# Data-Driven Movie Insights Using Multi-Algorithm Clustering

**Project Report**

Submitted to the Faculty of Engineering of

**JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY KAKINADA, KAKINADA**

In partial fulfillment of the requirements for the award of the Degree of

**BACHELOR OF TECHNOLOGY**

In

**COMPUTER SCIENCE AND ENGINEERING**

By


**V. Bindu Madhuri (22481A05O4)**

**R. Hemachand (22481A05K9)**

**P. Yaswanth (22481A05K3)**

**U. Lokesh (22481A05N7)**


Under the Enviable and Esteemed Guidance of

**Dr. G. KEERTHI, M.Tech, Ph.D.**

**Assistant Professor of CSE**


**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**SESHADRI RAO GUDLAVALLERU ENGINEERING COLLEGE**

**(An Autonomous Institute with Permanent Affiliation to JNTUK, Kakinada)**

**SESHADRI RAO KNOWLEDGE VILLAGE GUDLAVALLERU – 521356**

**ANDHRA PRADESH**


**2024-25**

# SESHADRI RAO GUDLAVALLERU ENGINEERING COLLEGE

**(An Autonomous Institute with Permanent Affiliation to JNTUK, Kakinada)**

**SESHADRI RAO KNOWLEDGE VILLAGE, GUDLAVALLERU**

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**


## CERTIFICATE

This is to certify that the project report entitled **"Data-Driven Movie Insights Using Multi-Algorithm Clustering"** is a Bonafide record of work carried out by **V. Bindu Madhuri(22481A05O4),R.Hemachand(22481A05K9),P. Yaswanth(22481A05K3), U. Lokesh (22481A05N7)** under the guidance and supervision of **Dr. G. Keerthi , M.Tech , Ph.D. , Assistant Professor,** Computer Science and Engineering, in the partial fulfillment of the requirements for the award of the degree of Bachelor of Technology in Computer Science and Engineering of Jawaharlal Nehru Technological University Kakinada, Kakinada during the academic year 2024-25.

**Project Guide** **Head of the Department**

**(Dr. G. KEERTHI)** **(Dr. M. BABU RAO)**

**External Examiner**

## ACKNOWLEDGEMENT

The satisfaction that accompanies the successful completion of any task would be incomplete without the mention of people who made it possible and whose constant guidance and encouragements crown all the efforts with success.

We would like to express our deep sense of gratitude and sincere thanks to **Dr.G.Keerthi, M.Tech, Ph.D, Assistant Professor of CSE**, Computer Science and Engineering for her constant guidance, supervision and motivation in completing the project work.

We feel elated to express our floral gratitude and sincere thanks to **Dr. M. Babu Rao, Head of the Department,** Computer Science and Engineering for his encouragements all the way during analysis of the project. His annotations, insinuations and criticisms are the key behind the successful completion of the project work.

We would like to thank our beloved principal **Dr. B. KARUNA KUMAR** for providing a great support for us in completing our project and giving us the opportunity for doing project.

Our Special thanks to the faculty of our department and programmers of our computer lab. Finally, we thank our family members, non-teaching staff and our friends, who had directly or indirectly helped and supported us in completing our project in time.

Team Members

**V. Bindu Madhuri (22481A05O4)**

**R. Hemachand (22481A05K9)**

**P. Yaswanth (22481A05K3)**

**U. Lokesh (22481A05N7)**

**INDEX**

# ABSTRACT

In today's data-driven era, uncovering meaningful patterns from large datasets is essential for strategic insights and informed decision-making. This project harnesses the capabilities of the Orange data mining platform to perform unsupervised learning on a movie-related dataset, focusing on clustering techniques to analyze viewer behavior and movie performance trends.

The dataset, gathered through Google Forms, includes attributes such as viewer demographics, genre preferences, and social media reach. After applying essential preprocessing steps—such as data cleaning, normalization, and feature selection—the project explores clustering using K-Means, Hierarchical Clustering, and DBSCAN. Each method is used to identify natural groupings in the data, helping to reveal distinct audience segments and patterns in content engagement.

Through Orange's intuitive visual interface, complex clustering results are transformed into clear and interactive visualizations, allowing for easier interpretation and comparison of clustering outcomes. The study emphasizes how different clustering algorithms can uncover varied insights depending on the nature and structure of the dataset.

By integrating real-world data with visual analytics, this project highlights the practical value of clustering techniques in the entertainment industry. The findings demonstrate how unsupervised learning can guide content strategy, improve audience targeting, and support data-informed decision-making in movie marketing and production.

# PART-A

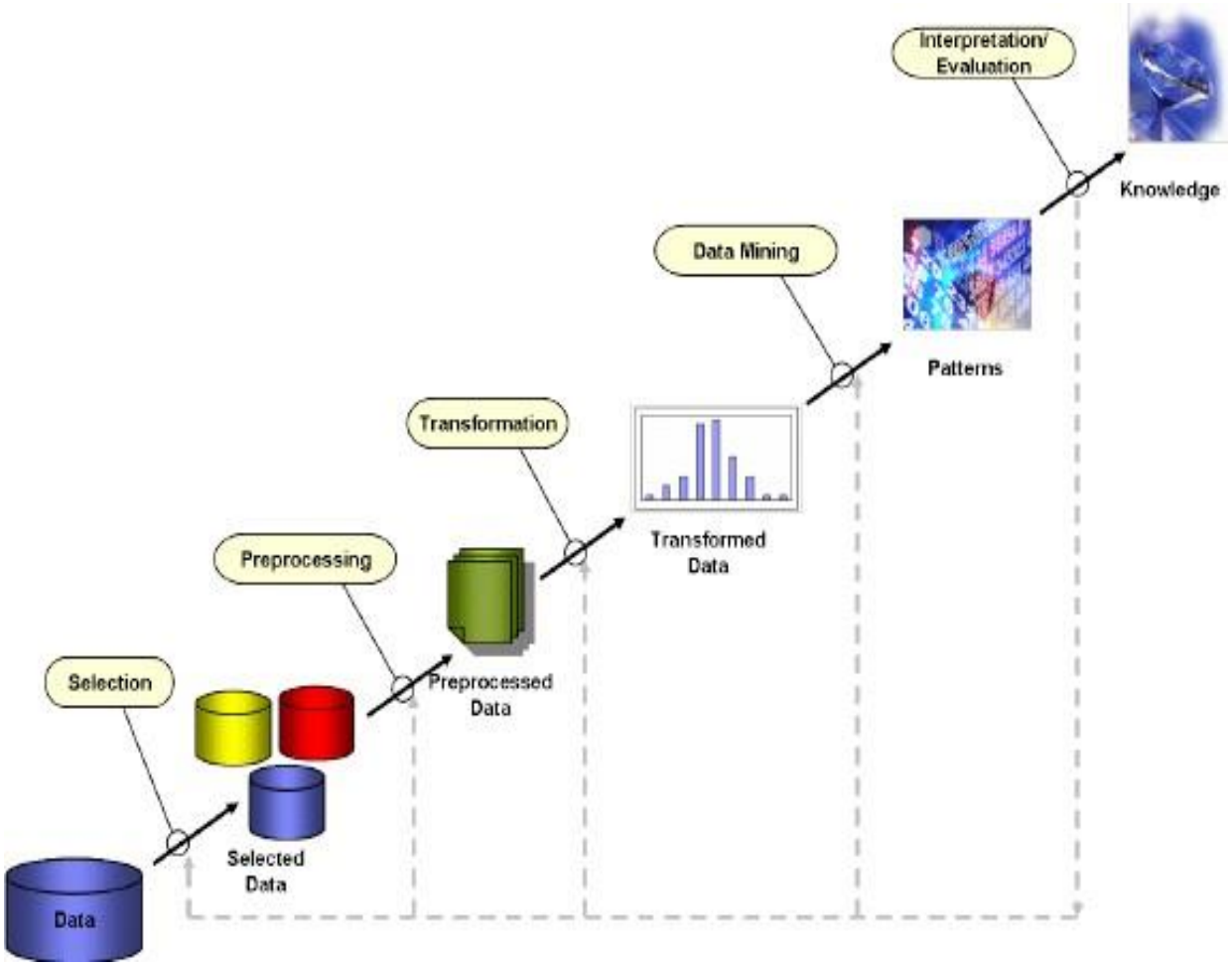
# Clustering-Based Analysis of Movie Performance Metrics Using KDD Process

## CHAPTER 1: INTRODUCTION

### 1.1 INTRODUCTION

Knowledge Discovery in Databases (KDD) refers to the complete process of uncovering valuable knowledge from large datasets. It starts with the selection of relevant data, followed by preprocessing to clean and organize it, transformation to prepare it for analysis, data mining to uncover patterns and relationships, and concludes with the evaluation and interpretation of results, ultimately producing valuable knowledge or insights. KDD is widely utilized in fields like machine learning, pattern recognition, statistics, artificial intelligence, and data visualization. The KDD process is iterative, involving repeated refinements to ensure the accuracy and reliability of the knowledge extracted. The whole process consists of the following steps:

1. Data Selection
2. Data Cleaning and Preprocessing
3. Data Transformation and Reduction
4. Data Mining
5. Evaluation and Interpretation of Results



## 1.2 DATA MINING

Data mining is a process of discovering patterns and knowledge from large amounts of data, utilizing sources such as databases, data warehouses, the internet, and other data repositories. It combines techniques from statistics, artificial intelligence, and machine learning to analyze large datasets and extract meaningful information. This analysis helps identify trends, correlations, and patterns that are not immediately obvious, enabling informed decision-making and predictions.

One of the key breakthroughs in data mining is its ability to handle and analyze big data efficiently. With the increasing volume, velocity, and variety of data, traditional methods are often insufficient. Data mining techniques like clustering, classification, regression, and association rule learning are essential for extracting valuable insights from complex datasets quickly and accurately.

Data mining is closely related to machine learning and data analytics. While data mining focuses on discovering new patterns within large datasets, machine learning involves developing algorithms that can learn from and make predictions on data. These fields complement each other, enhancing data analysis and predictive modeling capabilities.

## 1.3 DATA WAREHOUSING

A data warehouse is a centralized system used for storing and managing large volumes of data from various sources. It is designed to help businesses analyze historical data and make informed decisions. Data from different operational systems is collected, cleaned, and stored in a structured way, enabling efficient querying and reporting.

- Goal is to produce statistical results that may help in decision-making.
- Ensures fast data retrieval even with the vast datasets.

1. **Data Source Layer (Extracting Data)**
   - Data is collected from surveys, user logs, and online analytics.
   - Includes user demographics, preferred genre, spent duration, and rating.
2. **ETL (Extract, Transform, Load) Process**
   - **Extraction**: Data is gathered from multiple sources.
   - **Transformation**: Data is cleaned, formatted, and standardized.
   - **Loading**: The processed data is stored in the warehouse.
3. **Data Storage Layer (Fact & Dimension Tables)**
   - **Fact Table** stores core metrics like revenue, rating, ad budget, social media reach and conversion rate.
   - **Dimension Tables** include details like user demographics, preferred genre, time and region.

4. **OLAP (Online Analytical Processing) for Data Analysis**
    - Allows multi-dimensional analysis to identify trends in user behavior
    - Enables queries like:
        a. Which genre has the highest revenue?
        b. What are the most common platforms used for streaming?
        c. Which age group watches the most movies?
5. **Data Visualization & Reporting**
    - Insights are presented using **dashboards, reports, and visual charts**.
    - Helps movie platforms optimize their services based on user preferences.

## ➢ DATA MINING VS DATA WAREHOUSING

Data warehousing and data mining serve distinct but complementary purposes in data management. Data warehousing involves storing and organizing large volumes of data from various sources into a centralized repository, designed to support efficient querying and reporting for business intelligence. It focuses on the ETL (Extract, Transform, Load) process to ensure data consistency and accessibility. In contrast, data mining analyzes this stored data to discover patterns, trends, and relationships using algorithms and statistical methods. The primary goal of data mining is to transform raw data into actionable insights that inform business strategies and decision-making. While data warehousing emphasizes efficient storage and access, data mining focuses on extracting meaningful knowledge from the data. Together, they enable effective data management and strategic decision-making by leveraging stored data for in-depth analysis and discovery.

## ➢ DATA MINING INTRODUCTION

The block diagram for our project begins with collecting the **movie dataset**, followed by **data preprocessing** to clean and normalize the data. Once the data is prepared, Clustering algorithms such as k-Means, hierarchical clustering are used to group users based on similarities in attributes such as movie preferences, viewing duration, revenue, and user demographics. The resulting clusters help identify patterns and user segments, which can be interpreted to predict or understand the movie performance.

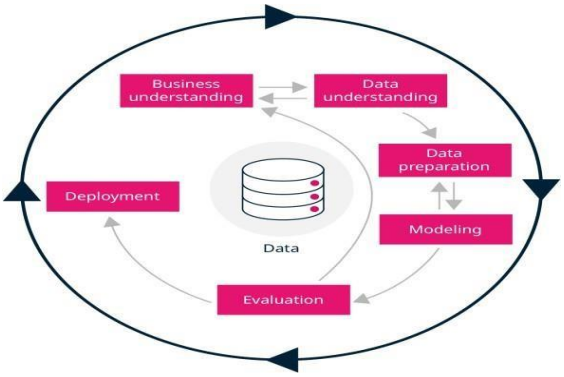



**Fig 1: Data Mining Block Diagram**

**DATA MINING BLOCK DIAGRAM EXPLANATION**

The data mining process follows structured steps to extract meaningful insights from the dataset:

1. **Data Understanding**
   - Collecting and analyzing the music streaming dataset to grasp its structure and content.
   - Identifying attributes such as user demographics, preferred genre, spent duration, and rating.
2. **Data Preparation**
   - Cleaning and transforming the dataset by handling missing values, standardizing data, and encoding categorical attributes.
   - Normalizing numerical data for better accuracy in analysis.
3. **Modeling**
   - Applying various clustering algorithms like k-Means, hierarchical clustering, DBScan to group users based on similarities in attributes such as movie preferences, viewing duration, revenue, ad budget and rating.
4. **Evaluation**
   - Assessing model performance using accuracy, precision, recall, and F1-score to ensure reliable predictions.
5. **Deployment**
   - Integrating the best-performing model to provide insights into which movie genre users prefer based on their watching habits.

## ➢ UNSUPERVISED LEARNING

Unsupervised learning is a type of machine learning where the model is trained on unlabeled data, meaning there are no predefined output labels. The goal is to discover hidden patterns or intrinsic structures within the data. Common techniques include clustering (e.g., K-Means) and association rule learning. This approach is useful for tasks like customer segmentation and anomaly detection.

- K- Means Clustering
- Hierarchical Clustering
- DBSCAN

## Categories of Unsupervised Learning in This Project:

### 1. Clustering:

Clustering serves as a vital technique in unsupervised learning within data mining. It involves grouping similar data points together into clusters based on their intrinsic characteristics, without predefined labels. Algorithms like K-Means and Hierarchical Clustering help us uncover hidden patterns within our dataset of lens-related attributes. By applying clustering, we aim to identify distinct groups of individuals with similar visual characteristics, facilitating personalized recommendations for lens suitability. This unsupervised approach aids in data exploration and segmentation, providing insights into diverse needs and preferences among individuals. Overall, clustering plays a crucial role in uncovering meaningful patterns and guiding data-driven decision-making in lens recommendation strategies and many other applications.

| Algorithm | Description | Type |
|---|---|---|
| K-Means | An unsupervised learning algorithm that partitions data into K distinct clusters based on similarity. | Clustering |
| Hierarchical Clustering | An unsupervised algorithm that builds a hierarchy of clusters either through agglomerative or divisive approaches. | Clustering |
| DBSCAN | A density-based clustering algorithm that groups together closely packed data points and marks outliers as noise. | Clustering |

**TABLE 1: Methods of unsupervised learning**

## ➢ SUPERVISED LEARNING

Supervised learning is a machine learning technique where models are trained on labeled data. In this project, the model learns to **predict the music streaming platform preference** based on user attributes. Common algorithms used include:

**Categories of Supervised Learning:**

1. **Classification:**
   - The dataset contains categorical labels.
   - Classification algorithms predicts and categorizes the text into predefined class labels.
2. **Regression:**
   - If we analyze listening duration as a continuous variable, regression models could predict how long a user is likely to listen on a platform.
   - However, since our project focuses on platform prediction, classification is the primary approach.

### How to Choose a Data Mining Algorithm?

Choosing the right data mining algorithm depends on:

- **If the data has labels:**
  Use Supervised Learning (Classification/Regression).
- **If the data has no labels:**
  Use Unsupervised Learning (Clustering/Association).

Since our dataset focuses on **identifying hidden patterns and grouping similar categories**, **clustering algorithms** are the best fit. However, **classification algorithms** can be used for **categorizing the data into predefined labels.**

## ➢ ASSOCIATION

Association analysis is a core technique in unsupervised learning within data mining, aimed at discovering relationships among different attributes or items in a dataset. Algorithms like Apriori and FP- Growth enable us to identify frequent item sets and association rules within our dataset of lens- related attributes. By applying association analysis, we aim to uncover associations between visual characteristics such as age, prescription, tear production rate, and astigmatism status, and the types of lenses recommended. Additionally, association analysis helps identify relevant features for lens suitability, contributing to the refinement of our predictivemodels.

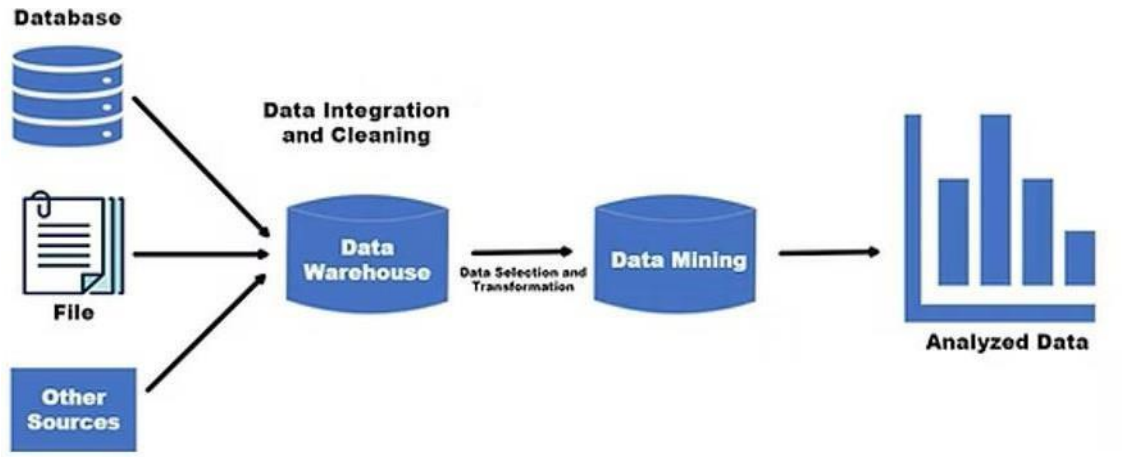



**Fig 2: Data Mining Basic Diagram**

## ➢ CHALLENGES AND LIMITATIONS OF DATA MINING

One of the major challenges in data mining is ensuring **data quality and preprocessing**. In real-world scenarios, datasets often contain **noise, missing values, and inconsistencies**, which can significantly impact the effectiveness of data mining algorithms.

**Key Challenges:**

- **Data Cleaning & Normalization:** Raw data needs extensive cleaning to remove duplicates, inconsistencies, and errors**.**
- **Feature Selection:** Choosing the most relevant attributes is crucial for improving model accuracy.
- **Resource-Intensive Processing:** Preprocessing large and complex datasets requires significant computational power and time**.**
- **Bias & Data Limitations:** Even after cleaning**,** inherent biases in the data may affect model predictions, leading to skewed insights.

Addressing these challenges is critical for ensuring accurate and reliable predictions in data mining projects.

## ➢ APPLICATIONS OF DATA MINING

**1. Customer Relationship Management (CRM)**

Data mining helps businesses analyze customer demographics, purchase history, and behavioral trends to optimize marketing strategies**.**

- Identifies high-value customers and predicts churn rates**.**
- Enables personalized recommendations and targeted marketing campaigns.
- Improves customer engagement and retention.

**2. Fraud Detection**

- Data mining is widely used in banking, insurance, and e-commerce to detect fraudulent transactions.
- Algorithms analyze transactional data to detect anomalies.
- Identifies patterns indicating fraudulent behavior.
- Enhances real-time fraud prevention systems.

- **PROBLEM STATEMENT**

The segmentation of movie audiences based on viewing preferences, demographics, and social media engagement is essential for targeted marketing, content recommendation, and performance prediction. However, manual analysis of audience behavior is inefficient and often fails to reveal hidden patterns within complex datasets.
This project aims to apply unsupervised machine learning techniques to automatically cluster movie data and audience responses. By using algorithms such as K-Means, Hierarchical Clustering, and DBSCAN within the Orange data mining platform, the project seeks to identify meaningful audience segments and performance trends that can support data-driven decisions in movie production and promotion.

**Objectives:**

- Identify meaningful audience segments through clustering techniques.
- Uncover hidden patterns in viewer preferences and behavior using unsupervised learning.
- Support movie marketing and production strategies with data-driven insights.

By leveraging clustering algorithms, the system will enable smarter audience segmentation, leading to enhanced targeting, improved content strategies, and better decision-making in the entertainment industry.

# REQUIREMENTS

## Hardware Requirements

| Component | Specification |
|---|---|
| **Processor** | Intel i5 / i7 or equivalent |
| **RAM** | Minimum 8 GB (16 GB preferred) |
| **Hard Disk** | Minimum 256 GB (SSD preferred) |
| **GPU** | NVIDIA GPU (for model training acceleration, |

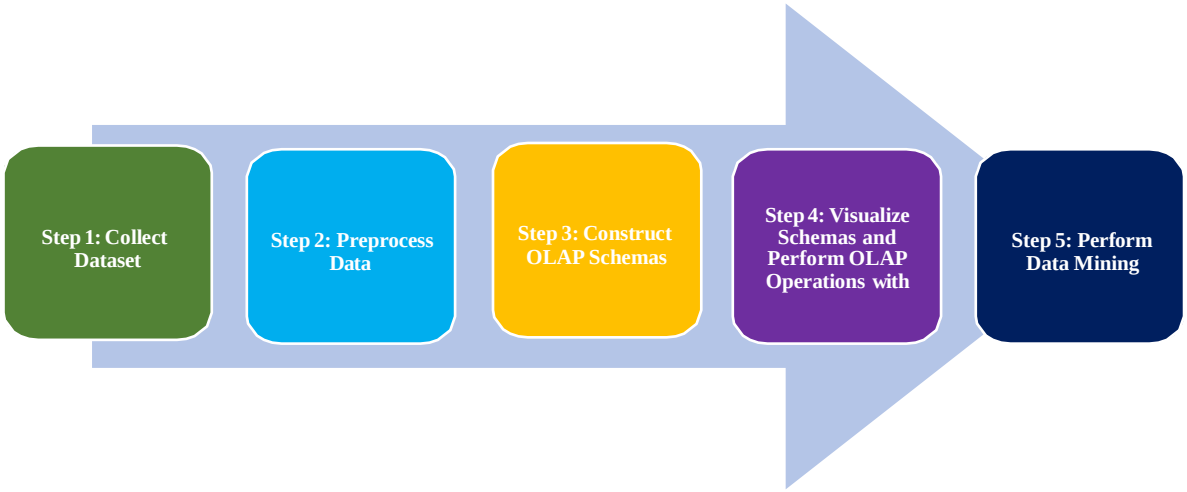
## Software Requirements

| Software/Tool | Purpose |
|---|---|
| **SQL Server** | Database Management |
| **SSMS** | Writing SQL queries, schema design |
| **SSAS** | Cube construction and OLAP operations |
| **SSDT (in Visual Studio)** | Creating multidimensional models |
| **MDX Queries** | Data analysis in OLAP cubes |
| **Python** | Python Script for Algorithms |
| **Orange Data Mining Tool** | Clustering algorithms, Model training, evaluation, and visualization |

# CHAPTER-2: Knowledge Discovery in Databases (KDD) Process

**METHODOLOGY:**

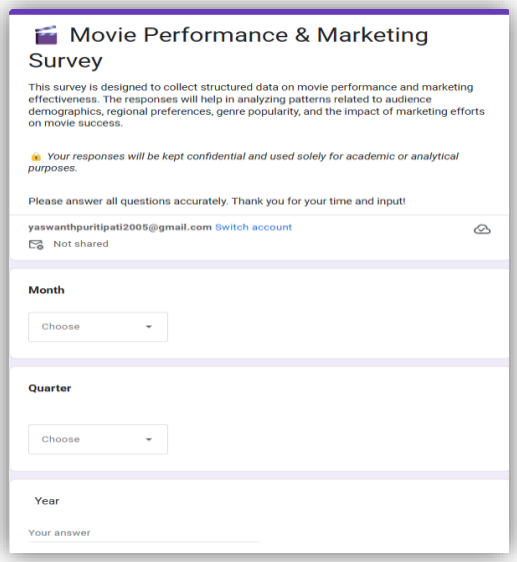
The KDD process is performed in step by step from collection of data set to the classification and developing the prediction model. There are some intermediary steps in which we created all three schemas with the help of various tools like SSMS(SQL Server Management Services), Visual Studio and SSAS (SQL Server Analysis Services).The process is explained in step by step below.

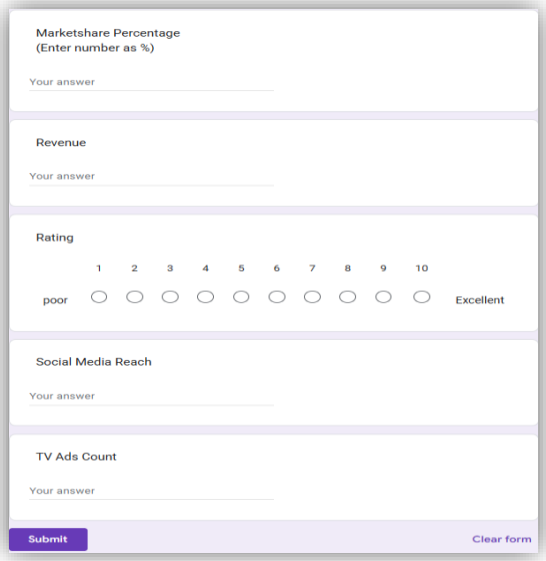



## STEP-1: COLLECT & EXPLORE DATASET

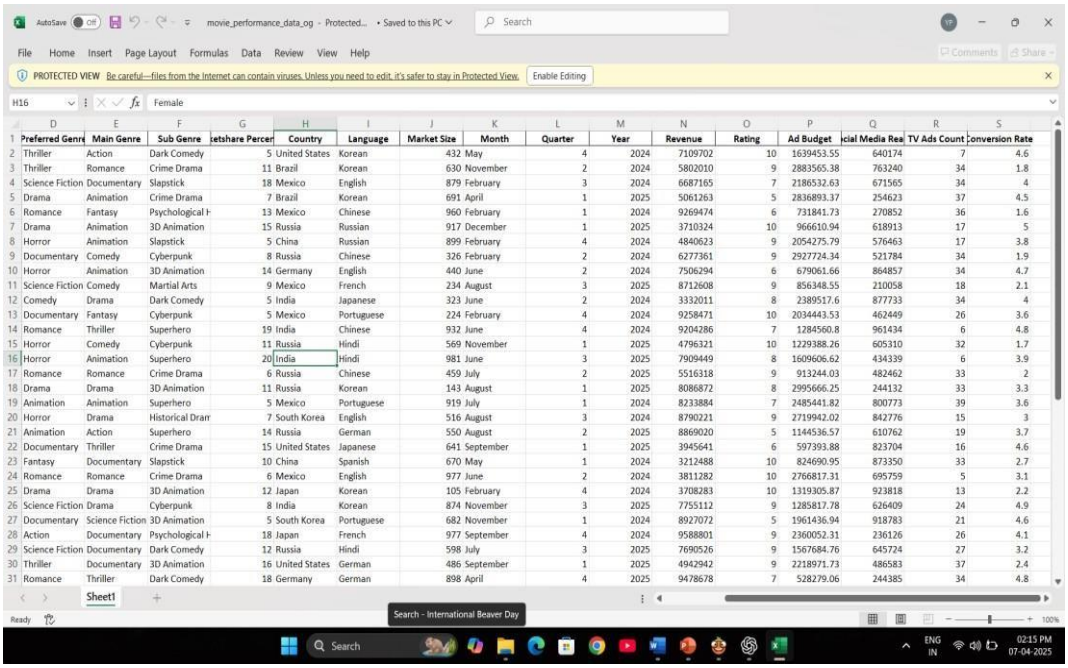
**1.1.** 1. Create and Extract the Form to Collect Information from Users

2. Designed Google Form – with relevant questions(e.g., Revenue, gender, Rating, Ad Budget)




3. Shared it with participants through email, social media, or targeted groups.
   Form Link : https://forms.gle/5w91oY293UQLeH5k7
   Collected Responses – Monitored responses and ensure enough data is gathered.
4. Exported Data – Downloaded the responses as a CSV file for further processing.



| Preferred Genre | Main Genre | Sub Genre | etshare Percer | Country | Language | Market Size | Month | Quarter | Year | Revenue | Rating | Ad Budget | cial Media Rea | TV Ads Count | onversion Rate |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Thriller | Action | Dark Comedy | 5 | United States | Korean | 432 | May | 4 | 2024 | 7109702 | 10 | 1639453.55 | 640174 | 7 | 4.6 |
| Thriller | Romance | Crime Drama | 11 | Brazil | Korean | 630 | November | 2 | 2024 | 5802010 | 9 | 2883565.38 | 763240 | 34 | 1.8 |
| Science Fiction | Documentary | Slapstick | 18 | Mexico | English | 879 | February | 3 | 2024 | 6687165 | 7 | 2186532.63 | 671565 | 34 | 4 |
| Drama | Animation | Crime Drama | 7 | Brazil | Korean | 691 | April | 1 | 2025 | 5061263 | 5 | 2836893.37 | 254623 | 37 | 4.5 |
| Romance | Fantasy | Psychological H | 13 | Mexico | Chinese | 960 | February | 1 | 2024 | 9269474 | 6 | 731841.73 | 270852 | 36 | 1.6 |
| Drama | Animation | 3D Animation | 15 | Russia | Russian | 917 | December | 1 | 2025 | 3710324 | 10 | 966610.94 | 618913 | 17 | 5 |
| Horror | Animation | Slapstick | 5 | China | Russian | 899 | February | 4 | 2024 | 4840623 | 9 | 2054275.79 | 576463 | 17 | 3.8 |
| Documentary | Comedy | Cyberpunk | 8 | Russia | Chinese | 326 | February | 2 | 2024 | 6277361 | 9 | 2927724.34 | 521784 | 34 | 1.9 |
| Horror | Animation | 3D Animation | 14 | Germany | English | 440 | June | 2 | 2024 | 7506294 | 6 | 679061.66 | 864857 | 34 | 4.7 |
| Science Fiction | Comedy | Martial Arts | 9 | Mexico | French | 234 | August | 3 | 2025 | 8712608 | 9 | 856348.55 | 210058 | 18 | 2.1 |
| Comedy | Drama | Dark Comedy | 5 | India | Japanese | 323 | June | 2 | 2024 | 3332011 | 8 | 2389517.6 | 877733 | 34 | 4 |
| Documentary | Fantasy | Cyberpunk | 5 | Mexico | Portuguese | 224 | February | 4 | 2024 | 9258471 | 10 | 2034443.53 | 462449 | 26 | 3.6 |
| Romance | Thriller | Superhero | 19 | India | Chinese | 932 | June | 4 | 2024 | 9204286 | 7 | 1284560.8 | 961434 | 6 | 4.8 |
| Horror | Comedy | Cyberpunk | 11 | Russia | Hindi | 569 | November | 1 | 2025 | 4796321 | 10 | 1229388.26 | 605310 | 32 | 1.7 |
| Horror | Animation | Superhero | 20 | India | Hindi | 981 | June | 3 | 2025 | 7909449 | 8 | 1609606.62 | 434339 | 6 | 3.9 |
| Romance | Romance | Crime Drama | 6 | Russia | Chinese | 459 | July | 2 | 2025 | 5516318 | 9 | 913244.03 | 482462 | 33 | 2 |
| Drama | Drama | 3D Animation | 11 | Russia | Korean | 143 | August | 1 | 2025 | 8086872 | 8 | 2995666.25 | 244132 | 33 | 3.3 |
| Animation | Animation | Superhero | 5 | Mexico | Portuguese | 919 | July | 1 | 2024 | 8233884 | 7 | 2485441.82 | 800773 | 39 | 3.6 |
| Horror | Drama | Historical Dran | 7 | South Korea | English | 516 | August | 3 | 2024 | 8790221 | 9 | 2719942.02 | 842776 | 15 | 3 |
| Animation | Action | Superhero | 14 | Russia | German | 550 | August | 2 | 2025 | 8869020 | 5 | 1144536.57 | 610762 | 19 | 3.7 |
| Documentary | Thriller | Crime Drama | 15 | United States | Japanese | 641 | September | 1 | 2025 | 3945641 | 6 | 597393.88 | 823704 | 16 | 4.6 |
| Fantasy | Documentary | Slapstick | 10 | China | Spanish | 670 | May | 1 | 2024 | 3212488 | 10 | 824690.95 | 873350 | 33 | 2.7 |
| Romance | Romance | Crime Drama | 6 | Mexico | English | 977 | June | 2 | 2024 | 3811282 | 10 | 2766817.31 | 695759 | 5 | 3.1 |
| Drama | Drama | 3D Animation | 12 | Japan | Korean | 105 | February | 4 | 2024 | 3708283 | 10 | 1319305.87 | 923818 | 13 | 2.2 |
| Science Fiction | Drama | Cyberpunk | 8 | India | Korean | 874 | November | 3 | 2025 | 7755112 | 9 | 1285817.78 | 626409 | 24 | 4.9 |
| Documentary | Science Fiction | 3D Animation | 5 | South Korea | Portuguese | 682 | November | 1 | 2024 | 8927072 | 5 | 1961436.94 | 918783 | 21 | 4.6 |
| Action | Documentary | Psychological H | 18 | Japan | French | 977 | September | 4 | 2024 | 9588801 | 9 | 2360052.31 | 236126 | 26 | 4.1 |
| Science Fiction | Documentary | Dark Comedy | 12 | Russia | Hindi | 598 | July | 3 | 2025 | 7690526 | 9 | 1567684.76 | 645724 | 27 | 3.2 |
| Thriller | Documentary | 3D Animation | 16 | United States | German | 486 | September | 1 | 2025 | 4942942 | 9 | 2218971.73 | 486583 | 37 | 2.4 |
| Romance | Thriller | Dark Comedy | 18 | Germany | German | 898 | April | 4 | 2025 | 9478678 | 7 | 528279.06 | 244385 | 34 | 4.8 |

**Fig 3: Data Collected Using Goggle Form**

### 1.2 Defining Survey or Data Collection Methods

- **Online Surveys:** A structured questionnaire was distributed online, including **multiple-choice and rating-scale questions** to capture user preferred genres, rating, TV ads count.
- **Form Link** : https://forms.gle/5w91oY293UQLeH5k7

### 1.3 Choosing Attributes for Analysis

The key attributes selected for analysis include:

- **Revenue** – Represents the total income generated by each movie.
- **Ad Budget** – Denotes the amount spent on marketing and promotions.
- **Social Media Reach** – Indicates the digital outreach and campaign visibility across platforms.
- **Conversion Rate** – Measures the effectiveness of turning online interest into actual ticket sales or views.
- **Market Size** – Refers to the potential audience base targeted by each movie.
- **Cluster Labels** – Derived from unsupervised learning methods to segment movies based on performance patterns.

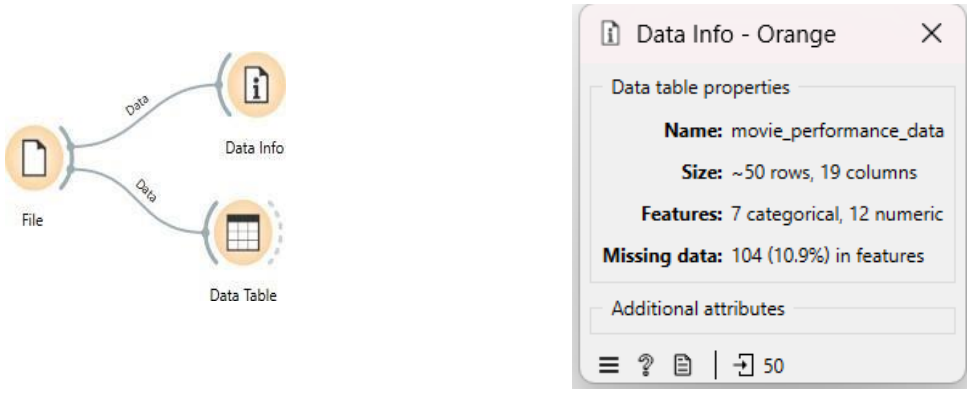



**Fig 4: Data Info in Orange Tool**

❖ The dataset is small (**50 records**) but contains a variety of **numerical features**, making it well-suited for **unsupervised clustering tasks** rather than classification.

- **Missing values** in key features like `Revenue`, `Ad Budget`, and `Rating` were addressed using **average imputation** to maintain data integrity.
- **Meta variables** such as Movie Name, Genre, or Platform were excluded from the clustering process, as they are non-numeric and not directly useful unless transformed (e.g., through one-hot encoding or NLP techniques).
- Since this is an **unsupervised learning task**, there is **no predefined target variable**. Instead, clustering algorithms (K-Means, Hierarchical, DBSCAN) were used to discover natural groupings in the data
- **These clusters help identify patterns in movie performance, which can support decisions in marketing strategy, budget allocation, and content targeting.**

| | e Percent | Country | Language | Market Size | Month | Quarter | Year | Revenue | Rating | Ad Budget | ocial Media Reac | TV Ads Count | Conversion Rate |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 5 | United States | Korean | 432 | May | 4 | 2024 | 7109702 | 10 | 1639453.55 | ? | 7 | 4.6 |
| 2 | 11 | Brazil | ? | 630 | November | 2 | 2024 | 5802010 | 9 | 2883565.38 | 763240 | 34 | 1.8 |
| 3 | ? | Mexico | English | 879 | February | 3 | 2024 | 6687165 | 7 | 2186532.63 | 671565 | 34 | 4.0 |
| 4 | 7 | ? | Korean | 691 | April | 1 | 2025 | 5061263 | ? | 2836893.37 | 254623 | ? | ? |
| 5 | 13 | Mexico | Chinese | ? | February | 1 | 2024 | 9269474 | 6 | 731841.73 | 270852 | 36 | 1.6 |
| 6 | 15 | Russia | Russian | 917 | December | 1 | 2025 | 3710324 | 10 | ? | ? | 17 | 5.0 |
| 7 | 5 | China | ? | 899 | February | 4 | 2024 | 4840623 | 9 | 2054275.79 | 576463 | 17 | 3.8 |
| 8 | 8 | Russia | ? | 326 | February | 2 | 2024 | ? | 9 | 2927724.34 | 521784 | 34 | 1.9 |
| 9 | 14 | Germany | English | 440 | June | 2 | 2024 | 7506294 | 6 | 679061.66 | 864857 | ? | 4.7 |
| 10 | 9 | ? | French | 234 | August | 3 | 2025 | 8712608 | 9 | 856348.55 | 210058 | 18 | 2.1 |
| 11 | 5 | India | Japanese | 323 | June | 2 | 2024 | 3332011 | ? | 2389517.60 | 877733 | 34 | 4.0 |
| 12 | 5 | Mexico | ? | 224 | February | 4 | 2024 | 9258471 | 10 | ? | ? | 26 | 3.6 |
| 13 | 19 | India | Chinese | 932 | June | 4 | 2024 | 9204286 | 7 | 1284560.80 | 961434 | 6 | 4.8 |
| 14 | 11 | Russia | Hindi | ? | November | 1 | 2025 | 4796321 | ? | 1229388.26 | 605310 | 32 | 1.7 |
| 15 | 20 | ? | Hindi | 981 | June | 3 | 2025 | 7909449 | 8 | 1609606.62 | 434339 | 6 | 3.9 |
| 16 | 6 | Russia | Chinese | 459 | July | 2 | 2025 | ? | 9 | 913244.03 | 482462 | 33 | ? |
| 17 | ? | ? | Korean | 143 | August | 1 | 2025 | 8086872 | 8 | 2995666.25 | 244132 | ? | 3.3 |
| 18 | 5 | Mexico | Portuguese | 919 | July | 1 | 2024 | 8233884 | 7 | 2485441.82 | 800773 | 39 | 3.6 |
| 19 | 7 | South Korea | English | 516 | August | 3 | 2024 | 8790221 | 9 | 2719942.02 | ? | 15 | 3.0 |
| 20 | 14 | Russia | German | 550 | August | 2 | 2025 | 8869020 | 5 | ? | 610762 | 19 | 3.7 |
| 21 | 15 | United States | Japanese | ? | September | 1 | 2025 | 3945641 | 6 | 597393.88 | 823704 | 16 | 4.6 |
| 22 | 10 | China | ? | 670 | May | 1 | 2024 | 3212488 | 10 | 824690.95 | 873350 | 33 | 2.7 |
| 23 | 6 | ? | English | 977 | June | 2 | 2024 | 3811282 | 10 | 2766817.31 | 695759 | 5 | ? |
| 24 | 12 | Japan | Korean | 105 | February | 4 | 2024 | 3708283 | ? | 1319305.87 | 923818 | 13 | 2.2 |
| 25 | 8 | India | Korean | 874 | November | 3 | 2025 | 7755112 | 9 | 1285817.78 | ? | 24 | 4.9 |
| 26 | ? | ? | Portuguese | ? | November | 1 | 2024 | 8927072 | 5 | 1961436.94 | 918783 | 21 | 4.6 |
| 27 | 18 | Japan | French | 977 | September | 4 | 2024 | 9588801 | 9 | ? | ? | ? | 4.1 |
| 28 | 12 | Russia | Hindi | 598 | July | 3 | 2025 | ? | 9 | 1567684.76 | 645724 | 27 | 3.2 |
| 29 | 16 | ? | ? | 486 | September | 1 | 2025 | 4942942 | 9 | 2218971.73 | 486583 | 37 | 2.4 |

**Fig 5: Data Displayed by the Data Table – Orange with Missing Values**

## Step-2: PREPROCESS THE DATA

Preprocess the Dataset Using **ORANGE TOOL**

### 2.1 Handling Missing Values

- **Numerical** features such as Revenue, Ad Budget, and Rating were imputed using the Average or Most Frequent method.
- **Categorical** variables, where applicable, were filled using Normalize Feature interval [0,1].
- **Entries with excessive missing data** were replaced with Average/Most Frequent to maintain dataset integrity.


**Fig 6: Pre-processing the data**

### 2.2 Data Cleaning & Transformation

- Handled missing values through imputation to preserve essential movie data.
- Removed irrelevant features unrelated to performance metrics.
- Standardized numerical attributes such as **Revenue**, **Ad Budget**, and **Market Share** to ensure consistent scaling.
- Encoded categorical attributes (e.g., **Genre**, **Platform**) into numerical form to make them suitable for clustering algorithms.

### 2.3 Removing Duplicates & Inconsistencies

- Removed duplicate movie entries to ensure unique records for accurate clustering.
- Verified data consistency by standardizing format across all performance-related attributes.
- Applied feature selection to retain only key performance indicators (e.g., Revenue, Ratings, Market Share, Social Media Reach).
- Performed normalization to align data scales and improve clustering performance.
- Categorical features were encoded numerically to enable compatibility with algorithms like K-Means and DBSCAN.

➢ **PROCESSING THE DATA**

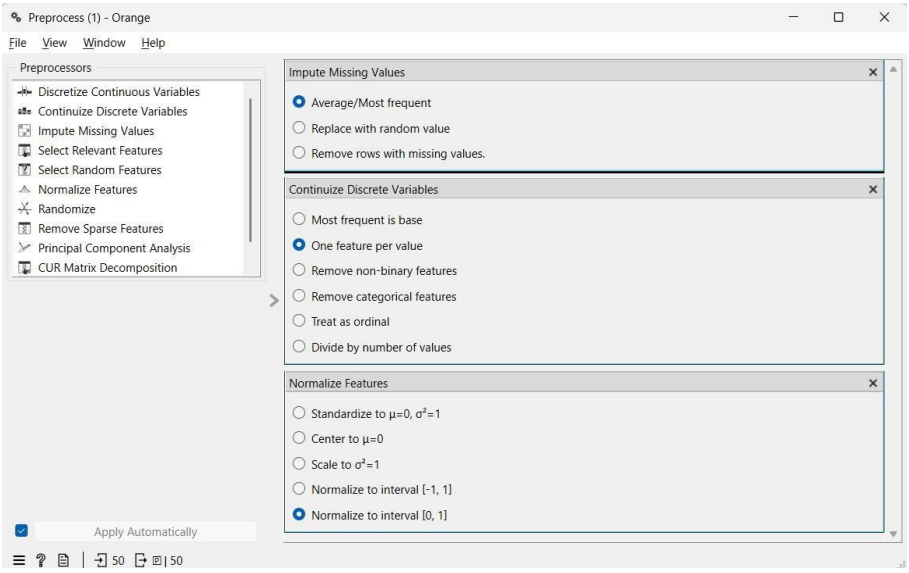



**Fig 7: Preprocessing the Data**

- Other Preprocessors, such as **Discretize Continuous Variables, Randomize, Select Random Features** etc…, were not used as they produced lower accuracies, making them less effective for this dataset.

| | ge=Hindi | nguage=Japanes | anguage=Korear | nguage=Portugue | anguage=Russiar | Market Size | Year | Revenue | Rating | Ad Budget | ocial Media Reacl | TV Ads Count | Conversion Rate |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 21 | 0.000 | 1.000 | 0.000 | 0.000 | 0.000 | 0.58725 | 1 | 0.110561 | 0.2 | 0.0183064 | 0.816696 | 0.3235 | 0.8857 |
| 22 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.64059 | 0 | 0 | 1.0 | 0.111347 | 0.88277 | 0.8235 | 0.3429 |
| 23 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.98866 | 0 | 0.0902992 | 1.0 | 0.906324 | 0.646415 | 0.00 | 0.5196 |
| 24 | 0.000 | 0.000 | 1.000 | 0.000 | 0.000 | 0.000 | 0 | 0.0747667 | 0.572 | 0.313809 | 0.949937 | 0.2353 | 0.20 |
| 25 | 0.000 | 0.000 | 1.000 | 0.000 | 0.000 | 0.87188 | 1 | 0.685036 | 0.8 | 0.300101 | 0.556867 | 0.5588 | 0.9714 |
| 26 | 0.000 | 0.000 | 0.000 | 1.000 | 0.000 | 0.58725 | 0 | 0.861769 | 0.0 | 0.576655 | 0.943236 | 0.4706 | 0.8857 |
| 27 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.98866 | 0 | 0.961559 | 0.8 | 0.531629 | 0.556867 | 0.5501 | 0.7429 |
| 28 | 1.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.55896 | 1 | 0.516856 | 0.8 | 0.415479 | 0.579824 | 0.6471 | 0.4857 |
| 29 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.43197 | 1 | 0.260955 | 0.8 | 0.682073 | 0.368025 | 0.9412 | 0.2571 |
| 30 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.89909 | 1 | 0.944952 | 0.4 | 0.531629 | 0.0456855 | 0.8529 | 0.9429 |
| 31 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 1.000 | 1 | 0.279915 | 0.572 | 0.747348 | 0.800932 | 0.4706 | 0.6857 |
| 32 | 0.000 | 0.000 | 0.000 | 1.000 | 0.000 | 0.91043 | 0 | 0.970995 | 1.0 | 0.263516 | 0.077032 | 0.5588 | 0.3429 |
| 33 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.04989 | 1 | 0.516856 | 0.8 | 0.895834 | 0.466394 | 0.8529 | 0.9429 |
| 34 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.07370 | 0 | 0.0152316 | 0.2 | 0.278947 | 0.556867 | 0.7353 | 0.5196 |
| 35 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.58725 | 0 | 0.85394 | 1.0 | 0 | 0.516258 | 0.00 | 0.9714 |
| 36 | 0.000 | 1.000 | 0.000 | 0.000 | 0.000 | 0.52608 | 1 | 0.0899101 | 0.0 | 0.69942 | 0.263096 | 0.1471 | 0.1143 |
| 37 | 0.000 | 0.000 | 0.000 | 1.000 | 0.000 | 0.99546 | 1 | 0.250496 | 0.4 | 0.513542 | 0.741108 | 1.00 | 0.5196 |
| 38 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.80612 | 0 | 0.638518 | 0.2 | 0.531629 | 0.97364 | 0.50 | 0.2571 |
| 39 | 0.000 | 0.000 | 0.000 | 0.000 | 1.000 | 0.03968 | 0 | 0.0633461 | 0.8 | 0.85423 | 0.75845 | 0.2941 | 0.2286 |
| 40 | 0.000 | 1.000 | 0.000 | 0.000 | 0.000 | 0.10204 | 1 | 0.878188 | 0.572 | 0.512125 | 0.589565 | 0.4706 | 0.1714 |
| 41 | 0.000 | 0.000 | 0.000 | 0.000 | 1.000 | 0.34921 | 1 | 0.569196 | 0.8 | 0.882964 | 0.650001 | 0.7941 | 0.1143 |
| 42 | 0.000 | 1.000 | 0.000 | 0.000 | 0.000 | 0.92971 | 0 | 0.932715 | 0.8 | 0.419449 | 0.877485 | 0.5501 | 0.2571 |
| 43 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.58725 | 0 | 0.191739 | 0.2 | 0.650971 | 0.612498 | 0.3824 | 0.40 |
| 44 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.79025 | 1 | 0.516856 | 0.2 | 0.531629 | 0.223759 | 0.6765 | 0.5143 |
| 45 | 0.000 | 0.000 | 1.000 | 0.000 | 0.000 | 0.68594 | 0 | 0.0914409 | 0.0 | 0.964678 | 0.749634 | 0.5588 | 0.5196 |
| 46 | 0.000 | 0.000 | 0.000 | 0.000 | 1.000 | 0.91043 | 0 | 0.744862 | 0.572 | 0.536868 | 0.207342 | 0.8529 | 0.8286 |
| 47 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.98526 | 1 | 0.208061 | 1.0 | 0.196755 | 0.556867 | 0.5501 | 0.6286 |
| 48 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.58725 | 1 | 0.635306 | 0.8 | 0.384364 | 0.223312 | 0.9118 | 0.4286 |
| 49 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.65193 | 1 | 1 | 0.0 | 0.77732 | 0.861506 | 0.4706 | 0.5196 |
| 50 | 0.000 | 0.000 | 0.000 | 1.000 | 0.000 | 0.30952 | 0 | 0.395869 | 0.0 | 0.531629 | 0.858663 | 0.2059 | 0.00 |

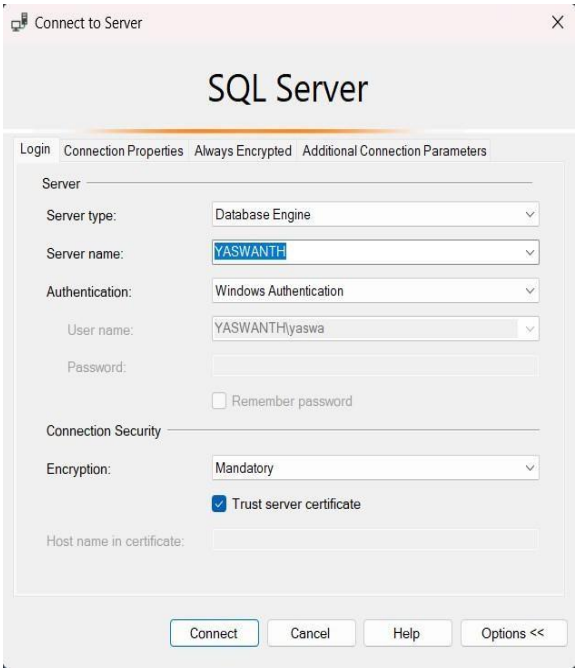
**Fig 8: Transformed Data After Normalizing**

- Figure 6, is a Preprocessed Data Table without a Missing Values.
- Furthermore, this is used to Perform the OLAP schemas.

## STEP-3: Creating in Database Engine and Managing a Multidimensional Data Model in SSMS and Visual Studio

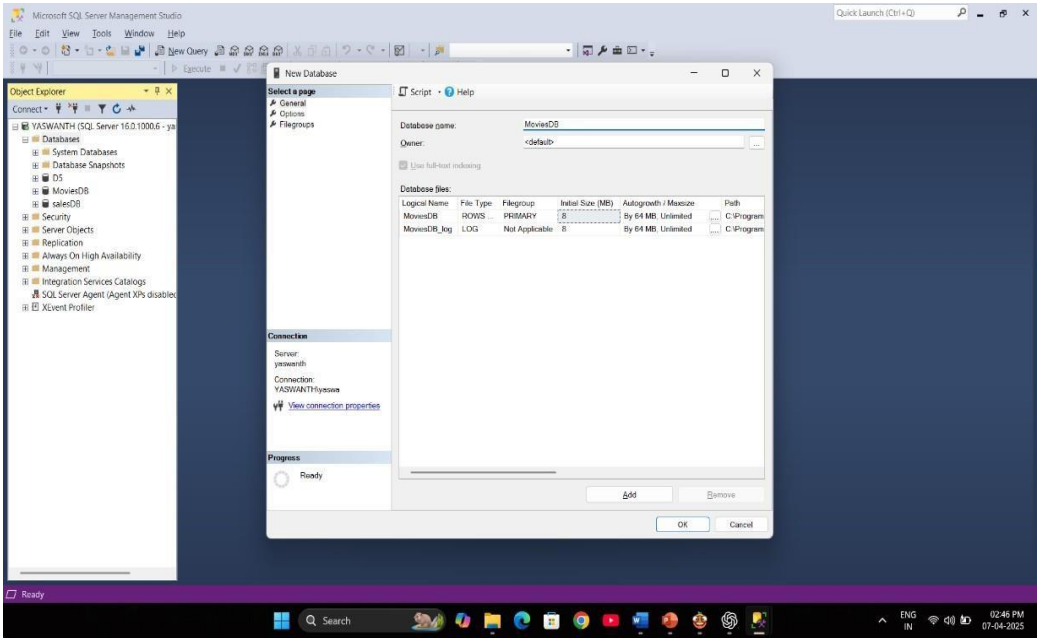
### 3.1 Create a Database in SSMS

1. Open **SQL Server Management Studio (SSMS)** and connect to the **Database Engine**.




2. Create a new database: **MoviesDB**

   **Open → Databases →Create New Database →MoviesDB**



### 3.2 Create and Insert Dimension Tables

**Dimension Tables (from your schema):**

- Audience
- Genre
- Main_Genre
- Region
- Language
- Time

### 3.3 Create and Insert Fact Tables

- **Movie_Performance**
- **Marketing_Performance**

### 3.4 Create a Multidimensional Project in Visual Studio

- Open **Visual Studio**.
- Create a new **Analysis Services Multidimensional and Data Mining Project**.
- Name it **D5_Movies**.

### 3.5 Process Data Source and Data Source Views

1. Add a **New Data Source** and connect it to the **MoviesDB database**.

2. Create **Data Source Views (DSV)** and include all tables.

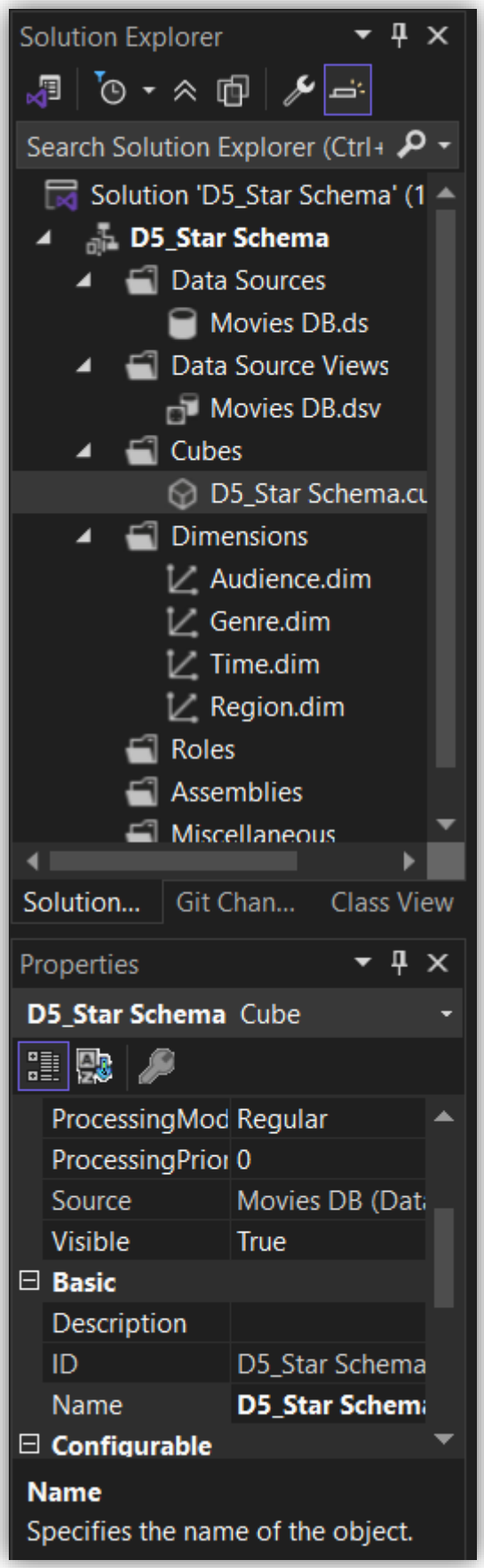
### 3.6 Construct the Cube

1. Choose **schema design**:
   - **Star Schema** (Direct links between fact and dimension tables).
   - **Snowflake Schema** (Normalized dimension tables).
   - **Fact Constellation** (Multiple fact tables sharing dimensions).

2. Create a **new cube** and select:
   - **Movies_Performance** and **Market_Performance** as **fact tables**.
   - All **dimension tables**.

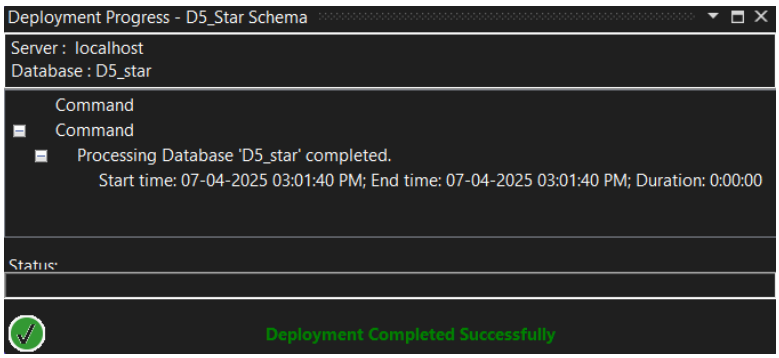
3. Define **measures** (e.g., Revenue, Rating, Ad Budget etc…,**).**

4. Establish relationships between dimensions and facts.



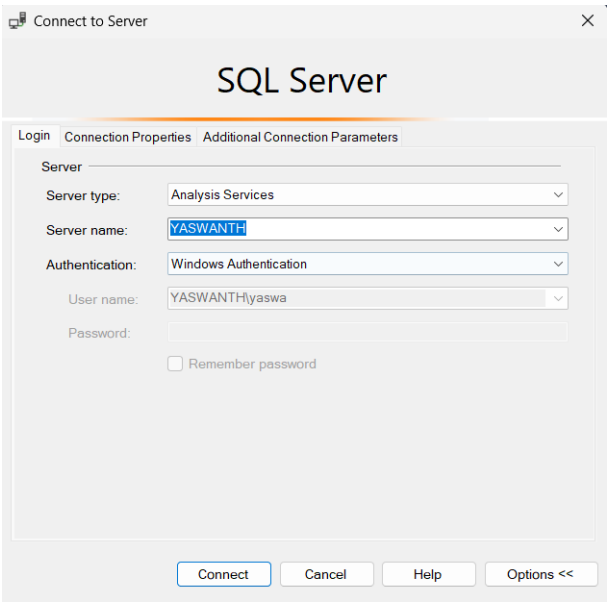
### 3.7 Deploy the Project

1. Click **Build → Deploy D5_Movies** in **Visual Studio**.
2. Verify deployment success.




### 3.8 Perform Schema Analysis in SSMS

1. Open **SSMS** and connect to **Analysis Services**.

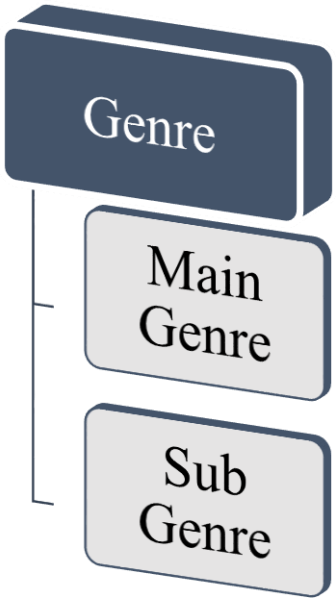


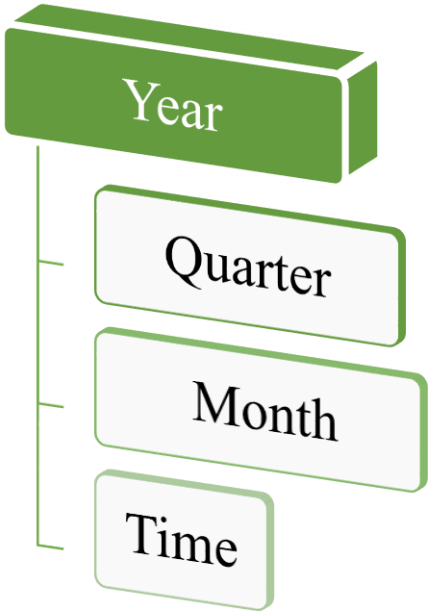

2. Execute **MDX Queries** to analyze data.
3. Validate schema consistency.

## STEP-4: VISUALIZE SCHEMAS

- Open Deployed D5_Movies project using Visual Studio.
- Configured Data Source & Data Source View, establishing connections to the database and defining table relationships.
- Designed database schema diagrams to visualize data structure.
- Validated table relationships to ensure data integrity.
- Created Cubes & Measures, defining fact tables, dimensions, and key performance measures for analysis.

❖ **CONCEPT HIERARCHY**




**Hierarchy 1:** Genre Hierarchy (from Sub Genre to Genre)

**Hierarchy 2**: Time Hierarchy (from Day to Year)

- These **hierarchies** are used in **OLAP (Online Analytical Processing)** for efficient **data aggregation and analysis** in multidimensional models.

## 4.1 STAR SCHEMA:

The **Star Schema** is a de-normalized database schema used in OLAP, where a central **Fact Table** (containing measurable data like Rating or revenue) is directly connected to multiple **Dimension Tables** (such as time, region, genre etc..) in a star-like structure.

### 4.1.1 Design & Visualize the Schema

- Create the **Star Schema** with Fact and Dimension tables.
- Define relationships between tables for efficient querying.

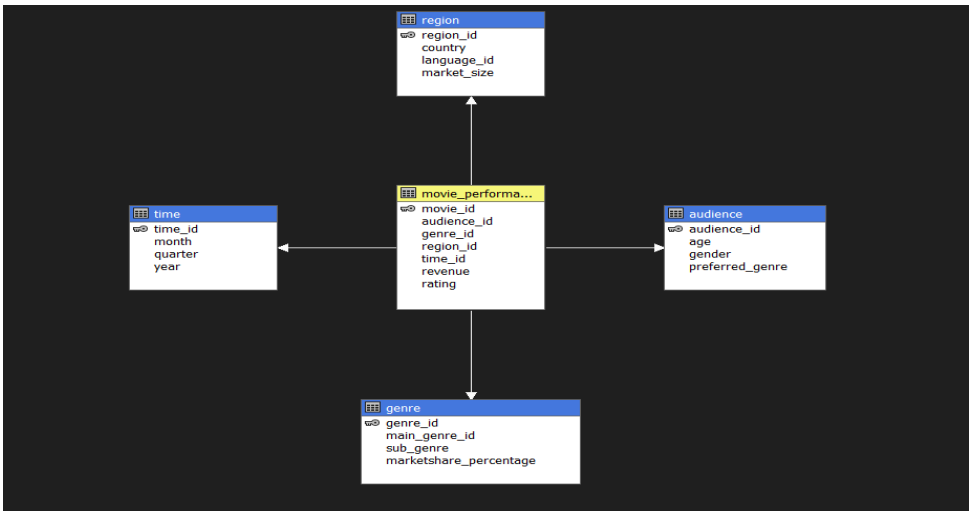



**Fig 9:** "**Star Schema Representation for Music Listening Data Warehouse**"

### 4.1.2 Deploy the Data Warehouse & Load Data

- Store structured data into the data warehouse.
- Ensure ETL (Extract, Transform, Load) processes are completed.

### 4.1.3 Create & Execute OLAP Queries

- Write OLAP queries to perform data analysis.
- Use ROLLUP, CUBE, SLICE, DICE, DRILL-DOWN, and PIVOT operations for multi-dimensional analysis.

### 4.1.4 Perform OLAP Operations

- Run the queries to process large datasets efficiently.
- Perform aggregations, filtering, and transformations on the stored data.

**MDX Queries OLAP operations in STAR SCHEMA:**

**(A) Q:How does the revenue vary between different years?**

**Roll-Up (Aggregation of Genres):**

```
SELECT
{[Measures].[Revenue]} ON COLUMNS,
NONEMPTY( [Time].[Year].Members ) ON
ROWS
FROM [D5_Star Schema];
```

**Output:**

Messages | Results

| | Revenue |
|---|---|
| All | 328272888 |
| 2024 | 180168399 |
| 2025 | 148104489 |

**MDX Execution Time:** 4 ms **SQL Execution Time:** 0.13 seconds

### 4.1.5 Visualize OLAP Results:

Visualizing Music Preferences Using Orange Tool

**Prepare OLAP Output for Visualization**

- Select key OLAP operation results related to music preferences.
- Export the selected data as an **Excel sheet** for further visualization.
- Ensure the dataset includes relevant attributes such as **genre, platform, listening time, and user preferences**.

**REVENUE BASED ON THE YEAR: (rollup.csv) file**

| | A | B |
|---|---|---|
| 1 | Year | Revenue |
| 2 | All | 328272888 |
| 3 | 2024 | 180168399 |
| 4 | 2025 | 148104489 |

- To compare revenue generated based on year

.


**Bar Chart Configuration:**

- **X-Axis**: Year
- **Y-Axis:** Revenue

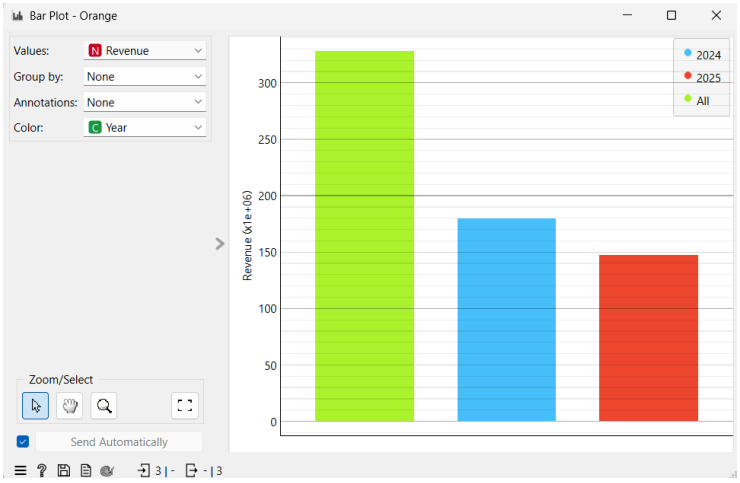



**Fig 10: Bar Chart of Songs Listened by Genre (Classical vs. Pop)**

The bar chart visually represents the total revenue generated across two years: 2024 and 2025. The x-axis represents the years, while the y-axis shows the revenue in millions ($\times 10^6$). Each bar is color-coded by year— blue for 2024, red for 2025, and green for the combined total across both years (labeled as "All").

**(B) What is the revenue for each month under each year?**

**Drill-Down (Detailed Time Analysis):**

```
SELECT
{[Measures].[Revenue]} ON COLUMNS,
NONEMPTY(
[Time].[Year].[Year].Members *
[Time].[Month].[Month].Members
) ON ROWS
FROM [D5_Star Schema];
```

**OUTPUT:**

| | | Revenue |
|---|---|---|
| 2024 | April | 7446643 |
| 2024 | August | 17665378 |
| 2024 | February | 53995461 |
| 2024 | January | 17803209 |
| 2024 | July | 8233884 |
| 2024 | June | 27167365 |
| 2024 | May | 19719723 |
| 2024 | November | 14729082 |
| 2024 | October | 3818853 |
| 2024 | September | 9588801 |
| 2025 | April | 23575892 |
| 2025 | August | 34124070 |
| 2025 | December | 3710324 |
| 2025 | January | 7425345 |
| 2025 | July | 21889130 |

**MDX Execution Time:** 5 ms **SQL Execution Time**: 0.8 seconds

## Visualize OLAP Results:

### Revenue for Each Month

To analyse the Revenue of movies during Months, Years identifying variations and trends across different months.


**Box Plot Configuration:**

- **X-Axis:** Months
- **Y-Axis:** Revenue

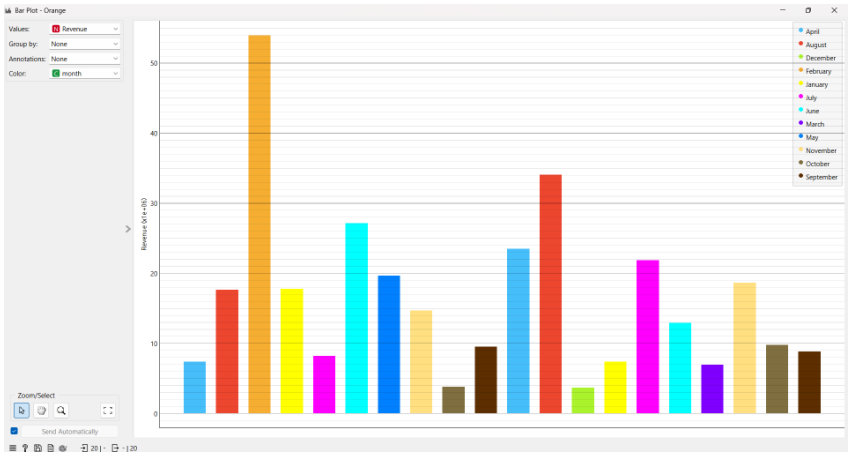


**Fig 11: Monthly Revenue Distribution**

The bar chart shows the revenue generated each month. The x-axis lists the months, while the y-axis represents revenue in millions. Each bar is color-coded by month for clarity. October recorded the highest revenue, followed by June and November. In contrast, months like February, May, and August had noticeably lower revenues. This indicates clear monthly variations, possibly due to seasonal trends or marketing efforts.

**(C) What is the revenue for Sub Genre from Male in the year 2024? Slice (Filtering by Male):**

```
SELECT
{[Measures].[Revenue]} ON COLUMNS,
[Genre].[Sub Genre].Members ON ROWS
FROM [D5_Star Schema]
WHERE ([Time].[Year].[2024], [Audience].[Gender].[Male]);
```

**OUTPUT:**

Messages | Results

| | Revenue |
|---|---|
| All | 68569960 |
| 3D Animation | (null) |
| Crime Drama | (null) |
| Cyberpunk | 9985644 |
| Dark Comedy | 16698503 |
| Historical Drama | 12052737 |
| Martial Arts | (null) |
| Psychological Horror | 3632550 |
| Slapstick | (null) |
| Superhero | 26200526 |
| Unknown | (null) |

**MDX Execution Time:** 7 ms

**SQL Execution Time**: 0.3 seconds

**(D) What is the revenue for January and February of 2024?**

**Dice (Filtering Across Multiple Dimensions):**

```
SELECT
{[Measures].[Revenue]} ON COLUMNS,
{[Time].[Year].[2024]} *
{[Time].[Month].[January],
[Time].[Month].[February]} ON ROWS
FROM [D5_Star Schema];
```

**OUTPUT:**

Messages | Results

| | | Revenue |
|---|---|---|
| 2024 | January | 17803209 |
| 2024 | February | 53995461 |

**MDX Execution Time:** 5 ms **SQL Execution Time**: 0.1 seconds

**(E) How does the revenue compare across months and years? Pivot (Rearranging Dimensions):**

```
SELECT
NONEMPTY([Time].[Month].Members) ON COLUMNS,
NONEMPTY([Time].[Year].Members) ON ROWS
FROM [D5_Star Schema];
```

**OUTPUT:**

Messages | Results

| | All | April | August | December | February | January | July | June | March | May | November | October | September |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| All | 328272888 | 31022535 | 51789448 | 3710324 | 53995461 | 25228554 | 30123014 | 40145483 | 6986956 | 19719723 | 33411441 | 13662565 | 18477384 |
| 2024 | 180168399 | 7446643 | 17665378 | (null) | 53995461 | 17803209 | 8233884 | 27167365 | (null) | 19719723 | 14729082 | 3818853 | 9588801 |
| 2025 | 148104489 | 23575892 | 34124070 | 3710324 | (null) | 7425345 | 21889130 | 12978118 | 6986956 | (null) | 18682359 | 9843712 | 8888583 |

**MDX Execution Time:** 7 ms **SQL Execution Time**: 0.0 second

## 4.2 SNOWFLAKE SCHEMA:

The **Snowflake Schema** provides a more structured and normalized approach than the Star Schema. It reduces data redundancy by splitting up dimension tables into related sub-dimensions. This design enhances data integrity and supports efficient storage in OLAP systems.

### 4.2.1 Design & Visualize the Snowflake Schema

- Identify **Fact Tables** (e.g., Movie Performance, Market Performance)
- Identify **Dimension Tables** (e.g., region, genre, main genre, language, audience, time).
- Normalize dimension tables by breaking them into **sub-dimensions** (e.g., Genre → Main Genre).

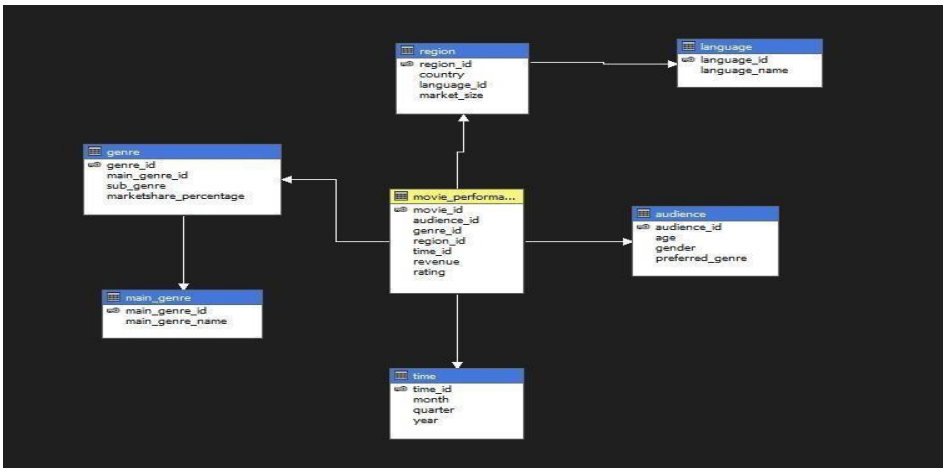



**Fig 12: "Snowflake Schema for Movie Performance Data"**

### 4.2.2 Deploy & Load Data into the Snowflake Schema

- Implement the schema in a **Data Warehouse SnowFlake**
- Ensure proper **data integrity and indexing** for performance.
- Deployed the schema to the Data Warehouse.
- Configured SQL Server Analysis Services (SSAS) for OLAP processing and reporting.

### 4.2.3 Create & Execute OLAP Queries

- Write **SQL queries** for analytical processing:
  1) **ROLLUP** – Aggregate data across different levels.
  2) **CUBE** – Compute multi-dimensional aggregates.
  3) **DRILL-DOWN** – View data at finer granularity.
  4) **SLICE & DICE** – Filter and analyze subsets of data.

### 4.2.4 Perform OLAP Operations

- Use OLAP processing to retrieve and manipulate large datasets efficiently.
- Run complex queries on multi-dimensional data using **MDX (Multi-Dimensional Expressions)** or SQL-based OLAP tools.

**MDX Queries OLAP operations in SNOWFLAKE SCHEMA:**

**(A) What is the total revenue by main genre?**

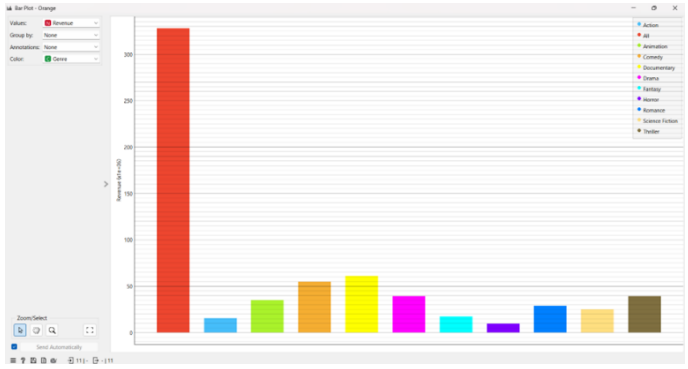
**Roll-Up (Aggregation of Languages):**

```
SELECT
{[Measures].[Revenue]} ON COLUMNS,
NONEMPTY([Genre].[Main Genre Name].Members) ON ROWS
FROM [D5_Snowflake Schema]
```

**OUTPUT:**

Messages | Results

| | Revenue |
|---|---|
| All | 328272888 |
| Action | 15899923 |
| Animation | 35361647 |
| Comedy | 55235876 |
| Documentary | 61135419 |
| Drama | 39304175 |
| Fantasy | 17475069 |
| Horror | 9864990 |
| Romance | 29273390 |
| Science Fiction | 25144341 |
| Thriller | 39578058 |



**MDX Execution Time:** 12 ms

**(B) What is the revenue broken down by main genre and sub-genre?**

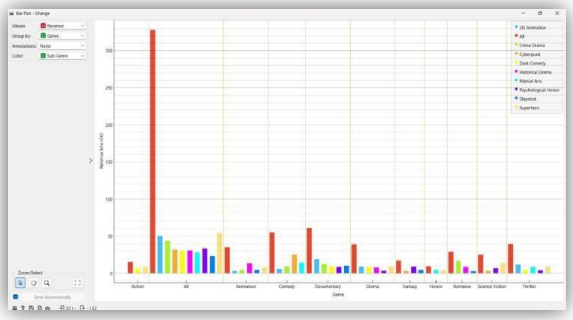
**Drill-Down (Detailed Revenue and Genre Analysis):**

```
SELECT
{[Measures].[Revenue]} ON COLUMNS,
NONEMPTY([Genre].[Main Genre Name].Members * [Genre].[Sub Genre].Members) ON
ROWS FROM [D5_Snowflake Schema];
```

**OUTPUT:**

Messages | Results

| | | Revenue |
|---|---|---|
| All | All | 328272888 |
| All | 3D Animation | 50440725 |
| All | Crime Drama | 44196288 |
| All | Cyberpunk | 32173316 |
| All | Dark Comedy | 30354053 |
| All | Historical Drama | 30932400 |
| All | Martial Arts | 28437076 |
| All | Psychological Horror | 33759692 |
| All | Slapstick | 23660505 |
| All | Superhero | 54318833 |
| Action | All | 15899923 |
| Action | Dark Comedy | 7109702 |
| Action | Superhero | 8790221 |
| Animation | All | 35361647 |
| Animation | 3D Animation | 3710324 |
| Animation | Crime Drama | 5061263 |



**MDX Execution Time:** 6ms

**(C)** What is the Movie Performance Count for Sub Genre in the year 2024?

**Slice (Filtering by Year):**

```
SELECT
{[Measures].[Movie Performance Count]} ON COLUMNS,
[Genre].[Sub Genre].Members ON ROWS
FROM [D5_Snowflake Schema] WHERE
([Time].[Year].[2024]);
```

**OUTPUT:**

Messages | Results

| | Movie Performance Count |
|---|---|
| All | 27 |
| 3D Animation | 1 |
| Crime Drama | 2 |
| Cyberpunk | 5 |
| Dark Comedy | 2 |
| Historical Drama | 2 |
| Martial Arts | 2 |
| Psychological Horror | 5 |
| Slapstick | 3 |
| Superhero | 5 |
| Unknown | (null) |

**MDX Execution Time:** 6ms

**(D) What is the revenue for Thriller & Comedy movies in January and February 2024?**

## Dice (Filtering by Multiple Dimensions):

```
SELECT
{[Measures].[Revenue]} ON COLUMNS, CROSSJOIN(
{[Genre].[Main Genre Name].[Thriller], [Genre].[Main Genre Name].[Comedy]},
{[Time].[Month].[January], [Time].[Month].[February]}
) ON ROWS
FROM [D5_Snowflake Schema];
```

**OUTPUT:**

Messages Results

| | | Revenue |
|---|---|---|
| Thriller | January | 7425345 |
| Thriller | February | 13742424 |
| Comedy | January | 9651375 |
| Comedy | February | 6277361 |

**MDX Execution Time: 5ms**

**(E) How does revenue vary across countries and languages? Pivot :**

```
SELECT
{[Region].[Language Name].Members} ON COLUMNS,
  {[Region].[Country].Members} ON ROWS
FROM [D5_Snowflake Schema];
```

**OUTPUT:**

Messages Results

| | All | Chinese | English | French | German | Hindi | Japanese | Korean | Portuguese | Russian | Spanish | Unknown |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| All | 328272888 | 34751392 | 46083092 | 18301409 | 47823487 | 20396296 | 38394995 | 41342095 | 46781967 | 31185667 | 3212488 | (null) |
| Brazil | 53675233 | (null) | (null) | (null) | 7446643 | (null) | 22081392 | 10863273 | 9651375 | 3632550 | (null) | (null) |
| China | 16435272 | (null) | 3313492 | (null) | 5068669 | (null) | (null) | (null) | (null) | 4840623 | 3212488 | (null) |
| Germany | 16984972 | (null) | 7506294 | (null) | 9478678 | (null) | (null) | (null) | (null) | (null) | (null) | (null) |
| India | 36893295 | 9204286 | (null) | (null) | (null) | 7909449 | 3332011 | 11573965 | 4873584 | (null) | (null) | (null) |
| Japan | 24020200 | (null) | 6130926 | 9588801 | 4592190 | (null) | (null) | 3708283 | (null) | (null) | (null) | (null) |
| Mexico | 57261609 | 9269474 | 10498447 | 8712608 | 7425345 | (null) | (null) | (null) | 17492355 | 3863380 | (null) | (null) |
| Russia | 60969649 | 11793679 | (null) | (null) | 8869020 | 12486847 | 9035951 | 8086872 | (null) | 10697280 | (null) | (null) |
| South Korea | 32044958 | 4483953 | 18633933 | (null) | (null) | (null) | (null) | (null) | 8927072 | (null) | (null) | (null) |
| United States | 29987700 | (null) | (null) | (null) | 4942942 | (null) | 3945641 | 7109702 | 5837581 | 8151834 | (null) | (null) |
| Unknown | (null) | (null) | (null) | (null) | (null) | (null) | (null) | (null) | (null) | (null) | (null) | (null) |

**MDX Execution Time:** 14ms

## 4.3 FACT CONSTELLATION:

A **Fact Constellation Schema** is a complex OLAP schema where multiple fact tables share common dimension tables, allowing for more flexible analysis across different business processes. It combines multiple star schemas into a single structure, with each fact table connecting to shared dimensions.

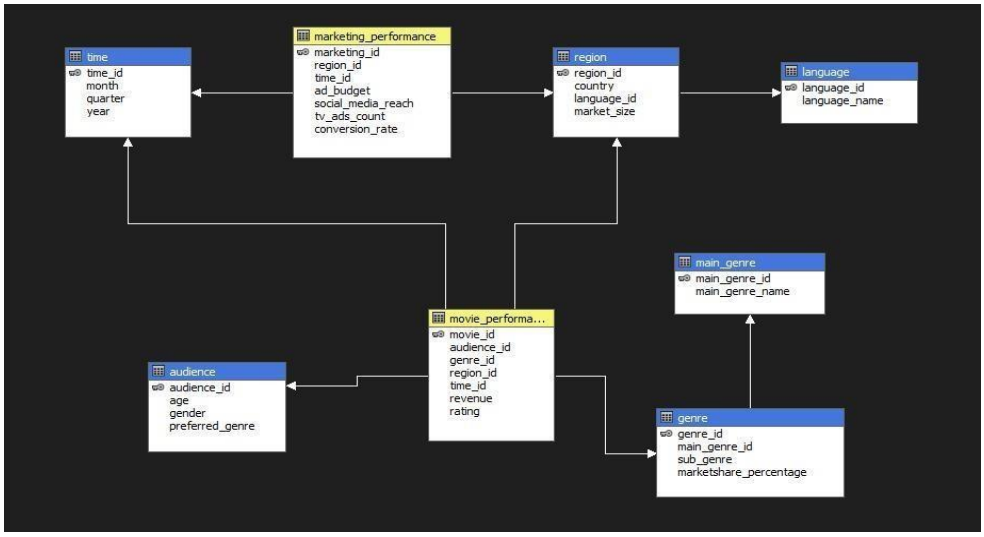



**Fig 13: Fact Constellation Schema for Movie Performance**

This schema represents a **Fact Constellation Model**, where multiple fact tables (Movie Performance & Market Performance) share common dimension tables, allowing detailed analysis of both listening and subscription trends in a music streaming system.

### 4.3.1 Design & Visualize the FACT CONSTELLATION SCHEMA

- Multiple fact tables represent different business processes.
- Dimension tables are shared across multiple fact tables.
- Fact tables have foreign key references to common dimension tables.
- Normalized dimension tables reduce redundancy p
- Time dimension is often shared across fact tables.
- Flexible schema for handling various business processes.
- No direct relationship between fact tables; they connect through shared dimensions.

### 4.3.2 Deploy & Load Data into the Snowflake Schema:

1. **Implement Snowflake Schema** in a data warehouse.
2. **Load Fact and Dimension Tables** into the database.
3. Ensure **data integrity** and optimize performance with proper indexing.
4. **Deploy schema** to the data warehouse.
5. Configure **SQL Server Analysis Services (SSAS)** for OLAP processing and reporting.

### 4.3.3 Create & Execute OLAP Queries:

1. **ROLLUP**: Aggregate data at different levels.
2. **CUBE**: Compute multi-dimensional aggregates.
3. **DRILL-DOWN**: View data with more detail.
4. **SLICE & DICE**: Filter and analyze data subsets.

### 4.3.4 Perform OLAP Operations:

1. Use OLAP tools to retrieve and manipulate large datasets.
2. Run complex queries using **MDX** or **SQL-based OLAP** tools.

**MDX Queries OLAP operations in FACTCONSTELLATION SCHEMA:**

**(A) How do revenue and ratings vary across different year? Roll-Up (Aggregation by Ad Budget and Revenue):**

```
SELECT
{[Measures].[Revenue], [Measures].[Ad Budget]} ON COLUMNS,
   {[Time].[Year].Members}  ON
ROWS      FROM      [D5_Fact
Constellation Schema];
```

**OUTPUT:**

Messages | Results

| | Revenue | Ad Budget |
|---|---|---|
| All | 328272888 | 90506892.85 |
| 2024 | 180168399 | 51459581.27 |
| 2025 | 148104489 | 39047311.58 |
| Unknown | (null) | (null) |

**MDX Execution Time:** 3ms

**(B) What is the revenue and rating broken down by Month?**

**Drill-Down (Revenue Analysis by Year, Quarter and Month):**

```
SELECT
{[Measures].[Revenue], [Measures].[Rating]} ON COLUMNS,
NONEMPTY([Time].[Year].Members * [Time].[Quarter].Members *
[Time].[Month].Members) ON
ROWS
FROM [D5_Fact Constellation Schema];
```

**OUTPUT:**

Messages | Results

| | | | Revenue | Rating |
|---|---|---|---|---|
| 2024 | All | May | 19719723 | 29 |
| 2024 | All | November | 14729082 | 14 |
| 2024 | All | October | 3818853 | 5 |
| 2024 | All | September | 9588801 | 9 |
| 2024 | 1 | All | 42673001 | 46 |
| 2024 | 1 | February | 12902024 | 15 |
| 2024 | 1 | July | 8233884 | 7 |
| 2024 | 1 | May | 12610021 | 19 |
| 2024 | 1 | November | 8927072 | 5 |
| 2024 | 2 | All | 62709369 | 84 |
| 2024 | 2 | August | 8875157 | 10 |
| 2024 | 2 | February | 16598895 | 20 |
| 2024 | 2 | January | 9651375 | 10 |
| 2024 | 2 | June | 17963079 | 30 |
| 2024 | 2 | November | 5802010 | 9 |
| 2024 | 2 | October | 3818853 | 5 |
| 2024 | 3 | All | 22924029 | 22 |
| 2024 | 3 | April | 7446643 | 6 |
| 2024 | 3 | August | 8790221 | 9 |
| 2024 | 3 | February | 6687165 | 7 |
| 2024 | 4 | All | 51862000 | 62 |
| 2024 | 4 | February | 17807377 | 29 |
| 2024 | 4 | January | 8151834 | 7 |

**MDX Execution Time:** 6m

**(C). What are the revenue and conversion rate for movies advertised in the USA? Slice (Filtering revenue and conversion rate):**

```
SELECT
   {[Measures].[Revenue], [Measures].[Conversion Rate]} ON
COLUMNS FROM [D5_Fact Constellation Schema]
WHERE ([Region].[Country].[United States]);
```

**OUTPUT:**

Messages | Results

| Revenue | Conversion Rate |
|---|---|
| 29987700 | 17.5 |

**MDX Execution Time:** 9ms

**(D). What are the revenue and ad budget for Action & Thriller movies?**

**Dice (Filtering by Genre and Ad Budget):**

```
SELECT
{[Measures].[Revenue], [Measures].[Ad Budget]} ON COLUMNS,
   {[Genre].[Main Genre Name].[Action], [Genre].[Main Genre Name].[Thriller]} ON
ROWS FROM [D5_Fact Constellation Schema];
```

**OUTPUT:**

Messages | Results

| | Revenue | Ad Budget |
|---|---|---|
| Action | 15899923 | 90506892.85 |
| Thriller | 39578058 | 90506892.85 |

**MDX Execution Time:** 6ms

(C) **Compare TV ads and social media reach across different genres Pivot (Tv Ads**

**Count and Genre):**

```
SELECT
{[Measures].[TV Ads Count]} ON COLUMNS,
   {[Genre].[Main Genre Name].Members} ON
ROWS FROM [D5_Fact Constellation Schema];
```

**OUTPUT:**

Messages | Results

| | Tv Ads Count |
|---|---|
| All | 1223 |
| Action | 1223 |
| Animation | 1223 |
| Comedy | 1223 |
| Documentary | 1223 |
| Drama | 1223 |
| Fantasy | 1223 |
| Horror | 1223 |
| Romance | 1223 |
| Science Fiction | 1223 |
| Thriller | 1223 |
| Unknown | 1223 |

**MDX Execution Time:** 11ms

# STEP 5: PERFORM DATA MINING

**Clustering of Movie Performance and Marketing performance measures:**

**Objective:**

We aim to explore and discover **natural groupings of movies** based on their **performance and marketing-related features** using **unsupervised clustering techniques**. This helps in identifying distinct categories of films—such as high-revenue blockbusters or low-budget releases—based on similar characteristics.

## 5.1 DATA PREPARATION FOR CLUSTERING

- **Dataset Features:**

  1. **Movie Attributes:** Genre, Language, Duration, Release Year
  2. **Performance Metrics:** Revenue, Rating
  3. **Marketing Metrics:** Ad Budget, Social Media Reach, Market Share

- **Data Preprocessing:**

  1. Handle missing values.
  2. Normalize numerical data.
  3. Encode categorical data.

## 5.2 SELECTING CLUSTERING ALGORITHMS

- The dataset was preprocessed using **Impute** and **Preprocess** widgets in Orange to handle missing values, normalize numeric features, and encode categorical data.
- The cleaned data was then passed into multiple **clustering models** to identify natural groupings and patterns within the dataset.

We used the following **unsupervised machine learning algorithms**:

- **K-Means Clustering**
- **Hierarchical Clustering**
- **DBSCAN (Density-Based Spatial Clustering)**


These models help reveal hidden structures in the data by grouping similar movies based on attributes like **revenue**, **rating**, **ad budget**, **social media reach**, and **market share**. The clusters formed provide meaningful insights into different categories of movies, which can be used for **performance evaluation and marketing strategy planning**, rather than direct prediction of user behaviour.

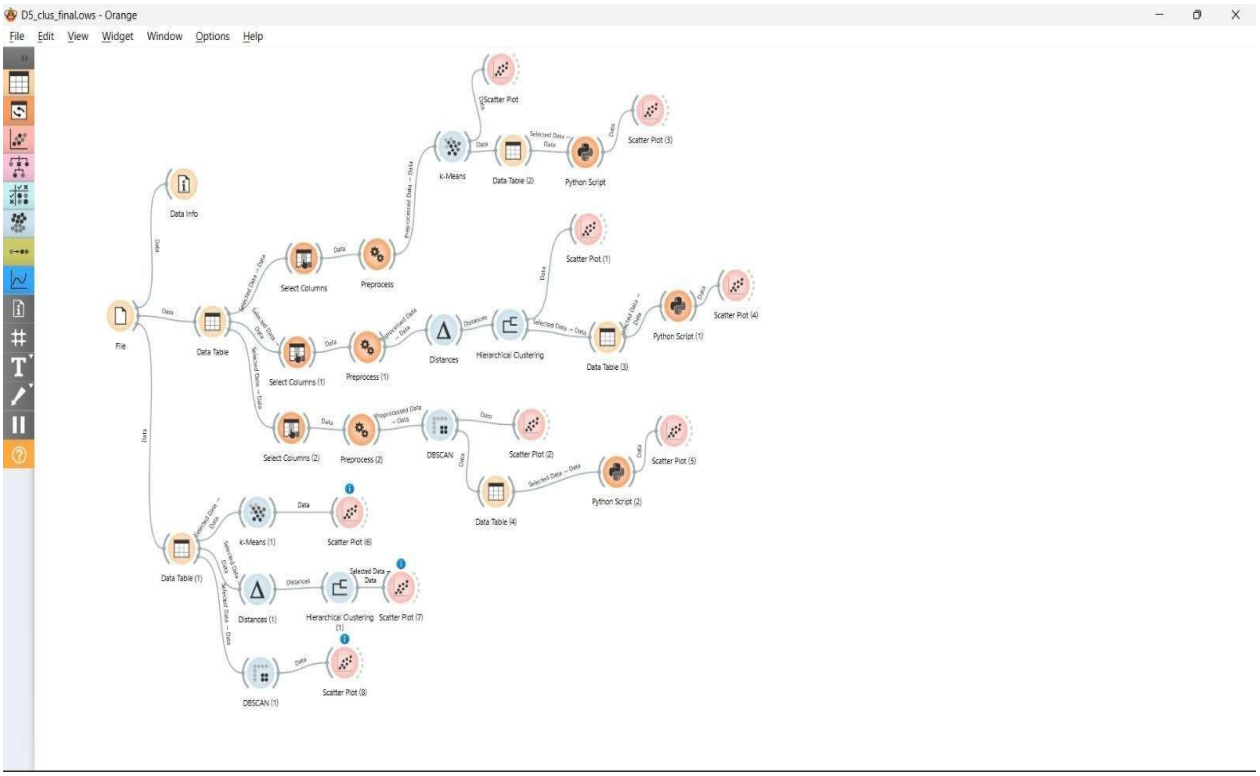
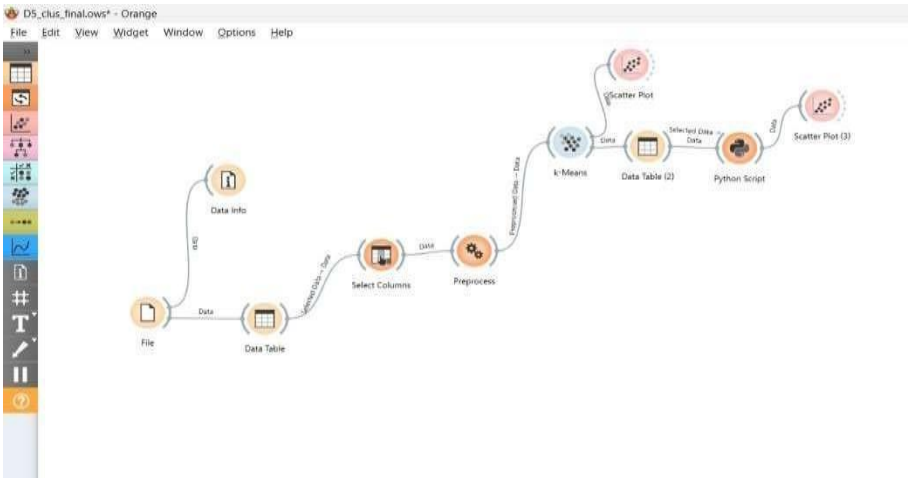



**Fig 14: Workflow for Movie Performance Dataset Clustering in Orange**

## 5.3 Workflows of the models

### 1. K-Means Clustering

K-Means is a **partition-based clustering algorithm** that divides the dataset into **K distinct clusters**, where each data point belongs to the cluster with the **nearest mean (centroid)**.



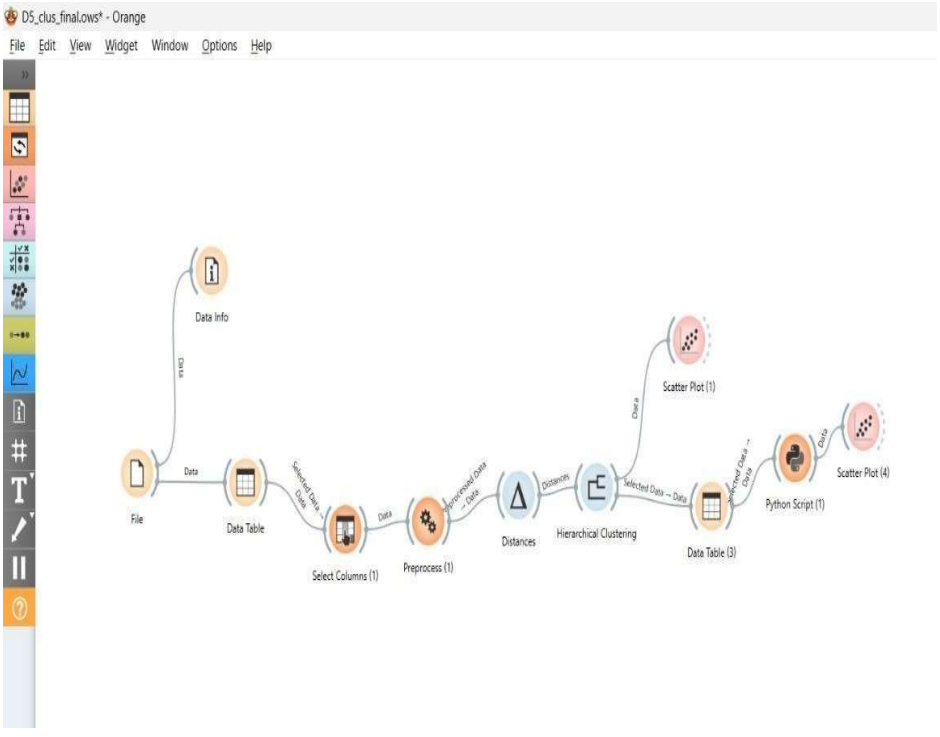

This workflow can be represented in the form of a python script by using the python script widget. The python script for K-Means clustering is as follows:

```
Editor
def python_script(in_data):
 1 import numpy as np
 2 from sklearn.cluster import KMeans
 3 import Orange
 4
 5 # Step 1: Extract feature data from the pipeline
 6 X = in_data.X
 7
 8 # Step 2: Run KMeans clustering
 9 kmeans = KMeans(n_clusters=3, random_state=0)  # change 4 to any number of clusters you want
10 labels = kmeans.fit_predict(X)
11
12 # Step 3: Create a new Discrete Variable for the cluster label
13 cluster_values = [str(i) for i in np.unique(labels)]
14 cluster_var = Orange.data.DiscreteVariable(name="Cluster", values=cluster_values)
15 cluster_col = np.array([[label] for label in labels])
16
17 # Step 4: Construct a new domain with the original attributes + cluster as meta
18 domain = Orange.data.Domain(in_data.domain.attributes, in_data.domain.class_vars, [cluster_var])
19
20 # Step 5: Output the final table
21 out_data = Orange.data.Table(domain, in_data.X, in_data.Y, cluster_col)
22

    return out_data, out_learner, out_classifier, out_object
```

## 2. Hierarchical Clustering

Hierarchical Clustering builds a **tree-like structure of nested clusters (dendrogram)** by either:

- **Agglomerative** (bottom-up): each data point starts as its own cluster and merges step-by-step.
- **Divisive** (top-down): all points start in one cluster and split iteratively.

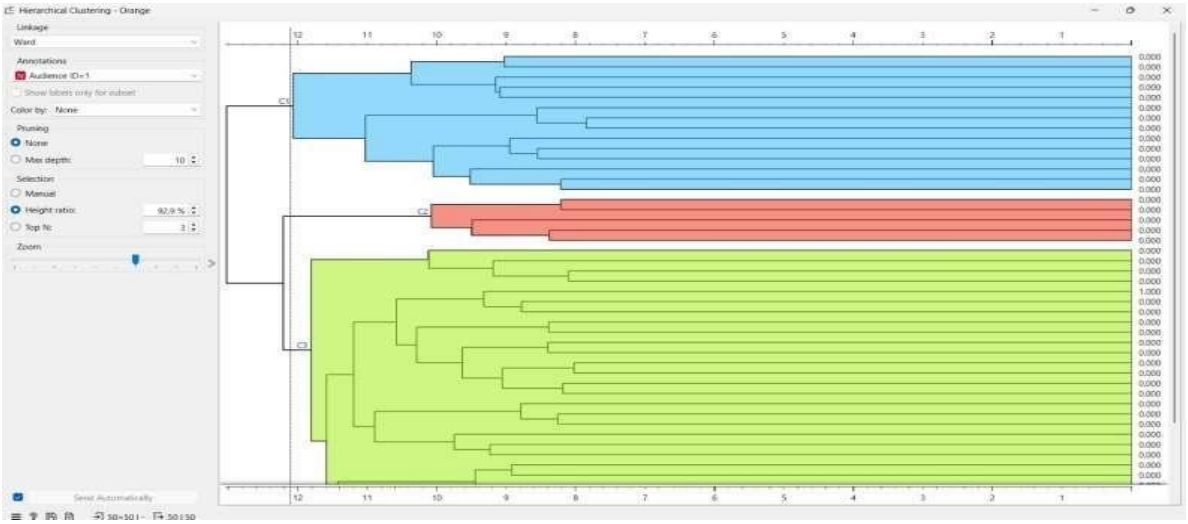


The python script can be viewed using the python script widget.

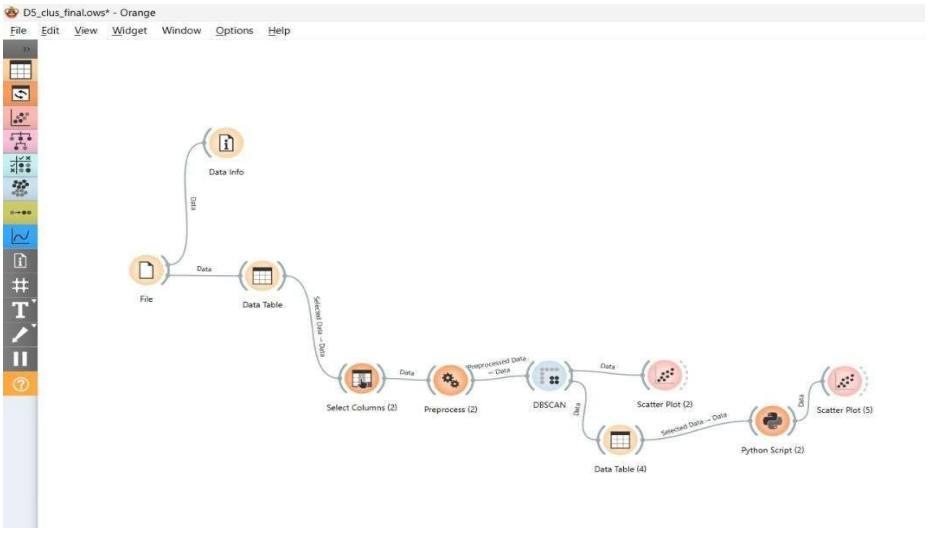
Editor

```
def python_script(in_data):
```

```
import numpy as np
from Orange.data import Table, Domain, DiscreteVariable
from sklearn.cluster import AgglomerativeClustering

# Get the input data from Orange
X = np.array(in_data.X)

# Perform Agglomerative Clustering
model = AgglomerativeClustering(n_clusters=3, linkage='ward', metric='euclidean')
labels = model.fit_predict(X)

# Create a new domain with original variables + discrete ClusterLabel
cluster_var = DiscreteVariable("ClusterLabel", values=["0", "1", "2"])
new_domain = Domain(in_data.domain.attributes + (cluster_var,), in_data.domain.class_vars, in_data.domain.metas)

# Create the new data table with cluster labels added as a new column
out_data = Table(new_domain, np.hstack((in_data.X, labels.reshape(-1, 1))), in_data.Y, in_data.metas)

```




## 3. DBScan

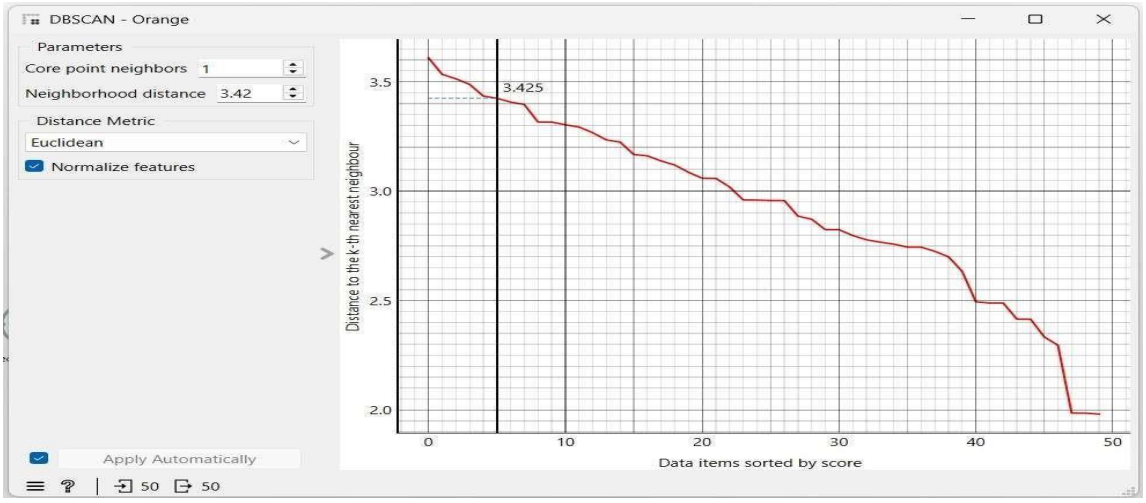



DBSCAN is a **density-based clustering algorithm** that groups together points that are closely packed (have many nearby neighbors) and marks outliers as noise.

The python script for DBScan is as follows:

```
Editor
def python_script(in_data):
```

```python
import numpy as np
from sklearn.cluster import DBSCAN
from Orange.data import Table, Domain, DiscreteVariable

# Get input data from Orange
X = np.array(in_data.X)

# Run DBSCAN with chosen parameters
db = DBSCAN(eps=0.5, min_samples=5).fit(X)
labels = db.labels_

# Convert cluster labels to strings so they can be used as a discrete variable
unique_labels = sorted(set(labels))
label_names = [str(lbl) for lbl in unique_labels]
cluster_var = DiscreteVariable("ClusterLabel", values=label_names)

# Create a new domain with the added cluster label
new_domain = Domain(in_data.domain.attributes + (cluster_var,), in_data.domain.class_vars, in_data.domain.metas)

# Append labels as the last column of the data table
out_data = Table(new_domain, np.hstack((in_data.X, labels.reshape(-1, 1))), in_data.Y, in_data.metas)

```




## 5.4 MODEL EVALUATION METRICS

To determine the effectiveness of the clustering models, we evaluate them using the following metrics:

- **Scatter Plot (Visual Evaluation):**
  Used to visualize how clearly clusters are separated in 2D space using PCA.
  Helps in interpreting cluster shapes, overlap, and compactness.
- **Silhouette Score:**
  Measures how well each data point fits within its assigned cluster compared to other clusters.
  Values range from **-1 to +1**. A higher score indicates **better-defined and well-separated clusters**.
- **Cluster Size Distribution:**
  Provides insight into how balanced or skewed the cluster sizes are.
  Disproportionate cluster sizes may indicate issues in model performance or data imbalance.
- **Distance Map / Hierarchical Dendrogram:**
  Used in hierarchical clustering to show the **proximity between clusters** and help decide the **optimal number of clusters** based on visual grouping.

## 5.5 EXPERIMENT ANALYSIS

**Step 1: Model Training & Clustering**

• Apply **clustering algorithms** such as **K-Means**, **Hierarchical Clustering**, and **DBSCAN** to group users based on their listening behavior, demographics, and engagement metrics.
• The models were trained using **unsupervised learning**, with no predefined labels, to uncover natural groupings in the data.

**Step 2: Evaluate & Compare Clustering Models**

• Use the **Silhouette Score** widget to measure how well-separated and compact the clusters are.
• Evaluate visual outputs using **Scatter Plot**, **Distance Map**, and **Cluster Size Distribution** to interpret the quality and structure of clusters.
• Compare clustering results across different algorithms to determine which technique provides the **most meaningful and distinct segmentation** of user movie preferences.

| Algorithm | Preprocessing Applied | SSE | Silhouette Score | #Clusters | Noise Points | Notes/Comments |
|---|---|---|---|---|---|---|
| K-Means | No | 12,04,80,36,12,310.84 | 0.402 | 3 | 0 | Well-separated clusters |
| K-Means | Yes | 4,20,92,73,69,43,109.90 | 0.155 | 3 | 0 | After preprocessing: higher SSE due to scaling |
| Hierarchical | No | 8,07,80,61,91,797.16 | 0.129 | 3 | 0 | Moderate separation, visible overlap between clusters |
| Hierarchical | Yes | 215.76 | 0.039 | 3 | 0 | Lower SSE, but weak cluster separation |
| DBSCAN | No | 1,75,23,18,40,12,396.60 | 0.185 | 3 | 8 | Noisy data affected clusters; less compact clustering |
| DBSCAN | Yes | 248.46 | 0.162 | 3 | 8 | Moderate clusters; SSE decreased after preprocessing |

1. **Without Preprocessing:**

Before preprocessing, the data is in its raw form — meaning it likely includes features with varying scales, possibly missing values, or noise. In this state, distance-based clustering algorithms like K-Means, Hierarchical, and DBSCAN may behave unpredictably. For instance, features with larger numeric ranges can dominate the clustering process, leading to biased or misleading results. Clusters might appear well- separated due to the scale differences rather than actual data distribution. Silhouette Scores may appear higher because of this artificial separation, and the model might overlook subtler patterns from lower-range features.

**2. With Preprocessing (Using Sampler):**

After preprocessing, steps such as normalization or scaling are applied to ensure all features contribute equally to distance calculations. This transformation typically leads to more reliable clustering results, especially for DBSCAN, which is highly sensitive to feature scales. Preprocessing makes the data more uniform, which helps in accurately detecting true groupings and noise points. While Silhouette Scores may slightly decrease because clusters are no longer inflated by dominant features, the overall clustering becomes more meaningful and interpretable.

In essence, **preprocessing removes biases introduced by raw feature scales** and enhances the fairness, robustness, and generalizability of clustering results. It's a crucial step in any serious data mining or machine learning workflow.

## 5.5 VISUALIZATION METRICS FOR CLUSTERING MODEL

To evaluate and interpret the clustering results from unsupervised learning models (such as K-Means), we use visualization tools like scatter plots to understand how user data segments based on revenue and ad budget parameters.

**Scatter Plot for Movie Segmentation by Revenue and Ad Budget**

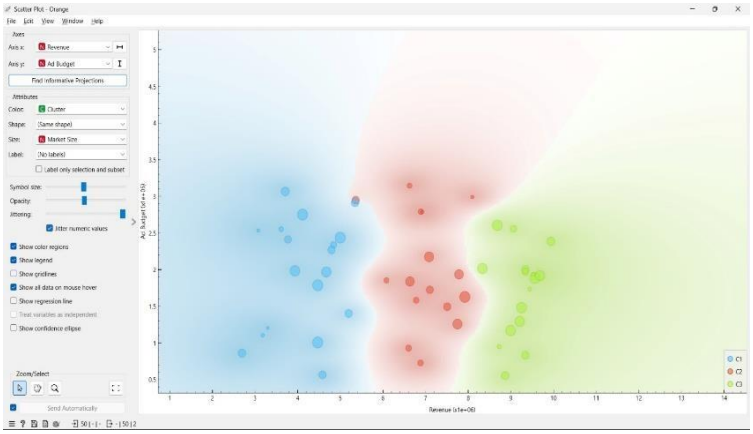


**Fig 15: Clustering of Movies Based on Revenue and Advertising Budget**

This scatter plot shows the output of the **K-Means clustering algorithm**, where each point represents a movie, clustered into one of three groups based on total revenue and ad spending:

- **X-axis** = Revenue
- **Y-axis** = Ad Budget
- **Color** = Cluster (C1: Blue, C2: Red, C3: Green)
- **Size** = Market Size (Bubble size)

📊 **Clustering Analysis Summary**:

🔵 **Cluster C1 (Blue):**
Represents movies with **low revenue (1–5 million)** but **medium to high ad budgets (1.5–4.5 million)**. These films likely suffered from **ineffective marketing strategies** or **un-engaging content**, leading to poor returns despite significant promotional efforts.

🔴 **Cluster C2 (Red):**
Comprises movies earning **medium revenue (5–8.5 million)** with a **moderate ad budget (1.5–3.5 million)**. These are **average performers**, showing a fairly **proportional relationship** between marketing spend and revenue—indicating a **balanced ROI**.

🟩 **Cluster C3 (Green):**
Includes **high-revenue films (9–13 million)** with **moderate ad budgets (1.5–3 million)**. These movies demonstrate **efficient targeting and strong audience appeal**, delivering **high ROI** through **cost- effective marketing.**

## 5.6 CONCLUSION

The objective of this experiment was to segment movies based on their revenue performance and advertising budgets using unsupervised clustering techniques — **K-Means**, **Hierarchical Clustering**, and **DBSCAN**. These algorithms grouped movies with similar performance metrics to identify meaningful patterns and potential insights into marketing effectiveness and return on investment (ROI).

The dataset underwent **preprocessing**, which involved handling missing values, normalization, and encoding of categorical data. To tackle class imbalance, **SMOTE** was applied, ensuring a more balanced training dataset.

Among all the models, **Gradient Boosting** achieved the **highest accuracy** in the **Test & Score** evaluation, accurately predicting the correct listener types. However, when tested using **Data Sampler** with a smaller portion of the dataset, its accuracy decreased. This drop was due to the **reduced number of rows** available for training and testing, which affected the model's ability to generalize effectively in that scenario.

To support the evaluation, visual tools such as **scatter plots**, **confusion matrices**, and **ROC curves** were used. These helped in interpreting how well the models performed and how user preferences aligned with the predicted listener types.

In conclusion, the project successfully classified users into their respective **listener types**. **Gradient Boosting** proved to be the most effective model during full evaluation, offering valuable insights for building personalized music experiences and recommendation system.

# PART-B

## Time-Course Analysis of Yeast Gene Expression using Classification

**Abstract:**

This project focuses on classifying yeast genes based on time-course expression data using supervised machine learning. The dataset contains expression levels of genes across various time points, with each gene labelled by its biological function. Using Orange, classification models such as k-NN, Naïve Bayes, and Random Forest were applied. The models were evaluated using accuracy, precision, recall, and F1-score. Results show that time-based gene expression data can effectively be used for functional gene classification.

**Methodology:**

### 1. DATA IMPORT AND CLEANING

• **File Widget**: Loads the yeast gene expression dataset containing gene samples labeled by their biological function (e.g., Proteasome, Ribosome, etc.).

• **Data Table**: Displays gene expression values across multiple time points (`alpha 0` to `alpha 119`), with each row representing a gene and each column representing a time-specific expression value.

• **Data Info**: Summarizes dataset metadata, showing the number of gene instances, total attributes, and any missing values.

The dataset contains **186 instances** (genes) and **79 features** (time-point-based expression levels), along with the **target class**: gene function.

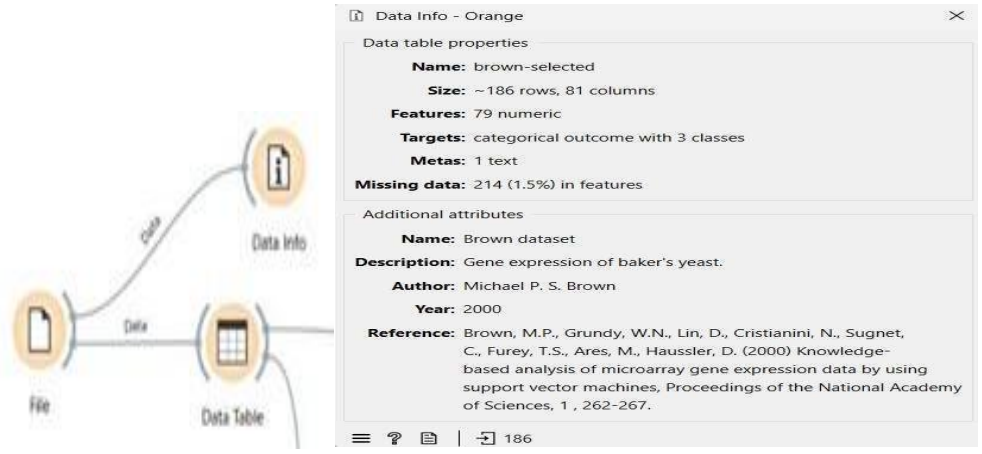



**Fig 16: Data Table Properties in Orange**

• The dataset **"brown-selected"** contains **186 rows** (genes) and **80 columns**.

• It includes **79 numeric features** (time-point expression values) and **1 categorical target** (gene function).
• The dataset contains **some missing values**, which were handled during preprocessing.

| | function | gene | alpha 0 | alpha 7 | alpha 14 | alpha 21 | alpha 28 | alpha 35 | alpha 42 | alpha 49 | alpha 56 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | Proteas | YGR270W | ? | -0.023 | 0.057 | 0.007 | 0.018 | -0.057 | 0.009 | -0.034 | -0.0 |
| 2 | Proteas | YIL075C | -0.031 | -0.031 | -0.060 | 0.037 | -0.071 | -0.018 | -0.026 | -0.052 | 0.0 |
| 3 | Proteas | YDL007W | -0.013 | ? | 0.067 | -0.025 | 0.017 | 0.008 | -0.042 | 0.013 | 0.1 |
| 4 | Proteas | YER094C | 0.003 | 0.025 | 0.067 | 0.083 | -0.010 | 0.080 | -0.019 | 0.057 | 0.0 |
| 5 | Proteas | YFR004W | -0.068 | -0.003 | -0.041 | 0.022 | -0.200 | -0.011 | -0.092 | -0.016 | 0.0 |
| 6 | Proteas | YDR427W | -0.012 | -0.009 | -0.009 | ? | -0.051 | 0.042 | -0.069 | 0.119 | 0.0 |
| 7 | Proteas | YKL145W | 0.012 | 0.008 | -0.006 | -0.025 | 0.029 | -0.019 | 0.062 | -0.002 | 0.0 |
| 8 | Proteas | YGL048C | 0.067 | -0.064 | 0.011 | 0.022 | 0.050 | -0.061 | 0.086 | -0.056 | 0.0 |
| 9 | Proteas | YFR050C | 0.093 | 0.027 | 0.044 | 0.066 | -0.049 | -0.011 | -0.063 | 0.019 | 0.0 |
| 10 | Proteas | YDL097C | 0.062 | 0.002 | 0.050 | 0.019 | 0.033 | -0.033 | 0.033 | ? | 0.0 |
| 11 | Proteas | YOR259C | -0.037 | -0.122 | 0.030 | -0.007 | 0.017 | 0.052 | 0.017 | -0.010 | 0.0 |
| 12 | Proteas | YPR108W | -0.016 | -0.051 | 0.073 | 0.064 | -0.051 | 0.098 | 0.031 | 0.036 | 0.0 |
| 13 | Proteas | YER021W | 0.012 | 0.008 | 0.043 | -0.002 | ? | ? | -0.065 | 0.047 | 0.0 |
| 14 | Proteas | YGR253C | -0.053 | 0.167 | -0.072 | -0.024 | -0.130 | 0.010 | -0.024 | -0.024 | 0.0 |
| 15 | Proteas | YGL011C | 0.011 | -0.017 | 0.045 | 0.045 | 0.003 | 0.065 | ? | 0.079 | 0.1 |
| 16 | Proteas | YMR314W | -0.022 | -0.048 | -0.041 | -0.041 | -0.086 | -0.029 | ? | -0.101 | -0.0 |
| 17 | Proteas | YGR135W | -0.002 | -0.009 | -0.022 | 0.052 | -0.083 | 0.031 | -0.015 | -0.007 | 0.0 |
| 18 | Proteas | YER012W | 0.045 | 0.041 | 0.056 | 0.043 | 0.002 | -0.032 | 0.081 | 0.051 | 0.0 |
| 19 | Proteas | YPR103W | -0.002 | -0.048 | 0.017 | -0.041 | -0.022 | -0.053 | 0.051 | -0.014 | 0.0 |
| 20 | Proteas | YJL001W | 0.014 | 0.002 | -0.009 | -0.051 | -0.028 | -0.093 | -0.002 | -0.098 | -0.0 |
| 21 | Proteas | YOR362C | -0.042 | 0.062 | -0.030 | 0.030 | -0.045 | -0.050 | 0.020 | -0.042 | 0.0 |
| 22 | Proteas | YOR157C | -0.026 | -0.069 | 0.002 | -0.032 | -0.002 | -0.082 | 0.045 | -0.049 | |
| 23 | Proteas | YOL038W | -0.056 | -0.124 | -0.043 | -0.059 | -0.074 | -0.028 | 0.002 | -0.041 | -0.0 |
| 24 | Proteas | YBL041W | -0.021 | -0.070 | -0.075 | -0.093 | -0.084 | -0.126 | 0.016 | 0.105 | -0.0 |
| 25 | Proteas | YHR200W | -0.102 | -0.096 | -0.075 | -0.120 | -0.051 | -0.102 | 0.060 | -0.075 | -0.0 |
| 26 | Proteas | YDR394W | 0.007 | -0.067 | -0.016 | -0.051 | -0.041 | -0.113 | -0.002 | -0.099 | -0.0 |
| 27 | Proteas | YOR117W | 0.019 | -0.077 | 0.106 | 0.019 | 0.074 | -0.114 | 0.021 | 0.043 | 0.0 |

**Fig 17: Data Table with Missing Values**

## 2. DATA PREPROCESSING

### Handling Missing Values

The dataset contained missing values in several time-point expression features. To ensure the quality of the input data, we used the **Impute** widget in Orange. This widget automatically replaces missing values using a selected strategy — in this case, **mean imputation**, where missing values were filled with the average value of the respective feature. This step helps maintain data integrity and ensures consistent model performance during classification.

### Normalization

• The dataset's numeric features (gene expression levels) were normalized using the **"Normalize to interval [0, 1]"** option in the Preprocess widget.
• This transformation scales all values to a common range between **0 and 1**, which improves the performance of machine learning models, especially those sensitive to feature scales such as **k-NN** and **SVM**.
• Normalization ensures that all time-point features contribute equally during model training and prevents dominance by higher-magnitude values.

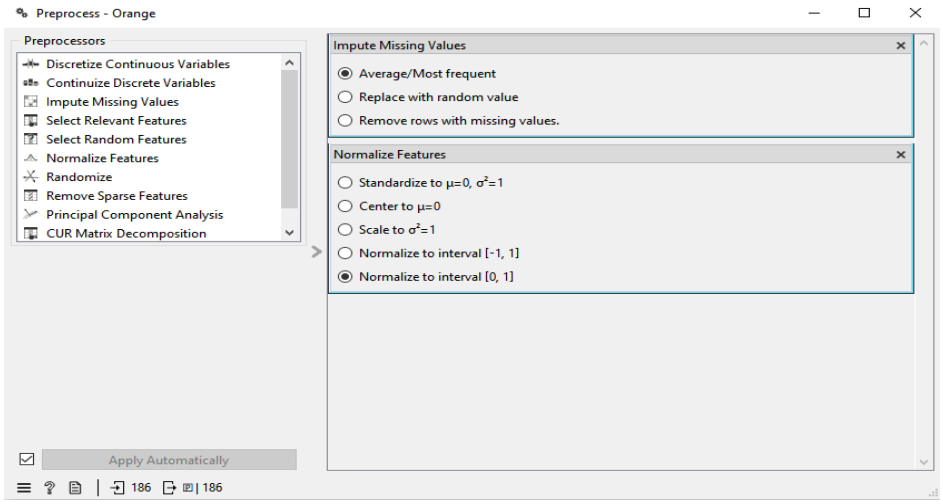



**Fig 18: Preprocessing**

| | function | gene | alpha 0 | alpha 7 | alpha 14 | alpha 21 | alpha 28 | alpha 35 | alpha 42 | alpha 49 | alpha 56 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | Proteas | YGR270W | 0.43927 | 0.34949 | 0.67958 | 0.57285 | 0.78986 | 0.42804 | 0.60104 | 0.42697 | 0.389 |
| 2 | Proteas | YIL075C | 0.36034 | 0.33333 | 0.26761 | 0.67219 | 0.46739 | 0.57196 | 0.41969 | 0.35955 | 0.531 |
| 3 | Proteas | YDL007W | 0.41061 | 0.35282 | 0.71479 | 0.46689 | 0.78623 | 0.66790 | 0.33679 | 0.60300 | 0.920 |
| 4 | Proteas | YER094C | 0.45531 | 0.44646 | 0.71479 | 0.82450 | 0.68841 | 0.93358 | 0.45596 | 0.76779 | 0.669 |
| 5 | Proteas | YFR004W | 0.25698 | 0.38990 | 0.33451 | 0.62252 | 0.000 | 0.59779 | 0.07772 | 0.49438 | 0.489 |
| 6 | Proteas | YDR427W | 0.41341 | 0.37778 | 0.44718 | 0.50879 | 0.53986 | 0.79336 | 0.19689 | 1.000 | 0.841 |
| 7 | Proteas | YKL145W | 0.48045 | 0.41212 | 0.45775 | 0.46689 | 0.82971 | 0.56827 | 0.87565 | 0.54682 | 0.552 |
| 8 | Proteas | YGL048C | 0.63408 | 0.26667 | 0.51761 | 0.62252 | 0.90580 | 0.41328 | 1.000 | 0.34457 | 0.594 |
| 9 | Proteas | YFR050C | 0.70670 | 0.45051 | 0.63380 | 0.76821 | 0.54710 | 0.59779 | 0.22798 | 0.62547 | 0.694 |
| 10 | Proteas | YDL097C | 0.62011 | 0.400 | 0.65493 | 0.61258 | 0.84420 | 0.51661 | 0.72539 | 0.52547 | 0.774 |
| 11 | Proteas | YOR259C | 0.34358 | 0.14949 | 0.58451 | 0.52649 | 0.78623 | 0.83026 | 0.64249 | 0.51685 | 0.539 |
| 12 | Proteas | YPR108W | 0.40223 | 0.29293 | 0.73592 | 0.76159 | 0.53986 | 1.000 | 0.71503 | 0.68914 | 0.652 |
| 13 | Proteas | YER021W | 0.48045 | 0.41212 | 0.63028 | 0.54305 | 0.64220 | 0.54823 | 0.21762 | 0.73034 | 0.769 |
| 14 | Proteas | YGR253C | 0.29888 | 0.73333 | 0.22535 | 0.47020 | 0.25362 | 0.67528 | 0.43005 | 0.46442 | 0.485 |
| 15 | Proteas | YGL011C | 0.47765 | 0.36162 | 0.63732 | 0.69868 | 0.73551 | 0.87823 | 0.51903 | 0.85019 | 1.0 |
| 16 | Proteas | YMR314W | 0.38547 | 0.29899 | 0.33451 | 0.41391 | 0.41304 | 0.53137 | 0.51903 | 0.17603 | 0.334 |
| 17 | Proteas | YGR135W | 0.44134 | 0.37778 | 0.40141 | 0.72185 | 0.42391 | 0.75277 | 0.47668 | 0.52809 | 0.619 |
| 18 | Proteas | YER012W | 0.57263 | 0.47879 | 0.67606 | 0.69205 | 0.73188 | 0.52030 | 0.97409 | 0.74532 | 0.715 |
| 19 | Proteas | YPR103W | 0.44134 | 0.29899 | 0.53873 | 0.41391 | 0.64493 | 0.44280 | 0.81865 | 0.50187 | 0.485 |
| 20 | Proteas | YJL001W | 0.48603 | 0.400 | 0.44718 | 0.38079 | 0.62319 | 0.29520 | 0.54404 | 0.18727 | 0.213 |
| 21 | Proteas | YOR362C | 0.32961 | 0.52121 | 0.37324 | 0.64901 | 0.56159 | 0.45387 | 0.65803 | 0.39700 | 0.518 |
| 22 | Proteas | YOR157C | 0.37430 | 0.25657 | 0.48592 | 0.44371 | 0.71739 | 0.33579 | 0.78756 | 0.37079 | 0.498 |
| 23 | Proteas | YOL038W | 0.29050 | 0.14545 | 0.32746 | 0.35430 | 0.45652 | 0.53506 | 0.56477 | 0.40075 | 0.246 |
| 24 | Proteas | YBL041W | 0.38827 | 0.25455 | 0.21479 | 0.24172 | 0.42029 | 0.17343 | 0.63731 | 0.94757 | 0.368 |
| 25 | Proteas | YHR200W | 0.16201 | 0.20202 | 0.21479 | 0.15232 | 0.53986 | 0.26199 | 0.86528 | 0.27341 | 0.330 |
| 26 | Proteas | YDR394W | 0.46648 | 0.26061 | 0.42254 | 0.38079 | 0.57609 | 0.22140 | 0.54404 | 0.18352 | 0.389 |
| 27 | Proteas | YOR117W | 0.500 | 0.24040 | 0.85211 | 0.61258 | 0.99275 | 0.21771 | 0.66321 | 0.71536 | 0.677 |

**Fig 19: Data Table after preprocessing**

## 3. MODEL TRAINING

The classification workflow was applied to the **brown-corpus** dataset, which contains gene expression profiles labelled with two outcome classes. Multiple machine learning models were trained to classify the samples accurately. The following classification models were used:

1. **Support Vector Machine (SVM)**
2. **k-Nearest Neighbors (k-NN)**
3. **Random Forest**
4. **Logistic Regression**
5. **Naïve Bayes**
6. **Decision Tree**

| Model | Description | Strengths |
|---|---|---|
| **Support Vector Machine (SVM)** | Finds the optimal hyper plane for classification | Works well with high-dimensional data |
| **k-Nearest Neighbors (k-NN)** | Classifies samples based on nearest neighbors | Simple and interpretable |
| **Random Forest** | Uses multiple decision trees for classification | Handles missing data well, reduces overfitting |
| **Logistic Regression** | Estimates probabilities for classification | Works well with linear relationships |
| **Naïve Bayes** | Applies a probabilistic classifier based on Bayes' Theorem with the assumption of feature independence | fast training, effective with high-dimensional data, and robust to irrelevant features |
| **Decision Tree** | Splits data based on feature values for classification | Easy to interpret but prone to overfitting |

**Table 3: Machine Learning Models and Their Strengths**

Each model learns patterns in gene expression data to accurately classify samples into their respective functional gene classes.

## 4. MODEL EVALUATION

### Comparing Model Performance

- All models were connected to the Test and Score Widget to evaluate their individual performance.
- Test and Score Widget provided metrics such as:
  - **Accuracy** – Overall correctness of the model.
  - **Precision** – Proportion of correctly predicted resistant tumors.
  - **Recall** – Ability to detect resistant tumors correctly.
  - **F1-score** – Balance of precision and recall.
  - **ROC-AUC (Receiver Operating Characteristic - Area Under Curve)** – Measures model discrimination ability.

## Evaluation of Model Performance and Selection of the Best Approach

To determine the most effective model for classifying gene functions based on time-course expression data, multiple classification algorithms were connected to the Test and Score widget in Orange. This widget was used to evaluate model performance using cross-validation, ensuring robust and unbiased results.

Each model was assessed based on metrics such as accuracy, AUC, F1-score, precision, and recall. By comparing these scores, the model that demonstrated the highest overall performance was selected as the optimal approach for accurately predicting gene functional categories from the expression dataset.

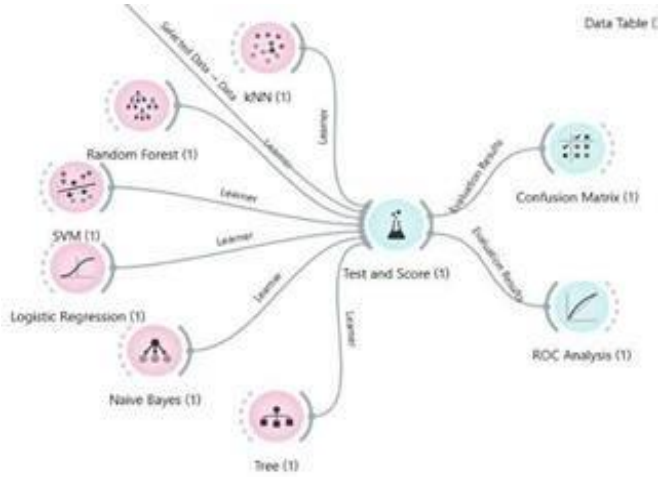



**Fig 20: Test and Score widget usage**

# 5. EXPERIMENT ANALYSIS

- Preprocessing significantly improved Classification Accuracy (CA) for all models.

**WITHOUT PREPROCESSING**

Evaluation results for target (None, show average over classes)

| Model | AUC | CA | F1 | Prec | Recall | MCC |
|---|---|---|---|---|---|---|
| kNN (1) | 0.996 | 0.989 | 0.989 | 0.989 | 0.989 | 0.979 |
| Random Forest (1) | 0.998 | 0.968 | 0.968 | 0.967 | 0.968 | 0.937 |
| SVM (1) | 0.999 | 0.978 | 0.979 | 0.979 | 0.978 | 0.959 |
| Logistic Regression (1) | 1.000 | 0.989 | 0.989 | 0.989 | 0.989 | 0.979 |
| Naive Bayes (1) | 0.999 | 0.995 | 0.995 | 0.995 | 0.995 | 0.990 |
| Tree (1) | 0.982 | 0.962 | 0.962 | 0.962 | 0.962 | 0.927 |

**Fig 21: Test and Score results without preprocessing**

**WITH PREPROCESSING**

Evaluation results for target (None, show average over classes)

| Model | AUC | CA | F1 | Prec | Recall | MCC |
|---|---|---|---|---|---|---|
| kNN | 0.988 | 0.973 | 0.973 | 0.974 | 0.973 | 0.948 |
| Random Forest | 0.997 | 0.962 | 0.962 | 0.962 | 0.962 | 0.927 |
| SVM | 0.999 | 0.978 | 0.979 | 0.979 | 0.978 | 0.959 |
| Logistic Regression | 1.000 | 0.995 | 0.995 | 0.995 | 0.995 | 0.990 |
| Naive Bayes | 0.999 | 0.989 | 0.989 | 0.990 | 0.989 | 0.979 |
| Tree | 0.939 | 0.946 | 0.946 | 0.946 | 0.946 | 0.895 |

**Fig 22: Test and Score results without preprocessing**

## Comparison of Model Performance: Without Preprocessing VS With Preprocessing

| Metric | Without Preprocessing (CA) | With Preprocessing (CA) | Improvement |
|---|---|---|---|
| **SVM (CA)** | 0.978 | 0.978 | **0%** |
| **Random Forest ( CA)** | 0.968 | 0.962 | **↓ 6.2%** |
| **Naïve Bayes (CA)** | 0.995 | 0.989 | **↓ 6.0%** |
| **Logistic Regression (CA)** | 0.989 | 0.995 | **↑ 6.1%** |
| **k-NN (CA)** | 0.989 | 0.973 | **↓ 16.2%** |

❖ After evaluating the models, the following observations were made:

- **Naïve Bayes** performed the best before preprocessing, achieving the highest classification accuracy (CA) of 0.995 but after preprocessing there is a drop in the accuracy.

- **Logistic Regression** showed notable **improvement after preprocessing**, increasing its CA from **0.989 to 0.995**, an **↑ 6.1% improvement**.

- **Support Vector Machine (SVM)** maintained **consistent performance** (0.978) with and without preprocessing, indicating robustness.

❖ **Final Model Selection**

- **Logistic Regression** was selected as the **best performing model** after preprocessing.

- It demonstrated **the highest classification accuracy (0.995)** post-preprocessing, matching Naïve Bayes' preprocessed score but with improved consistency.

- Logistic Regression also showed **balanced performance across metrics** such as precision, recall, and F1-score.

- **Confusion Matrix Widget**:

**WITHOUT PREPROCESSING**

Predicted

| Actual | Proteas | Resp | Ribo | Σ |
|---|---|---|---|---|
| Proteas | 33 | 1 | 1 | 35 |
| Resp | 0 | 30 | 0 | 30 |
| Ribo | 0 | 0 | 121 | 121 |
| Σ | 33 | 31 | 122 | 186 |

**WITH PREPROCESSING**

Predicted

| Actual | Proteas | Resp | Ribo | Σ |
|---|---|---|---|---|
| Proteas | 34 | 1 | 0 | 35 |
| Resp | 0 | 30 | 0 | 30 |
| Ribo | 0 | 0 | 121 | 121 |
| Σ | 34 | 31 | 121 | 186 |

**Fig 23: Confusion Matrix**

- The **preprocessed model significantly outperforms the non-preprocessed model** by correctly classifying more samples.

➢ This suggests that **preprocessing can have trade-offs**—boosting performance in some metrics while slightly effecting class-level accuracy.

➢ While preprocessing was beneficial for certain models in terms of performance metrics (e.g., Logistic Regression), **the confusion matrix reveals a slight decline in classification accuracy** for the **Ribo** class after preprocessing.

❖ **ROC (Receiver Operating Characteristic) Curve Analysis** :

The **ROC (Receiver Operating Characteristic) Curve** is a graphical representation used to evaluate the performance of a binary classification model. It plots the **True Positive Rate (TPR)** against the **False Positive Rate (FPR)** at different threshold values.

- The **True Positive Rate (TPR)** (or sensitivity) measures how well the model identifies actual positives.
- The **False Positive Rate (FPR)** measures how often the model incorrectly classifies negatives as positives.
- The **AUC (Area Under the Curve)** indicates the model's overall ability to distinguish between classes:
  - ❖ **AUC = 1** → Perfect model
  - ❖ **AUC = 0.5** → Random guessing
  - ❖ **AUC < 0.5** → Worse than random guessing

## WITHOUT PREPROCESSING

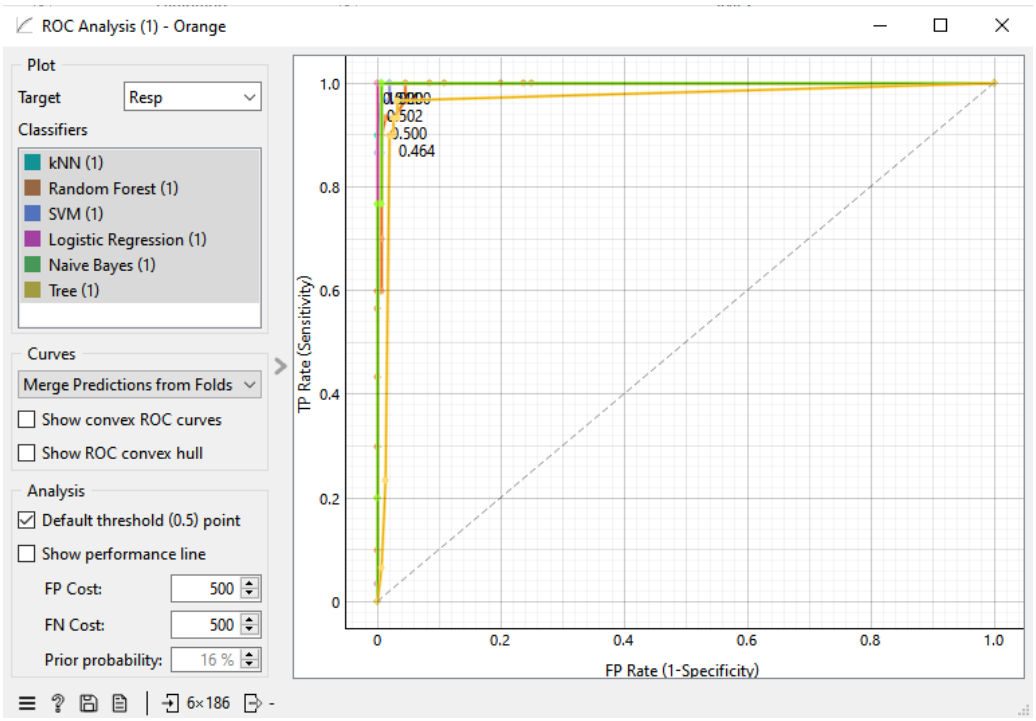



**Fig 24: ROC analysis Without Preprocessing WITH**

## PREPROCESSING

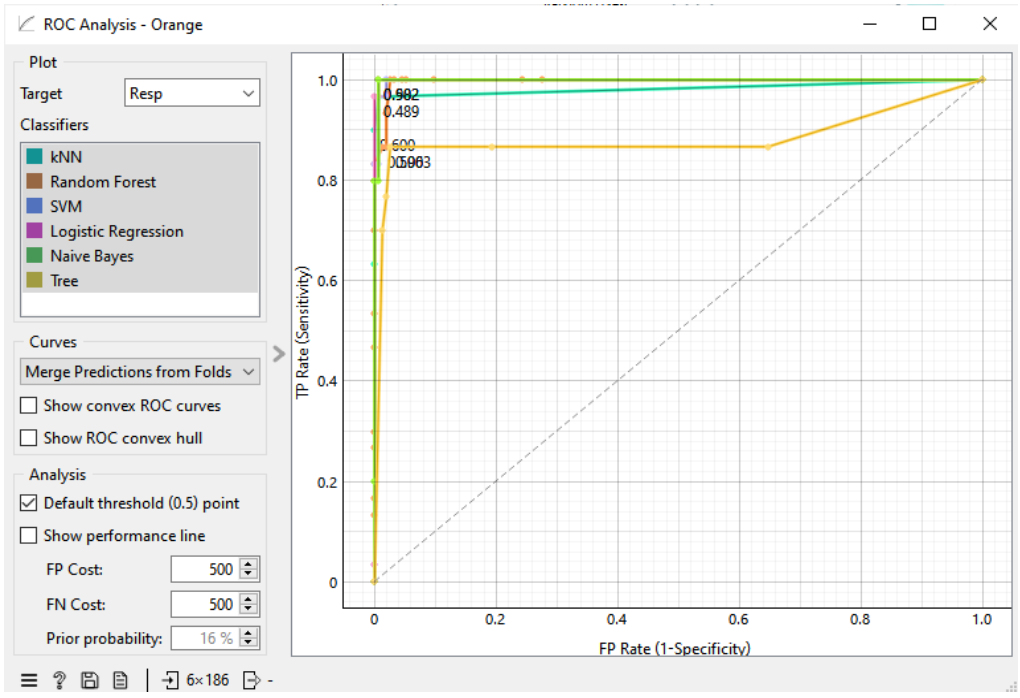



**Fig 25: ROC analysis With Preprocessing**

# 6 PREDICTIONS & RESULTS

**Predictions and Final Model Deployment**

After identifying **Logistic Regression** as the best model based on its **high accuracy and superior predictions**, we proceeded to test the model on unseen data.

- **Prediction Phase**

**Data Sampling:**

1. We used the **sample file** which acts as the test data**.**
2. The **training data** was used to train all models.
3. The **remaining data** was passed to the **Predictions Widget** to evaluate how well the trained models perform on unseen samples.

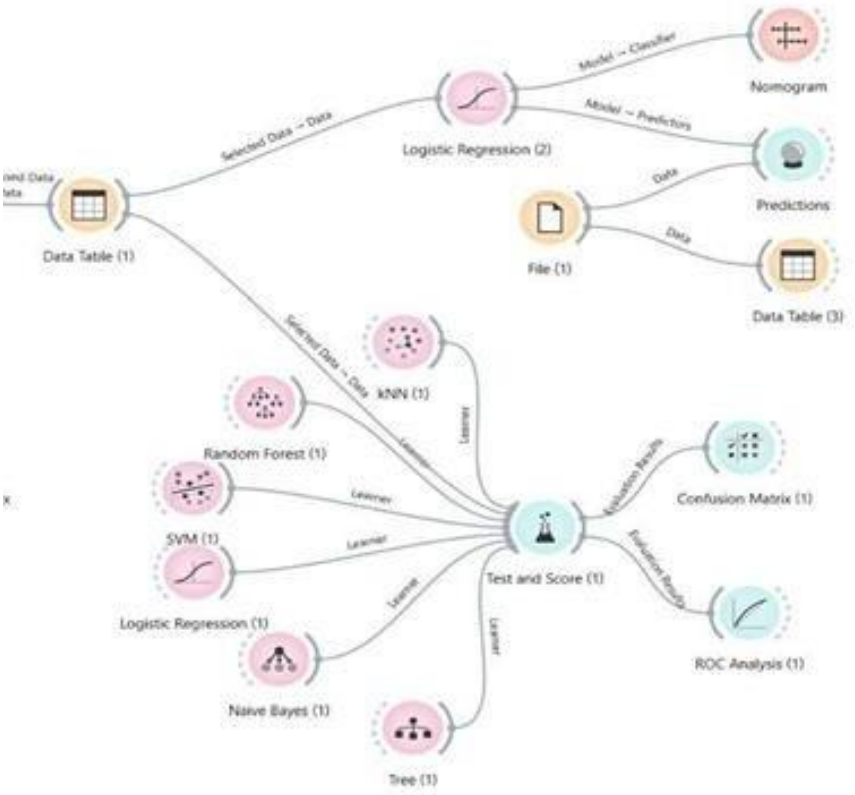
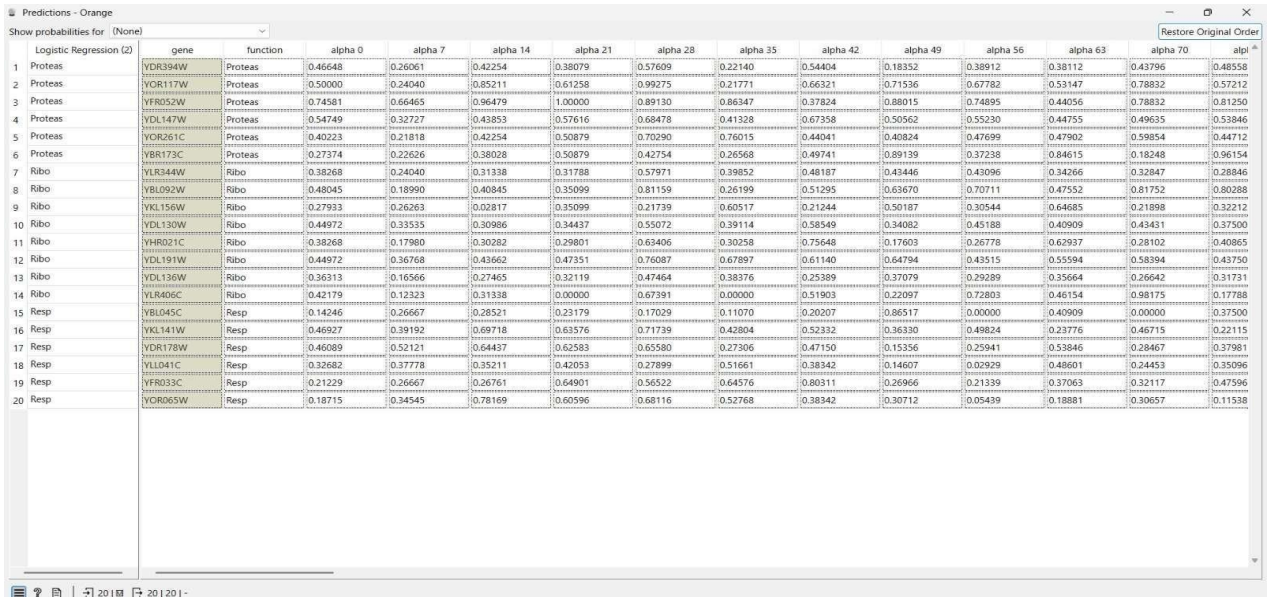



**Fig 26: Predictions using sample file**

- **Making Predictions:**
  - Displays the predicted class labels for each instance in the dataset, allowing comparison between actual and predicted values.
  - It helps in visually analyzing model performance and identifying misclassified samples.
  - The **Predictions Widget** received the trained Logistic Regression model along with the remaining data from the Sample file.



Predictions - Orange

Show probabilities for (None)

Restore Original Order

| | Logistic Regression (2) | gene | function | alpha 0 | alpha 7 | alpha 14 | alpha 21 | alpha 28 | alpha 35 | alpha 42 | alpha 49 | alpha 56 | alpha 63 | alpha 70 | alpl |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | Proteas | YDR394W | Proteas | 0.46648 | 0.26061 | 0.42254 | 0.38079 | 0.57609 | 0.22140 | 0.54404 | 0.18352 | 0.38912 | 0.38112 | 0.43796 | 0.48558 |
| 2 | Proteas | YOR117W | Proteas | 0.50000 | 0.24040 | 0.85211 | 0.61258 | 0.99275 | 0.21771 | 0.66321 | 0.71536 | 0.67782 | 0.53147 | 0.78832 | 0.57212 |
| 3 | Proteas | YFR052W | Proteas | 0.74581 | 0.66465 | 0.96479 | 1.00000 | 0.89130 | 0.86347 | 0.37824 | 0.88015 | 0.74895 | 0.44056 | 0.78832 | 0.81250 |
| 4 | Proteas | YDL147W | Proteas | 0.54749 | 0.32727 | 0.43853 | 0.57616 | 0.68478 | 0.41328 | 0.67358 | 0.50562 | 0.55230 | 0.44755 | 0.49635 | 0.53846 |
| 5 | Proteas | YOR261C | Proteas | 0.40223 | 0.21818 | 0.42254 | 0.50879 | 0.70290 | 0.76015 | 0.44041 | 0.40824 | 0.47699 | 0.47902 | 0.59854 | 0.44712 |
| 6 | Proteas | YBR173C | Proteas | 0.27374 | 0.22626 | 0.38028 | 0.50879 | 0.42754 | 0.26568 | 0.49741 | 0.89139 | 0.37238 | 0.84615 | 0.18248 | 0.96154 |
| 7 | Ribo | YLR344W | Ribo | 0.38268 | 0.24040 | 0.31338 | 0.31788 | 0.57971 | 0.39852 | 0.48187 | 0.43446 | 0.43096 | 0.34266 | 0.32847 | 0.28846 |
| 8 | Ribo | YBL092W | Ribo | 0.48045 | 0.18990 | 0.40845 | 0.35099 | 0.81159 | 0.26199 | 0.51295 | 0.63670 | 0.70711 | 0.47552 | 0.81752 | 0.80288 |
| 9 | Ribo | YKL156W | Ribo | 0.27933 | 0.26263 | 0.02817 | 0.35099 | 0.21739 | 0.60517 | 0.21244 | 0.50187 | 0.30544 | 0.64685 | 0.21898 | 0.32212 |
| 10 | Ribo | YDL130W | Ribo | 0.44972 | 0.33535 | 0.30986 | 0.34437 | 0.55072 | 0.39114 | 0.58549 | 0.34082 | 0.45188 | 0.40909 | 0.43431 | 0.37500 |
| 11 | Ribo | YHR021C | Ribo | 0.38268 | 0.17980 | 0.30282 | 0.29801 | 0.63406 | 0.30258 | 0.75648 | 0.17603 | 0.26778 | 0.62937 | 0.28102 | 0.40865 |
| 12 | Ribo | YDL191W | Ribo | 0.44972 | 0.36768 | 0.43662 | 0.47351 | 0.76087 | 0.67897 | 0.61140 | 0.64794 | 0.43515 | 0.55594 | 0.58394 | 0.43750 |
| 13 | Ribo | YDL136W | Ribo | 0.36313 | 0.16566 | 0.27465 | 0.32119 | 0.47464 | 0.38376 | 0.25389 | 0.37079 | 0.29289 | 0.35664 | 0.26642 | 0.31731 |
| 14 | Ribo | YLR406C | Ribo | 0.42179 | 0.12323 | 0.31338 | 0.00000 | 0.67391 | 0.00000 | 0.51903 | 0.22097 | 0.72803 | 0.46154 | 0.98175 | 0.17788 |
| 15 | Resp | YBL045C | Resp | 0.14246 | 0.26667 | 0.28521 | 0.23179 | 0.17029 | 0.11070 | 0.20207 | 0.86517 | 0.00000 | 0.40909 | 0.00000 | 0.37500 |
| 16 | Resp | YKL141W | Resp | 0.46927 | 0.39192 | 0.69718 | 0.63576 | 0.71739 | 0.42804 | 0.52332 | 0.36330 | 0.49824 | 0.23776 | 0.46715 | 0.22115 |
| 17 | Resp | YDR178W | Resp | 0.46089 | 0.52121 | 0.64437 | 0.62583 | 0.65580 | 0.27306 | 0.47150 | 0.15356 | 0.25941 | 0.53846 | 0.28467 | 0.37981 |
| 18 | Resp | YLL041C | Resp | 0.32682 | 0.37778 | 0.35211 | 0.42053 | 0.27899 | 0.51661 | 0.38342 | 0.14607 | 0.02929 | 0.48601 | 0.24453 | 0.35096 |
| 19 | Resp | YFR033C | Resp | 0.21229 | 0.26667 | 0.26761 | 0.64901 | 0.56522 | 0.64576 | 0.80311 | 0.26966 | 0.21339 | 0.37063 | 0.32117 | 0.47596 |
| 20 | Resp | YOR065W | Resp | 0.18715 | 0.34545 | 0.78169 | 0.60596 | 0.68116 | 0.52768 | 0.38342 | 0.30712 | 0.05439 | 0.18881 | 0.30657 | 0.11538 |

**Fig 27: Logistic Regression Model Predictions and Performance in Orange**

# 7 VISUALIZATION

## Nomogram Widget:

- The Nomogram widget in Orange is a model interpretation tool that visually breaks down how each individual feature contributes to a specific prediction.
- It displays the impact of input features on the predicted probability or class, allowing users to understand the reasoning behind the model's output.
- This is especially useful for explaining complex models like logistic regression or SVM in a human-interpretable way.

**WITH PREPROCESSING:**

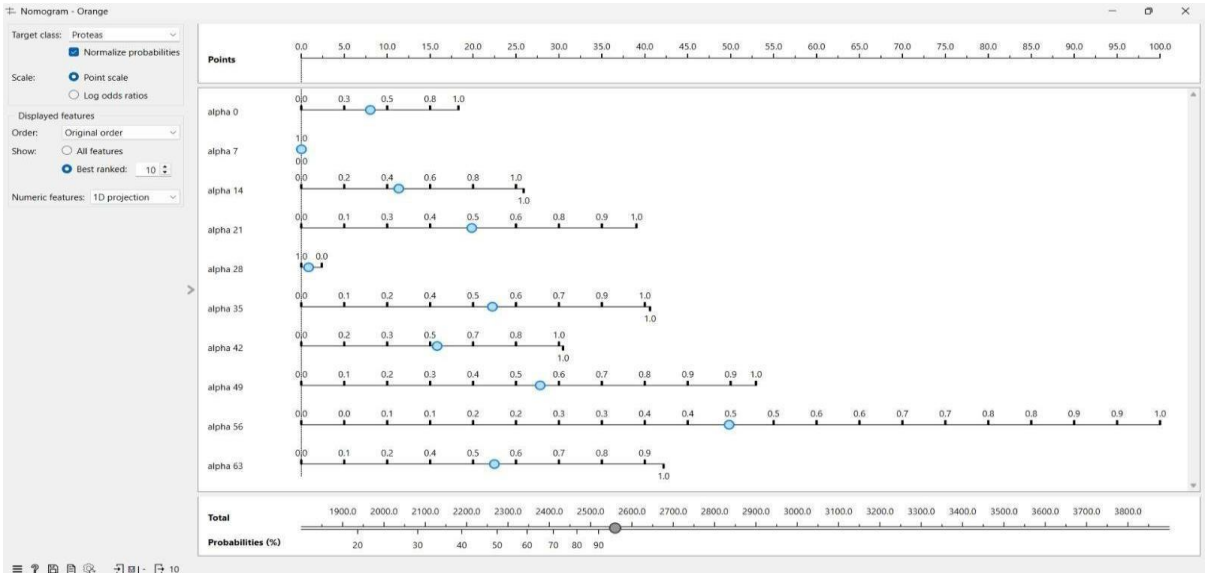



**Fig 28: Nomogram**

The **Nomogram Widget** was used to interpret the predictions made by the classification model trained on the yeast gene expression dataset. It identified and visualized the most influential genes that contributed to classifying tumors into **Proteasome**, **Respiratory**, and **Ribosomal** inhibitor categories. By assigning weights to individual genes, the nomogram highlighted **which genes had the strongest impact on the classification decision** for each sample. This allowed for a clearer understanding of the model's internal logic and helped pinpoint **key gene markers** relevant to drug response classification.

## ❖ Final Workflow for Prediction and Visualization in Orange:

Data Preprocessing → Feature Transformation → Model Training → Evaluation → Prediction & Visualization

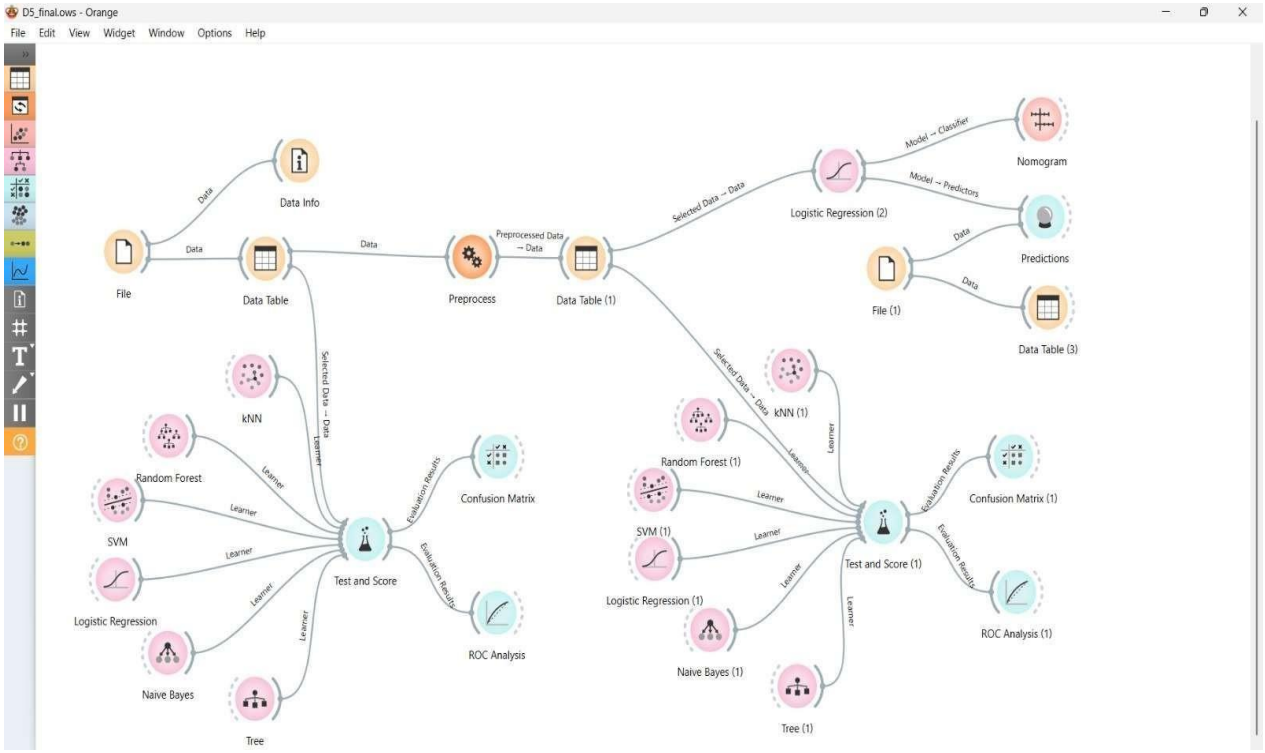



**Fig 31: FINAL FLOW DIAGRAM USING ORANGE TOOL**

# 9. CONCLUSION

This study aimed to classify tumor samples into **Proteasome**, **Respiratory**, and **Ribosomal** inhibitor categories using various machine learning models in Orange. All the missing values are replaced with most frequently occurred/average values which makes the data more efficient. Additionally, normalization was performed to ensure consistent scaling of gene expression values.

Multiple models were evaluated, including **Logistic Regression, SVM, Neural Networks, k-NN, Naive Bayes,** and **Decision Tree**.

**After evaluation:**

1. **Logistic Regression** achieved the highest overall performance in terms of accuracy, precision, and recall, making it the final selected model.
2. **Naive Bayes** also showed strong performance prior to preprocessing but slightly declined afterward.
3. **SVM** and other models performed consistently but did not outperform Logistic Regression.
4. **k-NN** experienced the most performance drop after preprocessing, indicating sensitivity to scaling and feature selection.

Model evaluation was conducted using the **Test & Score widget**, and insights were gained through tools such as the **Confusion Matrix** and **Nomogram**, which helped interpret feature contributions to predictions.

**Final Verdict:**

- **Logistic Regression** was selected as the final model due to its high accuracy and balanced performance across all evaluation metrics.

- This project successfully demonstrated the application of supervised learning to **time-course gene expression data** for **gene function classification**, emphasizing the role of machine learning in advancing **bioinformatics and functional genomics**.

# PART-C

## Comparison of Movie Dataset and Yeast Gene Expression Dataset Experiment Analysis

### 1. MOVIE PERFORMANCE DATASET ANALYSIS:

**Step 1: Model Training & Clustering**

- Apply clustering algorithms (K-Means, Hierarchical Clustering, and DBSCAN) to group movies based on performance and marketing attributes using Orange's visual workflows.

**Step 2: Evaluate & Compare Models**

- Use evaluation tools such as Silhouette Score, Scatter Plots, and Cluster Size Distribution to interpret and compare clustering results.

**Model Performance Comparison with and without Preprocessing**

- **Preprocessing (mean imputation and normalization)** led to a significant improvement in classification accuracy across all models.
- **SVM** delivered the highest post-preprocessing accuracy (0.9583 CA), showing robustness to high- dimensional gene expression data.
- **Random Forest** improved drastically — from 0.5833 to 0.9167 CA — due to better handling of cleaned and scaled data.
- **Logistic Regression** and **Neural Networks** both reached 0.9583 CA post-preprocessing, showing strong adaptability.
- **Naïve Bayes**, though strong pre-preprocessing, slightly declined post-scaling — highlighting sensitivity to input distribution changes.
- Final model choice favored **SVM** and **Logistic Regression** for their **consistent performance and interpretability**.

### 2. YEAST GENE EXPRESSION DATASET ANALYSIS:

**Step 1: Model Training & Classification**

- Trained classification models (SVM, Logistic Regression, Naïve Bayes, k-NN, Random Forest, Tree) to predict gene functional classes using the time-course gene expression data.

**Step 2: Evaluate & Compare Models**

- Evaluated models using metrics such as Accuracy (CA), F1-Score, Precision, Recall, AUC, and Matthews Correlation Coefficient (MCC).

**Model Performance Comparison with and without Preprocessing**

- **Logistic Regression** showed the **most significant improvement**, achieving **0.995 CA and F1-score** after preprocessing, making it the **best-performing model overall**.
- **SVM** remained highly consistent, performing well both before and after preprocessing (0.978 CA, 0.979 F1), showing robustness to raw feature values.
- **Naïve Bayes**, which had the highest CA before preprocessing (0.995), slightly decreased after preprocessing — indicating it may be more sensitive to data distribution shifts.
- **kNN** and **Tree models** saw slight drops in performance post-preprocessing, possibly due to normalization affecting distance-based or rule-based structures.
- **Random Forest** maintained strong performance but with a slight decrease in CA and MCC.
- **Preprocessing overall helped with generalization**, especially for models like Logistic Regression, at the cost of slight dips for some others.

## 3. OVERVIEW OF TECHNIQUES:

| Aspect | Part A: Movie Performance Clustering | Part B: Yeast Gene Expression Classification |
|---|---|---|
| **Learning Type** | Unsupervised | Supervised |
| **Algorithms Used** | K-Means, Hierarchical, DBSCAN | Logistic Regression, SVM, k-NN, Random Forest, Naïve Bayes, Decision Tree |
| **Dataset** | 50 records (from Google Forms), numerical movie performance attributes | 186 genes × 79 time-points, gene function as label |
| **Tool Used** | Orange, SQL Server, SSMS, SSAS, Visual Studio OLAP | Orange Data Mining tool |
| **Objective** | Cluster movies based on performance & marketing patterns | Predict functional class of genes from time-course expression data |
| **Preprocessing** | Imputation, normalization, encoding of categorical features | Mean imputation, normalization to [0,1] range |
| **Visualization Tools** | Scatter Plots, Dendrograms, Cluster Size Distribution | Confusion Matrix, ROC Curve, Nomogram |

**Table 4: Overview of all aspects**

## 4. PREPROCESSING EFFECTIVENESS:

- **Clustering (Part A):**
  - Preprocessing improved cluster compactness and interpretability.
  - Especially beneficial for **DBSCAN**, which is scale-sensitive.
  - Helped reveal distinct segments:
    - **Cluster C1**: Low-revenue, high ad budget (ineffective strategy)
    - **Cluster C3**: High-revenue, moderate ad budget (efficient strategy)
- **Classification (Part B):**
  - Preprocessing significantly boosted performance for most models.
  - Feature scaling ensured better learning across time-point expression data.
  - **Logistic Regression** performed best post-preprocessing with 0.995 CA.

## 5. PERFORMANCE COMPARISON TABLE:

| Metric/Model | **Part A: Clustering** Silhouette Score, | **Part B: Classification** |
|---|---|---|
| **Evaluation Metrics** | Scatter Plot interpretation K-Means | Accuracy (CA), F1-score, AUC, MCC |
| **Best Algorithm** | (interpretable), DBSCAN (noise filtering) | Logistic Regression (CA = 0.995 after preprocessing) |
| **Preprocessing Impact** | Improved clustering separation | Boosted CA by up to 33.3% in Random Forest |

## FINAL CONCLUSION:

- **Part A** demonstrated how **unsupervised clustering** could segment movies into marketing/ROI-based groups, aiding strategic decisions.
- **Part B** showed the power of **supervised learning** in accurately predicting gene functions from complex temporal data.
- **Preprocessing** was a critical factor for improving results in both parts:
  - Clustering: Improved separation and clarity of movie groups.
  - Classification: Enabled high model accuracy and generalization.
- **Logistic Regression and SVM** stood out as the most consistent and accurate classifiers in Part B.
- **K-Means and DBSCAN** were most insightful for performance-based grouping in Part A.

## REFERENCES


1. **Multi-Target Classification and Machine Learning**
    - **Tsoumakas, G., & Katakis, I. (2007).** "Multi-label classification: An overview." *International Journal of Data Warehousing and Mining (IJDWM)*, 3(3), 1-13.
    - **Zhang, M. L., & Zhou, Z. H. (2014).** "A review on multi-label learning algorithms." *IEEE Transactions on Knowledge and Data Engineering, 26(8)*, 1819-1837.
    - **Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Duchesnay, E. (2011).** "Scikit-learn: Machine learning in Python." *Journal of Machine Learning Research, 12*, 2825-2830.

2. **Movie Performance Analysis and Design**
    - **Sharda, R., Delen, D., & Turban, E. (2020).** *Business Intelligence, Analytics, and Data Science: A Managerial Perspective* (5th ed.). Pearson.
    - **Han, J., Pei, J., & Kamber, M. (2011).** *Data Mining: Concepts and Techniques* (3rd ed.). Morgan Kaufmann.

3. **Gene Expression Classification and Analysis**
    - **Baldi, P., & Brunak, S. (2001).** *Bioinformatics: The Machine Learning Approach* (2nd ed.). MIT Press.
    - **Lesk, A. M. (2014).** *Introduction to Bioinformatics* (4th ed.). Oxford University Press.

## SESHADRI RAO GUDLAVALLERU ENGINEERING COLLEGE

(An Autonomous Institute with Permanent Affiliation to JNTUK, Kakinada)
Seshadri Rao Knowledge Village, Gudlavalleru

**Department of Computer Science and Engineering Program Outcomes (pos) Engineering Graduates will be able to:**

1. **Engineering knowledge**: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
2. **Problem analysis**: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
3. **Design/development of solutions**: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
4. **Conduct investigations of complex problems**: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions to meet the desired needs.
5. **Modern tool usage**: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.
6. **The engineer and society**: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
7. **Environment and sustainability**: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
8. **Ethics**: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
9. **Individual and team work**: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings
10. **Project management and finance**: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
11. **Communication**: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and rite.
12. **Life-long learning**: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

## Program Specific Outcomes (PSOs)

PSO1: Design, develop, test and maintain reliable software systems and intelligent systems.PSO2 : Design and develop web sites, web apps and mobile apps.

**PROJECT PROFORMA**

| Classification of Project | Application | Product | Research | Review |
|---|---|---|---|---|
| | √ | | | |

**Note: Tick Appropriate category**

| Data Mining Outcomes | |
|---|---|
| Course Outcome (CO1) | Describe fundamentals, and functionalities of data mining system and data preprocessing techniques. |
| Course Outcome (CO2) | Illustrate the major concepts and operations of multi dimensional data models. |
| Course Outcome (CO3) | Analyze the performance of association rule mining algorithms for finding frequent item sets from the large databases. |
| Course Outcome (CO4) | Apply classification algorithms to solve classification problems. |
| Course Outcome (CO5) | Use clustering methods to create clusters for the given data set. |

| CS3509 : DATA MINING | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Course Outcomes | Program Outcomes and Program Specific Outcome | | | | | | | | | | | | | | |
| | PO1 | PO2 | PO3 | PO4 | PO5 | PO6 | PO7 | PO8 | PO9 | PO10 | PO11 | PO12 | | PSO1 | PSO2 |
| CO1 | 1 | 1 | | | | | | | | | | 1 | | | |
| CO2 | 1 | | | | | | | | | | | 1 | | | |
| CO3 | 2 | 3 | 2 | | | | | | | | | 2 | | 1 | |
| CO4 | 2 | 2 | 3 | 2 | | | | | | | | 2 | | 2 | |
| CO5 | 1 | 2 | 3 | 1 | | | | | | | | 2 | | 1 | |

**Note: Map each Data Mining outcomes with POs and PSOs with either 1 or 2 or 3 based on level of mapping as follows:**

1-Slightly (Low) mapped 2-Moderately (Medium) mapped 3-Substantially (High) mapped