Project Report

Classification of Video Games

Abhishek Patel

1. Introduction

This project focuses on predicting the genre of a video game using machine learning techniques. The classification is based on features such as platform, critic/user scores, publisher, and regional/global sales. The task is formulated as a multi-class classification problem with twelve possible genre labels (e.g., Action, Sports, Role-Playing).

The goal is to evaluate how different preprocessing methods, dimensionality reduction techniques, and classifiers affect the accuracy and stability of predictions. We apply normalization methods (Min-Max scaling and Z-score), dimensionality reduction (PCA and LDA), and classifiers including Logistic Regression, Random Forest, SVM, KNN, and three Bayesian variants (Naive, Multivariate, Nonparametric). Through repeated experiments, we also aim to analyze the variance in accuracy caused by changing train-test splits and imbalanced class distributions.

The motivation behind this project lies in the real-world relevance of genre classification. Accurate genre prediction can support game recommendation systems, improve market analysis, and help developers tailor game designs based on data-driven insights. Moreover, working on a diverse dataset with both numerical and categorical features provides a practical context for applying core pattern recognition techniques.

Overall, this project integrates key concepts from CSE 802—including density estimation, dimensionality reduction, and classifier evaluation—into a comprehensive workflow. By comparing classifier performance under different preprocessing pipelines and experimental setups, we aim to draw meaningful conclusions about model effectiveness, feature importance, and classification challenges in high-dimensional, real-world datasets.

2. Description of Dataset Source

- **Dataset Name**: Video Game Sales
- **Source**: Kaggle - https://www.kaggle.com/datasets/sidtwr/videogames-sales-dataset
- **Number of Samples**: Approximately 16,500 video game entries

Dimensions and Structure

- **Total Columns**: 16 (15 input features + 1 target variable)
- **Data Types**: A mix of numerical and categorical features **Categorical Features**
- Platform: The platform on which the game was released (e.g., PS4, Xbox One, PC)
- Publisher: Name of the game publisher
- Rating: ESRB content rating (e.g., E, T, M)

Numerical Features

- Year_of_Release: Year the game was released
- Critic_Score: Average review score from critics (0–100)
- Critic_Count: Number of critic reviews
- User_Score: Average user score (converted to a 0–10 scale)
- User_Count: Number of user reviews
- NA_Sales, EU_Sales, JP_Sales, Other_Sales: Regional sales in millions
- Global_Sales: Total worldwide sales (sum of all regions) **Target Variable**

- **Column:** Genre
- **Type:** Categorical
- **Number of Classes:** 12 (Action, Adventure, Fighting, Misc, Platform, Puzzle, Racing, Role-Playing, Shooter, Simulation, Sports, Strategy**)**

3. Description of Analysis Conducted

3.1 Data Preprocessing

The dataset contained both numeric and categorical attributes, as well as some missing values. The following steps were performed:

- Missing values in numeric columns (e.g., Critic_Score, User_Count) were filled with the column mean.
- Categorical columns (e.g., Publisher, Rating) were filled with the most frequently occurring value.

- Non-essential columns such as Name and Developer were removed.
- Categorical values in Platform, Rating, and Publisher were converted into numerical form using one-hot encoding. To reduce dimensionality, only the top 10 publishers were kept, and others grouped under “Other.”

3.2 Feature Normalization

To ensure all features were on comparable scales, the following normalization techniques were applied:

- **Min-Max Scaling**: Scales features to the [0, 1] range.
- **Z-Score Normalization**: Standardizes features to have zero mean and unit variance.

These versions of the dataset were used separately to observe their impact on classification performance.

3.3 Dimensionality Reduction

Two primary techniques were used to reduce feature complexity:

- **Principal Component Analysis (PCA)**: A projection method that transforms features into a smaller number of uncorrelated components that preserve most of the variance in the data.
- **Linear Discriminant Analysis (LDA)**: A supervised projection method that reduces dimensionality while maximizing class separability. LDA was particularly useful for evaluating how well genre classes can be distinguished after projection.

Both PCA and LDA were applied to the original and normalized datasets (Min-Max and ZScore), producing several reduced feature sets for experimentation.

3.4 Iterative Data Splitting

Rather than relying on a single split, the data was partitioned and tested multiple times to assess model consistency:

- For each run, the dataset was split into training, validation, and test sets, with test sizes of 10%, 20%, and 30%.
- From the remaining training portion, 10% was set aside as a validation set.
- The splitting was repeated using different random seeds to simulate different scenarios and avoid bias from a fixed partition.
- After each run, metrics like accuracy and error rate were recorded.

- At the end, mean and variance of accuracy and error were calculated to analyze model stability and robustness.

This multi-run evaluation ensured that classifier performance was not an artifact of a specific split, providing a more general performance estimate.

3.5 Classifier Training

Five types of classifiers were evaluated on each version of the dataset:

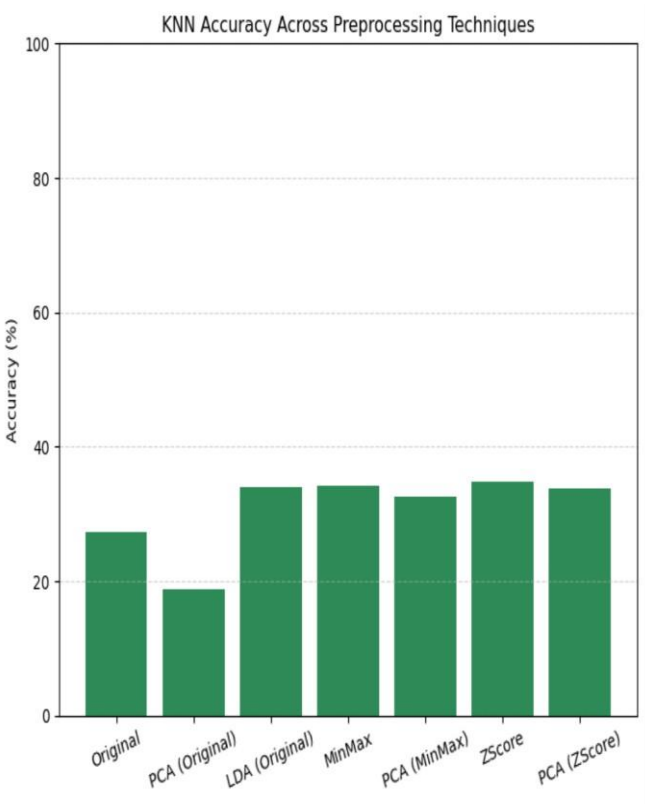
- Logistic Regression
- Random Forest
- LinearSVC
- k-Nearest Neighbors (KNN)
- Custom Bayesian Classifiers:
    - Naive Bayes: Assumes features are independent and normally distributed.
    - Multivariate Gaussian Bayes: Uses full covariance matrices between features.
    - Nonparametric Bayes: Uses kernel density estimation to model flexible distributions.

Each model was trained using all feature variations (original, minmax-normalized, zscore_normalized, PCA-reduced, minmax-normalized_PCA-reduced, zscore-normalized_PCAreduced, LDA-reduced) and evaluated using accuracy and confusion matrices.

4. Presentation of Results

We present the results of the best and worst performing classifiers, including their confusion matrices and classification accuracy across multiple iterations.
(All other results, including accuracy trends, variance analysis, and performance graphs, can be found in the code.)

1) **KNN (For Data with only Zscore normalization applied)**




```
Run 1: Train = 13375, Val = 1672, Test = 1672 → Accuracy: 34.87%
Run 2: Train = 11703, Val = 1672, Test = 3344 → Accuracy: 32.72%
Run 3: Train = 10031, Val = 1672, Test = 5016 → Accuracy: 30.90%

Mean Accuracy: 32.83%
Accuracy Variance: 2.63
Mean Error Rate: 67.17%
Error Rate Variance: 2.63

Z-Score Normalized Data Test Accuracy on KNN: 34.87%
```

Confusion Matrix (% per actual class):

	0	1	2	3	4	5	6	7	8	9	10	11
0	54.02	4.89	2.87	6.90	3.16	0.29	4.60	4.31	8.91	2.59	6.32	1.15
1	28.03	28.03	5.30	6.06	0.00	3.79	5.30	9.85	3.03	4.55	5.30	0.76
2	26.14	9.09	27.27	4.55	1.14	1.14	6.82	10.23	5.68	1.14	4.55	2.27
3	19.77	5.65	3.95	37.29	2.82	2.82	6.21	9.04	0.56	5.08	6.78	0.00
4	29.27	4.88	2.44	7.32	17.07	7.32	7.32	4.88	4.88	1.22	13.41	0.00
5	16.67	8.33	5.00	15.00	8.33	28.33	6.67	3.33	1.67	1.67	5.00	0.00
6	34.56	3.68	3.68	7.35	5.88	2.21	21.32	0.00	2.21	2.94	16.18	0.00
7	35.62	6.88	5.00	6.88	1.88	1.88	1.25	31.25	4.38	0.62	1.25	3.12
8	33.05	4.24	5.08	6.78	4.24	1.69	3.39	9.32	24.58	0.85	3.39	3.39
9	15.12	13.95	2.33	12.79	0.00	8.14	10.47	2.33	2.33	18.60	10.47	3.49
10	14.16	1.37	3.20	10.96	4.11	1.83	8.68	4.11	3.20	1.37	45.66	1.37
11	24.24	10.61	1.52	10.61	4.55	3.03	7.58	6.06	0.00	0.00	12.12	19.70

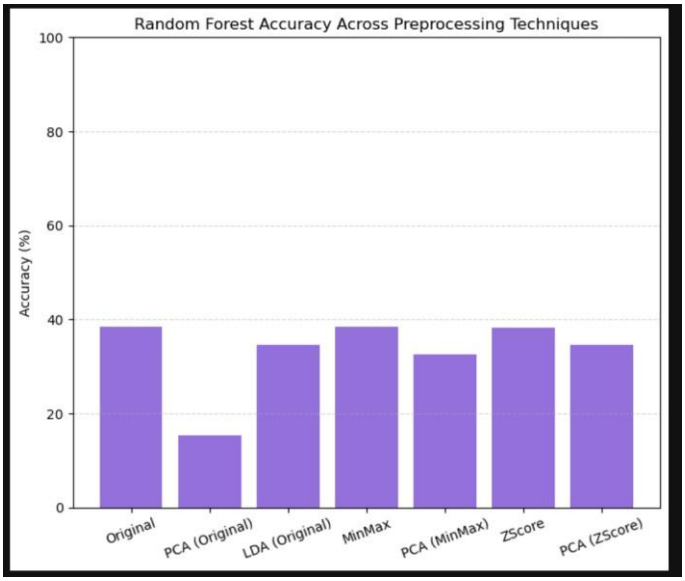
0 - Action **1** - Adventure **2** – Fighting **3** – Misc **4** – Platform **5** - Puzzle

6 – Racing **7** – Role playing **8** – Shooter **9** – Simulation **10** – Sports **11** - Strategy

Classification Report:

	precision	recall	f1-score	support
Action	0.36	0.54	0.43	348
Adventure	0.30	0.28	0.29	132
Fighting	0.29	0.27	0.28	88
Misc	0.35	0.37	0.36	177
Platform	0.22	0.17	0.19	82
Puzzle	0.30	0.28	0.29	60
Racing	0.25	0.21	0.23	136
Role-Playing	0.37	0.31	0.34	160
Shooter	0.31	0.25	0.27	118
Simulation	0.31	0.19	0.23	86
Sports	0.49	0.46	0.47	219
Strategy	0.37	0.20	0.26	66
accuracy			0.35	1672
macro avg	0.33	0.29	0.30	1672
weighted avg	0.34	0.35	0.34	1672

2) **Random Forest (For Data with only Min max normalization applied)**




```
Run 1: Train = 13375, Val = 1672, Test = 1672 → Accuracy: 37.98%
Run 2: Train = 11703, Val = 1672, Test = 3344 → Accuracy: 38.40%
Run 3: Train = 10031, Val = 1672, Test = 5016 → Accuracy: 36.04%

Mean Accuracy: 37.47%
Accuracy Variance: 1.05
Mean Error Rate: 62.53%
Error Rate Variance: 1.05

MinMax Normalized Data Test Accuracy on Random Forest: 38.40%
```

Confusion Matrix (% per actual class):

	0	1	2	3	4	5	6	7	8	9	10	11
0	53.57	4.51	1.60	5.53	3.78	2.47	2.91	7.86	8.30	1.60	5.97	1.89
1	27.73	31.25	0.78	6.64	3.12	2.34	2.73	9.77	5.08	1.56	6.25	2.73
2	24.28	8.67	23.12	4.05	0.58	0.00	5.20	15.03	6.94	1.16	7.51	3.47
3	20.00	7.50	1.94	30.56	3.06	4.17	4.72	5.28	1.67	6.67	12.22	2.22
4	26.32	2.63	2.63	6.32	25.26	3.68	10.00	4.21	3.16	2.11	11.05	2.63
5	12.50	8.04	7.14	13.39	6.25	19.64	8.04	2.68	1.79	6.25	11.61	2.68
6	19.58	2.92	2.50	5.83	5.42	1.67	26.67	1.67	5.42	3.75	22.08	2.50
7	25.90	5.25	5.25	4.59	2.95	1.64	1.31	36.39	5.90	1.64	6.23	2.95
8	30.22	2.61	3.73	2.61	2.24	2.61	2.99	5.60	37.31	0.37	6.34	3.36
9	19.21	5.96	2.65	7.95	1.32	3.97	3.97	7.95	4.64	25.17	11.26	5.96
10	9.34	3.61	1.91	7.64	3.61	1.70	7.01	1.91	1.70	2.34	58.60	0.64
11	24.43	3.82	3.82	5.34	0.00	2.29	6.11	11.45	3.82	3.82	14.50	20.61

Classification Report:

	precision	recall	f1-score	support
Action	0.40	0.54	0.46	687
Adventure	0.35	0.31	0.33	256
Fighting	0.33	0.23	0.27	173
Misc	0.38	0.31	0.34	360
Platform	0.32	0.25	0.28	190
Puzzle	0.22	0.20	0.21	112
Racing	0.31	0.27	0.29	240
Role-Playing	0.37	0.36	0.37	305
Shooter	0.40	0.37	0.39	268
Simulation	0.31	0.25	0.28	151
Sports	0.50	0.59	0.54	471
Strategy	0.26	0.21	0.23	131
accuracy			0.38	3344
macro avg	0.35	0.32	0.33	3344
weighted avg	0.38	0.38	0.38	3344

5. Analysis of Results

The overall classification accuracy across models was limited by two major challenges in the dataset:

- Severe class imbalance, where some genres (like Action and Sports) had hundreds of examples while others (like Puzzle and Strategy) had very few.
- Feature overlap across genres, especially in review scores and sales figures, which made it difficult for classifiers to distinguish between certain classes.

Key learning and insights from the project and experiments conducted are as follows. **Classifier Performance**

- Best-performing classifier on original data: Random Forest, due to its robustness to feature scale and non-linear decision boundaries.
- Worst-performing classifier on original data: Naive Bayesian, which performed poorly due to unrealistic independence assumptions and lack of feature discrimination.
- Among all Bayesian variants:
    - Multivariate Bayesian Classifier achieved the highest accuracy.
    - Naive Bayes consistently underperformed. **Dimensionality Reduction**

- LDA (Linear Discriminant Analysis) proved to be the most effective dimensionality reduction technique, improving accuracy across all classifiers by projecting data in a class-aware manner.
- PCA (Principal Component Analysis) lowered classifier accuracy when applied to raw data likely because it discards class label information during projection.
- However, PCA performed better when applied on normalized data, producing more stable results and compact representations. **Normalization Impact**
- Both Min-Max Scaling and Z-Score Normalization produced similar accuracy levels across classifiers.
- Normalization was essential only for KNN, which resulted in improving accuracy.
- For models like Random Forest and Logistic Regression, normalization had little effect, and performance remained stable on original data.

Effect of Training Data Size

- All classifiers performed best when the training data size was large (i.e., using a 90% train and 10% test split).
- Comparative degradation in performance was observed when using less training data, particularly with a 70% train and 30% test split.
- Although the overall change in accuracy was not drastic, improvements of up to 5% were observed when increasing the size of the training set.
- This confirms the expected trend that classifiers benefit from more examples during training, particularly in multi-class settings with imbalanced distributions.

Class Imbalance Effects

- The imbalance in class distribution significantly affected genre classification.
- Dominant classes like Action and Sports were overrepresented in predictions and were less confused in confusion matrix, while minority classes like Puzzle, Strategy, and Platform were often misclassified and were more confused in confusion matrix.
- Treating all classes equally (without weighting or balancing) led to poorer performance, especially in macro-averaged metrics.

Genre-Level Prediction Quality

- Most accurately predicted genres as per the confusion matrix and report:
  Action, Sports, Shooter
  These classes had more samples and distinct patterns.
- Most confused/poorly predicted genres as per the confusion matrix and report:
  Puzzle, Strategy, Platform
  These suffered from data scarcity and overlapping feature distributions.

6. Summary and Conclusions

This project addressed a multi-class classification problem using a video game dataset, aiming to predict the genre of each game based on numerical and categorical attributes such as sales, scores, and platform. A variety of machine learning classifiers, normalization techniques, and dimensionality reduction methods were applied to evaluate their impact on classification accuracy and stability.

Key insights from the experimental analysis include:

- The overall classification accuracy was limited primarily due to class imbalance and overlapping feature distributions among genres.
- Random Forest was the most effective classifier on the original dataset, offering high accuracy and resilience without the need for feature scaling.
- LDA (Linear Discriminant Analysis) consistently enhanced model performance and was the most successful dimensionality reduction technique.
- Normalization was crucial only for K-Nearest Neighbors, while other classifiers performed similarly across normalized and unnormalized data.
- Naive Bayes struggled due to its simplistic assumptions, and PCA reduced accuracy unless it was applied after normalization.
- All classifiers achieved better results when trained on larger datasets (e.g., 90% training data, 10% test)

Beyond classifier metrics, the study revealed structural challenges in the dataset. Genres like Puzzle and Strategy had few examples and poorly defined boundaries, leading to low precision and high confusion. Conversely, genres such as Action and Sports benefited from higher representation and clearer patterns in the data.

These findings emphasize the importance of data quality and distribution in multi-class problems. While machine learning models can perform well under ideal conditions, real-world datasets require careful handling of imbalance, appropriate feature transformations, and robust evaluation.

In future work, expanding the feature set (e.g., including textual descriptions or metadata) and applying advanced balancing techniques could further improve performance, especially for underrepresented genres.

7. References • Dataset source - https://www.kaggle.com/datasets/sidtwr/videogames-sales-dataset

- Classification of Games using Support Vector Machine https://arxiv.org/pdf/2105.05674

- 10 Popular Machine Learning Algorithms for Solving Classification Problems https://medium.com/@howtodoml/10-popular-ml-algorithms-for-solvingclassificationproblems-b3bc1770fbdc