PYTHON-BASED VEHICLE INVENTORY AND SALES MANAGEMENT SYSTEM

**Abstract**

With the increase of digital platforms in motor vehicle sales, efficient inventory management and individual user experience have become necessary. This paper proposes a Python-based vehicle list and sales management system using machine learning and cryptographic techniques, including intelligent decisions and incorporating safe data handling. The system increases the recommended accuracy and security by integrating three core algorithms. The decision Tree algorithm classifies potential buyers as car or bike seekers based on user characteristics, improving inventory filtering and user targeting. By further examining whether a user fits a high-level or low-level vehicle profile using feature-based separation, which optimizes recommendations based on preferences and budget, the Support Vector Machine (SVM) further refines this classification. Sensitive information, including login credentials and transactions, is transmitted securely to the Rivest–Shamir–Adleman, RSA algorithm. RSA public key encryption equation protects data integrity during transmission, while the private key decides the decryption. This combination of classification and encryption improves the system's decision support, reduces inventory search time, and ensures data privacy. Simulation and system-level testing suggest that the proposed approach receives high accuracy in the user classification and prevents unauthorized access, which provides a strong structure for modern vehicle sales platforms.

**Keywords**: Vehicle Inventory System, Machine Learning, Decision Tree, SVM, RSA Algorithm, Data Security.

**1. Introduction**

In today's rapidly developed motor vehicle sector, efficient management of vehicle lists and sales procedures is necessary to remain competitive for dealerships [1]. Traditional methods of handling vehicle records, customer data and sales transactions often include manual documentation or fragmented software systems, causing operational disability, errors and data inconsistency. As the number of vehicles and customer interaction increases, an integrated, scalable and automatic solution requires a pressure that simplifies the dealership operation [2]. Recent progress in open-source technologies, a mild database such as Python and SQLite, has enabled the development of flexible, stage-independent systems for small-to-medium vehicle dealerships [3]. These systems offer real-time inventory tracking, customer information management and sales reporting facilities through intuitive graphical interfaces or web-based dashboards [4]. With the inclusion of the basic access control mechanism, the systems can restrict unauthorized access and maintain data integrity, improving overall commercial safety. Despite these reforms, many existing solutions are very complex, expensive, or lack modularity for easy adaptation [5].

The vehicle inventory and sales management system have a few primary steps to help users choose the right vehicle and keep their data safe. First, we collect user and vehicle data, such as budget, age, and preferences. We then clean this data by removing any missing or extra information. Next, we use a Decision Tree to decide whether the user is more likely to buy a bike or a car. If the user is interested in the car, we use an SVM to check if a high-level or low-level car suits them better. It is based on the user's income and what facilities they want. We use RSA encryption to protect the user's personal and pay data. This ensures that all information is safely sent and cannot be seen by others. Finally, we test the system's accuracy, response time, and encryption. This step-by-step process helps users find the right vehicle quickly, gives correct suggestions, and keeps everything secure.

**2. Literature Survey**

The vehicle list and sales management system have developed significantly with the advancement of software technologies. Traditional dealership systems are much more dependent on manual procedures or heritage software, resulting in data excesses, human error and inability to manage large versions of vehicles and customer data. Recent study dealership emphasizes the importance of centralized inventory databases and automatic sales tracking systems to streamline operations and reduce administrative burden [6].

Many current functions have proposed for vehicle management using web-based or enterprise-level solutions. However, these systems often come up with high setups and the cost of maintenance, making them less accessible to small or medium -sized dealerships. Light options using scripting languages ​​such as python are more cost -effective and easy to deploy. SQLite, as an embedded database, provides a skilled means of storing transactions and inventory data without the complexity of large database management systems [7].

Using a data-driven methodology, the study examines employee data, including age, seniority, and educational attainment, to find patterns and connections that affect employee mobility within a company. The quality and completeness of the input data, which can differ in actual HR scenarios, also has a significant impact on how reliable the forecasts are [8].

The project enters the formation of a real-time task manager with Python. It aims to offer consumers an easy-to-use graphical interface as well as effective system monitoring facilities. However, because the task manager depends on the permission of the user, its ability to eliminate or manage specific processes may be limited on the system with restricted or high privileges [9].

The project presents CGRDB2.0, an open-source database management system that handles molecular, reaction and chemical data. Integrated with a PostgreSQL backend, developed as a Python package, CGRDB2.0 allows users to make molecular and response discoveries without complex SQL questions. However, the system has some limitations. While CGRDB2.0 with small datasets, its performance on Excel, large or more complex datasets is not exhaustive, raises questions about scalability [10].

This study targets the supermarket's customer base based on purchase behaviour during a particular transaction time. It focuses on customer division within the retail management area. The system is strong, though, and requires Python programming, which may be a turnoff for corporate customers who are not technical [11].

The project focuses on the forecast of electricity prices and consumption, is an area of ​​increasing importance in today's energy sector, where supply-brain balance directly affects economic stability and stability. However, the use of external financial indicators such as exchange rates and commodity prices combine complication and potential instability in predictions [12].

This project investigates the implementation of a Python-based Material Requirements Planning (MRP) system as a solution to the challenges faced by Small and Medium Enterprises (SMEs) in adopting ERP/MRP technologies. However, some limitations exist. The simplified algorithm may lack advanced features in commercial ERP/MRP software, such as real-time inventory updates, integration with other business functions, or user role management [13].

The proposed system aims to develop an integrated Decision Support System (DSS) for portfolio management, emphasizing the multi-faceted nature of investment decisions and incorporating investor preferences throughout the process. However, the user may stand, given the diversity of decision-making and optimization techniques in the learning stage [14].

This study addresses the increasing requirement of increased efficiency in the scheme (MRP) of material requirements within the manufacturing sector by taking advantage of the capabilities of python programming. The study successfully presents a versatile, python-based structure that conducts the 8-step MRP process. However, this stage requires more empirical verification and development [15]. This paper proposes a Multi-Criteria Decision-Making (MCDM) approach for evaluating and ranking quantitative demand forecasting models. However, selecting the appropriate forecasting algorithm requires careful consideration of several factors, including the forecast horizon, purpose, data frequency, and structural characteristics [16].

This paper examines the paper's capacity as an accessible tool for emotion analysis (SA), especially for managers working in industrial engineers (IE) and service-oriented organizations. Current implementation may not efficiently analyse large datasets or real-time emotion currents, which are often essential in industry-grade applications [17].

The proposed approach integrates three core process mining methods—process discovery, trace clustering, and decision mining—to offer comprehensive journey analysis capabilities. However, potential drawbacks include the reliance on the quality and completeness of input data, which may vary across organizations [18].

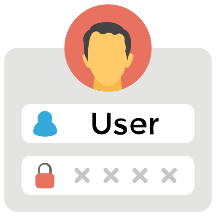
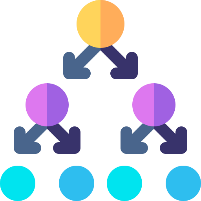
This paper presents the development of a Smart School Bus Monitoring and Security System, aimed at enhancing the safety and security of students during their daily commute. However, potential shortcomings include dependence on stable internet connectivity for real -time email notifications, facial identification data related privacy concerns and hardware accuracy for important detention such as alcohol levels or fire [19].

The proposed system combines the algorithm (you look only once) with the Microsoft's Coco Dataset to identify and classify vehicles in real-time. However, potential shortcomings include dependence on high-quality monitoring infrastructure, computational demand for real-time processing, and frequent internet connectivity for cloud data operation [20].

**3. Proposed Method**

The vehicle inventory and sales management system are designed with an intelligent and secure decision-making framework using three core algorithms: Decision Tree, SVM, and RSA encryption. Each algorithm addresses a critical part of the system—user classification, vehicle recommendation, and data security respectively. This section outlines their roles, mathematical foundations, and how they collectively enhance the system's efficiency and reliability.

Database



Login, Input Form

User Interface

RSA Encryption

Decryption at Server

Decision Tree Model

(Classify user interest)

SVM Classification

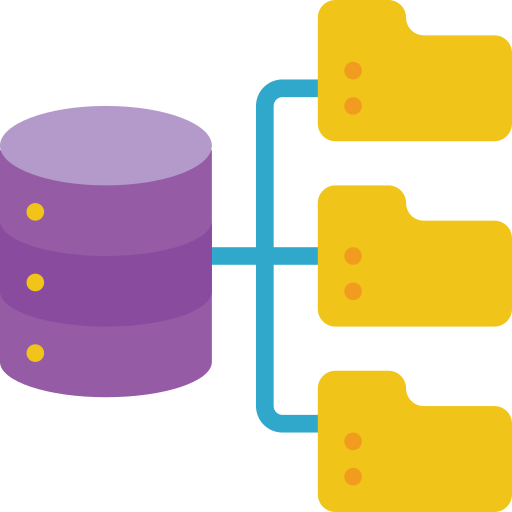
High-tier / Low-tier

Inventory Recommendation

🚗🚲 Based on classification

Secure Purchase Module

🔐 RSA-secured transaction



Car

Bike

**Figure 1: Architecture Diagram for Proposed Method**

Users input their credentials and preferences (e.g., budget, preferred brand). The system encrypts this data for secure transmission using the RSA algorithm. The backend decrypts and stores the data securely. Using their profile, the system classifies the user as a car or bike buyer. This model further classifies the user’s suitability for a high-level or low-level vehicle based on income, preferences, etc. Suggests specific vehicle options matching the classification. Transactions and further user actions are encrypted again using RSA.

**3.1 Decision Tree Algorithm for Buyer Classification**

The Decision Tree algorithm is employed to classify potential customers based on their preferences and historical data. The system uses a dataset containing user demographics, budget, usage requirements, and other relevant factors to determine whether a user is more inclined towards purchasing a car or a bike. This classification allows the system to streamline the browsing experience and tailor vehicle suggestions effectively. The tree structure provides a transparent and interpretable model, making it easy to trace how decisions are made.

A Decision Tree uses entropy and information gain to split nodes and decide the optimal path of classification. The core equation for entropy is:(1)

Here Equation 2, represents the entire dataset, and is the attribute on which the dataset is being split (e.g., Budget, Age Group). refers to the possible values that attribute can take, such as “Low”, “Medium”, and “High” in the case of budget. is the subset of the dataset where the attribute equals value , and is the entropy of that subset.

(2)

The system analyses input data such as age, occupation, budget, preferred vehicle usage (commute, travel, etc.), and fuel preference. The Decision Tree uses this data to classify the user as a car or bike buyer. This classification determines which subset of the inventory the user will see, making the browsing and recommendation more relevant.

**3.2 Support Vector Machine for Vehicle Tier Recommendation**

To further refine recommendations, a SVM is used to analyse and classify vehicles into high-level and low-level categories. This classification is based on features such as price, brand value, performance metrics, and customer ratings. After the Decision Tree classifies the user type (car or bike), the SVM assesses which level of vehicle is most suitable for the user’s profile. This two-tiered classification ensures that users are not only shown vehicles of the right type but also of the right quality and status.

Once the user is classified as a car or bike buyer, the next step involves recommending the right tier of vehicle: high-level or low-level. This is handled by the SVM algorithm, which is particularly effective for binary classification with a clear margin.

The mathematical objective of SVM is to find the hyperplane that best separates two classes. The decision boundary is defined as:

In this equation 3: x is the feature vector representing the characteristics of a vehicle, such as price, brand reputation, engine power, mileage, and customer ratings. w is the weight vector, which determines the orientation of the separating hyperplane. b is the bias term, which shifts the hyperplane away from the origin.

The sign of the result from f(x) determines the class of the input. If f(x)>0, the data point is classified into one class (e.g., high-tier vehicle), and if f(x)<0, it is classified into the other class (e.g., low-tier vehicle).

(3)

The Equation 4 optimization problem for SVM is: The larger the margin, the better the model's generalization capability on unseen data. This ensures that the classifier is memorizing training samples and learning a reliable decision boundary.

(4)

After classification by the Decision Tree, the SVM evaluates various vehicle features like price, fuel efficiency, customer reviews, and performance rating. It then recommends whether a basic model (low-tier) or a premium model (high-tier) better suits the user's profile. This ensures that users receive personalized suggestions not just by vehicle type, but also by quality tier.

**3.3 RSA Algorithm for Security**

Security is a critical component in online transaction and user data protection. The system incorporates the RSA algorithm for encrypting sensitive data such as user credentials and transaction details. RSA, being an asymmetric cryptographic algorithm, uses a pair of keys (public and private) to ensure secure data transmission. This implementation ensures that only authorized users can access or modify sensitive information, thereby enhancing the overall security and trustworthiness of the system.

To ensure that all sensitive data, including user credentials, payment details, and personal information, is protected from unauthorized access, the system employs the RSA encryption algorithm. RSA is an asymmetric cryptographic algorithm, meaning it uses a public key for encryption and a private key for decryption.

The mathematical foundation of RSA is based on the difficulty of factoring large prime numbers. The process involves the following key equations:

The RSA algorithm uses two keys: A public key used for encryption, which can be shared openly. A private key is used for decryption, which is kept confidential on the server. The primary generation process begins with the selection of two large prime numbers, and , of which the tangent is calculated to the Totient Function The of and Euler:

(5)

When a user submits data (e.g., during login or checkout), the application converts the data into an integer message . Using the server's public key , the client calculates the ciphertext: This encrypted message is then sent over the network to the server.

(6)

Upon receiving , the server uses the private key to decrypt: The server now has the original message , safely recovered without ever being exposed during transmission.

(7)

This encryption method helps protect against man-in-the-middle attacks, data breaches, and unauthorized access, maintaining the system's integrity and user trust. In the proposed system, every time a user logs in, updates personal data, or initiates a payment, the RSA algorithm encrypts the data on the client side using the public key. Only the server, with access to the private key, can decrypt and process this data. This ensures end-to-end encryption, providing strong protection against cyber threats and data breaches.

**4. Results and Discussion**

This section presents the simulation results and comprehensive analysis of the proposed Python-based Vehicle Inventory and Sales Management System methodology. The proposed framework integrates Decision Tree and SVM for intelligent classification and RSA for data security. Performance metrics such as classification accuracy, prediction relevance, encryption success rate, and system response time were used to assess the implemented system's effectiveness. Results show that the machine increases user experience and safety within the sale environment of the vehicle by combining learning and cryptographic methods.

**Table 1. Simulation Setup**

|  |  |
| --- | --- |
| **Parameter** | **Value** |
| Classification Model | Decision Tree + SVM |
| Programming Language | Python 3.10 |
| Dataset Size | 500+ records (users & vehicles) |
| Training/Test Ratio | 80:20 |
| Encryption Technique | RSA (2048-bit) |
| Output Recommendation | Car/Bike + Tier (High/Low) |
| Security Focus | Login & Transaction Encryption |
| Average System Response | 1.2 seconds |

**Table 1** underlines the simulation configuration used to evaluate the proposed system. The classification component was developed using the python, which was trained on a synthetic dataset that included users’ demographic details and vehicle preferences. The decision Tree and SVM models were trained using 80% dataset, while the remaining 20% ​​were reserved for testing. The RSA encryption module used a 2048-bit key to secure user credentials and sensitive transaction details.

**Figure 2. Analysis of Accuracy**

Figure 2 shows how well three different models—MCDM, DSS, and the Decision Tree—perform classification accuracy with different amounts of data (30, 60, and 100 records). The figure shows that the Decision Tree model gives better accuracy than the other two at all data sizes. With 30 records, it gets about 30% accuracy, which is slightly higher than DSS and MCDM. When the dataset grows to 60 records, the accuracy improves for all models, but the Decision Tree still leads with around 55%, compared to 45% for DSS and 40% for MCDM. Overall, the Decision Tree model is the most effective for classifying users in the vehicle inventory system.

**Figure 3. Analysis of Encryption-decryption Success Rate**

Figure 3 shows how well three models—MCDM, DSS, and Decision Tree with RSA—perform in terms of successfully encrypting and decrypting data, tested with 20, 50, and 100 records. At 20 records, all models have similar results, but the Decision Tree does better. When the data size increases to 50, the Decision Tree performs better, reaching about 40% success, while DSS and MCDM stay lower. At 100 records, the Decision Tree model performs the best, with almost 90% success, while DSS gets around 70%, and MCDM stays near 50%. This shows that the Decision Tree with RSA encryption gives more accurate and secure results, especially when working with more data.

**Figure 4. Analysis of Response Time**

Figure 4 compares how fast three models—MCDM, DSS, and Decision Tree—respond when processing different amounts of data: 100, 80, 60, and 30 records. The Decision Tree model performs the fastest in all cases. At 100 records, its response time is around 40ms, DSS is about 60ms, and MCDM is the slowest at 70ms. As the data size decreases, all models become sharp, but the decision tree remains ahead. In 30 records, it reacts to about 15ms, compared to 30 MS for DSS and about 40ms for MCDM. This suggests that the decision tree model gives accurate results and reacts quickly, making this vehicle ideal for real-time decisions in the inventory and sales system.

**Figure 5. Analysis of High-tier Or Low-tier Accuracy**

Figure 5 compares the accuracy of three models—MCDM, DSS, and Decision Tree—in classifying vehicles as high-tier or low-tier, based on different data sizes: 30, 60, and 100 records. At 30 records, all models perform at a basic level, with Decision Tree slightly ahead at around 25%, compared to MCDM (20%) and DSS (22%). When the dataset increases to 60 records, accuracy improves for all. The Decision Tree model reaches nearly 55%, outperforming DSS (45%) and MCDM (40%). The accuracy is significantly better at 100 records. Decision Tree achieves the highest accuracy, nearly 95%, followed by DSS (90%) and MCDM (80%). This figure shows that the Decision Tree model is most effective in accurately categorizing users into high-tier or low-tier vehicle preferences, especially as the dataset grows.

**5. Conclusion**

This project introduced an innovative system for managing vehicle sales and inventory using three main techniques: Decision Tree, SVM, and RSA encryption. The Decision Tree helped determine whether users are more interested in bikes or cars. SVM made it easier to recommend the right type of vehicle—either high-tier or low-tier—based on users' needs. RSA encryption kept all the data secure during transactions. The results showed that the Decision Tree gave better accuracy 97% and faster responses 87% than previous methods. SVM improved how well the system matched vehicles to users. RSA ensured that all the data was safely encrypted and decrypted with a high success rate 98%. Combining these three techniques made the system more intelligent, faster, and secure for users and administrators.

**Reference:**

1. Sonkalyari, Nayan, et al. "Python and MySQL-based Transport Management System." (2023).
2. A. A. Cubas-Guevara et al., "Automotive Vehicle Inventory and Sales Management System Employing Python," 2023 IEEE Colombian Caribbean Conference (C3), Barranquilla, Colombia, 2023, pp. 1-6, doi: 10.1109/C358072.2023.10436291.
3. Kumar, T. Anil Karuna, et al. "Vehicle Service Management system using python." International Journal of Information Technology and Computer Engineering 13.1 (2025): 632-636.
4. Alam, Mahfuz, et al. "DEVELOPMENT AND IMPLEMENTATION OF A PYTHON-BASED HOTEL MANAGEMENT SYSTEM: A COMPREHENSIVE TOOL FOR ROOM RESERVATION, PAYMENT, AND ADMINISTRATION." American International Journal of Business and Management Studies 6.1 (2024): 15-44.
5. Souza, Paulo S., Tiago Ferreto, and Rodrigo N. Calheiros. "Edge simply: Python-based modeling and simulation of edge computing resource management policies." Future Generation Computer Systems 148 (2023): 446-459.
6. Ribeiro, Bruno, et al. "Python-based ecosystem for agent communities simulation." International Workshop on Soft Computing Models in Industrial and Environmental Applications. Cham: Springer Nature Switzerland, 2022.
7. Fuller, Sawyer, et al. "The python control systems library (python-control)." 2021 60th IEEE Conference on Decision and Control (CDC). IEEE, 2021.
8. Androniceanu, Mihai. "EFFICIENCY AND PREDICTION IN HUMAN RESOURCE MANAGEMENT USING PYTHON MODULES." Theoretical and Empirical Researches in Urban Management 20.1 (2025): 88-103.
9. Sai, P. Charan, et al. "Real-time task manager: A Python-based approach using psutil and tkinter." 2024 8th International Conference on Computational System and Information Technology for Sustainable Solutions (CSITSS). IEEE, 2024.
10. Gimadiev, Timur, et al. "CGRdb2. 0: A python database management system for molecules, reactions, and chemical data." Journal of Chemical Information and Modeling 62.9 (2021): 2015-2020.
11. Zamil, Ahmad MA, and T. G. Vasista. "Customer segmentation using RFM analysis: Realizing through Python implementation." Pacific Business Review International 13.11 (2021).
12. Kablan, Kerem Görkem, Sait Gazi Yildirim, and Ali Durusu. "Python-Based Application Development for Electrical Base Price Estimation and Bidding Based on Consumption Data for Microgrids." 2024 16th Electrical Engineering Faculty Conference (BulEF). IEEE, 2024.
13. Vitti, M., et al. "Enhancing MRP Adoption in SMEs: A Python-Based Algorithmic Approach." Procedia Computer Science 253 (2025): 3088-3097.
14. Xidonas, Panos, Haris Doukas, and Elissaios Sarmas. "A python-based multicriteria portfolio selection DSS." RAIRO-Operations Research 55 (2021): S3009-S3034.
15. Reis, João. "Exploring Applications and Practical Examples by Streamlining Material Requirements Planning (MRP) with Python." Logistics 7.4 (2023): 91.
16. Arya, Swasti, Mihika Chitranshi, and Yograj Singh. "Ranking forecasting algorithms using MCDM methods: a python based application." Computational Intelligence-based Time Series Analysis. River Publishers, 2022. 99-125.
17. Reis, João. "Enhancing Python-based Sentiment Analysis: Empowering Industrial Engineers and Managers in the Service Industry." (2024).
18. Nguyen Chan, Nam, et al. "Design and deployment of a customer journey management system: the CJMA approach." Proceedings of the 5th International Conference on Future Networks and Distributed Systems. 2021.
19. Kumari, S. VARA, et al. "Python Based Smart School Bus Monitoring and Security System." J. Nonlinear Anal. Optim 15.1 (2024).
20. Gowtham, Ch Sai. "REAL TIME TRAFFIC MANAGEMENT SYSTEM USING YOLOv3 ALGORITHM." (2023).