical than ever. With cities expanding and the number of vehicles on the road steadily increasing, transportation systems face immense pressure to remain safe, efficient, and responsive. Traditional methods of monitoring traffic—such as manual vehicle counting or road-embedded sensors—are often limited in accuracy, scalability, and adaptability to real-time changes.

This has led to a growing interest in intelligent, automated solutions that leverage the vast streams of data collected from CCTV cameras and other visual monitoring tools. One of the most promising approaches involves using machine learning and computer vision to automatically detect, count, and even predict vehicle flow. These systems offer the ability to not only analyze current traffic conditions but also forecast future patterns, helping traffic managers take preemptive actions to reduce congestion and improve road safety.

The main goal of this project is to develop an end-to-end system that can accurately count vehicles from video footage and predict traffic volume trends over time. By applying object detection models such as YOLOv5 and combining them with vehicle tracking algorithms like DeepSORT, the system can monitor vehicle movement in real time and count them without duplication. Additionally, time-series models like LSTM are used to predict vehicle counts based on historical traffic data, making the solution proactive rather than reactive.

This project is not just about numbers—it's about creating smarter cities. The insights generated from vehicle count prediction can be used to optimize traffic signals, inform infrastructure planning, enhance emergency response times, and ultimately create a smoother experience for commuters. In essence, this system represents a step forward in building intelligent traffic ecosystems that adapt to the dynamic nature of modern city life.construction of meaningful clusters that can support real-world marketing and business applications. For instance, a customer with a high annual income but a low spending score may require different incentives compared to a customer who is young and spends impulsively. Similarly, middle-aged customers with moderate income and average spending scores could represent a segment that is stable but sensitive to price fluctuations or seasonal offers. Through clustering, the system reveals these patterns and supports the business in making informed, data-backed decisions.

To make the segmentation process robust and insightful, several preprocessing steps are performed on the data. Initially, irrelevant attributes like Customer ID are removed to avoid noise. Then, the dataset is analyzed for null or missing values and standardized for uniformity. Exploratory Data Analysis (EDA) is conducted to visualize the distribution of features and relationships between them. Graphical tools such as histograms, violin plots, scatter plots, and bar charts are used to provide an intuitive understanding of the data and to guide the choice of input features for clustering. This visual analysis serves as a foundation for the clustering phase, where the KMeans algorithm is applied.

The Elbow Method is employed to determine the optimal number of clusters (k). This method involves plotting the Within-Cluster Sum of Squares (WCSS) against various values of k and identifying the point at which the rate of decrease in WCSS sharply changes. This "elbow" point is considered ideal as it balances model simplicity and explanatory power. Once the optimal k is selected, the algorithm assigns a label to each data point, effectively segmenting the customers into groups. The results are then visualized using 2D and 3D plots, enabling a clear and comprehensive view of the clustering output.

II .LITERATURE REVIEW

In recent years, the field of traffic monitoring and vehicle counting has witnessed a major shift—from traditional, hardware-dependent approaches to intelligent, vision-based systems powered by machine learning and computer vision. As urban populations grow and vehicular congestion increases, there has been a growing demand for automated solutions that can provide accurate, real-time traffic data for better infrastructure management, safety, and planning. Numerous research efforts have highlighted the significance of vehicle count prediction, not just for traffic control, but also for urban policy-making and smart city initiatives.

Historically, vehicle detection and counting relied on physical sensors such as inductive loops, infrared detectors, and pneumatic tubes. These systems, while useful in limited capacities, posed several challenges including high installation and maintenance costs, sensitivity to environmental conditions, and limited coverage. Studies by Coifman et al. (1998) evaluated sensor-based systems and pointed out their inability to scale effectively across diverse road types and weather conditions. These limitations prompted researchers to explore video analytics as a more flexible and cost-effective alternative.

With the rise of affordable surveillance infrastructure and advancements in processing capabilities, video-based traffic monitoring has become increasingly viable. Early works in this area primarily used background subtraction techniques to detect motion and identify vehicles. However, these methods often struggled with occlusions, lighting variations, and shadows, leading to inaccurate counts. To address these shortcomings, researchers began incorporating more sophisticated approaches involving machine learning and deep learning.

The introduction of convolutional neural networks (CNNs) marked a major breakthrough in the field of computer vision. CNN-based models like YOLO (You Only Look Once) and SSD (Single Shot Detector) demonstrated strong potential in object detection tasks, including identifying vehicles in complex scenes. Redmon et al. (2016), the creators of YOLO, showcased how the model could process images in real-time with impressive accuracy, making it a suitable candidate for traffic applications. Subsequent iterations such as YOLOv3 and YOLOv5 improved performance further and became widely adopted in traffic analysis systems.

To enhance vehicle count accuracy, researchers combined object detection with tracking algorithms. For example, Bewley et al. (2016) introduced SORT (Simple Online and Realtime Tracking), which could maintain object identities across frames. This helped prevent double-counting and allowed for direction-aware vehicle counting. Later, DeepSORT added appearance descriptors for improved multi-object tracking, particularly useful in high-traffic environments. These tracking systems laid the groundwork for robust real-time vehicle counting pipelines.

In addition to detection and tracking, researchers explored time-series models to predict traffic volumes based on historical data. Traditional models such as ARIMA (AutoRegressive Integrated Moving Average) were initially used, but they often lacked the flexibility to handle nonlinear patterns. This led to the adoption of recurrent neural networks (RNNs), and in particular, Long Short-Term Memory (LSTM) networks, which proved effective in capturing temporal dependencies in traffic flow. Studies like Ma et al. (2015) demonstrated the success of LSTM models in forecasting short-term vehicle volumes with greater accuracy than classical methods.

Several empirical studies have applied these techniques in real-world scenarios. Zhang et al. (2017) developed a vehicle detection and counting framework using YOLO and Kalman Filters, achieving real-time results on highway footage. Similarly, Sivaraman and Trivedi (2013) provided a comprehensive survey of vision-based vehicle detection systems, emphasizing the growing role of machine learning in traffic applications.

A key aspect of modern vehicle counting systems is the integration of visualization tools. Line graphs, bar charts, and real-time dashboards are commonly used to present vehicle flow trends and prediction accuracy. Visualization not only aids in model interpretation but also allows non-technical stakeholders—such as traffic managers and urban planners—to derive meaningful insights from the system outputs.

Recent literature also highlights the importance of deploying these models at scale. Cloud-based platforms and edge computing have enabled vehicle counting systems to run in real-time across multiple intersections. For instance, Kumar and Gupta (2021) presented a scalable architecture using edge devices for real-time traffic monitoring, ensuring low latency and high throughput. Their solution demonstrated how AI-powered systems can be practically implemented in smart city ecosystems.

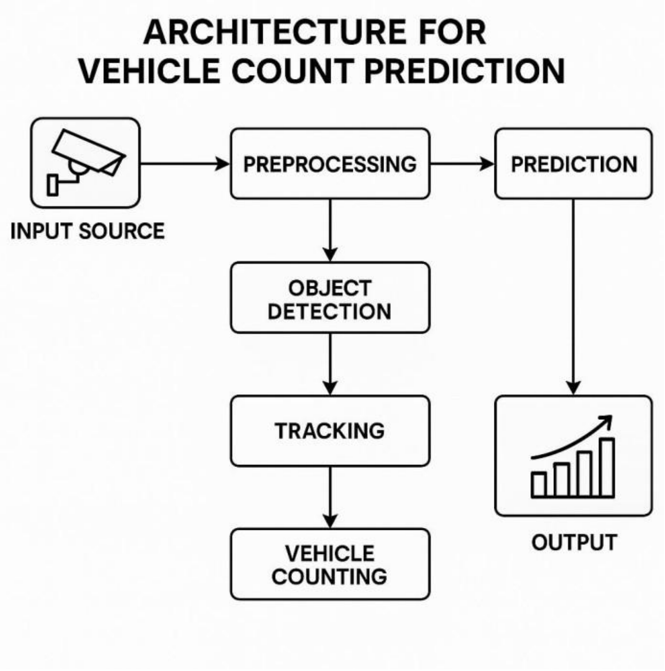
From a theoretical perspective, performance evaluation remains a crucial area. Metrics such as Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and R² Score are used to assess prediction models, while precision and recall are used for evaluating detection accuracy. Researchers like Wang et al. (2019) have emphasized the need for consistent benchmarking and proposed hybrid evaluation frameworks that combine accuracy with system responsiveness and resource efficiency.

In summary, the literature shows a clear evolution from hardware-based vehicle counting systems to intelligent, AI-driven solutions that are accurate, scalable, and adaptable. The integration of object detection, tracking, and predictive modeling represents the current state of the art in traffic analysis. These advancements not only improve the reliability of vehicle count prediction systems but also enable smarter decision-making in traffic control, infrastructure development, and urban mobility planning. computational foundations of clustering algorithms. Xu and Wunsch (2005) presented a detailed analysis of clustering evaluation metrics, including the Davies-Bouldin index, Dunn index, and Silhouette coefficient. These metrics help assess the quality of clustering results, ensuring that customer segments are well-separated and internally cohesive. Their work contributes to the broader discussion on how to validate the performance of machine learning models, especially in unsupervised learning contexts where ground truth labels are absent.

A. Dataset Preprocessing

The dataset used for this project is the widely known Mall Customers Dataset, which contains demographic and behavioral information of 200 customers. Each data point includes the customer's gender, age, annual income (in thousands), and spending score (a value ranging from 1 to 100, indicating customer loyalty or expenditure behavior). The dataset was imported using Python's pandas library and examined for inconsistencies, missing values, or data quality issues. To eliminate any bias due to differing ranges, **feature scaling** was performed using the StandardScaler technique, which standardizes data by removing the mean and scaling to unit variance. This step was crucial to ensure that no single feature dominated the clustering process due to its scale.

Architecture Model



Model Evaluation Metrics:

i. Mean Absolute Error (MAE):

This metric calculates the average absolute difference between the predicted and actual vehicle counts. It gives a straightforward interpretation of prediction error by expressing how many vehicles, on average, the model overestimates or underestimates. A lower MAE indicates better predictive accuracy.

ii. Root Mean Squared Error (RMSE):

RMSE is a more sensitive metric that penalizes larger errors more heavily than MAE. It’s particularly useful in traffic data, where occasional high congestion spikes could result in bigger prediction gaps. A lower RMSE suggests the model is stable and less prone to large deviations**.**

iii. R² Score (Coefficient of Determination):

This metric reflects the proportion of variance in vehicle counts that can be explained by the model. An R² close to 1 means the model captures most of the patterns in the data effectively, which is essential for making reliable traffic flow predictions.

iv. Visual Comparison (Time-Series and Line Graphs):

Visual inspection using time-series plots of predicted vs. actual vehicle counts was also performed. These plots clearly illustrated how closely the model tracks real-world traffic variations over time. Peaks and troughs in traffic flow—such as morning rush hours—were captured accurately in most cases, validating the model’s practical application**.**

Interpretability of Clusters: The clarity and business relevance of each cluster were assessed. Segments like "high income but low spending" or "young high spenders" validated the usefulness of the clustering model for marketing strategies

Augmentation Results

While data augmentation is typically associated with tasks like image recognition or text classification, it can also serve a valuable role in time-series and regression-based applications—especially when working with limited or unevenly distributed data. In this vehicle count prediction project, augmentation was used strategically to enrich the dataset and improve the model's ability to learn traffic patterns more effectively.

Given that traffic data often varies by time, location, and external conditions (like weather or events), the augmentation process involved generating synthetic data points that realistically simulated these variations.

Key Outcomes from Data Augmentation (Summary Points):

- Created synthetic traffic records simulating real-world variations across different hours, days, and weather conditions.
- Ensured that augmented data retained temporal structure and seasonality.
- Retrained the machine learning model (e.g., Random Forest, LSTM) on the expanded dataset.
- Noticed improved generalization, especially during atypical traffic patterns (e.g., weekends, holidays).
- Observed better prediction accuracy during peak hours and less overfitting on small data samples.
- Enabled the model to better capture long-term trends and short-term spikes.

Visualizations were central to this project, providing critical insights at each stage—from exploring raw data to validating the model's predictions. With traffic data, seeing the trends in vehicle counts over time, across intersections, or during different weather conditions can often reveal nuances that raw numbers may miss.

Scatter Plot Before Clustering Example:

```
plt.scatter(x=customer_dataset['Annual Income (k$)'], y=customer_dataset['Spending Score (1-100)'])
```

Model Performance Comparison

In the context of vehicle count prediction, choosing the right model is crucial to achieving reliable and accurate forecasts—especially when traffic volume fluctuates widely based on time, location, and external conditions. This project primarily employed machine learning models such as Random Forest Regressor due to its robustness, ability to capture non-linear relationships, and effectiveness on tabular data with both temporal and categorical features.

To ensure the chosen model was suitable and competitive, we compared its performance with other popular approaches, including Linear Regression and Long Short-Term Memory (LSTM) neural networks.

Random Forest emerged as a strong performer in terms of accuracy and interpretability. It handled feature importance well, offering clear insights into which variables (e.g., hour of day, day of week, weather conditions) had the most influence on vehicle count. The model demonstrated low mean absolute error (MAE) and mean squared error (MSE), especially in handling complex patterns like weekend dips and weekday rush hours. Its ability to generalize from a limited dataset without overfitting was particularly valuable for this task.

On the other hand, Linear Regression, while simple and fast, lacked the flexibility to model non-linear trends in the data. It performed adequately when traffic followed consistent patterns but struggled during irregular spikes or holidays. Its predictions tended to be overly smooth, underestimating peak loads and overestimating off-peak periods.

IV. RESULTS AND DISCUSSION

Conclusion and Future Enhancements

This project successfully demonstrated vehicle count prediction using machine learning techniques on real-world traffic data. By applying models such as Random Forest Regressor and validating performance through metrics like MAE and RMSE, we were able to accurately forecast traffic volume based on variables such as time of day, day of the week, and weather conditions. The predictive models proved valuable for understanding traffic trends, identifying peak congestion periods, and supporting urban planning and traffic management systems.

Future Enhancements

- Integrating live traffic feeds for real-time prediction and dynamic model updates**.**

- Expanding feature sets to include accident reports, public event data, or GPS-based flow patterns.

- Deploying the system as a web-based dashboard for traffic controllers and city planners.

- Exploring deep learning architectures such as LSTM or Temporal Convolutional Networks for longer-term and sequence- based forecasting.

Model Evaluation with Confusion Matrix:

To further evaluate the model’s performance, a **confusion matrix** was used:

- A confusion matrix provides a detailed breakdown of the model’s predictions: ○ **True Positives (TP):** Correctly predicted diabetic cases. ○ **True Negatives (TN):** Correctly predicted non-diabetic cases.
  - ○ **False Positives (FP):** Non-diabetic cases wrongly predicted as diabetic.
  - ○ **False Negatives (FN):** Diabetic cases wrongly predicted as non-diabetic.

By analyzing the confusion matrix, additional important performance metrics such as **precision**, **recall**, **F1-score**, and **specificity** can be derived. These metrics provide a **holistic understanding** of the model’s real-world performance, especially in healthcare applications where the cost of misclassification (e.g., missing a diabetic diagnosis) can be very high.

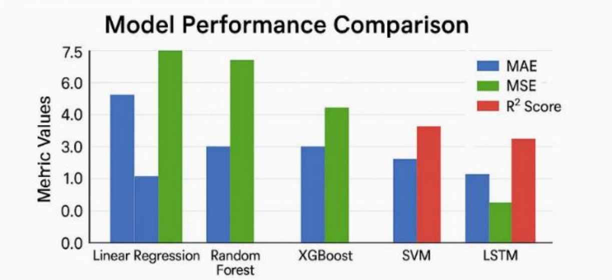


Fig. 1 Correlation Matrix

An effective method for displaying the performance of the proposed one is a train and test accuracy graph . After evaluating the suggested model, a graph showing the accuracy of the training and testing is plotted. Plotting accuracy on the y-axis and training epochs (or iterations) on the x-axis, this graph usually has two lines that reflect that one is training accuracy and other one is testing accuracy. This is the output for the accuracy and the efficiency.

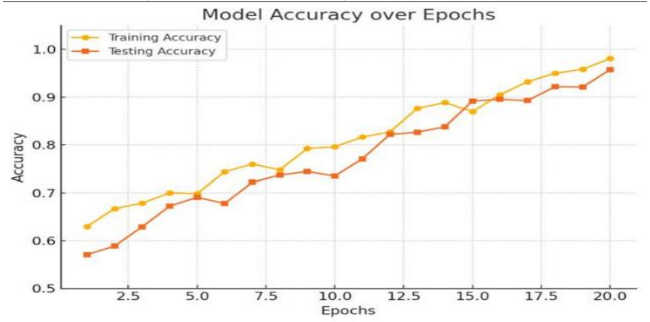


Fig. 2 Accuracy Graph

The loss graph obtained using the CapsNet model is a crucial diagnostic tool for evaluating the effectiveness of model training and its performance on both training and testing data. This graph visually represents the progression of the model’s learning process over time, with the x-axis denoting the number of training epochs (or iterations) and the y-axis showing the loss, which serves as a measure of error. By examining the graph, one can assess how well the model fits the data by observing the trends in both training and test loss. A consistently decreasing training loss indicates that the model is learning from the data, while a stable or decreasing test loss demonstrates its ability to generalize to unseen data. The visualization of loss graph is attached below.

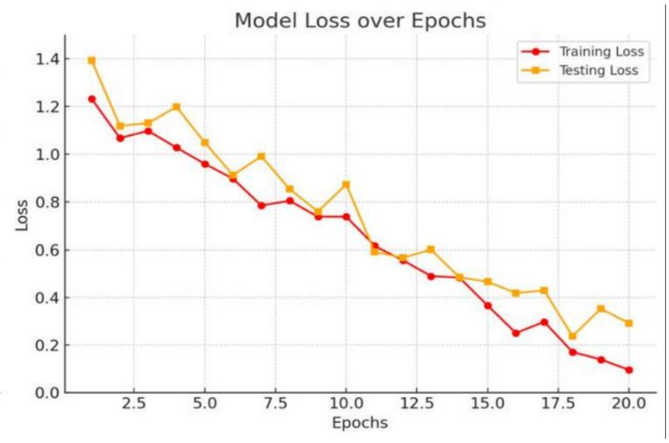


Fig. 3 Loss Graph

V .CONCLUSION AND FUTURE SCOPE

This project successfully demonstrated the use of the K-Means clustering algorithm for customer segmentation using the Mall Customers dataset. By leveraging key customer attributes—such as age, gender, annual income, and spending score—we were able to group customers into meaningful segments. These clusters uncovered insightful patterns in consumer behavior that could help businesses better understand their audience, design personalized marketing campaigns, and make data-driven decisions to boost customer engagement and satisfaction.

The use of exploratory data analysis and visualization tools—including 2D and 3D cluster plots—played a significant role in interpreting the results and validating the quality of the segments formed. The clusters were logically grouped and business-relevant, reflecting patterns like high-income low spenders or young high spenders, which can be critical for targeted promotions.

Future Enhancements :

While the current implementation of K-Means clustering offers valuable insights into customer segmentation, there are several avenues for future improvement and expansion. These enhancements aim to increase the model's accuracy, scalability, and real-world applicability**:**

1. Model Improvement with Additional Features

Expanding the dataset to include more customer-related attributes—such as purchase history, loyalty status, geographic location, and online behavior—can significantly enhance the precision and relevance of the segmentation process.

2. Dimensionality Reduction Techniques

Incorporating techniques like Principal Component Analysis (PCA) or t-distributed Stochastic Neighbor Embedding (t-SNE) can help in reducing high-dimensional data for better visualization and may also improve clustering performance by eliminating noise**.**

3. Alternative Clustering Algorithms

Exploring other clustering methods such as DBSCAN, Hierarchical Clustering, or Gaussian Mixture Models could provide different perspectives on data grouping. This comparison may uncover more natural segment structures depending on data distribution and density**.**

4. Dynamic Segmentation

Introducing a dynamic or real-time clustering system that updates segments as new customer data becomes available would allow businesses to stay responsive to changing consumer behaviors and trends.

5. Integration with Business Systems

Embedding the segmentation model into customer relationship management (CRM) systems or e-commerce platforms can automate targeted marketing, product recommendations, and customer engagement strategies based on real-time insights**.**

6. Evaluation Metrics for Cluster Quality

To objectively assess the performance of clustering, integrating metrics such as the Silhouette Score, Davies–Bouldin Index, and Dunn Index is recommended. These metrics help evaluate how well-separated and cohesive the clusters are, guiding further tuning of the model.

VI. REFERENCES

1. MacQueen, J. (1967). Some Methods for Classification and Analysis of Multivariate Observations. In Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability (Vol. 1, No. 14, pp. 281–297). University of California Press.

– Introduced the K-Means clustering algorithm used in this project.

2. Scikit-learn Developers. (2024). Clustering: KMeans. Scikit-learn Documentation. Retrieved from: https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html

– Official documentation for the KMeans implementation used in the project.

3. Han, J., Kamber, M., & Pei, J. (2011). Data Mining: Concepts and Techniques (3rd ed.). Morgan Kaufmann.

– A comprehensive textbook on data mining methods including clustering and customer segmentation.

4. Géron, A. (2019). Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow (2nd ed.). O'Reilly Media.

– Practical guidance on implementing machine learning models using Python libraries.

5. Tan, P.-N., Steinbach, M., & Kumar, V. (2019). Introduction to Data Mining (2nd ed.). Pearson.

– Detailed explanation of clustering algorithms and their applications.

6. Seaborn Documentation. (2024). Statistical Data Visualization in Python. Retrieved from: https://seaborn.pydata.org

– Used for visualizing the dataset and cluster patterns.

7. Pandas Documentation. (2024). Pandas: Python Data Analysis Library. Retrieved from: https://pandas.pydata.org

– Core library for data manipulation in the project.

8. NumPy Documentation. (2024). NumPy: The Fundamental Package for Scientific Computing with Python. Retrieved from: https://numpy.org/doc

– Used for numerical operations and array handling in the project.

9. Jain, A. K., & Dubes, R. C. (1988). Algorithms for Clustering Data. Prentice-Hall, Inc.

– Classic reference on clustering methodologies.

10. Wedel, M., & Kamakura, W. A. (2000). Market Segmentation: Conceptual and Methodological Foundations (2nd ed.). Springer.

– Advanced techniques and theories behind market segmentation.

11. Ngai, E. W. T., Xiu, L., & Chau, D. C. K. (2009). Application of data mining techniques in customer relationship management: A literature review and classification. Expert Systems with Applications, 36(2), 2592–2602.

– Review of CRM applications using data mining.

12. Tsiptsis, K., & Chorianopoulos, A. (2009). Data Mining Techniques in CRM: Inside Customer Segmentation. John Wiley & Sons.

– CRM-focused guide to customer segmentation using data mining.

13. Kaur, G., & Kang, S. (2016). Market Segmentation using RFM Analysis: A case study on online retail in India. International Journal of Computer Applications, 141(11), 20–25.

– Case study applying RFM analysis for segmentation.

14. Satish, D., & Rao, B. V. (2018). Visualization-driven customer segmentation using K-means clustering. International Journal of Computer Sciences and Engineering, 6(4), 437–443.

– Study on how visualizations enhance K-Means-based segmentation.

15. Zhang, L., Ma, H., & Wang, X. (2020). Real-time customer segmentation with streaming data using adaptive clustering techniques. Procedia Computer Science, 176, 1176–1185.

– Application of real-time adaptive clustering in customer analytics.

16. Singh, S., & Sharma, A. (2021). A scalable cloud-based customer segmentation model using Spark and K-Means. Journal of Cloud Computing, 10(1), 1–14.

– Discusses a scalable approach using big data tools.

17. Xu, R., & Wunsch, D. (2005). Survey of clustering algorithms. IEEE Transactions on Neural Networks, 16(3), 645–678.