澳門大學

UNIVERSIDADE DE MACAU

UNIVERSITY OF MACAU

UNIVERSITY OF MACAU
FACULTY OF SCIENCE AND TECHNOLOGY

MATH3017 Course Project Report

Video Game Sales Analysis

Students' Number & Name

DC127235, YANG ZHEYU

DC126940, YANG DISHEN

DC127244, CHEN RISHENG

DC127018, SHI JIAYU

DC127217, XU WENXIN

Date of Submission:30/04/2024

Video Game Sales Analysis

Summary

The sales of video games are affected by many factors in different regions. In our report, we are aimed to create models to predict the sales and select the best algorithm by selecting them.

For task 1, after preprocessing and visualization, we find that the sales differ greatly in different regions. We apply **Ridge, Lasso** and **Elastic network regression** for each region on our dataset after tuning the hyperparameter by **K-folds cross validation** and **Grid search**, where we find that Elastic can not optimize Ridge because of poor performance of Lasso.

For task 2, we output the top ten coefficients of regression to find that **Publishers** and **Platforms** of the video games are the most important features, which is verified by **PCA**. Then we apply **Random forest** to classificate and predict the sales, where Random Forest fits best when comparing with above regressions.

For task 3, considering the sensitivity analysis, we apply our Model on similar dataset Walmart sales. The model also fits well, thus we can ensure the generalization of our model.

Keyword Ridge regression, Lasso regression, Elastic network regression, Random forest classification, Cross-validation, PCA

Table of Content

1 Introduction

1.1 Tasks Background

The video game industry is a rapidly growing and highly competitive market. For game developers and publishers, accurately predicting the sales of a new game is crucial for making informed marketing and production decisions. This study aims to develop a reliable predictive model that utilizes various game attributes, such as platform, genre, and ratings, to forecast global sales.

We will employ and compare several commonly used regression and machine learning methods, including Ridge Regression, Lasso Regression, Elastic Net, and Random Forest. These models will be trained and tested on a dataset containing sales data for over 1,000 video games. By comparing the predictive performance of these models using metrics such as mean squared error (MSE) and coefficient of determination (R^2), we can determine which approach is best suited for this prediction task.

The findings of this research can help gaming companies better understand the key factors influencing game sales and provide valuable decision support for future game development and publishing. Moreover, this work demonstrates an application of machine learning techniques in the field of video games.

1.2 Tasks Restatement

- Develop predictive models to forecast the sales of video games in different regions, specifically focusing on the North American, European, Japanese, and global markets. By leveraging historical sales data and game attributes, we seek to build models that provide reliable sales estimates for upcoming video game releases.

- By analyzing a comprehensive dataset containing information on video game characteristics, such as platform, genre, publisher, critic scores, and user ratings, we intend to uncover the most important variables driving sales performance. Understanding these influential factors will provide valuable insights for game developers and publishers to optimize their game design and marketing strategies.

- Compare and evaluate the performance of different predictive models, including Ridge Regression, Lasso Regression, Elastic Net, and Random Forest. By applying these models to the video game sales dataset, we will assess their predictive accuracy, generalization ability, and robustness. Through a rigorous comparison of these models, we aim to identify the most suitable approach for predicting video game sales in different regions.

2 Preparation

This project is from MATH3017 Data-Driven Sampling Methods provided by University of Macau, and taught by Prof. Hongwei Yuan. The project is based on the program, which is named “Video Game Sales” on Kaggle, and the data set is

“Video_Games_Sales_as_at_22_Dec_2016.csv ”. Pycharm is the compiler applied for this project, and we used Python 3.11 for the coding. The program related to some important packages like sklearn. For the sensitive analysis, we also use the data set “Walmart. csv” to test our models.

3 Our work

In this project, we first preprocessed the original data file of "Video_Games_Sales_as_at_22_Dec_2016.csv", and then carried out data visualization and statistical analysis. Ridge regression model and Lasso regression model were established, and Elastic network regression was used to combine the two models. At the same time, we used another classification model: random forest to predict the results. After obtaining the results, we changed the number of end trees of random forest through the ridge trace map, and replaced other data sets applied to our model for model sensitivity analysis. At the end of the conclusion, the generalization of the model is guaranteed.

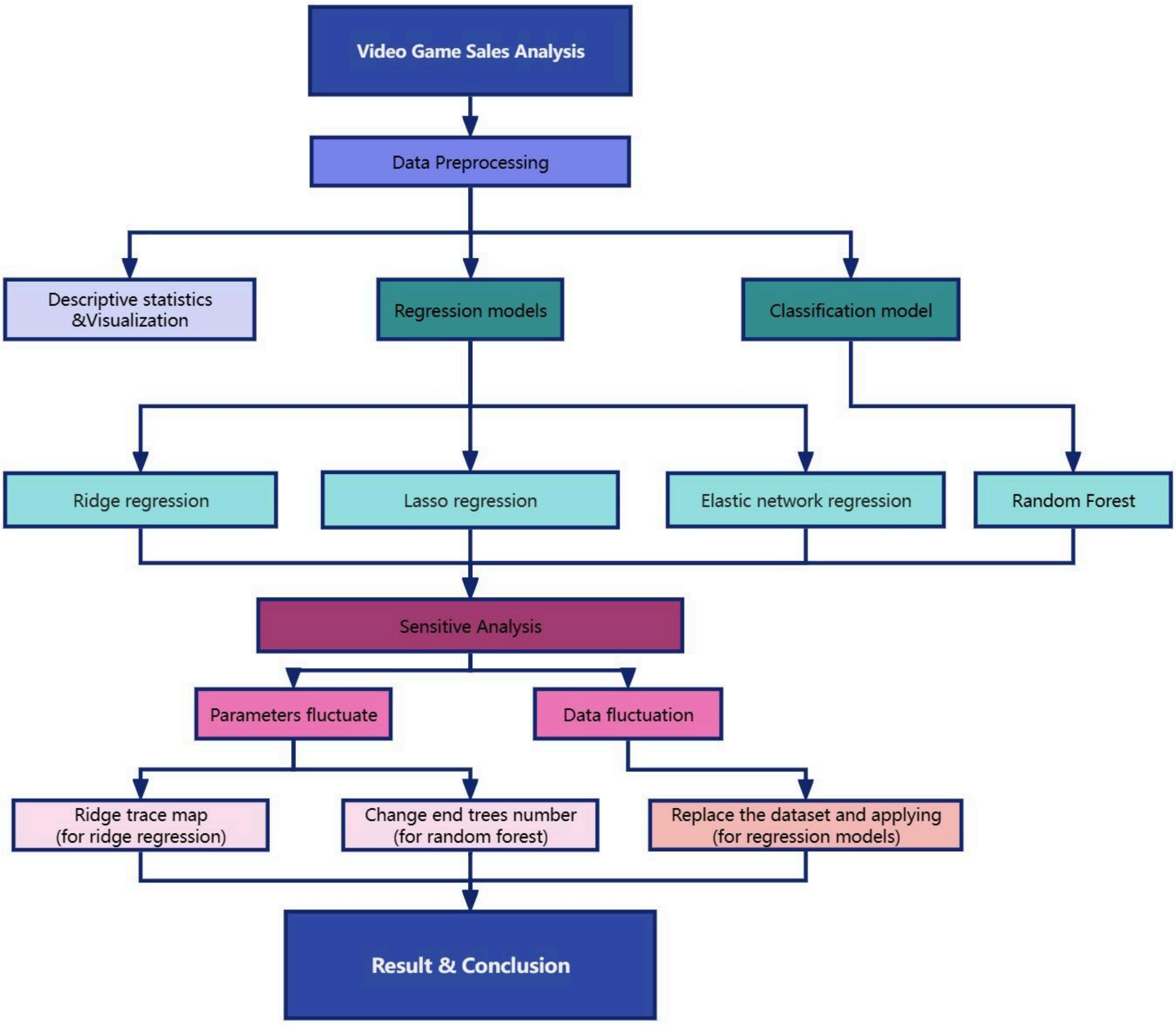

Video Game Sales Analysis
Data Preprocessing
Descriptive statistics &Visualization
Regression models
Classification model
Ridge regression
Lasso regression
Elastic network regression
Random Forest
Sensitive Analysis
Parameters fluctuate
Data fluctuation
Ridge trace map (for ridge regression)
Change end trees number (for random forest)
Replace the dataset and applying (for regression models)
Result & Conclusion

Figure 1. Our work in the whole project

4 Notations

Table 1. Notations

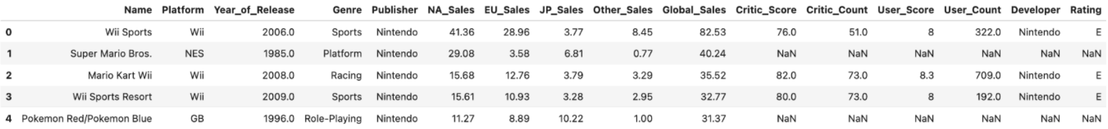
Symbol	Definition
Y	response variable for observations
ϵ	error value
β	parameter
α	parameter
λ	regularization penalty

5 Data Preprocessing & Visualization

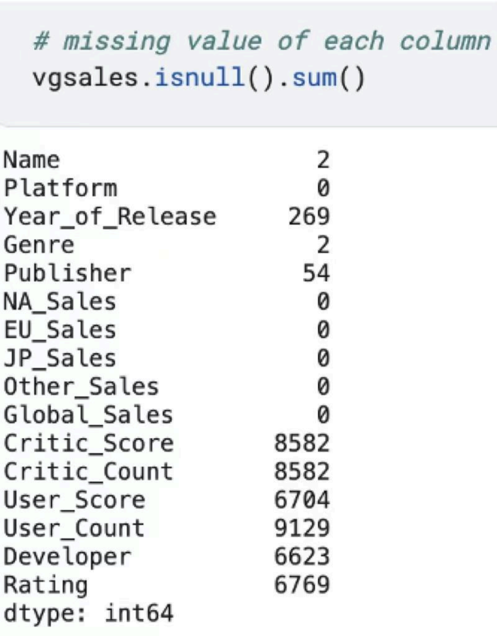
5.1 Data Preprocessing

- **Data Sets Overview**:
  The dataset contains information about video games, including their sales performance, various attributes and missing values.

Table 2. Data Sets First Five Lines



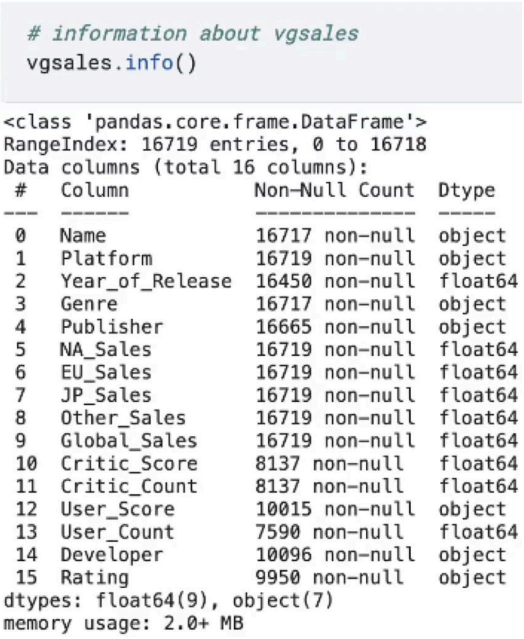
	Name	Platform	Year_of_Release	Genre	Publisher	NA_Sales	EU_Sales	JP_Sales	Other_Sales	Global_Sales	Critic_Score	Critic_Count	User_Score	User_Count	Developer	Rating
0	Wii Sports	Wii	2006.0	Sports	Nintendo	41.36	28.96	3.77	8.45	82.53	76.0	51.0	8	322.0	Nintendo	E
1	Super Mario Bros.	NES	1985.0	Platform	Nintendo	29.08	3.58	6.81	0.77	40.24	NaN	NaN	NaN	NaN	NaN	NaN
2	Mario Kart Wii	Wii	2008.0	Racing	Nintendo	15.68	12.76	3.79	3.29	35.52	82.0	73.0	8.3	709.0	Nintendo	E
3	Wii Sports Resort	Wii	2009.0	Sports	Nintendo	15.61	10.93	3.28	2.95	32.77	80.0	73.0	8	192.0	Nintendo	E
4	Pokemon Red/Pokemon Blue	GB	1996.0	Role-Playing	Nintendo	11.27	8.89	10.22	1.00	31.37	NaN	NaN	NaN	NaN	NaN	NaN



```
# missing value of each column
vgsales.isnull().sum()
```

```
Name                 2
Platform             0
Year_of_Release    269
Genre                2
Publisher           54
NA_Sales             0
EU_Sales             0
JP_Sales             0
Other_Sales          0
Global_Sales         0
Critic_Score      8582
Critic_Count      8582
User_Score        6704
User_Count        9129
Developer         6623
Rating            6769
dtype: int64
```

Figure 2. Missing Value



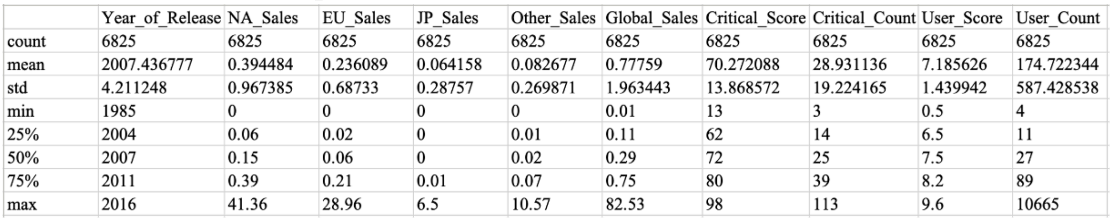
```
# information about vgsales
vgsales.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 16719 entries, 0 to 16718
Data columns (total 16 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   Name             16717 non-null  object
 1   Platform         16719 non-null  object
 2   Year_of_Release  16450 non-null  float64
 3   Genre            16717 non-null  object
 4   Publisher        16665 non-null  object
 5   NA_Sales         16719 non-null  float64
 6   EU_Sales         16719 non-null  float64
 7   JP_Sales         16719 non-null  float64
 8   Other_Sales      16719 non-null  float64
 9   Global_Sales     16719 non-null  float64
 10  Critic_Score     8137 non-null   float64
 11  Critic_Count     8137 non-null   float64
 12  User_Score       10015 non-null  object
 13  User_Count       7590 non-null   float64
 14  Developer        10096 non-null  object
 15  Rating           9950 non-null   object
dtypes: float64(9), object(7)
memory usage: 2.0+ MB
```

Figure 3. Data Information

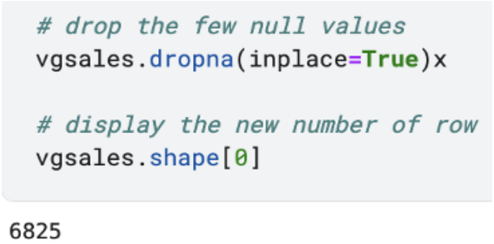
- **Descriptive Statistical Analysis**:
  Calculate measures of central tendency (mean) and dispersion (standard deviation, range) for each numerical feature.

Table 3. Descriptive Statistics of Video Game Sales Dataset



	Year_of_Release	NA_Sales	EU_Sales	JP_Sales	Other_Sales	Global_Sales	Critical_Score	Critical_Count	User_Score	User_Count
count	6825	6825	6825	6825	6825	6825	6825	6825	6825	6825
mean	2007.436777	0.394484	0.236089	0.064158	0.082677	0.77759	70.272088	28.931136	7.185626	174.722344
std	4.211248	0.967385	0.68733	0.28757	0.269871	1.963443	13.868572	19.224165	1.439942	587.428538
min	1985	0	0	0	0	0.01	13	3	0.5	4
25%	2004	0.06	0.02	0	0.01	0.11	62	14	6.5	11
50%	2007	0.15	0.06	0	0.02	0.29	72	25	7.5	27
75%	2011	0.39	0.21	0.01	0.07	0.75	80	39	8.2	89
max	2016	41.36	28.96	6.5	10.57	82.53	98	113	9.6	10665

- **Missing Value Handling**:
  Removing records with missing values.



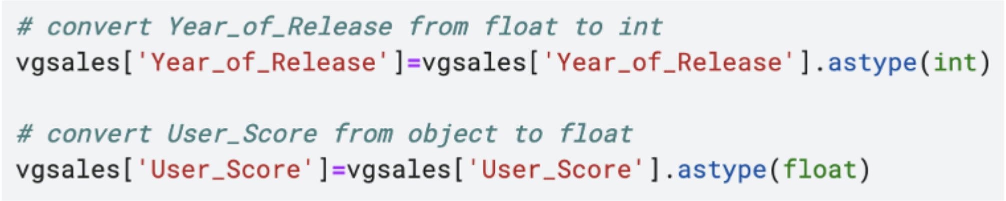
```
# drop the few null values
vgsales.dropna(inplace=True)x

# display the new number of row
vgsales.shape[0]
```

```
6825
```

Figure 4. Codes for Handling Missing Values

- **Data Type Adjustment**:
  Converting features to their appropriate data types. For instance, ‘Year_of_Release’ is converted to integer type. 'User_Score' is converted from object to numeric type.

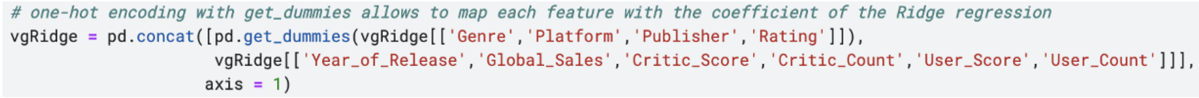


```
# convert Year_of_Release from float to int
vgsales['Year_of_Release']=vgsales['Year_of_Release'].astype(int)

# convert User_Score from object to float
vgsales['User_Score']=vgsales['User_Score'].astype(float)
```

Figure 5. Codes for Data Type Adjustment

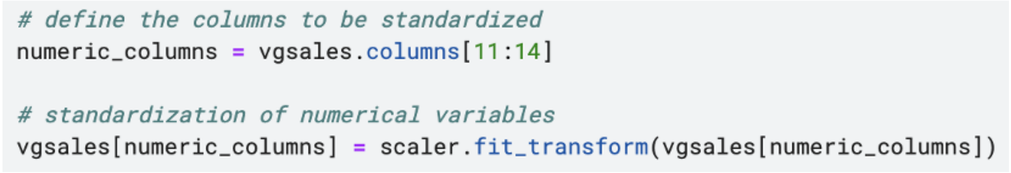
- **One-Hot Encoding**:
  Identify categorical features that need to be encoded, such as Platform, Genre, Publisher, and Rating. Perform one-hot encoding on these categorical features to convert them into binary vectors.



```
# one-hot encoding with get_dummies allows to map each feature with the coefficient of the Ridge regression
vgRidge = pd.concat([pd.get_dummies(vgRidge[['Genre','Platform','Publisher','Rating']]),
                     vgRidge[['Year_of_Release','Global_Sales','Critic_Score','Critic_Count','User_Score','User_Count']]],
                    axis = 1)
```

Figure 6. Codes for One-Hot Encoding

- **Data Standardization**:
  Applying standardization techniques to scale the numerical features to a common range and standardization helps to eliminate the effect of different scales and ensures that all features contribute equally to the model.

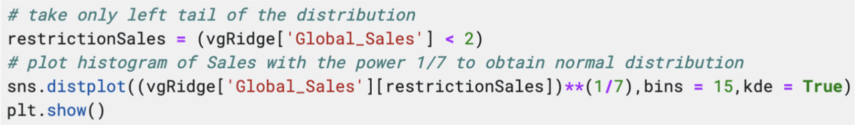


```
# define the columns to be standardized
numeric_columns = vgsales.columns[11:14]

# standardization of numerical variables
vgsales[numeric_columns] = scaler.fit_transform(vgsales[numeric_columns])
```

Figure 7. Codes for Data Standardization

- **Extreme Values Handling**:
  Identifying and handling extreme values or outliers in the dataset. In our dataset, these are represented by games with exceptionally high global sales. The global sales are shown as an example and other sales are handled similarly. First, we make a plot to inspect the distribution of the target Y. It can be seen that the distribution is right-skewed, so we take only the left tail of the distribution and plot the histogram of Sales with the power one seventh to obtain normal distribution. This helps stabilize variance and make the data more suitable for linear regression models.



```
# take only left tail of the distribution
restrictionSales = (vgRidge['Global_Sales'] < 2)
# plot histogram of Sales with the power 1/7 to obtain normal distribution
sns.distplot((vgRidge['Global_Sales'][restrictionSales])**(1/7),bins = 15,kde = True)
plt.show()
```

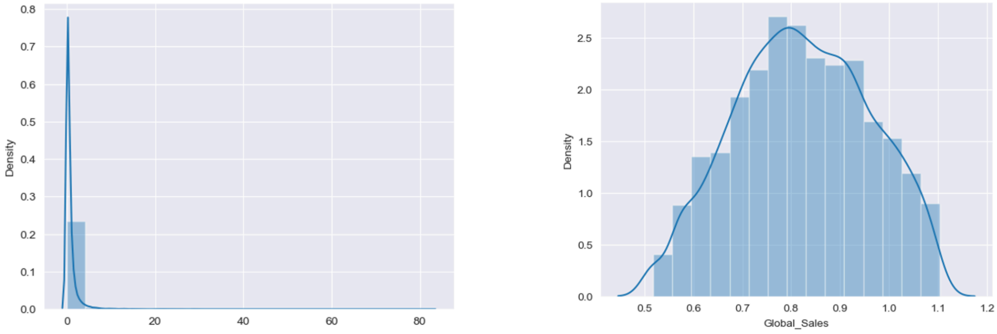
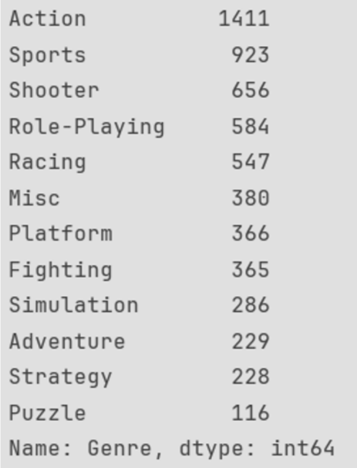


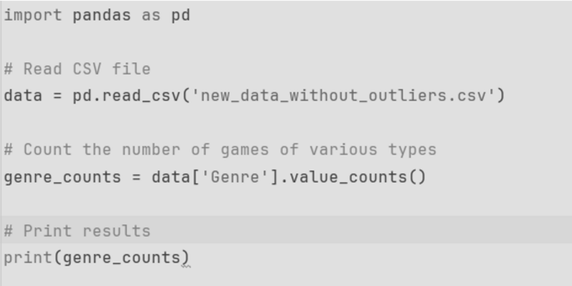
Figure 8. Results for Extreme Values Handling

5.2 Visualization

- **Number of Games of Various Types:**
  The Action category has the highest number of games, followed by the Sports and Shooter categories.



```
Action          1411
Sports           923
Shooter          656
Role-Playing     584
Racing           547
Misc             380
Platform         366
Fighting         365
Simulation       286
Adventure        229
Strategy         228
Puzzle           116
Name: Genre, dtype: int64
```



```
import pandas as pd

# Read CSV file
data = pd.read_csv('new_data_without_outliers.csv')

# Count the number of games of various types
genre_counts = data['Genre'].value_counts()

# Print results
print(genre_counts)
```

Figure 9. Result

Figure 10. Code

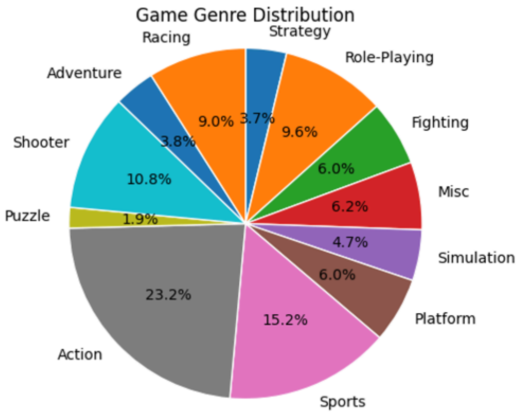
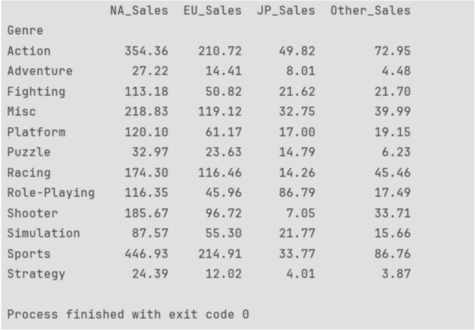


Figure 11. Distribution of Video Game Genres

- **Game Sales in Different Markets:**
  Action games are most popular in the North American and European markets, while the Role-Playing genre is more popular in the Japanese market.



```
              NA_Sales  EU_Sales  JP_Sales  Other_Sales
Genre
Action          354.36    210.72     49.82        72.95
Adventure        27.22     14.41      8.01         4.48
Fighting        113.18     50.82     21.62        21.70
Misc            218.83    119.12     32.75        39.99
Platform        120.10     61.17     17.00        19.15
Puzzle           32.97     23.63     14.79         6.23
Racing          174.30    116.46     14.26        45.46
Role-Playing    116.35     45.96     86.79        17.49
Shooter         185.67     96.72      7.05        33.71
Simulation       87.57     55.30     21.77        15.66
Sports          446.93    214.91     33.77        86.76
Strategy         24.39     12.02      4.01         3.87

Process finished with exit code 0
```

Figure 12. Result

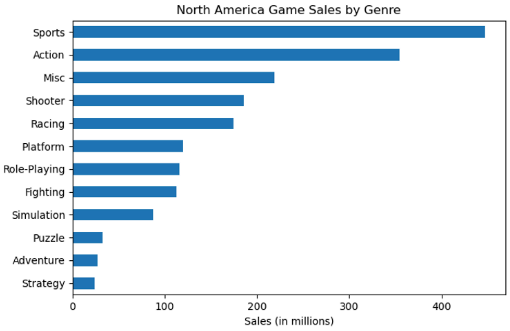


Figure 13. Game Sales in NA

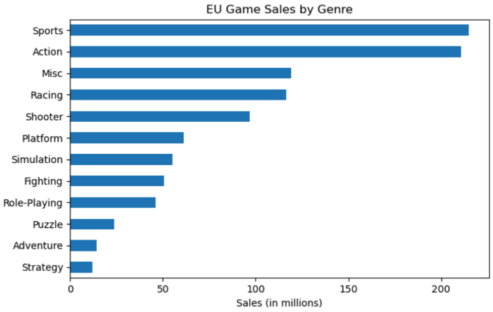


Figure 14. Game Sales in EU

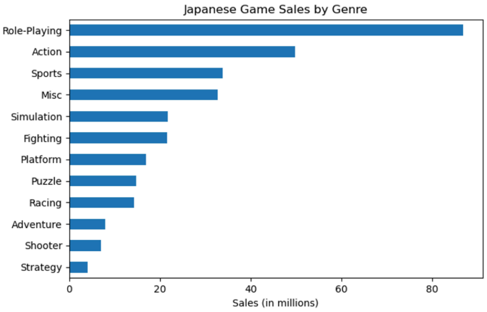


Figure 15. Game Sales in JP

- **Number of Popular Games On Each Platform:**
  The PS2 has the most popular games, followed by the X360 and PS3.

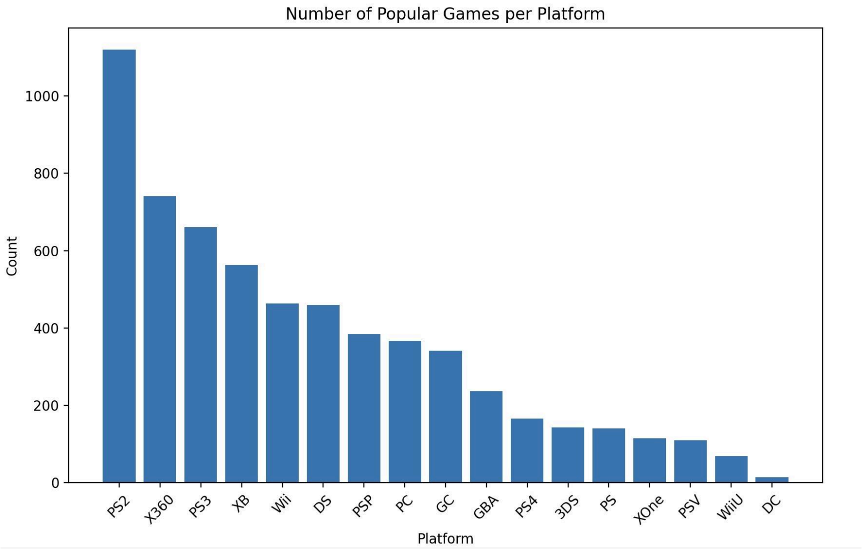
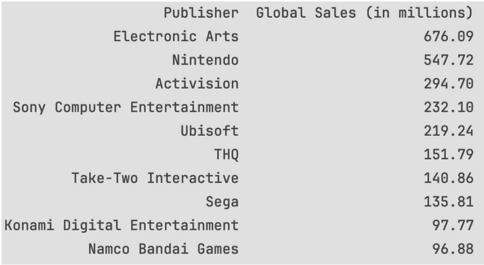


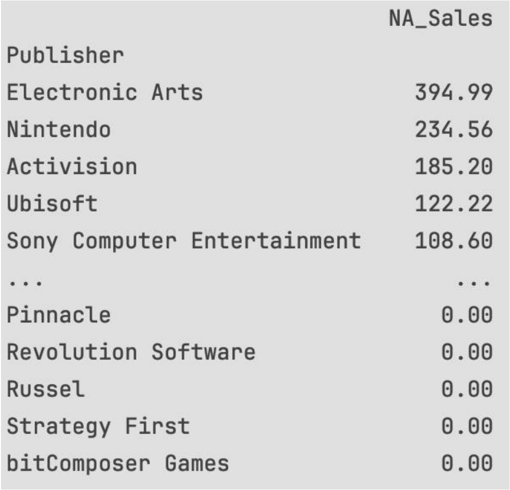
Figure 16. Number of Popular Games per Platform

- **Sales of Global Gaming Companies:**
  The top five companies in terms of sales are Electronic Arts, Nintendo, Activision, Sony Computer Entertainment and Ubisoft. In terms of market segments, it can be seen that in the North American market and the European market, the top five selling companies are almost the same as the top five selling companies in total. In the Japanese market, the top five companies have changed, and the top five companies are all Japanese video game publishers.



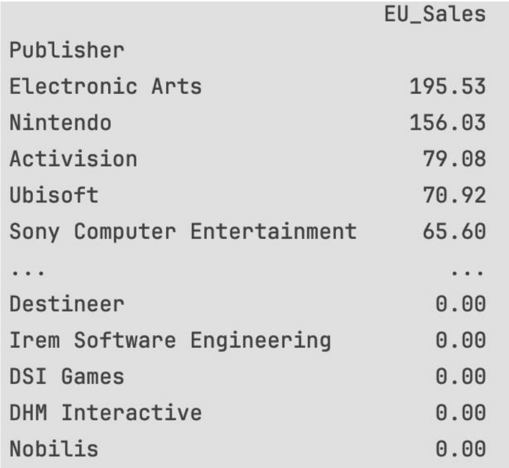
```
                   Publisher  Global Sales (in millions)
             Electronic Arts                      676.09
                    Nintendo                      547.72
                  Activision                      294.70
 Sony Computer Entertainment                      232.10
                     Ubisoft                      219.24
                         THQ                      151.79
        Take-Two Interactive                      140.86
                        Sega                      135.81
Konami Digital Entertainment                       97.77
          Namco Bandai Games                       96.88
```

Figure 17. Sales of Global Gaming Company



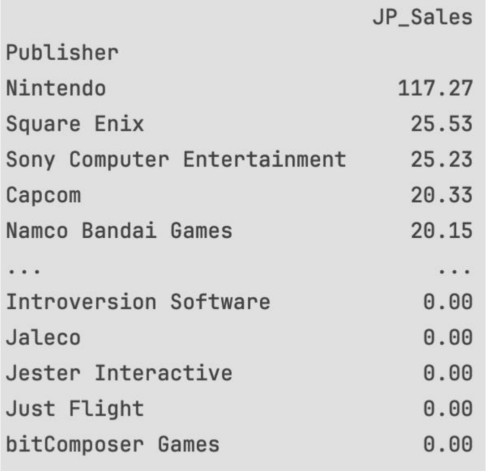
```
                             NA_Sales
Publisher
Electronic Arts                394.99
Nintendo                       234.56
Activision                     185.20
Ubisoft                        122.22
Sony Computer Entertainment    108.60
...                               ...
Pinnacle                         0.00
Revolution Software              0.00
Russel                           0.00
Strategy First                   0.00
bitComposer Games                0.00
```

Figure 18. Sales of North America Gaming Company



```
                              EU_Sales
Publisher
Electronic Arts                 195.53
Nintendo                        156.03
Activision                       79.08
Ubisoft                          70.92
Sony Computer Entertainment      65.60
...                                ...
Destineer                         0.00
Irem Software Engineering         0.00
DSI Games                         0.00
DHM Interactive                   0.00
Nobilis                           0.00
```

Figure 19. Sales of Europe Gaming Company



```
                              JP_Sales
Publisher
Nintendo                        117.27
Square Enix                      25.53
Sony Computer Entertainment      25.23
Capcom                           20.33
Namco Bandai Games               20.15
...                                ...
Introversion Software             0.00
Jaleco                            0.00
Jester Interactive                0.00
Just Flight                       0.00
bitComposer Games                 0.00
```

Figure 20. Sales of Japanese Gaming Company

6 Ridge Regression

6.1 Introduce to Ridge Regression

Ridge regression is a technique for linear regression that addresses multicollinearity and overfitting. It adds a penalty term to the cost function, shrinking the coefficient estimates towards zero. This helps stabilize the estimates and maintains all predictors in the model.

Ridge regression is useful when predictor variables are highly correlated and prevents overfitting. It strikes a balance between complexity and generalization. It is computationally efficient and suitable for large datasets. Overall, ridge regression is a valuable tool for robust and reliable linear regression models.

Ridge regression is based on the traditional linear regression algorithm and adds a regular term L2.

For linear regression models：

$$Y = X\beta + \epsilon \text{ , where } \epsilon \sim N(0, \sigma^2) \qquad (1)$$

The least squares model loss function is as follows:

$$L_{ols}(\hat{\beta}) = \sum_{i=1}^{n} (y_i - x'\hat{\beta})^2 = \left\| y - X\hat{\beta} \right\|^2 \qquad (2)$$

The residual sum of squares $L_{ols}(\hat{\beta})$ should be as small as possible, but the variance in the model will have an impact when the variables are highly correlated.The ridge regression formula is as follows:

$$L_{ridge}(\hat{\beta}) = \sum_{i=1}^{n} (y_i - x_i'\hat{\beta})^2 + \lambda \sum_{j=1}^{m} \hat{\beta}^2 = \left\| y - X\hat{\beta} \right\|^2 + \lambda \left\| \hat{\beta} \right\|^2 \qquad (3)$$

$\hat{\beta}_{ridge} = (X'X + \lambda I)^{-1}(X'Y)$.When λ approaches 0, $\hat{\beta}_{ridge}$ approaches $\hat{\beta}_{OLS}$ infinitely; When the coefficients are reduced it will lead to lower variance, which will lead to lower error values. When λ approaches infinity ∞, $\widehat{\beta}_{ridge}$ approaches 0 infinitely.Therefore, ridge regression reduces the complexity of the model, but it does not reduce the number of variables, but only narrows their influence.

6.2 Model Establishment

In our model, we randomly split the dataset into 4 parts and choose one of them to be the test set. Before using the package sklearn.linear_model, we used K-folds Cross Validation and Grid Search to find a relatively best alpha to start our machine learning. Then we output the result.

6.2.1 Tuning the Alpha

• **K-Fold Cross-Validation.** K-fold cross-validation partitions data into k subsets. Each fold serves as validation while the rest are for training. This process repeats n times, ensuring robust evaluation. It improves model reliability and generalization by efficiently utilizing data for assessment and development. Here use cv(average mse) to estimate the performance of alpha.

$$CV_{(n)} = \frac{1}{n}\sum_{i=1}^{n} MSE_i, \quad CV_{(n)} = \frac{1}{n}\sum_{i=1}^{n} \left(\frac{y_i - \widehat{y}_i}{1-h_i}\right)^2 \qquad (4)$$

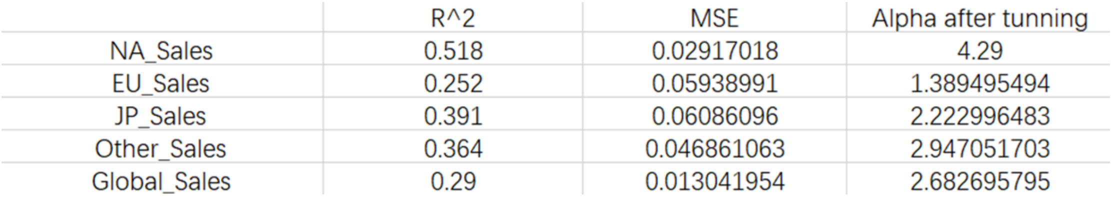
• **Grid Search.** Grid search is a hyperparameter optimization technique in machine learning. It systematically searches through a specified grid of hyperparameters, evaluating model performance for each combination. By exhaustively exploring options, it helps find the best hyperparameters for optimal model performance, enhancing accuracy and generalization.

In our model, we set the value of k to 5 and the value of n to 3. And set the grid range as logspace(-1,1).

6.2.2 Output the Result

In our model, we set the value of k to 5 and the value of n to 3. And set the grid range as logspace(-1,1). Then we get the result.

Table 4. Result of Ridge regression



	R^2	MSE	Alpha after tunning
NA_Sales	0.518	0.02917018	4.29
EU_Sales	0.252	0.05938991	1.389495494
JP_Sales	0.391	0.06086096	2.222996483
Other_Sales	0.364	0.046861063	2.947051703
Global_Sales	0.29	0.013041954	2.682695795

As we are using the real data, the performance of Ridge regression is acceptable.

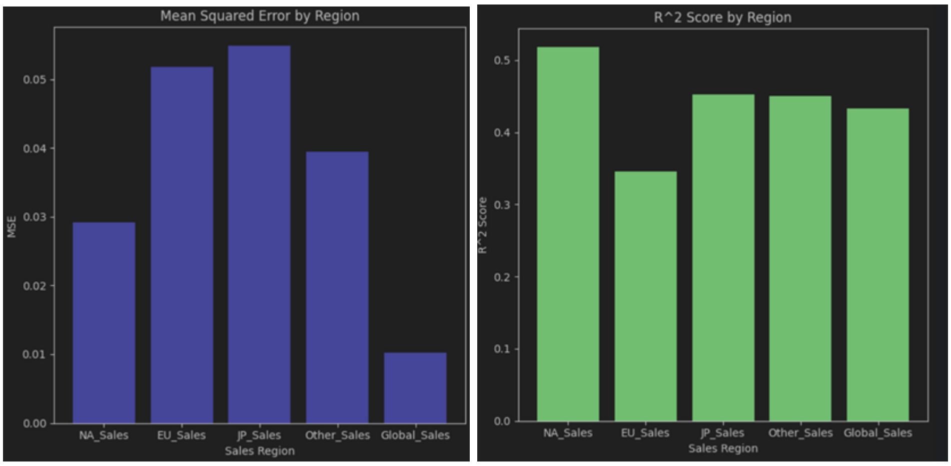
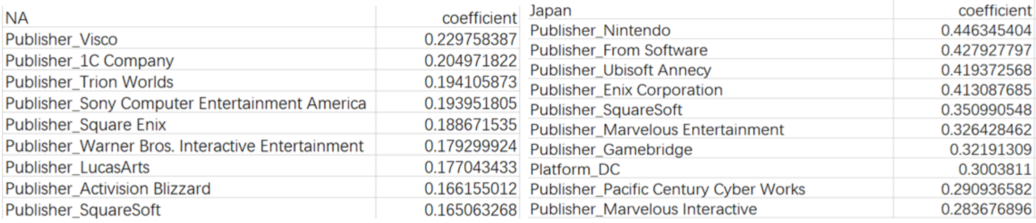


Figure 21 & 22. Performance of Ridge regression

We also output the 10 largest coefficients of features influencing the game sales. As we have used one-hot code to deal with the dataset before, mainly the features here consist of publishers and platform name.If we think about the games we buy, we pay more attention to the publisher of the game, so this makes sense. For instance, Nintendo publisher's coefficient is relatively large in Japan, and indeed the publisher's influence in Japan is very large

Table 5. Top 10 coefficient of Video Games sales in NA and Japan

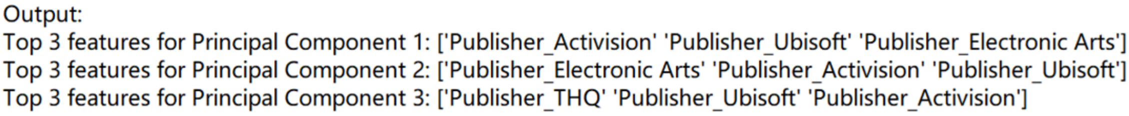


NA	coefficient	Japan	coefficient
Publisher_Visco	0.229758387	Publisher_Nintendo	0.446345404
Publisher_1C Company	0.204971822	Publisher_From Software	0.427927797
Publisher_Trion Worlds	0.194105873	Publisher_Ubisoft Annecy	0.419372568
Publisher_Sony Computer Entertainment America	0.193951805	Publisher_Enix Corporation	0.413087685
Publisher_Square Enix	0.188671535	Publisher_SquareSoft	0.350990548
Publisher_Warner Bros. Interactive Entertainment	0.179299924	Publisher_Marvelous Entertainment	0.326428462
Publisher_LucasArts	0.177043433	Publisher_Gamebridge	0.32191309
Publisher_Activision Blizzard	0.166155012	Platform_DC	0.3003811
Publisher_SquareSoft	0.165063268	Publisher_Pacific Century Cyber Works	0.290936582
		Publisher_Marvelous Interactive	0.283676896

6.2.3 Verify the Result

We use the Principal Component Analysis algorithm to check if our output is rational.

- **Principal Component Analysis.** PCA is a dimensionality reduction technique in machine learning and statistics. It identifies the principal components, or directions, in a dataset that capture the most variance. By projecting data onto these components, PCA reduces complexity while preserving essential information, aiding in visualization, feature selection, and noise reduction.



```
Output:
Top 3 features for Principal Component 1: ['Publisher_Activision' 'Publisher_Ubisoft' 'Publisher_Electronic Arts']
Top 3 features for Principal Component 2: ['Publisher_Electronic Arts' 'Publisher_Activision' 'Publisher_Ubisoft']
Top 3 features for Principal Component 3: ['Publisher_THQ' 'Publisher_Ubisoft' 'Publisher_Activision']
```

Figure 23. PCA output

From the figure of PCA output, we can conclude the publisher is the most important feature affecting the video games sales.

7 Lasso and Elastic Regression

7.1 Introduction to Lasso and Elastic Network

Lasso regression is a linear regression technique that performs both variable selection and regularization. It adds a penalty term to the cost function, which is the sum of squared residuals, proportional to the absolute values of the regression coefficients. This encourages sparsity in the coefficient estimates, effectively eliminating irrelevant variables from the model. Lasso regression is useful for feature selection and dealing with high-dimensional datasets.

Lasso regression penalizes the sum of the absolute values of the coefficients.The Lasso regression formula is as follows:

$$L_{lasso}(\widehat{\beta}) = \sum_{i=1}^{n}(y_i - x_i'\widehat{\beta})^2 + \lambda \sum_{j=1}^{m}\left|\widehat{\beta}_j\right| \tag{5}$$

If the λ value is zero, it is equivalent to the basic OLS model. However, given appropriate values of λ, Lasso regression can make some coefficients go to zero. Larger values of λ shrink more features to zero. This can completely eliminate some features and can provide a subset of predictions that helps mitigate issues of multicollinearity and model complexity. If the variable does not shrink to zero, it means that its importance is high, so it can be used to filter feature variables.

Elastic Net regression is a combination of ridge and lasso regression. It adds a penalty term that is a mixture of the ridge and lasso penalties. This allows it to handle both multicollinearity and variable selection. Elastic Net is particularly effective when there are many correlated predictors and a subset of them is relevant. The elastic network model is as follows:

$$L_{enet}(\widehat{\beta}) = \frac{\sum_{i=1}^{n}(y_i - x_i'\widehat{\beta})^2}{2n} + \lambda(\frac{1-\alpha}{2}\sum_{j=1}^{m}\widehat{\beta}_j^2 + \alpha\sum_{j=1}^{m}\left|\widehat{\beta}_j\right|) \tag{6}$$

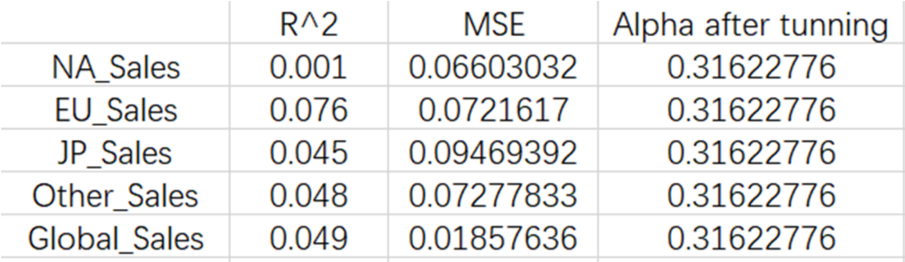
α = 0 corresponds to the ridge regression model, and α = 1 corresponds to the Lasso regression model. This article sets α = 0.5 to optimize the model. On the one hand, it can inherit the stability of ridge regression, and on the other hand, it can perform group effects in the case of highly correlated variables. Elastic network regression combines the advantages of ridge regression and Lasso regression to reduce some coefficients to achieve the purpose of feature variable screening.

Both Lasso and Elastic Net are powerful techniques for linear regression that provide regularization and feature selection capabilities, making them valuable tools in predictive modeling and data analysis.

7.2 Model Establishment

7.2.1 Lasso Regression

As we apply the same tuning method to the Lasso model as the Ridge Model, the hyper-parameter can not converge. After changing several times for the Grid Search range, we find that the upper bound of searching does not make sense to alpha and the lower bound -0.5 is always regarded as the best parameter choice. Also, if the lower bound we set is smaller than -0.5, the algorithm would not converge. Here our search range is in logspace, -0.5 represents the value 10^-0.5 of alpha.



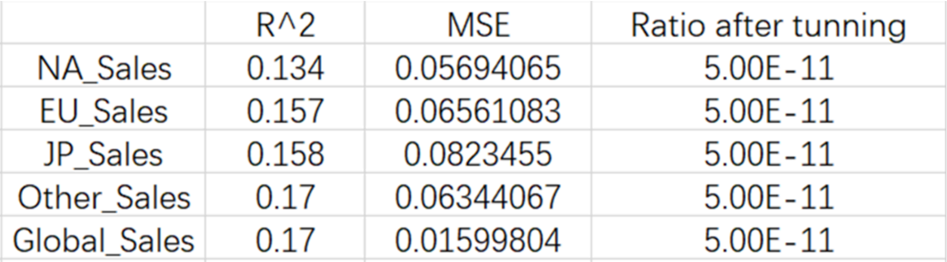
	R^2	MSE	Alpha after tunning
NA_Sales	0.001	0.06603032	0.31622776
EU_Sales	0.076	0.0721617	0.31622776
JP_Sales	0.045	0.09469392	0.31622776
Other_Sales	0.048	0.07277833	0.31622776
Global_Sales	0.049	0.01857636	0.31622776

Figure 24. Output of Lasso regression

From the result of Lasso, we can find it is poor fit, we should not consider this method.

7.2.2 Elastic Network Regression

We know that Elastic network regression combines Ridge and Lasso to perform regression, but from these above results we know that lasso does not perform well on our dataset. So in our Elastic network regression model, we mainly use Ridge regression by fixing the alpha from the Ridge tuning part and setting the ratio Grid search range to linear (5^-11, 6^-11). Even though our search space is very small, we find that the ratio tuning is still the minimum.



	R^2	MSE	Ratio after tunning
NA_Sales	0.134	0.05694065	5.00E-11
EU_Sales	0.157	0.06561083	5.00E-11
JP_Sales	0.158	0.0823455	5.00E-11
Other_Sales	0.17	0.06344067	5.00E-11
Global_Sales	0.17	0.01599804	5.00E-11

Figure 25. Performance of Elastic network regression

Hence we conclude that inapplicability of Lasso leads to its negative contribution to Elastic network regression.

Ridge regression is the best algorithm among these three regressions.

8 Random Forest Classification

8.1 Introduction

Since ridge regression did not work well, we tried to reformulate the initial prediction problem into a more manageable classification problem by applying the random forest algorithm to classify the sales categories.

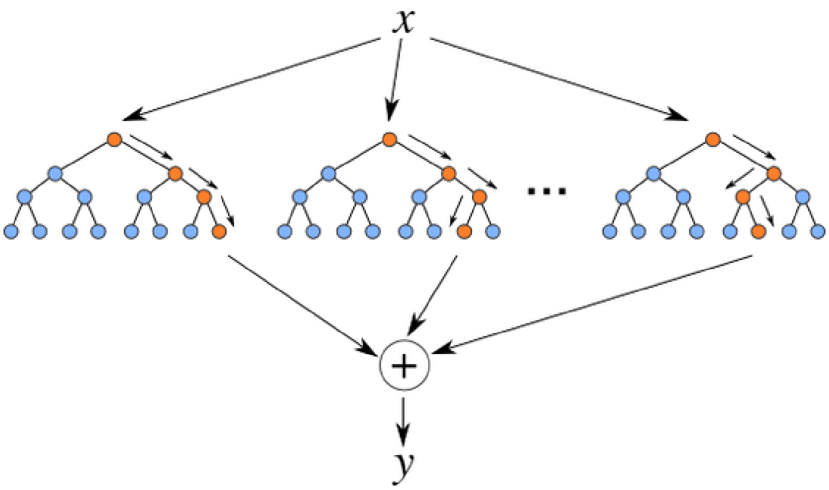


Figure 26. Principle of random forest

8.2 Pre-processing Steps

Firstly, we compute the conditions under which titles having “Global Sales” within a certain range will be labeled from 0 to 3. Then we draw the pieplot to show the distribution of “Sales category” on log scale with the new classes. It can be noticed from the output figure that the classes remain unbalanced, so we tried to address this problem using a resampling method.

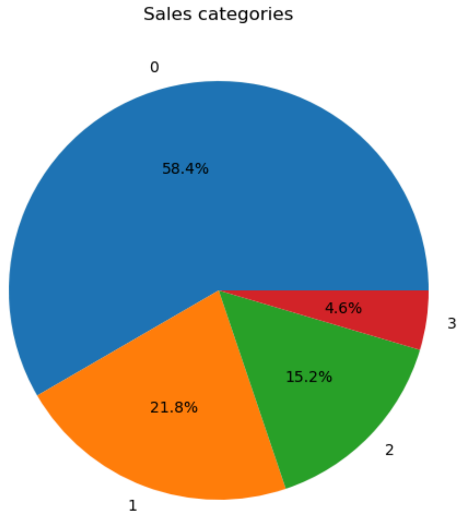
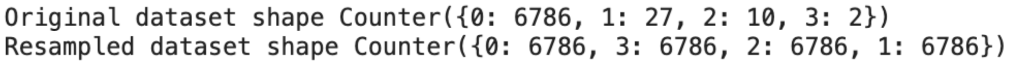


Figure 27. Distribution of Sales category on log scale

We employed the Random Over Sampler, a resampling technique that randomly duplicates instances from the minority classes to match the distribution of the majority class, and the shape of the pre-processed dataset is shown below.



```
Original dataset shape Counter({0: 6786, 1: 27, 2: 10, 3: 2})
Resampled dataset shape Counter({0: 6786, 3: 6786, 2: 6786, 1: 6786})
```

Figure 28. Output of the shape of the datasets

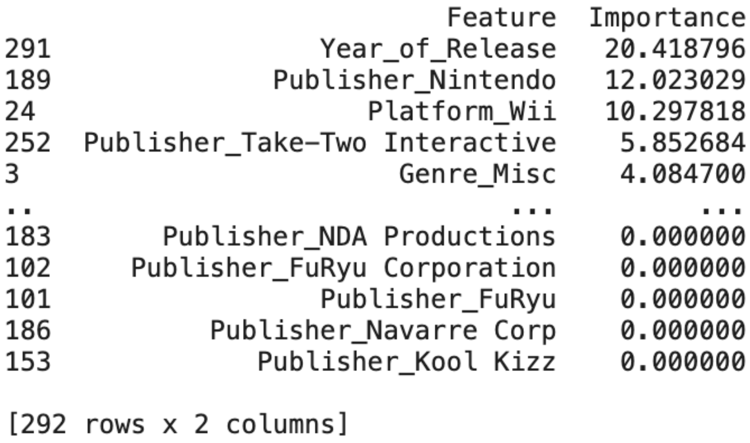
8.3 Apply and Result

8.3.1 Random Forest Classifier

After training a random forest classifier to predict the sales categories in the test set, we noticed that its performance far surpassed that of ridge regression, with the classifier achieving an accuracy of 97%. This result can be attributed to the preprocessing steps that were taken earlier. Firstly, classification problems are easier to handle than regression; secondly, the random oversampling made the sales categories balanced, which facilitated problem-solving.

8.3.2 Feature Importance

Finally, the significance of the features is calculated using the mean and standard deviation based on the results of each feature tree that makes up the forest. As can be seen, year is the most important feature. Nintendo and the gaming platform Wii also has a significant impact on sales.



```
                              Feature  Importance
291                    Year_of_Release   20.418796
189                 Publisher_Nintendo   12.023029
24                        Platform_Wii   10.297818
252  Publisher_Take-Two Interactive    5.852684
3                           Genre_Misc    4.084700
..                                 ...         ...
183          Publisher_NDA Productions    0.000000
102        Publisher_FuRyu Corporation    0.000000
101                    Publisher_FuRyu    0.000000
186            Publisher_Navarre Corp    0.000000
153               Publisher_Kool Kizz    0.000000

[292 rows x 2 columns]
```

Figure 29. Output of feature importance

9 Sensitive Analysis

9.1 Conception of Sensitive Analysis

Sensitivity analysis is a method of studying and analyzing the sensitivity of changes in the state or output of a system or model to changes in system parameters or surrounding conditions. Sensitivity analysis is often used in optimization methods to study the stability of the optimal solution when the original data is inaccurate or changes. Sensitivity analysis can also determine which parameters have a greater impact on the system or model. Therefore, sensitivity analysis is important in the evaluation of various options.

There are generally two types of sensitivity analysis. One is to let the parameters fluctuate to see if the model is normal. For example, the ridge trace plot converges as alpha increases; the other is to let the data fluctuate, similar to replacing the data.

9.2 Parameter Fluctuation

9.2.1 Ridge Trace Map

Ridge estimation is a biased estimation regression method specially used for collinear data analysis. It is essentially an improved least squares estimation method. The selection of ridge parameters in the regression coefficients of ridge estimation is a very important issue, and the most commonly used method is the ridge trace method.

The general principles for selecting k values in the ridge trace method are:

- The ridge estimate of each regression coefficient is basically stable;
- For regression coefficients with unreasonable signs when estimated by the least squares method, the sign of the ridge estimate will become reasonable;
- The regression coefficient has no absolute value that is not economically meaningful;
- The sum of squares of the residuals does not increase too much.

The determination of k by the ridge trace method lacks a strict and convincing theoretical basis, and there is a certain degree of subjectivity and artificiality. This seems to be an obvious shortcoming of the ridge trace method. However, the artificiality of determining the k value by the ridge trace method is exactly where the organic combination of qualitative analysis and quantitative analysis comes into play.

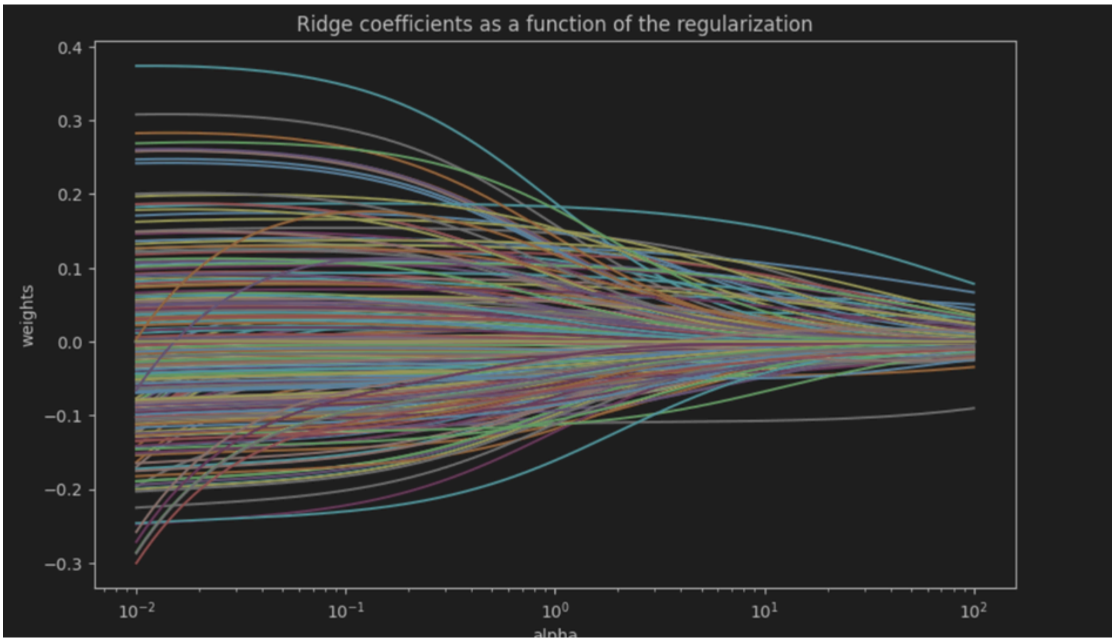


Figure 30. Ridge Trace Map

In the ridge trace map, we need to select the alpha where the ridge trace is relatively stable. Here is the Ridge trace map of our database in the model. We can see that when alpha is about equal to 1, the trajectory of the ridge trace map gradually began to become stable. As can be seen from the ridge trace map, our model has good sensitivity.

9.2.2 Random Forest

After the model is built, we usually need to perform sensitivity analysis to screen out unwanted historical matching parameters, so that the computational cost can be reduced. A random forest is a well-known statistical learning tool that maps a list of input parameters to a predicted response. Even for highly nonlinear data, the constructed random forest model can efficiently rank these input parameters.

In our project study, we selected different numbers of end trees for ergodic studies to strictly compare the capabilities of random forests in sensitivity analysis.We found that by implementing the optimal number of decision trees, random forest configuration parameters, and appropriate experimental methods for fractional factor design, the historical matching results based on random forests showed better results in this case.

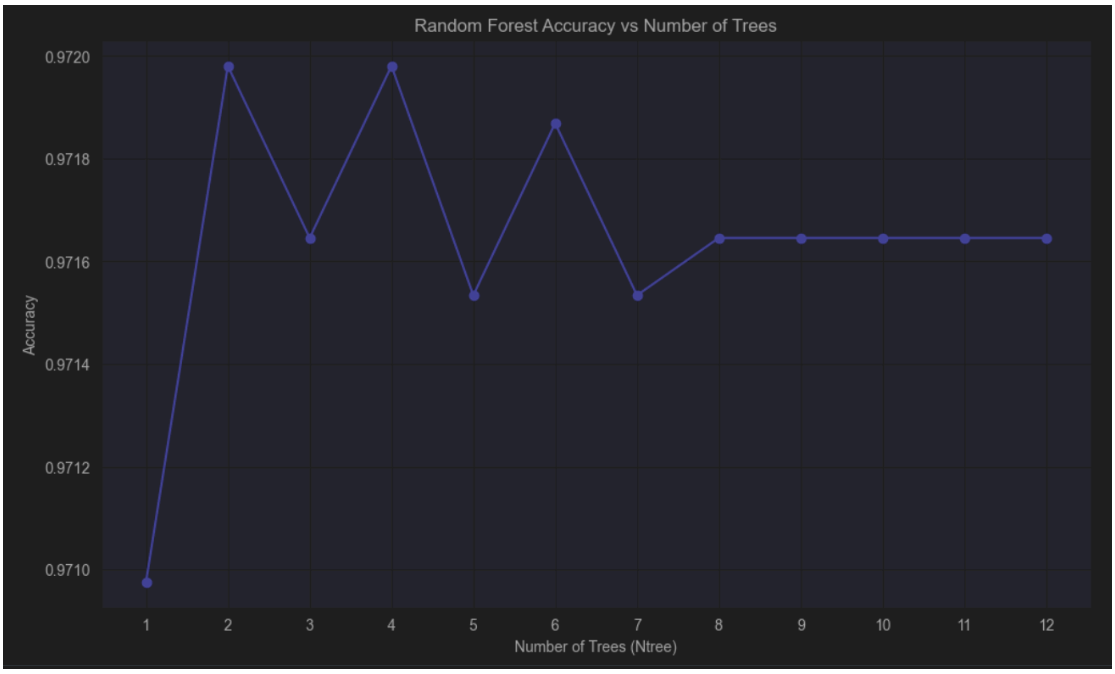


Figure 31. Random Forest Accuracy vs Number of Trees

For our random forest model, we tried to put different integer values into the last end tree. In the adjustment process from 1 to 12, we found that when the number of end trees was 2, the accuracy of our model reached the highest, about 0.97. When the number of end trees was 8, the accuracy of our model was about 0.97. The accuracy of the model is leveling off.

9.3 Data Fluctuation

9.3.1 Other Dataset : Walmart Prediction

Walmart, one of the largest retail stores in the United States, wants to accurately forecast sales and demand. Every day there are certain events and holidays that affect sales. Here is the historical sales data of Walmart's 45 stores, including CPI, unemployment index, etc.

To test the sensitivity of our model, we selected this set of Walmart sales data, the data set is "Walmart.csv".We bring this into the three regression models we have previously built to explore the sensitivity of our model and its broad applicability to other models.

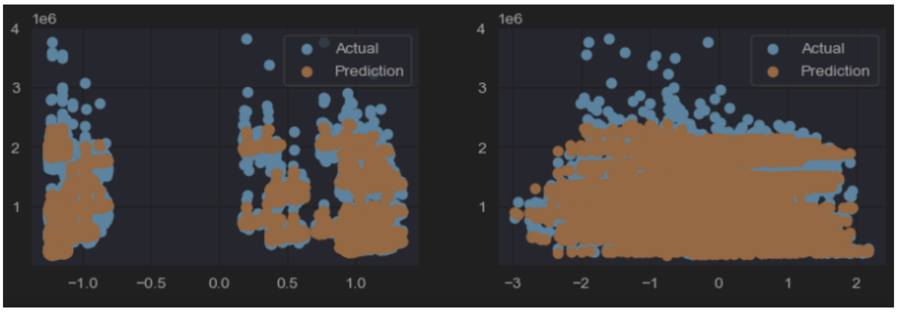
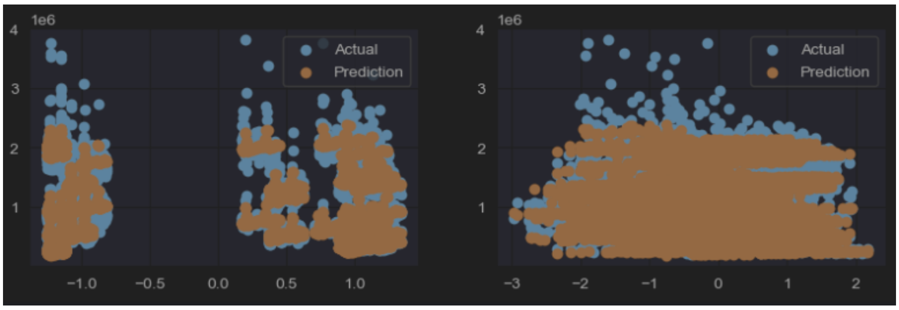
9.3.2 Compare with three regression model

We bring this into the three regression models we have previously built to explore the sensitivity of our model and its broad applicability to other models.

Table 6. Ridge, Lasso, Elastic Net regression function

Ridge	*Ridge Formula: Sum of Error + Sum of the squares of coefficients* $L = \underline{\sum(\hat{Y}i - Yi)}^2 + \underline{\lambda\sum \beta^2}$
Lasso	*Lasso = Sum of Error + Sum of the absolute value of coefficients* $L = \underline{\sum(\hat{Y}i - Yi)^2} + \lambda\sum \underline{\lvert\beta\rvert}$
Elastic Net	*Elastic Net Formula: Ridge + Lasso* $L = \underline{\sum(\hat{Y}i - Yi)^2} + \underline{\lambda\sum \beta^2} + \underline{\lambda\sum \lvert\beta\rvert}$

Table 7. Distributions of actual and predicted values for three regressions

Ridge	
Lasso	

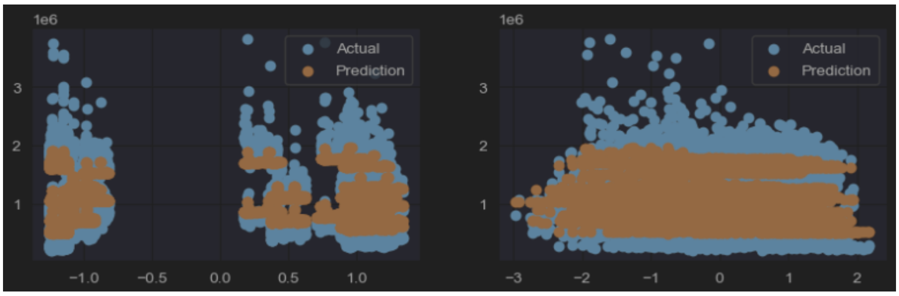
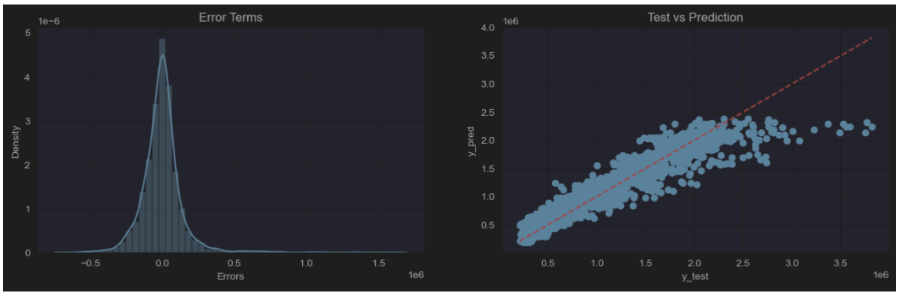
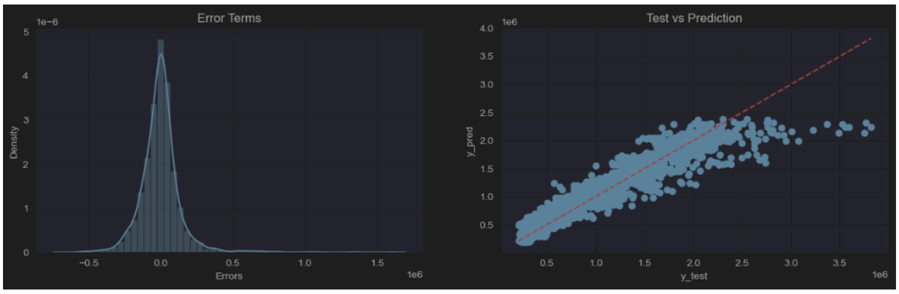
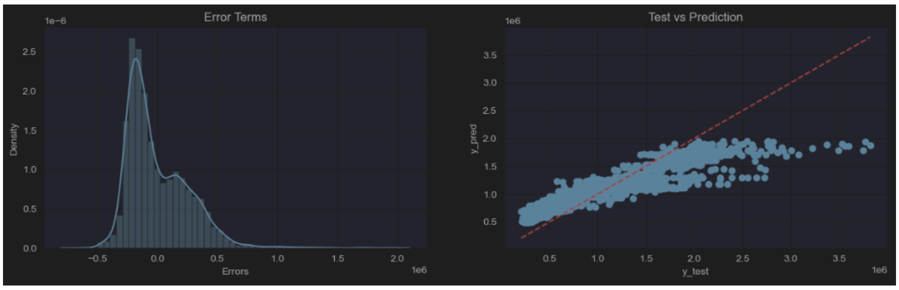
Elastic Net	

Table 8. Error terms and test vs prediction for three regressions

Ridge	
Lasso	
Elastic Net	

For these tables, they show the results of the models are:

- The Intercept of the Regression Model was found to be 1047603.298112138
- Best alpha parameter found: {'alpha': 3.2374575428176433}
- Cross-validation scores: [0.93593586 0.92071592 0.93025392 0.93220418 0.92477374]
- Mean cross-validation score: 0.93

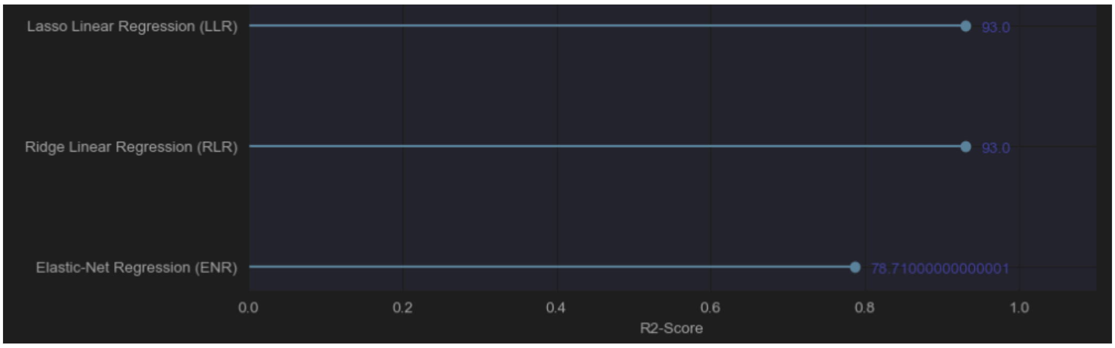
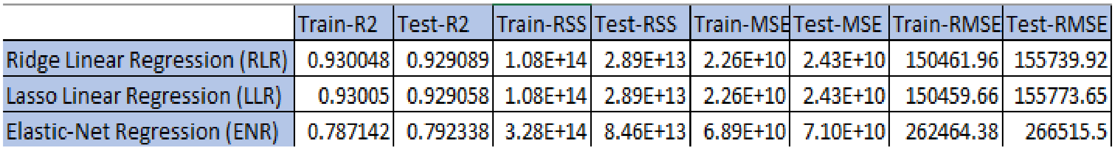


Figure 32. Comparison of three regression R square values

Table 9. Score of R-squared, RSS, MSE, RMSE of train and test



	Train-R2	Test-R2	Train-RSS	Test-RSS	Train-MSE	Test-MSE	Train-RMSE	Test-RMSE
Ridge Linear Regression (RLR)	0.930048	0.929089	1.08E+14	2.89E+13	2.26E+10	2.43E+10	150461.96	155739.92
Lasso Linear Regression (LLR)	0.93005	0.929058	1.08E+14	2.89E+13	2.26E+10	2.43E+10	150459.66	155773.65
Elastic-Net Regression (ENR)	0.787142	0.792338	3.28E+14	8.46E+13	6.89E+10	7.10E+10	262464.38	266515.5

The R-squared of our training set and test set are both around 0.93, indicating that the generalization of our model is relatively good.

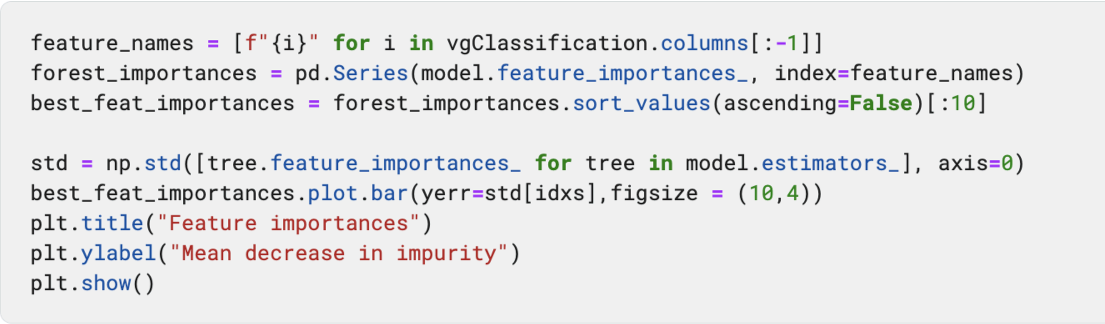
10 Conclusion

10.1 Regressions model conclusion

In the regression model, because the final result of lasso is around 0.3, the effect is poor for this set of data. Because the construction of elastic network regression requires the use of lasso and ridge results at the same time, which also affects the accuracy of the elastic network, we choose the best ridge regression among the three. From the output results only for our data, we can see

- **Different regions have different factors**
  By entering different characteristics, the sales volume of different regions is predicted
- **The ridge regression shows that publish is the most important**
  Here we also output the maximum 10 sets of coefficients for sales
  Nintendo, for example, is No. 1 in Japan, but it's not doing as well in North America
- **Other factors also make sense**
- **The factors affecting the world are the most complex**

10.2 Random Forest is best



```
feature_names = [f"{i}" for i in vgClassification.columns[:-1]]
forest_importances = pd.Series(model.feature_importances_, index=feature_names)
best_feat_importances = forest_importances.sort_values(ascending=False)[:10]

std = np.std([tree.feature_importances_ for tree in model.estimators_], axis=0)
best_feat_importances.plot.bar(yerr=std[idxs],figsize = (10,4))
plt.title("Feature importances")
plt.ylabel("Mean decrease in impurity")
plt.show()
```

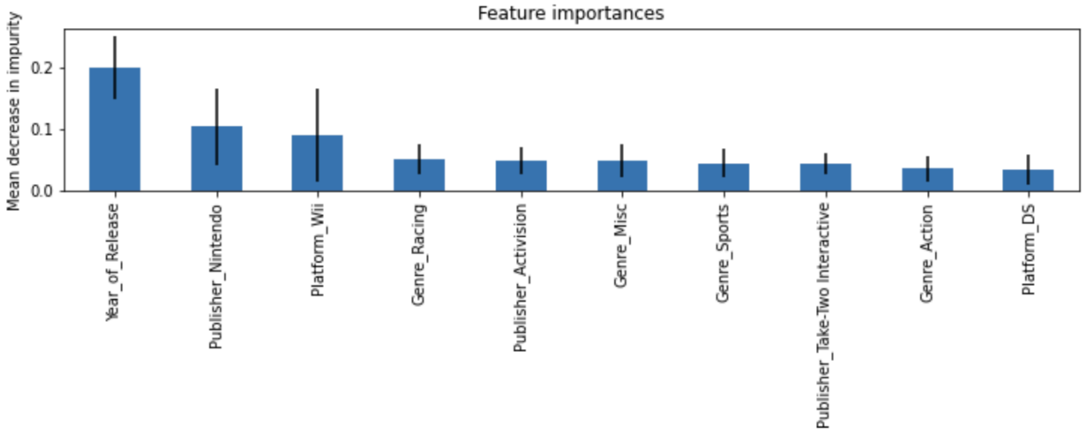


Figure 33. Feature importances of random forest result

The classification effect of random forest can reach 0.97, indicating that for this set of data, the classification effect of random forest will be slightly better than the effect of regression model

11 Acknowledgement

The authors would like to thank Prof. Hongwei Yuan and Prof. Lihu Xu for the kind guidance and assistance for the authors in this Project.

12 Reference

[1] Laufer, B., Docherty, P. D., Murray, R., Krueger-Ziolek, S., Jalal, N. A., Hoeflinger, F., Rupitsch, S. J., Reindl, L., & Moeller, K. (2023). Sensor Selection for Tidal Volume Determination via Linear Regression—Impact of Lasso versus Ridge Regression.

[2] SALEH, A. K. Md. E., NAVRÁTIL, R., & NOROUZIRAD, M. (2018). Rank theory approach to ridge, LASSO, preliminary test and Stein-type estimators: A comparative study. *Canadian Journal of Statistics*, *46*(4), 690–704.

[3] Gabauer, D., Gupta, R., Marfatia, H. A., & Miller, S. M. (2024). Estimating U.S. housing price network connectedness: Evidence from dynamic Elastic Net, Lasso, and

ridge vector autoregressive models. *International Review of Economics & Finance*, *89*, 349–362.

[4] García-Nieto, P. J., García-Gonzalo, E., & Paredes-Sánchez, J. P. (2021). Prediction of the critical temperature of a superconductor by using the WOA/MARS, Ridge, Lasso and Elastic-net machine learning techniques. *Neural Computing & Applications*, *33*(24), 17131–17145.

[5] Liu, H., & Yu, B. (2013). Asymptotic properties of Lasso+mLS and Lasso+Ridge in sparse high-dimensional linear regression.