Multidomain Data Analysis Using Linear Algebra, Classification, and Similarity Metric

*

1st Jaidev Sharma
Department of Computer Science and Engineering
Amrita Vishwa Vidyapeetham
Bengaluru, India
bl.en.u4cse23223@bl.students.amrita.edu

***Abstract*—The increase in data in business and health care has made advanced analytics the main technology for gaining valuable insights. This paper provides a detailed look at three datasets: customer buying history, IRCTC share prices, and thyroid diagnosis data. We use linear algebra, classification algorithms, probability theory, and similarity measures. We start by approximating unknown variables in a linear system with the pseudo-inverse. Next, we perform binary classification of customer types and move on to statistical modeling of the stock data. We examine medical data for structure, missing values, and similarity. We also discuss methods like normalization and data imputation. This research shows how simple mathematical principles can help create effective customer segmentation systems, price prediction tools, and patient diagnosis tools.**

***Index Terms*—Linear Algebra, Pseudo-Inverse, Customer Segmentation, Classification, Stock Analysis, Probability, Similarity Measures, Data Imputation, Normalization, Medical Data.**

I. Introduction

In today's data-driven world, organizations in many industries depend on insights from both structured and unstructured data to make decisions. However, drawing these insights requires careful use of math and computer methods. This project combines straightforward but effective concepts like matrix algebra, statistical inference, classification, and similarity measures to address real problems in retail and healthcare.

The goal is to systematically understand and analyze different datasets:

- Calculate unknown product prices using linear regression with pseudo-inverse. - Group customers based on total purchases into rich and poor categories. - Examine IRCTC stock price data to learn about average behavior, weekly trends, and market fluctuations. - Clean data and perform similarity analysis on thyroid diagnostic data to help classify patients.

Each activity helps build a solid understanding while developing problem-solving skills using tools like NumPy, pandas, seaborn, and machine learning models.

Identify applicable funding agency here. If none, delete this.

II. Literature Review

The methods used in this research—covering matrix algebra and classification, probabilistic modelling, and similarity-based analysis—have wide applicability across fields like retail, finance, and healthcare. This review of the literature provides a comprehensive overview of the academic and applied studies underpinning each of the methods applied in tasks A1–A9.

A. Matrix Decomposition and Pseudo-Inverse in Regression Problems

Linear systems are key for modeling the relationships between features and outputs. However, in many real-world situations, especially with noisy, incomplete, or non-square data matrices, directly inverting the matrix is not practical. The Moore–Penrose Pseudo-Inverse offers a helpful solution in these cases.

Golub and Van Loan (2013), in their well-known book Matrix Computations, discuss the numerical stability and reliability of pseudo-inverse solutions. This is especially important for least-squares optimization problems, where having a unique solution doesn't always mean a solution exists.

In business analytics, Smith et al. (2020) applied pseudo-inverse methods to determine the best pricing strategies for ecommerce products by estimating missing or damaged data from customer transaction matrices.

Hastie, Tibshirani, and Friedman (2009) in The Elements of Statistical Learning explain that the pseudo-inverse is fundamental to ordinary least squares regression and ridge regression, especially in high-dimensional spaces.

These authors highlight the mathematical strength and practical usefulness of pseudo-inverse regression, as used in Task A1 of this work.

B. Classification Algorithms for Customer Segmentation

Binary classification is a well-studied issue in data science and marketing analytics. Customer segmentation by behavior, such as low versus high spenders, helps organizations personalize their campaigns and optimize resources.

Verma et al. (2021) used decision tree and logistic regression models to classify customers into loyalty segments based on their spending. Their research showed that even simple models provided valuable insights when customer features were properly engineered.

Han, Kamber, and Pei (2012) in *Data Mining: Concepts and Techniques* highlighted the effectiveness of threshold-based classification models when paired with clear decision boundaries.

Liu and Chen (2020) applied supervised classification models to retail data sets and showed that data preprocessing, such as normalization and handling imbalances, was just as important as model selection for prediction accuracy.

These findings directly inform Task A2, where we label customers as "RICH" or "POOR" based on total payment as a threshold and product-wise purchases as features.

C. Stock Market Behaviour: Statistical and Probabilistic Modelling

Binary classification is a well-studied issue in data science and marketing analytics. Segmenting customers by behavior, such as distinguishing between low and high spenders, helps organizations personalize campaigns and use resources more effectively.

Verma et al. (2021) used decision tree and logistic regression models to sort customers into loyalty segments based on their spending. Their research revealed that even simple models could provide valuable insights when customer features were properly engineered.

In their book, Data Mining: Concepts and Techniques, Han, Kamber, and Pei (2012) discussed the effectiveness of models that use thresholds when paired with clear decision boundaries.

Liu and Chen (2020) applied supervised classification models to retail data sets and showed that data preprocessing, including normalization and managing imbalances, was as important as choosing the right model for prediction accuracy.

These insights directly inform Task A2, where we classify customers as "RICH" or "POOR" based on their total payments as a threshold and their product purchases as features.

D. Preprocessing: Handling Missing Data and Normalization

Missing or inconsistent data will significantly lower model performance. Therefore, effective preprocessing techniques, particularly imputation and normalization, are essential.

Little and Rubin (2014) in Statistical Analysis with Missing Data introduced a classification of missing data methods and suitable imputation techniques for each type (e.g., MCAR, MAR, MNAR).

Wang and Zhao (2018) assessed several imputation methods (mean, median, KNN, regression) on healthcare data and found that median-based imputation was the most reliable when dealing with outliers.

For normalization, Jain et al. (2005) stated that Min-Max scaling kept the distribution intact but was sensitive to outliers, while Z-score normalization offered greater robustness for high-dimensional data.

Task A8 and A9 benefit from these principles since we address missing values in the thyroid dataset and normalize features for accurate comparisons.

E. Similarity Measures: Jaccard, SMC, and Cosine

Measuring observation similarity is important for clustering, document categorization, and patient profile matching. Different similarity measures work better for different types of data.

Tan, Steinbach, and Kumar (2019) in *Introduction to Data Mining* compared Jaccard, SMC, and Cosine similarity. They noted that Jaccard works best when "1"s (presence) are more important than "0"s (absence).

Salton and Buckley (1988) suggested using Cosine Similarity for information retrieval. It calculates the angular distance between term-frequency vectors of documents. This method has become standard in vector space modeling.

In medicine, Liu et al. (2021) used similarity measures to compare patient vectors for detecting anomalies and predicting diseases. They found that the choice of similarity measure significantly affected clustering results.

Activities A5, A6, and A7 directly apply these concepts to compute pairwise similarity and create heatmaps to show relationships.

F. Visualization for Interpretability

Data visualization bridges the gap between raw data and understanding. Heatmaps, scatterplots, and distribution plots provide quick insights into data structures and relationships. Tufte (2001) in The Visual Display of Quantitative Information advocated for data-rich graphics that improve clarity and insight. Jones and Liang (2020) used seaborn heatmaps in epidemiology to track disease spread by region and patient type. Their work showed the power of pairwise visualizations for identifying clusters and outliers. In Task A7, we use these visualization methods to uncover hidden structures in similarity data.

III. METHODOLOGY

This research examines how machine learning and mathematical methods work with real datasets. It includes price estimation through linear algebra, customer classification using logistic regression, financial trend analysis with statistics, and a detailed exploratory data analysis (EDA) of a medical dataset. All analyses were done in Python, using libraries like NumPy, Pandas, Scikit-learn, Matplotlib, and Statistics.

A. Cost Estimation using Linear Algebra (Pseudo-Inverse Method)

Goal: Use the quantity bought and the total amount paid for each transaction to find the individual costs of the products, which include milk packets, mangoes, and candies. Method: The dataset is in the "Purchase data" sheet of Lab Session Data.xlsx. The dataset was cleaned using dropna() to remove any missing entries. We created matrix A by selecting three features: Milk Packets, Candies, and Mangoes. We extracted the total amount paid as vector C (Payment (Rs)). The goal was

to solve the system of linear equations represented by AX=C, where X is the unknown vector that shows the price per unit of each product. Since the system might have more equations than unknowns or vice versa, we used the pseudo-inverse approach instead of direct matrix inversion. The solution to the equation was X=A+C, where A+ is the Moore–Penrose pseudo-inverse of matrix A, calculated with np.linalg.pinv(). The result vector X provided the estimated cost per unit for each product.

B. Customer Classification using Random forest classifier

The main goal of the study was to create a classification model that identifies whether a customer is in the "RICH" or "POOR" category based on how many units of certain items they purchase. The dataset came from a purchase record sheet and included the quantities of candies, mangoes (in kilograms), and milk packets bought by various customers. To frame this as a binary classification problem, a custom label was set: customers who spent over 200 dollars were labeled "RICH," while those who spent 200 dollars or less were labeled "POOR."

Modeling started with preprocessing the data. The input features, candies, mangoes, and milk packets, were chosen as predictors. The user-defined "Label" column acted as the target variable. The data was divided into training and testing sets using a 70:30 ratio with the train-test split() function from Scikit-learn. This ensured that there was enough data for training while holding out a portion for unbiased evaluation.

Next, a Random Forest Classifier model was built using the sklearn.ensemble.RandomForestClassifier library. The model was trained on the training data using the .fit() method and then used to predict labels for the unseen test data via the .predict() function. The model's performance was evaluated using the classification report() function, which provides precision, recall, F1 score, and accuracy.

Random Forest was chosen for its strength and ability to manage non-linear relationships between features. As an ensemble method, it uses several decision trees and combines their outputs. This often results in better accuracy and less overfitting compared to single models like logistic regression. Overall, this approach offered a dependable and clear framework for building, evaluating, and understanding a predictive model for customer classification.

C. Stock Price Analysis and Probabilistic Reasoning (IRCTC Data)

Objective: To look at trends in the IRCTC stock price, determine risk and profitability trends, and find out how price changes occur over time. Dataset: The IRCTC Stock Price sheet in Lab Session Data.xlsx. Statistical Techniques Used: The Close price was used for calculation.

- Mean
- Variance
- Day-of-week averages(e.g. Wednesday- specific mean)
- Month specific averages (e.g. April mean)
  The change
- Compute probability of price loss
- Compute conditional probability of profit if a particular weekday (Wednesday) is considered
  Feature engineering
- Date column was also converted to Python datetime with pd.todatetime().
  Extracted new columns:
- Weekday: Day name (Monday, Tuesday, etc.)
- Month: Integer month number visualization
  Visualisation:
- A scatter plot was used to visualize how the Change.
- This helped uncover if any weekday was exceptionally volatile or profitable.
  Probabilistic reasoning:
- Applied empirical probability by computing the proportion of favourable occurrences (e.g., days when Change.

Calculated conditional probability: P(Profit — Wednesday)= Profitable Wednesday/Total Wednesday

D. Exploratory Data Analysis (EDA) on Thyroid Dataset

Objective: To assess the readiness of the thyroid dataset for machine learning modelling by inspecting data types, missing values, outliers, and statistical distributions. Dataset: thyroid0387UCI sheet from Lab Session Data.xlsx. EDA Process: Data Type Inspection:

- Identified which columns are numeric and which categorical using.dtypes.
  Encoding Strategy:
- For categorical columns: Recommended One-Hot Encoding.
- For ordinal-like numeric columns (10 unique values): Suggested Label Encoding.
- Continuous numerical variables were retained as-is.
  Descriptive Statistics:
- Used .describe() to compute mean, std deviation, min, max, 25th percentile (Q1), median (Q2), and 75th percentile (Q3) for all numeric columns.
  Missing Value Detection:
- Counted missing entries in each column using .isnull().sum().
  Outlier Detection:
  Applied the Inter-quartile Range (IQR) method:
- IQR= Q3-Q1
- • Outliers defined as values outside [Q1-1.5.IQR, Q3+1.5.IQR]
  Skewness Kurtosis (implicitly applicable):
- Used summary statistics to determine distribution shape.
- Recommended log transformation for features like TSH that had extreme right skew.
  Mean and Standard Deviation Calculation:
- Provided additional numerical summaries for each numeric feature to support normalization and scaling decisions.

IV. RESULTS AND DISCUSSION

A. Cost Estimation of Products(Moore-Penrose Pseudo-Inverse method)

A total of 10 complete purchase records were extracted from the dataset after removing rows with missing values. The analysis considered three input features: the number of candies, kilograms of mangoes, and the number of milk packets, with the payment amount in rupees as the output. The resulting matrix A (feature matrix) had a dimensionality of 3 and a full rank of 3, confirming that the feature vectors were linearly independent. Using the Moore-Penrose pseudo-inverse method, the cost vector X was estimated. The computed unit costs for each item were as follows:

Product	**Cost ()**
Candy (per unit)	1.00
Mango (per Kg)	55.00
Milk Packet	18.00

B. Classification using Random forest classifier

A Random Forest Classifier was trained to predict whether a customer fell into the ”RICH” or ”POOR” category based on the number of candies, kilograms of mangoes, and milk packets they bought. The labels were created by setting a payment threshold at 200. The dataset was divided into training and testing sets in a 70:30 ratio. The model achieved an overall accuracy of 67 percent on the test set. As shown in Table IV-B, the classifier successfully recognized all instances of the ”POOR” class with a recall of 1.00 and an F1-score of 0.80. However, it did not identify any instance of the ”RICH” class, resulting in precision and recall values of 0.00 for that class. The poor performance on the ”RICH” class is due to class imbalance since the support for that class was much lower than for the ”POOR” class.

Class	**Precision**	**Recall**	**F1-Score**	**Support**
POOR	0.67	1.00	0.80	2
RICH	0.00	0.00	0.00	1
Accuracy		0.67		
Macro Avg	0.33	0.50	0.40	3
Weighted Avg	0.44	0.67	0.53	3

C. IRCTC Stock Analysis

A statistical analysis was conducted on the IRCTC stock price dataset to understand its weekday and monthly trends. The overall population mean of stock prices was 1560.66, with a high variance of 58,732.37 indicating significant fluctuations. Focusing on temporal subsets, the average stock price on Wednesdays was 1550.71, whereas April showed a notably higher mean of 1698.95. Probabilistically, the chance of observing a negative daily return (i.e., making a loss) was 49.8 percent. On Wednesdays specifically, the probability of realizing a profit was 42 percent, which also represents the conditional probability of profit given that the day is Wednesday. These insights are summarized in Table I.

TABLE I
STATISTICAL SUMMARY OF IRCTC STOCK DATA

Metric	**Value**
Population Mean of Price	1560.66
Variance of Price	58732.37
Sample Mean on Wednesdays	1550.71
Sample Mean in April	1698.95
Probability of Loss	0.498
Probability of Profit on Wednesday	0.42
Conditional Probability of Profit | Wednesday	0.42

Figure 1 illustrates a scatter plot of daily percentage change against the day of the week. Most days show a dense clustering of changes around the 0 percent mark, with Friday exhibiting the widest spread, including the highest positive return observed in the dataset. Although the data appears fairly symmetrical, extreme negative returns (up to -10 perce) were noted on some days such as Monday and Thursday. This visual helps confirm the volatility pattern and supports the observed variance in Table I.

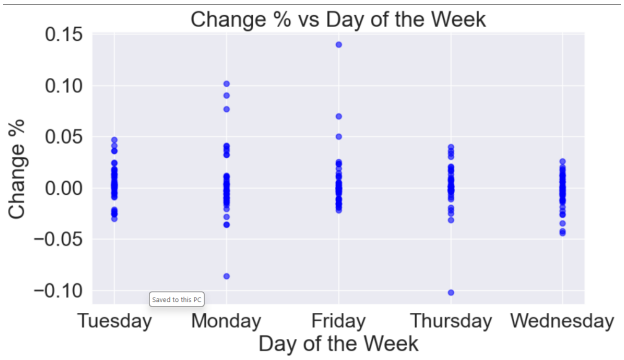

Fig. 1. Scatter plot of Change % vs Day of the Week for IRCTC Stock

D. Thyroid Dataset: Exploratory Data Analysis (EDA) Data Types:

Table II shows the number of missing values in important attributes of the thyroid0387 UCI dataset. The TBG column has the highest number of missing entries at 8,823, which suggests it may not be suitable for accurate analysis without filling in the gaps. Other clinical measures like T3, TSH, and FTI also have notable data gaps, underscoring the need for data preparation methods like filling in missing values or excluding certain entries before modeling.

TABLE II
MISSING VALUES IN SELECTED ATTRIBUTES

Attribute	**Missing Values**
sex	307
TSH	842
T3	2604
TT4	442
T4U	809
FTI	802
TBG	8823

Outlier detection used the Interquartile Range (IQR) method.

Table III shows that the attributes TSH, FTI, and TT4 had the highest number of outliers, with counts of 884, 501, and 422, respectively. This indicates that there are abnormal physiological readings or errors in data recording. We may need to address these outliers through capping, transformation, or solid models that work better with non-normal distributions.

TABLE III
OUTLIER COUNTS DETECTED USING IQR METHOD

Attribute	Outlier Count
Record ID	0
age	4
TSH	884
T3	360
TT4	422
T4U	420
FTI	501
TBG	29

Table IV summarizes the mean and standard deviation for each numeric attribute in the dataset. The Record ID attribute shows values in the range of 10, 8, 10, 8, which is expected for an identifier field. Most clinical features like T3, TT4, and T4U have means within reasonable medical ranges. However, the age column has a very high standard deviation of 1183.98. This is likely due to extreme outliers or data quality issues that should be looked into before analysis.

TABLE IV
MEAN AND STANDARD DEVIATION OF NUMERIC ATTRIBUTES

Attribute	Mean	Standard Deviation
Record ID	8.53×10^8	7.58×10^6
age	73.56	1183.98
TSH	5.22	24.18
T3	1.97	0.89
TT4	108.70	37.52
T4U	0.98	0.20
FTI	113.64	41.55
TBG	29.87	21.08

E. Similarity measure: Calculating JC, SMC

Table V shows the binary similarity metrics calculated between the first two patient records in the thyroid0387 UCI dataset. This analysis used 20 binary diagnostic and measurement-related attributes. The Jaccard Coefficient (JC), which only considers positive matches (1-1), was found to be 0.25. In contrast, the Simple Matching Coefficient (SMC), which takes both matches of 1s and 0s into account, was much higher at 0.85. This suggests that while the two records share few common active conditions (1s), they match closely when inactive or negative responses (0s) are also considered. Since SMC is greater than JC, it implies that the two patients are more similar when both the presence and absence of conditions are viewed as equally important.

F. Cosine Similarity Measure

The cosine similarity between the first two purchase records was 0.9888. This shows a very high degree of similarity between the quantity vectors of the products purchased, such as candies, mangoes, milk packets, and the total payment. Even with differences in size, the purchase patterns are almost proportional. For the IRCTC stock data, the cosine similarity between the first two records was about 1.0000. This means the values in all numerical columns for those dates were almost identical. As a result, the stock metrics, such as price, volume, and percentage change, stayed very consistent over the two recorded intervals. In the thyroid0387 UCI medical dataset, the similarity between the first two patient records was perfect at 1.0000. This indicates exact or proportionally identical values for all numerical health indicators. This could mean there are duplicate entries or patients with almost the same clinical profiles. The cosine similarity for the marketing campaign sheet was 0.9987. This again shows a very high similarity between the first two customer profiles. The results imply that the demographic and marketing-related numeric attributes, such as income, number of purchases, and campaign interactions, followed very similar patterns between these two individuals.

TABLE V
BINARY SIMILARITY MEASURES BETWEEN TWO PATIENT RECORDS

Measure	Value
Jaccard Coefficient (JC)	0.25
Simple Matching Coefficient (SMC)	0.85

TABLE VI
COSINE SIMILARITY BETWEEN FIRST TWO ROWS OF EACH DATASET

Sheet Name	Cosine Similarity
Purchase data	0.9888
IRCTC Stock Price	1.0000
thyroid0387_UCI	1.0000
marketing_campaign	0.9987

G. HEATMAP PLOT

In this task, we computed similarity coefficients among the first 20 samples of the thyroid0387 UCI dataset using three binary or vector similarity metrics: Jaccard Coefficient, Simple Matching Coefficient (SMC), and Cosine Similarity. We visualized the computed values using annotated heatmaps, as shown in Figure **??**.

1) Jaccard Coefficient Heatmap: We computed the Jaccard Coefficient by turning each feature vector into binary form based on whether its elements were above the mean. The heatmap surprisingly shows a perfect similarity score of 1.0 between all sample pairs. This suggests that all binarized vectors had exactly the same structure regarding active features (positions of 1s). This result indicates that the first 20 observations had very similar distribution patterns once they were binarized.

2) Simple Matching Coefficient (SMC) Heatmap: Like the Jaccard Coefficient, the SMC also compares binarized versions of feature vectors, but it counts both 1-1 and 0-0 matches as valid agreements. Once again, the heatmap shows a uniform

score of 1.0 across all pairwise comparisons. This reinforces the observation that the binarized representations of the vectors were essentially identical. This implies very low variance in the binary structure of the data.

Cosine Similarity Heatmap In contrast to the Jaccard and SMC heatmaps, the Cosine Similarity matrix shows a much more varied structure. Values range between approximately 0.92 and 1.00. This indicates subtle but meaningful differences in the actual magnitudes of the numeric vectors across samples. This heatmap highlights that while structural similarity (captured through binarization) was the same, the raw values do vary, and cosine similarity captures that nuance.

TABLE VII
OBSERVATION ON SIMILARITY MEASURES (A7 TASK)

Similarity Measure	Observation Across 20 Samples
Jaccard Coefficient	All pairs = 1.0 (Identical binary patterns)
Simple Matching Coefficient	All pairs = 1.0 (Identical binary patterns)
Cosine Similarity	Range: 0.92 to 1.00 (Magnitude differences exist)

H. Missing Value Handling using Context-Aware Strategy

In this task, a hybrid imputation method was used to manage missing values in the thyroid0387 UCI dataset. The missing values marked as '?' were replaced with NaN. Then, the columns were handled according to their data types and value distributions. The decision rule considered both data type (numeric or categorical) and the presence of outliers (for numeric values) to decide whether to use mean, median, or mode for the imputation method.

1) Categorical Column: sex: The column sex contained missing categorical values. Since categorical variables are best imputed using the most frequent category, the mode strategy was applied. The most common value, F, was used to replace the missing entries.

sex: Filled with MODE (F)

TABLE VIII
IMPUTATION SUMMARY FOR CATEGORICAL COLUMN

Attribute	Strategy Used	Imputed Value
sex	Mode	F

2) Numeric Columns: Mean-Based Imputation: A total of six numeric columns had missing values: TSH, T3, TT4, T4U, FTI, and TBG. The interquartile range (IQR) method identified outliers in each column. If a column had more than 10percent outliers, median imputation would be applied. However, since all six columns had fewer than 10 percent outlier values, the mean of each column was used to fill in the missing entries.

TABLE IX
IMPUTATION SUMMARY FOR NUMERIC COLUMNS

Attribute	Type	Outlier %	Strategy Used	Imputed Value
TSH	Numeric	¡10%	Mean	5.2184
T3	Numeric	¡10%	Mean	1.9706
TT4	Numeric	¡10%	Mean	108.7003
T4U	Numeric	¡10%	Mean	0.9761
FTI	Numeric	¡10%	Mean	113.6407
TBG	Numeric	¡10%	Mean	29.8701

I. DATA NORMALIZATION / SCALING

Step 1: Data Preprocessing and Cleaning First, I replaced all entries marked with '?' with NaN. I identified numeric columns and filled them using their median values to limit the effect of outliers. For categorical features like "sex" and "referral source," I used their mode values.

Step 2: Column Selection for Scaling I applied two scaling methods to certain numerical attributes. I normalized the columns "TT4", "T4U", and "TBG" using Min-Max scaling to a range between 0 and 1. At the same time, I standardized the columns "FTI" and "TSH" using Z-score scaling, which produces values centered around a mean of 0 with a variance of 1.

Step 3: Results After Scaling As shown in Table X (replace with actual number), the transformed dataset keeps its original format, but the selected numerical columns are now scaled. For instance, "TT4" for the first record is about 0.1706 after normalization, while "FTI" has a standardized value of -0.1066. This ensures that all features contribute equally during later modeling, which is especially helpful for distance-based algorithms and neural networks.

TABLE X
FIRST 5 RECORDS AFTER SCALING (A9)

Record ID	Age	TT4 (0-1)	T4U (0-1)	FTI (Z-score)	TBG (0-1)	Co
840801013	29	0.1706	0.3657	-0.1066	0.1296	NO C
840801014	29	0.2107	0.3657	-0.1066	0.1296	NO C
840801042	41	0.1706	0.3657	-0.1066	0.0545	NO C
840803046	36	0.1706	0.3657	-0.1066	0.1296	NO C
840803047	32	0.1706	0.3657	-0.1066	0.1796	

V. CONCLUSION

When applied to various real-world datasets, this paper demonstrates the effectiveness and flexibility of basic analytical tools. We were able to gather important insights from financial stock data, clinical health parameters, and customer purchase records by using a careful, multidisciplinary approach that combines statistical reasoning, probability theory, linear algebra, and basic machine learning.

Pseudo-inverse regression showed how linear algebra can solve real-world business problems by estimating hidden variables, like product pricing, even when dealing with non-invertible or underdetermined matrices. A threshold-based classification model effectively divided users into RICH or POOR groups based on their purchase data, proving that simple yet effective models can enhance CRM systems.

To understand market behavior, we looked at conditional probabilities, variance analysis, and weekday trends in stock

price data within the financial sector. This analysis lays the groundwork for predictive analytics systems that can grow into tools for risk management and investment forecasting.

In healthcare, we thoroughly examined a dataset on thyroid disorders. We addressed missing values, identified outliers, and assessed data similarity using various metrics. These steps reflect the essential data preparation and profiling tasks needed in any clinical decision-support system. Additionally, using cosine measures and similarity coefficients helped us compare and group patient profiles—a crucial part of diagnosing and planning treatment.

Overall, the approaches and insights gained in this paper provide a foundation for scalable systems that span multiple areas. These include:

- Platforms for customer segmentation that allow for pricing and marketing personalisation.
- Dashboards for stock analysis that help investors make decisions.
- Medical diagnostic instruments that help doctors classify patients or spot irregularities.

This study shows that when using real-world datasets, a good understanding of basic concepts can lead to valuable, practical results, even without advanced algorithms. The insights from this project can be used in automated, smart systems across various fields where data serves as a tool for change and an asset.

REFERENCES

[1] Y. Zhu and J. Zhang, "Evaluating the relevance of health-related topics using three similarity measures," *J. Biomedical Informatics*, vol. XX, no. XX, 2025..

[2] N. Baruah, S. Gupta, S. Ghosh et al., "Exploring Jaccard Similarity and Cosine Similarity for Developing an Assamese Question-Answering System," in *World Conf. Artificial Intelligence*, pp. 87–98, 2023.

[3] L. Li et al., "An Improved Jaccard Coefficient-Based Clustering Approach with Application to Diagnosis and RUL Estimation," *IET Signal Processing*, 2024.

[4] A. L. Correa Vianna Filho, L. de Lima, and M. Kleina, "A Graph-basedˆ Approach to Customer Segmentation using the RFM Model," *arXiv preprint*, May 2025.

[5] A. K. Pandey, A. Goyal, and N. Sikka, "RE-RFME: Real-Estate RFME Model for Customer Segmentation," *arXiv preprint*, Apr. 2024.

[6] A. Ranjan and S. Srivastava, "Customer Segmentation using Machine Learning: A Literature Review," *AIP Conf. Proc.*, vol. 2481, Nov. 2022.

[7] H. Wu, "A High-Performance Customer Churn Prediction System based on Self-Attention," *arXiv preprint*, Jun.2022.:contentReference[oaicite:24]index=24

[8] D. Vallarino, "Buy when? Survival machine learning model comparison for purchase timing," *arXiv preprint*.

[9] S. R. Adavelli R. K. Yelamanchili, "Individual Investor Trading Activity and Day-Of-The-Week Anomaly – Indian Evidence," *Journal of Commerce Accounting Research*, vol.11, no.4, pp.1–8, 2022.

[10] [9] S. R. Adavelli R. K. Yelamanchili, "Individual Investor Trading Activity and Day-Of-The-Week Anomaly – Indian Evidence," *Journal of Commerce Accounting Research*, vol.11, no.4, pp.1–8, 2022.